doi: 10.82124/CEAS-GNC-2026-088 CEAS-GNC-2026-088


# Constrained Model Reference Control for Helicopter Upper Modes

**Julian Schließus** Graduate Student, University of Stuttgart, 70174 Stuttgart, Germany. contact@julian-schliessus.de

**Christian Fischer** Development Engineer, 80331 Munich, Germany. chfm@gmx.de

**Süleyman Özkurt** Research Associate, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. sueleyman.oezkurt@ifr.uni-stuttgart.de

**Torbjørn Cunis** Lecturer, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. tcunis@ifr.uni-stuttgart.de

*ABSTRACT*

**Model reference control (MRC) is a widely used approach in helicopter flight control to impose desired dynamic behaviour – such as specified damping and natural frequencies – on the actual nonlinear and complex helicopter system. In this work, the entire closed-loop system consists of a trajectory planner, which defines simple and well-behaved reference dynamics, and the actual flight control law, which ensures that the helicopter follows these reference trajectories. The trajectory planner generates a trajectory from the target flight state specified by the pilot, e.g., the vertical speed, delivering feasible reference states for the lower-level controller. For unconstrained trajectories, the closed-loop stability can be verified using conventional linear control engineering methods such as, e.g., the Hurwitz criterion. However, these methods are not applicable if the trajectories are constrained by state limits such as the maximum and minimum vertical speed of the helicopter. For constrained trajectories, nonlinearities such as saturations are often inserted into the control loops to consider the helicopter's state limits. These nonlinearities complicate the proof of stability, and thus an alternative approach is required for the stability analysis. In this work, an analytical approach is presented to generate a trajectory that respects input and state constraints while applying a linear control law with the help of the invariant set theory and linear control methods. The method is applied on a model of the Airbus H145 to analyse its control performance.**

**Keywords:** Constrained Rotorcraft Control; Input and State Constraints; Linear Constrained Control; Maximal Control Invariant Set

## Nomenclature

| | | |
|---|---|---|
| $a$ | = | acceleration |
| $g_i$ | = | level function |
| $\boldsymbol{k}$ | = | linear gain vector |
| $K$ | = | control gain |
| $C_\infty$ | = | maximal control invariant set |
| $j$ | = | jerk |
| $\mathcal{K}$ | = | set of feasible linear control gains |
| $\boldsymbol{n}$ | = | outwards pointing normal vector |

| | | |
|---|---|---|
| $P$ | = | plant |
| $u$ | = | input |
| $u_\infty$ | = | feedforward input (for stat. error) |
| $v$ | = | velocity |
| $\boldsymbol{x}$ | = | state vector |
| $y_\infty$ | = | reference (set by pilot) |
| $\delta$ | = | boundary-layer thickness of $C_\infty$ |
| $\tau$ | = | time constant |

| | | |
|---|---|---|
| $\hat{P}$ | = | approximated plant model |
| $u_*$ | = | feedforward reference for robust controller |
| $\mathcal{U}$ | = | set of input constraints |
| $\mathcal{X}$ | = | set of state constraints |
| $y$ | = | output |
| $y_*$ | = | reference for robust controller |
| $\epsilon_i$ | = | tracking-error bound (component $i$) |

# 1 Introduction

The automation of helicopter flight control has become increasingly important due to the system's underactuated and poorly damped nature. Pilots must coordinate multiple control inputs to manage the complex rotorcraft dynamics, making automation via Automatic Flight Control Systems (AFCSs) essential for reducing workload and ensuring safety. Here, model reference control (MRC) is an often used approach to impose a well-damped and simple dynamical behaviour that improves the handling qualities of the rotorcraft. However, feasible and safe flight trajectories, that is, reference inputs for the underlying flight controller, must adhere to strict input and state constraints imposed by actuator limitations, regulatory requirements, and passenger comfort considerations.

Appropriate controller design that considers these constraints is necessary to comply with actuator limitations and regulatory requirements. However, conventional control techniques like PID-control, while effective for unconstrained systems, often fail to guarantee safety and feasibility when input and state constraints have to be met. Nonlinearities in input and state can render planned trajectories inconsistent with the helicopter's actual dynamics, potentially resulting in constraint violations and unsafe flight behaviour.

To overcome this issue, advanced methods such as Model Predictive Control, Reference Governor, and barrier Lyapunov functions, are commonly introduced to guarantee constraint satisfaction while ensuring system stability [1–6]. However, their practical adoption in real-time, safety-critical helicopter applications is hindered by issues such as computational complexity, memory requirements, or the lack of guaranteed feasibility for a set of initial conditions.

This work addresses these challenges by developing a trajectory planning method, based on maximal control invariant sets [7, 8] of the state, that enables a simple, real-time capable linear control design. The trajectory planning is informed by the analytical calculation of the largest set of feasible initial states, ensuring that generated trajectories remain consistent with the helicopter's dynamics while satisfying all prescribed constraints. Subsequently, a linear control law for the MRC framework can be derived. The proposed framework enables well-established methods for flight control design, ease of certification, and implementation in real-time AFCS environments of helicopters without comprising closed-loop stability.

The structure of the remaining paper is as follows: Section 2 defines the problem and outlines the objectives. Section 3 describes the methodology used to calculate the feasible states and the control law. Section 4 applies the proposed method to the double integrator which models the dynamics of the Vertical Speed (VS) mode. In Section 5, the method is extended to the double integrator with low-pass filter which models the dynamics used in the Altitude Hold (ALT) mode. Section 6 concludes the work and discusses future research directions.

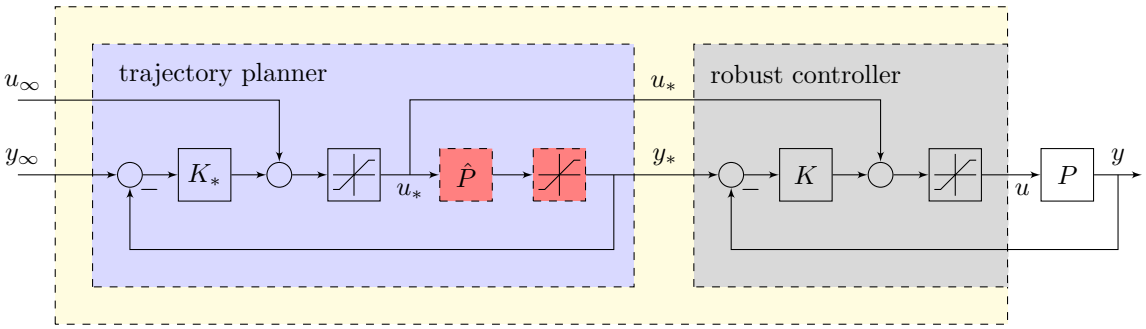



**Figure 1 MRC with saturation on the control input $u$ and on the states $x$.**

# 2 Problem Description

Since the development process of an helicopter AFCS is complex and time-consuming, it is common to divide the upper mode control into a guidance part (trajectory planner) and a control part (robust controller) using MRC [9, 10] as displayed in Fig. 1. With MRC, the advantage is that the plant $P$ does not need to be inverted, which is in some cases, e.g., non-minimum phase systems, not possible. Instead, a low-order approximation model $\hat{P}$ of the plant $P$ is used to generate a trajectory $(y_*, u_*)$ that can be followed by the plant $P$. Also, all states can be extracted from $\hat{P}$, which is useful when constraining the states.

**Definition 2.1 (Trajectory)** *The trajectory generated by the trajectory planner of a MRC framework consists of the pair $(y_*, u_*)$ (see Fig. 1). It consists of the desired system output $y_*$ and the feed-forward control input $u_*$. This pair defines a feasible reference path that the actual system $P$ aims to follow, ensuring consistency with the model dynamics $y_* = \hat{P}u_*$ under the given input and state constraints.*

The task of the control part (grey block in Fig. 1) is to reject perturbations, such as wind gusts, and to stabilise the system around the trajectory $(y_*, u_*)$ generated by the guidance part (purple block in Fig. 1). The guidance part is responsible for generating a feasible trajectory $(y_*, u_*)$ which is followed by the controller. Separating the tasks of trajectory generation and control allows test flights to be performed only to certify the control part. Afterwards, the design of the robust controller is frozen. The guidance part can be developed without the need of further test flights, since the guidance part does not have any feedback loop to the control part. Using such a separation reduces cost and time during the development process of the AFCS.

For linear systems, the stability of MRC can be proven with known conventional methods. However, when input- and state constraints are considered, the application of these methods is no longer sufficient to guarantee stability since nonlinearities in the form of saturation elements are included to the system. The influence of the constraints is shown in the following sections.

## 2.1 Input Constraints

Input constraints arise from the physical limitations of the helicopter's actuators, which can only drive at a limited rate and have limited amplitude authority. To consider the physical limits of the actuator, the control input $u$ is limited with a saturation block (grey background in Fig. 1) in front of the plant $P$. The same input saturation from the robust controller is applied to the trajectory planner (blue background in Fig. 1) in front of $\hat{P}$ when generating the trajectory. The introduced saturation of the control input $u$ has no influence on the consistency of the pair $(y, u)$ since the relation $y = Pu$ is still valid. This also applies to the pair $(y_*, u_*)$ since the relation $y_* = \hat{P}u_*$ is still valid.

## 2.2 State Constraints

One advantage of MRC described previously is that the states $\boldsymbol{x}_*$ of the trajectory $(y_*, u_*)$ can be accessed from the model $\hat{P}$. Due to the stability, handling quality, and ride quality requirements [11–13], the states of the helicopter need to be constrained. These constraints are realised as saturation limits and other nonlinearities on the states $\boldsymbol{x}$ of the helicopter. Specifically, the input constraints $\mathcal{U}$ follow from actuator hardware limits, while the state constraints $\mathcal{X}$ combine physical airframe limits, safety regulations (e.g., descent speed limits to prevent Vortex Ring State (VRS) [11]), and ride-quality standards [13]. This is highlighted in Fig. 1 by red-coloured dashed-bordered blocks, where the dashes signify that the block is nonlinear. Since the relation $u_* = \hat{P} y_*$ is violated through the nonlinearities in the model $\hat{P}$ and the saturation limits on the states of the helicopter, the consistency of the pair $(y_*, u_*)$ is not valid anymore. Consequently, the generated trajectory $(y_*, u_*)$ fails to represent the real helicopter dynamics and the robust controller may not be able to follow the trajectory $(y_*, u_*)$ anymore.

## 2.3 Example: Double Integrator

To illustrate this inconsistency, consider the Vertical Speed Hold Mode (VS) mode (an autopilot/upper mode), where the system dynamics can be described as a double integrating system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{a}_z \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} u \\ a_z \end{bmatrix} \tag{1}$$

with vertical acceleration $a_z$, vertical speed $v_z$, and the control input $u$. Now consider the scenario illustrated in Fig. 2, where the helicopter is initialised with a vertical speed $v_z$ at its lower bound $v_z = \underline{v}_z$ and the vertical acceleration $a_z$ at its lower bound $a_z = \underline{a}_z$. This corresponds to a descending helicopter, which is accelerating its descent. When considering $v_z$, it is clear that since $v_z$ is the integral of the vertical acceleration $a_z$, the vertical acceleration $a_z$ needs to become positive so that the vertical velocity $v_z$ does not become more negative and violate the state constraint $\underline{v}_z$. Because $a_z$ cannot jump to a positive value, as it is defined as the integral of $u$, the vertical velocity $v_z$ will violate the state constraint $v_z \geq \underline{v}_z$, as depicted in Fig. 2. Further, due to the high input $u$, the vertical acceleration increases to a value $a_z > \overline{a}_z$, since $a_z$ is the integral of $u$, violating the upper limit $\overline{a}_z$. As shown in Fig. 2, the saturation blocks ensure that the trajectory (green line) stays within the bounds and does not cross the limits (red dashed lines). The saturation just "freezes" the trajectory at its limit, not considering that the system dynamics are not capable of suddenly holding a state at a fixed value. As a result, the rotorcraft (blue line) diverts from the trajectory (green line) and violates the limits of the state constraints. When the trajectory diverts, dangerous oscillations on the constraint boundaries can occur if the trajectory does not match the helicopter's capabilities.

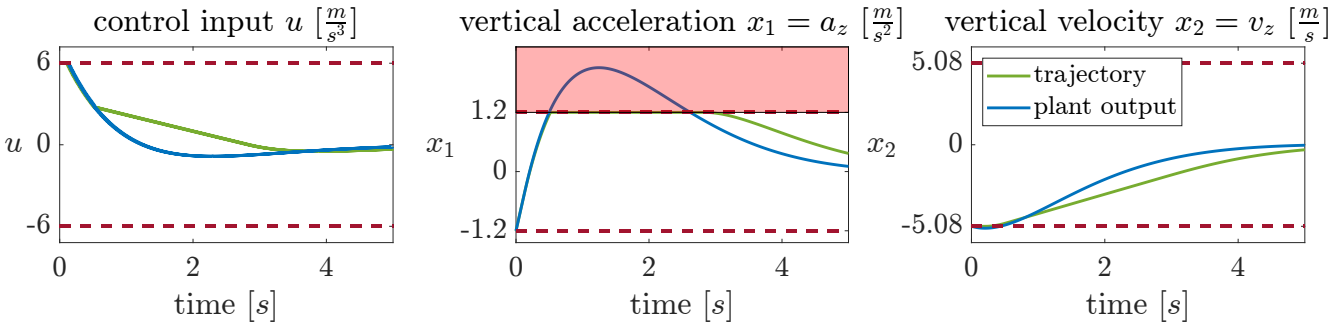



**Figure 2 Detachment of the trajectory (green line) from the helicopter dynamics (blue line) when the trajectory is saturated at the upper vertical speed constraint.**

## 2.4 Motivation and Objective

As shown previously, the conventional trajectory planner with state saturations cannot generate a trajectory $(y_*, u_*)$ that is consistent with the model $\hat{P}$ of the helicopter dynamics while respecting the state constraints. Inconsistencies in the trajectory can lead to detachment from the helicopter dynamics, resulting in dangerous oscillations and violations of safety and comfort requirements. This is due to nonlinearities in the model $\hat{P}$ and the state saturation limits.

Common approaches to ensure constraint satisfaction while maintaining stability include barrier Lyapunov backstepping [5, 6], Model Predictive Control [1, 2, 14], and Reference Governor [3, 4, 15]. However, due to the limited computational power and memory available on the helicopter, the need for real-time execution, and certification requirements for safety-critical systems, these methods are not suitable for the given problem. Therefore, the goal of this work is to present a method that defines trajectory and controller in a feasible manner such that a real-time system implementation and a system certification [11, 12] is possible.

# 3 Methodology

In the following, the method is presented to design the trajectory planner for the MRC framework. To guarantee the existence of a control law $u_*$ that complies with the state constraints $\mathcal{X}$ of the system, the feasible set of initial states $C_\infty$ has to be found. This is obtained analytically through the calculation of the maximal control invariant set $C_\infty$ (see Section 3.4). Afterward, a linear control law (with added input constraints and possible switching nonlinearities) is used to stabilise the system by steering the scalar input $u_*$. From the shape of $C_\infty$, a set of feasible linear control gains $\mathcal{K}$ is calculated that contains all valid linear control gain vectors $\boldsymbol{k}_*$. With this approach, the system can be controlled by manipulating only the control input $u_*$, leaving the states $\boldsymbol{x}_*$ unchanged, such that the trajectory $(\boldsymbol{x}_*, u_*)$ remains a consistent pair. To minimise the online computational effort, the maximal control invariant set $C_\infty$ and the $\mathcal{K}$-space are computed offline which can be loaded onto the helicopter as a look-up table. By providing the set of feasible initial states $C_\infty$ it can be decided whether an upper mode (i.e., autopilot) on the helicopter can be engaged or not. The system is guaranteed to stay within the state constraints $\mathcal{X}$ and the input constraints $\mathcal{U}$ by the definition of the feasible initial states through $C_\infty$ [16]. The following three components have to be considered for the entire methodology.

1) **Input Constraint** $u_* = \mathbf{sat}(u'_*)$
   Model the actuator limitations with a saturation function.
2) **Maximal Control Invariant Set** $C_\infty$ **and Feasible Initial State** $x_{*0}$
   Define the largest set of feasible initial states $C_\infty$ that guarantees the existence of a control input $u_*$ which respects the state constraints $\mathcal{X}$ of the system. The initial state is chosen as $x_{*0} \in C_\infty$.
3) **Control Law** $u'_* - u_\infty = K_*(y_\infty - y_*)$
   A linear control law (with added input constraints and possible switching nonlinearities) is used to stabilise the system by steering the scalar input $u$ so that the trajectory remains within $C_\infty$.

A comprehensive description of these components is given in the following subsections. For some dynamics, the maximal control invariant set $C_\infty$ can be found analytically. Compared to numerical approaches, the analytical calculation of $C_\infty$, at least for some dynamics, is more beneficial for the certification of the rotorcraft, while considering the online computation limitations. In addition, since the presented approach allows the implementation of a linear control law with constraint guarantees, certification will be further facilitated.

The main disadvantage of the approach is that for system dimensions $n > 3$, the maximal control invariant set $C_\infty$ can no longer be computed analytically (but numerically [17]). The offline determination of $C_\infty$ allows the implementation of liner control laws in certified model-based environments, which is

unavoidable considering the limited computational power and memory resources of the rotorcraft Flight Control Computers (FCCs) and the safety requirements [18]. As mentioned previously, the AFCS structure is assumed to be given in form of MRC, especially because the stabilising robust controller is given and assumed to be frozen in design.

## 3.1 Assumptions

As mentioned in Section 2, the focus of this paper lies in the trajectory planner (purple block in Fig. 1). The following assumptions define the scope of this work.

**Assumption 3.1 (Perfect Model)** *The approximation model is assumed to match the plant exactly, $\hat{P} = P$. In the full MRC architecture (Fig. 1), model mismatch and external disturbances are rejected by the robust controller [10, 19], whose design is frozen and outside the scope of this work, since this work focuses on trajectory generation. Under this assumption, the constraint guarantees derived for the model states $\boldsymbol{x}_*$ apply equally to the plant states $\boldsymbol{x}$.*

**Assumption 3.2 (Gain Matching)** *For presentation purposes, the trajectory-planner gain $\boldsymbol{k}_*$ is chosen to coincide with the robust-controller gain $\boldsymbol{k}$, i.e., $\boldsymbol{k}_* = \boldsymbol{k}$. This reduces the closed-loop structure to the simplified form in Fig. 3, bypassing the robust controller for undisturbed flight cases and perfect model knowledge, which allows to define $u_* = u$ as well as $y_* = y$. This assumption is made without loss of generality for the constraint analysis: by Definition 3.2, any $\boldsymbol{k}_* \in \mathcal{K}$ guarantees invariance of $C_\infty$ independently of the robust controller's gain.*

**Remark 3.1 (Interaction with the full MRC architecture)** *In the full MRC architecture (Fig. 1), the input to the plant is $u = u_* + \boldsymbol{k}^\top(\boldsymbol{x}_* - \boldsymbol{x})$, subject to $u \in \mathcal{U}$, where $u_*$ is the feedforward from the trajectory planner and $\boldsymbol{k}$ is the robust controller gain vector. The invariance guarantees of $C_\infty$ apply to the model states $\boldsymbol{x}_*$ driven by $u_*$ alone. For the plant states $\boldsymbol{x}$, two effects arise: (i) the robust controller correction $\boldsymbol{k}^\top(\boldsymbol{x}_* - \boldsymbol{x})$ consumes part of the actuator authority $\mathcal{U}$, reducing the effective input budget available to the trajectory planner; and (ii) the tracking error $\boldsymbol{x}_* - \boldsymbol{x}$ causes the plant states to deviate from the model states. If the robust controller bounds the tracking error per component, $|x_{*_i} - x_i| \leq \epsilon_i$, then plant-state constraint satisfaction can be approached by tightening the state bounds to $[\underline{x}_i + \epsilon_i, \overline{x}_i - \epsilon_i]$ and the input bounds to $[\underline{u} + \boldsymbol{k}^\top\boldsymbol{\epsilon}, \overline{u} - \boldsymbol{k}^\top\boldsymbol{\epsilon}]$ with $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_n]^\top$ when computing $C_\infty$. This conservative tightening is standard in robust invariant-set theory [16] but is not pursued in the present work, which focuses on the nominal trajectory-planning problem.*

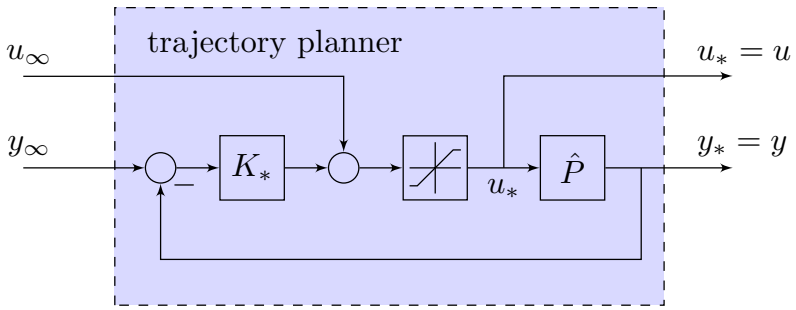



**Figure 3 Simplified structure with controller and input saturation.**

## 3.2 Considered Upper Mode Dynamics

To determine an analytical solution for the set of feasible initial states, the upper modes listed in Table 1 are considered. The dynamics of these flight modes can be represented by $\hat{P}$ which are low-order

approximations of the real linearised rotorcraft dynamics. For the vertical dynamics, the jerk $j_z = \dot{a}_z$ is steered by pitching the rotor blades with the collective rod. Therefore, to maintain a vertical speed $v_z$ a double integrating model $\hat{P}$ is introduced in the VS mode. To control the vertical position $p_z$ a triple integrator is thus used as model in the Altitude Hold (ALT) mode. For the VS- and ALT mode a first-order low-pass filter can be introduced to model actuator / helicopter inertia. [1] Since the helicopter models are based on systems of order $n \leq 3$, the analytical solution of $C_\infty$ is possible for the given problem which will be discussed later.

**Table 1 Overview of the considered upper modes.**

| Flight Mode | Controlled Variable | Regulated Variable | Dynamics $\hat{P}$ |
|---|---|---|---|
| VS | $j_z$ (actuator directly steered) | $v_z$ | $\frac{1}{s^2}$ or $\frac{1}{s(1+\tau s)}$ |
| ALT | $j_z$ (actuator directly steered) | $p_z$ | $\frac{1}{s^3}$ or $\frac{1}{s^2(1+\tau s)}$ |

## 3.3 Input Constraint

Since the amplitude and rate of helicopter actuators are limited, the control input $u$ is subject to constraints. Within this paper, the input constraint

$$\mathcal{U} = \left\{u \in \mathbb{R} : \underline{u} \leq u \leq \overline{u}\right\} \tag{2}$$

with lower and upper bound $\underline{u}$ and $\overline{u}$, respectively, of the actuator input is considered. A saturation block in Fig. 3 is used to clip the control input $u$ at the actuator boundaries.

## 3.4 Feasible Initial States

The trajectory is subject to state constraints imposed by governmental regulations and safety requirements. These may, for example, take the form of a maximum vertical speed or a maximum vertical acceleration. The state constraints are defined as

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \underline{x}_1 \leq x_1 \leq \overline{x}_1, \ldots, \underline{x}_n \leq x_n \leq \overline{x}_n\} \tag{3}$$

with lower and upper bound $\underline{x}_i$ and $\overline{x}_i$, respectively, of the state $x_i$. The chosen approach is to find the largest set of feasible initial states that guarantees the existence of a control input $u$ that respects the constraints of the system. With this approach, the system can be controlled by manipulating only the control input $u$, leaving the states $\boldsymbol{x}$ unchanged, such that the trajectory $(\boldsymbol{x}, u)$ remains a consistent pair, i.e., no non-linearities. The largest set of feasible initial states is named the maximal control invariant set. The maximal control invariant set $C_\infty$ can be constructed with the help of the level functions $g_i$ that enclose the set $C_\infty$. A more general definition of $C_\infty$ can be found in Doeser et al [8]. The definition of a level function, taken from [8], is given below.

**Characterization 3.1 (Level Function $g_i$)** *The level function $g_i$ characterises the shape of the maximal control invariant set $C_\infty$ for the i-th state component. The extremal time $t^*$ is the time at which the state $x_i(t)$ has an extremum (minimum or maximum), which can lie in the future or in the past depending on the initial state $x_{i_0}$. The level function $g_i$ can be described piecewise depending on the extremal time $t^*$*

[1] The dynamics of the double integrator with first-order low-pass filter can also be used as a model for other flight modes, e.g., Indicated Air Speed Hold Mode (IAS).

*as*

$$\underline{g}_i(x_{1_0}, \ldots, x_{i_0}, \overline{u}) = \begin{cases} \underline{x}_i - \min_t x_i(t) \leq 0, & \text{if } t^* \geq 0 \\ \underline{x}_i - x_{i0} \leq 0, & \text{otherwise} \end{cases}$$
$$\overline{g}_i(x_{1_0}, \ldots, x_{i_0}, \underline{u}) = \begin{cases} \max_t x_i(t) - \overline{x}_i \leq 0, & \text{if } t^* \geq 0 \\ x_{i0} - \overline{x}_i \leq 0, & \text{otherwise} \end{cases} \tag{4}$$

*where $\underline{g}_i$ and $\overline{g}_i$ are the lower and upper bounds of the $g_i$ level function, respectively, $\underline{x}_i$ and $\overline{x}_i$ are the lower and upper bounds of the state $x_i$ from the constraint $\mathcal{X}$, respectively, and $x_i$ is the $i$-th element of the state vector $\boldsymbol{x}$, determined by solving the differential equation of the system. The minimisation / maximisation of $x_i(t)$ is performed under a constant maximal / minimal control input $\overline{u}$ / $\underline{u}$. The overall level function $g_i$ is then formulated as*

$$g_i = \max\left(\overline{g}_i, \underline{g}_i\right), \tag{5}$$

*where $\underline{g}_i$ and $\overline{g}_i$ denote the lower and upper bound functions, respectively. The boundary of the maximal control invariant set $C_\infty$ is characterised by $g_i = 0$, and the interior by $g_i < 0$.*

The role of the level function $g_i$ is to define the shape of the maximal control invariant set $C_\infty$ from the system dynamics and the state constraints. Traditionally, the solution of the differential equation $\dot{x}$ is used to describe the trajectory of the system departing from an initial state $\boldsymbol{x}_0$ at a time $t$. Taking the extremum over non-negative time $t* \geq 0$, the extreme value might be the initial value and $\min_t x_i(t) = x_{i_0}$. Since the goal is to maximise the size of the set $C_\infty$ and the state constraints $\mathcal{X}$ shall not be violated, the extremum of the state $x_i(t)$ is transformed with a translation onto the bounds $\underline{x}_i$ and $\overline{x}_i$. This is done by requiring that $\min_t x_i(t) \geq \underline{x}_i$ and $\max_t x_i(t) \leq \overline{x}_i$, with min and max representing the extremum, respectively and the relation can then be reformulated as $g_i \leq 0$. The $g_i$ function can thus be seen as a transformation of the system trajectory $\boldsymbol{x}(t)$ onto the boundary of the set of state constraints $\mathcal{X}$ to maximise the size of the set of feasible initial states $C_\infty$. As a reminder, $C_\infty$ is defined as the set of initial states $\boldsymbol{x}_0$ for which there exists a control input $u$ that respects the state constraints $\mathcal{X}$ and the input constraints $\mathcal{U}$. Having defined the level function $g_i$, the maximal control invariant set $C_\infty$ can now be constructed.

**Definition 3.1 (Construction of $C_\infty$, [8, 16])** *The maximal control invariant set $C_\infty$ is defined as the intersection of the level functions $g_i$:*

$$C_\infty = \left\{\boldsymbol{x} \in \mathbb{R}^n : \forall i,\ \underline{g}_i(\boldsymbol{x}, \overline{u}) \leq 0 \ \text{ and } \ \overline{g}_i(\boldsymbol{x}, \underline{u}) \leq 0\right\} \tag{6}$$

*with $\underline{g}_i$ and $\overline{g}_i$ being the lower and upper bounds of the $g_i$ level function, respectively. For each intersection of the level functions $g_i$ in $C_\infty$, the following condition must additionally hold:*

$$\mathcal{U}_{\underline{g}_j, \overline{g}_k}(\boldsymbol{x}) = \left\{u \in \mathcal{U} : \underline{g}_j(x_1, ..., x_j, u) \leq 0,\ \overline{g}_k(x_1, ..., x_k, u) \leq 0\right\} \neq \emptyset \tag{7}$$

*meaning that at each intersection of the level functions $g_j$ and $g_k$, the intersection of the input constraints $\mathcal{U}_{\underline{g}_j, \overline{g}_k}$ is not empty and the level functions $g_j$ and $g_k$ are thus not in conflict with each other.*

**Remark 3.2 (Physical Interpretation of $C_\infty$)** *Because $\mathcal{X}$ reflects the physical limitations of the rotorcraft (see Section 2), $C_\infty$ represents the largest set of flight states from which the autopilot can be safely engaged. For higher-dimensional systems ($n \geq 3$), the shape of $C_\infty$ may be complex; in practice, tighter per-state bounds can be chosen within $C_\infty$ to obtain simple engagement criteria that are intuitive for pilots and straightforward to display on the Multifunction Display. The tightening of these bounds is a design choice that allows engineers to trade off between state dimensions depending on the required handling qualities and safety margins.*

**Remark 3.3 (Behaviour outside $C_\infty$)** *If the current state $\boldsymbol{x}_0 \notin C_\infty$, no admissible control input $u \in \mathcal{U}$ exists that can prevent a violation of the state constraints $\mathcal{X}$. This does not reflect an algorithmic limitation, but rather a physical one: the actuator authority is insufficient to counteract the current flight state. For instance, if the helicopter descends at maximum velocity $\overline{v}_z$ with maximum acceleration $\overline{a}_z$, the collective may lack sufficient authority to arrest the descent before exceeding the speed limit. Nonetheless, the saturated linear control law $u = sat(-\boldsymbol{k}^\top \boldsymbol{x})$ remains well-defined outside $C_\infty$ and continues to steer the system towards the origin as aggressively as $\mathcal{U}$ permits. The constraint guarantees are lost during the transient, but no discontinuity or instability is introduced. Whether such a best-effort mode is acceptable in practice is a certification decision and outside the scope of this work.*

In contrast to the search for the extremum of the trajectory $\boldsymbol{x}(t)$ using min/max as in Characterization 3.1, the max operator in Definition 3.1 selects the most restrictive level function among all $g_i$; the outer boundary of the maximal control invariant set $C_\infty$ is then characterised by $\max_i g_i = 0$, while the interior satisfies $\max_i g_i < 0$.

## 3.5 Linear Control Law

$C_\infty$ guarantees the existence of a control input $u$ that respects the constraints of the system, but it does not guarantee the stability of the system. Further, the suitable control input $u$ needs to be determined since only the existence of a control input $u$ is guaranteed. To provide an easy implementation of the flight control law, introduce the linear control law

$$u = -\boldsymbol{k}^\top \boldsymbol{x} \tag{8}$$

where $\boldsymbol{k}$ is the gain vector and $\boldsymbol{x}$ is the state vector.[2] In the case of the single integrating system, the system is guaranteed to be stable for $k > 0$; for higher-order systems, the $\mathcal{K}$-space is introduced, which describes the set of stable $\boldsymbol{k}$ that respect the bounds of the set $C_\infty$. To select the valid inputs $u$ that respect the constraints of $C_\infty$, the outer normal vector $\boldsymbol{n}$ of the set $C_\infty$ is constructed. When the angle between the vector field $\dot{\boldsymbol{x}}$ of the system dynamics and the outer normal vector $\boldsymbol{n}$ is greater than or equal to 90°, the system will not leave the set $C_\infty$, which can be expressed as $\boldsymbol{n}^\top \dot{\boldsymbol{x}} \leq 0$ where $\dot{\boldsymbol{x}}$ is the vector field of the system dynamics and $\boldsymbol{n}$ is the outer normal vector of the set $C_\infty$ pointing outwards. This expression guarantees that the system trajectory does not cross the boundary of the set $C_\infty$ and thus never leaves the set $C_\infty$. The outer normal vector $\boldsymbol{n}$ can be found by calculating the gradient of the level function $g_i$ on the boundary of $C_\infty$ at $g_i = 0$ (because the level function rises towards the boundary of the set, the gradient points outwards) [16, 21]. This can be expressed as

$$\boldsymbol{n}^\top \dot{\boldsymbol{x}} = \nabla g_i^\top \dot{\boldsymbol{x}} \leq 0 \tag{9}$$

Having defined a condition at the boundary of the set $C_\infty$ that selects a suitable control input $u$, the linear control law can be found by inserting the system dynamics $\dot{\boldsymbol{x}}$ into the condition and solving for the linear control gain vector $\boldsymbol{k}$. This yields a set of matrix inequalities that describe the space of valid control gains $\boldsymbol{k}$, called from now on the $\mathcal{K}$-space, that respect the constraints of the system.

[2] To reduce complexity of this work, only the regulator problem is considered, stabilising the system around the origin. However, reference tracking is also possible when introducing $\Delta\boldsymbol{x}$ in the control law, [20].

**Definition 3.2 ($\mathcal{K}$-space)** *The $\mathcal{K}$-space is defined below and represents the set of all linear control gain vectors $\boldsymbol{k}$ that respect the constraints of the system with the linear control law $u = -\boldsymbol{k}^\top \boldsymbol{x}$:*

$$\mathcal{K} = \left\{ \boldsymbol{k} \in \mathbb{R}^n \; : \; \begin{array}{l} \nabla \underline{g}_1^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^n : \underline{g}_1(\boldsymbol{x}_1, \overline{u}) \; = 0 \right\} \wedge \\ \ldots \quad \wedge \\ \nabla \underline{g}_n^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^n : \underline{g}_n(\boldsymbol{x}_{1:n}, \overline{u}) = 0 \right\} \wedge \\ \nabla \overline{g}_1^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^n : \overline{g}_1(\boldsymbol{x}_1, \underline{u}) \; = 0 \right\} \wedge \\ \ldots \quad \wedge \\ \nabla \overline{g}_n^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^n : \overline{g}_n(\boldsymbol{x}_{1:n}, \underline{u}) = 0 \right\} \end{array} \right\} \tag{10}$$

*where $\dot{\boldsymbol{x}}$ is the vector field of the system dynamics.*

The $\mathcal{K}$-space guarantees the stability of the system and thus invariance of the set $C_\infty$ for $\boldsymbol{u} = -\boldsymbol{k}^\top \boldsymbol{x}$. Because any controller weights $\mathbf{k} \in \mathcal{K}$-space prevents the system from leaving $C_\infty$ at the boundary $\partial C_\infty$, only the normal vectors on the boundary of the set $C_\infty$ are considered.

**Remark 3.4 (Absence of integrator windup)** *All control modes in the trajectory planner – the control law $u = -\boldsymbol{k}^\top \boldsymbol{x}$, the saturation, and the boundary-holding mode $u_* = x_1$ (Section 5.3) – are memoryless functions of $\boldsymbol{x}$ and therefore free of integrator windup; consequently, no windup-induced performance degradation arises. When $\hat{P} \neq P$, tracking-error accumulation through the plant integrators is rejected by the robust controller (Assumption 3.1 and Remark 3.1), whose own anti-windup design is frozen and outside the scope of this work.*

# 4 Vertical Speed Mode - Double Integrator System

The double integrator model can be used to describe various aspects of helicopter dynamics. For example, it can describe the influence of a change in the time derivative of the vertical acceleration $\dot{a}_z$ on the vertical velocity $v_z$ in the VS mode.

## 4.1 General Formulation

In a more general form, the double integrator system can be described in state-space representation as follows

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}}_{=\boldsymbol{A}} \boldsymbol{x}(t) + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{=\boldsymbol{B}} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{=\boldsymbol{C}} \boldsymbol{x}(t) \end{aligned} \tag{11}$$

with $y(t)$ being the output of the system (vertical speed $v_z$ in the VS mode). The states $\boldsymbol{x}(t)$ and the input $u(t)$ are defined in the subspaces

$$\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^2, \; u(t) \in \mathcal{U} \subseteq \mathbb{R}, \; \forall t \geq 0 \tag{12}$$

The input constraint $\mathcal{U}$ is enforced with a saturation block as mentioned in Section 3.3.

## 4.2 Feasible Initial States

The system is subject to state constraints in $x_1$ and $x_2$, which can be formulated as

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^2 : \underline{x}_1 \leq x_1 \leq \overline{x}_1, \underline{x}_2 \leq x_2 \leq \overline{x}_2\} \tag{13}$$

where $\underline{x}_.$ represents the lower bound and $\overline{x}_.$ the upper bound, respectively.

The shape of the set $C_\infty$ can be found by solving the differential equation of the system dynamics $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ with a constant $u$ at the bounds of the input constraint $\mathcal{U}$. The differential equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u$ can be solved as

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0 + \int_0^t \left[e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}u(\tau)\right] d\tau, \quad u \in \{\underline{u}, \overline{u}\} \tag{14}$$

where $\boldsymbol{x}_0$ is the initial state of the system and $u(t)$ is the control input. $\boldsymbol{x}_0$ describes the starting point of the trajectory $\boldsymbol{x}(t)$ that will be used to find the shape of the set $C_\infty$. The maximal control invariant set $C_\infty$ is defined by the extremal trajectory with $u = \underline{u}$ for $x_1 > 0$ and $u = \overline{u}$ for $x_1 < 0$. To obtain the shape of the extremal trajectory on the boundary of the set $C_\infty$, the differential equation in Eq. (11) is solved with the input $u = \underline{u}$ for $x_1 > 0$ and $u = \overline{u}$ for $x_1 < 0$, respectively. The solution of the differential equation can be found by first evaluating the matrix exponential $e^{\boldsymbol{A}t}$ and $e^{-\boldsymbol{A}\tau}$ and then evaluating the integral:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_{1_0} + ut \\ x_{1_0}t + x_{2_0} + u\frac{t^2}{2} \end{bmatrix}, \quad u \in \{\underline{u}, \overline{u}\} \tag{15}$$

which is the solution for $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ for constant $\underline{u}$ or $\overline{u}$. To maximise the size of the control invariant $C$[3] such that it becomes $C_\infty$, the trajectory needs to be at its extremum at $x_1 = 0$. With the definition of the $g_i$ level function from Characterization 3.1, the level functions $g_i$ for the double integrator system can be constructed as follows:

$$\begin{aligned}
\underline{g}_1(x_{1_0}) &= \underline{x}_1 - x_{1_0} \leq 0 \\
\overline{g}_1(x_{1_0}) &= x_{1_0} - \overline{x}_1 \leq 0 \\
\underline{g}_2(x_{1_0}, x_{2_0}, \overline{u}) &= \begin{cases} \underline{x}_2 - x_{2_0} + \frac{x_{1_0}^2}{2\overline{u}} & \leq 0, \quad \text{if } x_{1_0} < 0 \\ \underline{x}_2 - x_{2_0} & \leq 0, \quad \text{if } x_{1_0} \geq 0 \end{cases} \\
\overline{g}_2(x_{1_0}, x_{2_0}, \underline{u}) &= \begin{cases} x_{2_0} - \overline{x}_2 & \leq 0, \quad \text{if } x_{1_0} \leq 0 \\ x_{2_0} - \overline{x}_2 - \frac{x_{1_0}^2}{2\underline{u}} & \leq 0, \quad \text{if } x_{1_0} > 0 \end{cases}
\end{aligned} \tag{16}$$

with the lower bound level function $\underline{g}_i$ and the upper bound level function $\overline{g}_i$. The intersected level functions $g_i$ are shown for $g_i = 0$ in Fig. 4. As described in Definition 3.1, assembling all segments gives a definition of the set $C_\infty$ as the intersection of the level functions $g$, which presents the set of feasible initial states $C_\infty$ (i.e., the maximal control invariant set):

$$C_\infty = \left\{\boldsymbol{x}_0 \in \mathbb{R}^2 : \max\left(\overline{g}_1\left(x_{1_0}\right), \underline{g}_1\left(x_{1_0}\right), \overline{g}_2\left(x_{1_0}, x_{2_0}, \underline{u}\right), \underline{g}_2\left(x_{1_0}, x_{2_0}, \overline{u}\right)\right) \leq 0\right\} \tag{17}$$

The intersection of the level functions $g_1$ and $g_2$ is shown in Fig. 4, with the $g_1$-functions marked in blue and the $g_2$-functions marked in purple and magenta.

[3] The control invariant set $C$ is defined as the set that guarantees the existence of a control input $u$ such that all trajectories that start in $C$ stay in $C$ while respecting input constraints $\mathcal{U}$.

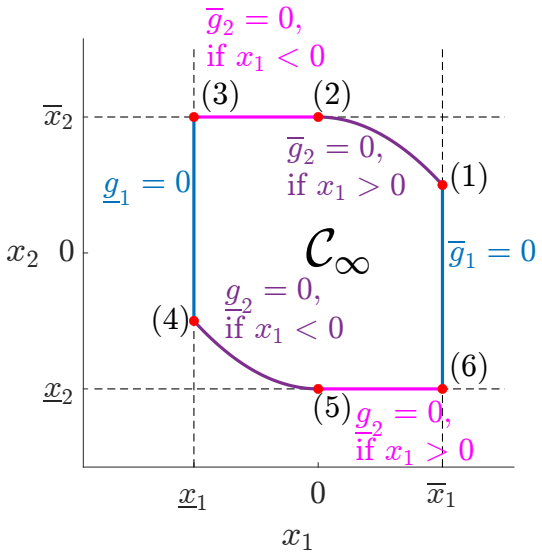



**Figure 4 Control invariant set $C_\infty$ with intersected level functions $g_i$ at $g_i = 0$.**

## 4.3 $\mathcal{K}$-Space - Control Law

Up to this point, the maximal control invariant set $C_\infty$ has only defined the set of initial states that guarantee the existence of a control input $u \in \mathcal{U}$ which keeps the system within the set $C_\infty$. In this section, a linear control law $u = -\boldsymbol{k}^\mathsf{T}\boldsymbol{x}$ is chosen and is saturated to guarantee the invariance of the set $C_\infty$. As described in Section 3.5, the valid set of linear control gains $\boldsymbol{k}$ can be found by solving $\nabla g_i{}^\mathsf{T}\dot{\boldsymbol{x}} \leq 0$. To find a valid set of control gains $\boldsymbol{k}$, the condition $\nabla g_i{}^\mathsf{T}\dot{\boldsymbol{x}} \leq 0$ must be evaluated for all $g_i$-functions. Based on that a set $\mathcal{K}$ of control gains $\boldsymbol{k}$ can be constructed to ensure the the invariance of the set $C_\infty$. First, the $\mathcal{K}_1$-space is constructed for the $g_1$-functions using Definition 3.2:

$$\mathcal{K}_1 = \left\{ \boldsymbol{k} \in \mathbb{R}^2 \ : \ \begin{array}{l} k_1\underline{x}_1 + k_2 x_2 \ \leq 0, \forall x_2 \in [\underline{x}_{2_0}, \overline{x}_2] \cap \\ -k_1\overline{x}_1 - k_2 x_2 \leq 0, \forall x_2 \in [\underline{x}_2, \overline{x}_{2_0}] \end{array} \right\} \tag{18}$$

Then the $\mathcal{K}_2$-space is constructed considering the $g_2$-functions of the double integrator system:

$$\mathcal{K}_2 = \left\{ \boldsymbol{k} \in \mathbb{R}^2 \ : \ \begin{array}{l} -k_1 x_1 - k_2 x_2 \leq -\overline{u}, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \underline{g}_2(\boldsymbol{x}_{1:2}, \overline{u}) = 0 \right\} \cap \\ k_1 x_1 + k_2 x_2 \ \leq \underline{u}, \forall \boldsymbol{x} \in \partial C_\infty \cap \ \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \overline{g}_2(\boldsymbol{x}_{1:2}, \underline{u}) = 0 \right\} \end{array} \right\} \tag{19}$$

The $\mathcal{K}_i$ spaces represent linear inequality systems which are visualised as blue $\mathcal{K}_1$ and purple $\mathcal{K}_2$ areas in Fig. 5a.

To obtain the full $\mathcal{K}$-space for the double integrator system, the $\mathcal{K}_1$-space and the $\mathcal{K}_2$-space must be intersected. The assembled $\mathcal{K}$-space, consisting of the $g_1$-functions and the $g_2$-functions, is defined as $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$ and formulated in Eq. (A.40). The intersection of the $\mathcal{K}_1$-space and the $\mathcal{K}_2$-space is shown in Fig. 5a, and the final $\mathcal{K}$-space is shown in Fig. 5b.

## 4.4 Simulation Results for the Vertical Speed Mode

In this section, a simulative study is shown in Fig. 6 which demonstrate that for a chosen $\boldsymbol{k}$ from the $\mathcal{K}$-space, the system drives always to the equilibrium if the initial states $\boldsymbol{x}_0$ from the feasible set $C_\infty$ are taken. In the VS mode, the vertical velocity $v_z$ is tracked. The input constraint is defined by the maximum actuator rate of $\overline{\dot{u}}_{\text{act}} = 10\ \frac{mm}{s}$. The factor $h$, which scales the actuator amplitude to the vertical

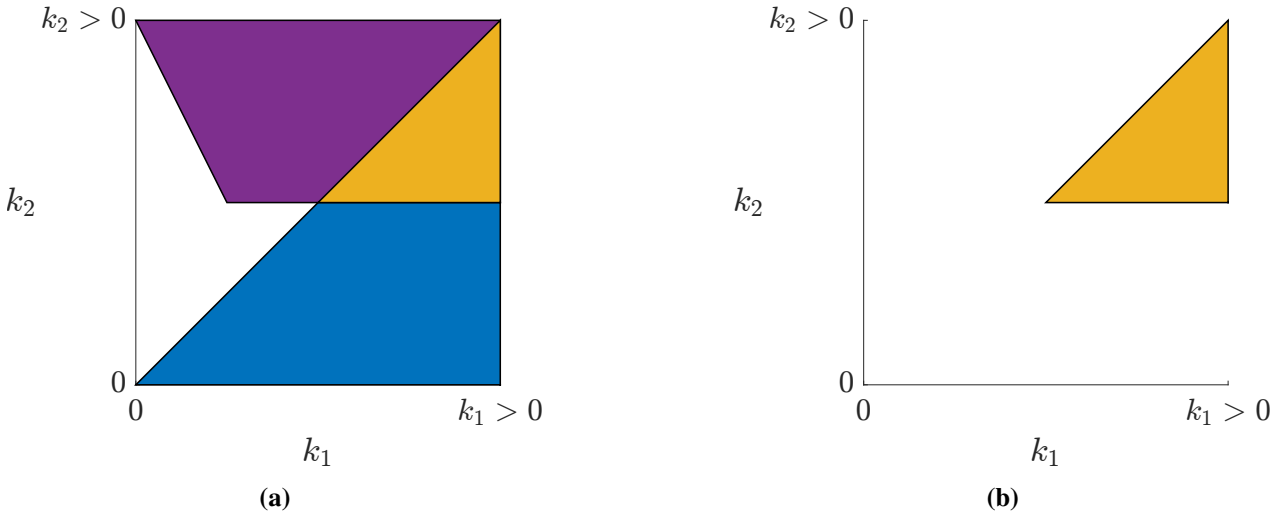



**Figure 5** $\mathcal{K}_1$ **space (blue),** $\mathcal{K}_2$ **space (purple), and the intersection of both spaces as** $\mathcal{K}$ **(yellow).**

acceleration, is set to $h = 0.1\ \frac{m}{s^2\ mm}$. Since the input of the VS mode is the time derivative of the vertical acceleration, the input constraint is

$$\overline{u} = \overline{\dot{a}}_z = h \cdot \overline{\dot{u}}_{\text{act}} = 0.1\ \frac{m}{s^2\ mm} \cdot 10\ \frac{mm}{s} = 1\ \frac{m}{s^3} \tag{20}$$

The state constraint for the first state, $x_1 = a_z$, is defined by the maximum actuator amplitude of $\overline{u}_{\text{act}} = 10\ mm$. Together with the factor $h$, this results in a maximum vertical acceleration of

$$\overline{x}_1 = \overline{a}_z = h \cdot \overline{u}_{\text{act}} = 0.1\ \frac{m}{s^2\ mm} \cdot 10\ mm = 1\ \frac{m}{s^2} \quad . \tag{21}$$

The second state, $x_2 = v_z$, is constrained by the maximum vertical speed of $\overline{v}_z = 1000\ \frac{ft}{min}$, which is the maximum descent speed to prevent vortex-ring state. For simplicity, the constraint is assumed to be symmetric, thus

$$\overline{x}_2 = \overline{v}_z = 1000\ \frac{ft}{min} \approx 5\ \frac{m}{s} \quad . \tag{22}$$

The linear control gain $\boldsymbol{k}$ is chosen from the $\mathcal{K}$-space of the system as $\boldsymbol{k} = \begin{bmatrix} 1.28 & 0.32 \end{bmatrix}^\top$. It can be observed that the input- and state constraints are respected by the system and the trajectories converge towards the origin.

# 5 Altitude Hold Mode - Double Integrator and PT1

The helicopter dynamics are often subject to lag due to inertia effects of the rotor-structure coupling or actuator lag. In the following, the lag considered for the ALT mode is described. This upper mode can be modelled using a first-order low-pass filter with a double integrator (see Table 1).

## 5.1 General Formulation

The general formulation of the system dynamics, i.e., $\hat{P} = \frac{1}{s^2(1+\tau s)}$, is given by the state-space representation:

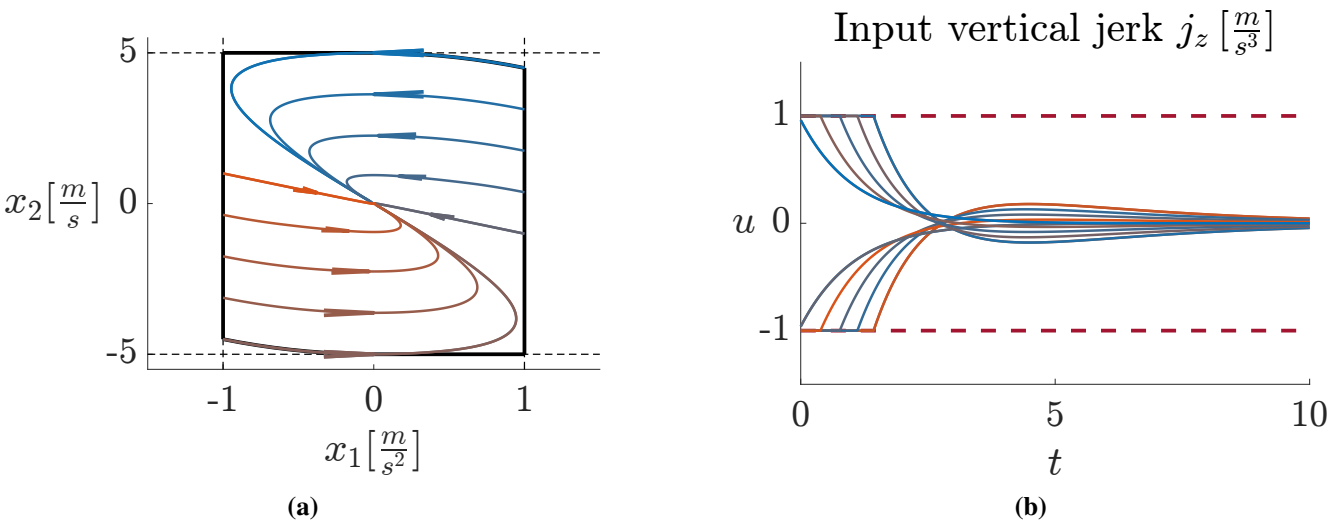



**Figure 6 (a) Phase plane and (b) control input $u$ of the double integrator system with the initial states $C_\infty$ and the linear control gains $\mathcal{K}$.**

$$\begin{aligned}\dot{x}_1 &= -\frac{1}{\tau}x_1 + \frac{1}{\tau}u \\ \dot{x}_2 &= x_1 \\ \dot{x}_3 &= x_2\end{aligned} \tag{23}$$

The states $\boldsymbol{x}(t)$ and the input $u(t)$ are defined in the subspaces

$$\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^3,\ u(t) \in \mathcal{U} \subseteq \mathbb{R},\ \forall t \geq 0 \tag{24}$$

The input constraint $\mathcal{U}$ is enforced with a saturation block as mentioned in Section 3.3.

## 5.2 Feasible Initial States

As in the previous sections, the differential equation of system dynamics are solved for a constant input $u(t) = \{\underline{u}, \overline{u}\}$:

$$x(t) = e^{At}(x_0 + \int_0^t e^{-A\tilde{t}}d\tilde{t}\,Bu),\ u \in \{\underline{u}, \overline{u}\} \tag{25}$$

Evaluating the integral, the solution of the differential equation is obtained as

$$\boldsymbol{x}(t) = \begin{bmatrix} u(1-e^{-\frac{t}{\tau}}) & +x_{10}\,e^{-\frac{t}{\tau}} & \\ u(t-\tau+\tau\,e^{-\frac{t}{\tau}}) & +x_{10}\tau(1-\,e^{-\frac{t}{\tau}}) & +x_{20} \\ u(\frac{t^2}{2}+\tau^2-\tau^2\,e^{-\frac{t}{\tau}}-t\,\tau) & +x_{10}\tau(-\tau+\tau\,e^{-\frac{t}{\tau}}+t) & +t\,x_{20}+x_{30} \end{bmatrix},\ u \in \{\underline{u}, \overline{u}\}. \tag{26}$$

To find the invariant set $C_\infty$ for considered system, the trajectory from Eq. (26) is solved for $x_2(t) = 0$ to find the extremum in $x_3(t)$, enabling calculation of the level functions $g_3(x) \leq 0$ according to Characterization 3.1. By assuming $u \neq x_{1_0}$, $u - x_{1_0} \neq 0$, and reformulating $x_2(t) = 0$ to separate the exponential term , yields

$$e^{-\frac{t}{\tau}} = \frac{u - \frac{ut}{\tau} - x_{1_0} - \frac{x_{2_0}}{\tau}}{u - x_{1_0}}. \tag{27}$$

Rearranging and introducing the Lambert $W$ function (see Section A.2) gives the time $t^*$ for the extremum of $x_3(t)$

$$t^* = \frac{-x_{2_0} + \tau u \left(1 + W_0\left(\tilde{x}\right)\right) - \tau x_{1_0}}{u} \tag{28}$$

with $\tilde{x} = \frac{u - x_{1_0}}{u} e^{\frac{\frac{x_{2_0}}{\tau} - u + x_{1_0}}{u}} > 0$. Inserting $t^*$ into $x_3(t)$ from Eq. (26) gives the extremum

$$\begin{aligned} x_3(t^*) = & \frac{1}{2\,u}\Big[\tau^2 u^2 - \tau^2 {x_{1_0}}^2 + 2\,u\,x_{3_0} - {x_{2_0}}^2 - 2\,\tau^2 u^2\, e^{\frac{x_{2_0} - \tau u + \tau x_{1_0} - \tau u W_0(\tilde{x})}{\tau u}} \\ & + \tau^2 u^2 W_0\left(\tilde{x}\right)^2 + 2\,\tau\,u\,x_{2_0} - 2\,\tau\,x_{1_0}\,x_{2_0} + 2\,\tau^2\,u\,x_{1_0}\, e^{\frac{x_{2_0} - \tau u + \tau x_{1_0} - \tau u W_0(\tilde{x})}{\tau u}}\Big]. \end{aligned} \tag{29}$$

According to Characterization 3.1, the level functions $g_3(x) \leq 0$ can be defined as listed in Eq. (A.44). Since the extremal time $t^*$ is constructed with the Lambert $W$ function, it is not possible to determine the feasible region of the level functions $g_3(x) \leq 0$ in the $x$-space algebraically as done in the previous sections and can thus only be formulated in terms of $t^*$. Still, $t^* \geq 0$ must hold, which can be used to determine the feasible region of the level functions $g_3(x) \leq 0$ by evaluating the $g_3$ function individually for each $x$.

According to Definition 3.1, the intersection of the $g_1$, $g_2$, and $g_3$ functions should lead to a maximal control invariant set $C_\infty$. However, beginning from dimensions of $n \geq 3$ the compatibility of the $g_i$ function needs to be checked. When the $g_i$ functions intersect, the normal vector condition must be satisfied to ensure the existence of a control input $u$ that keeps the trajectory within the set $C_\infty$. Here, a conflict between the $\underline{g}_1$ function and the $\overline{g}_3$ function, as well as a conflict between the $\overline{g}_1$ function and the $\underline{g}_3$ function arises. Therefore, the $g_{3_1}$ function is defined, which aims to prevent an intersection of $\underline{g}_1$ with $\overline{g}_3$ and $\overline{g}_1$ with $\underline{g}_3$ and to ensure that the extremal trajectory is at its extremum on $g_1$. Setting $x_1(t) = \overline{x}_1$ for $\underline{g}_{3_1}$ and $x_1(t) = \underline{x}_1$ for $\overline{g}_{3_1}$ and solving for $t$ gives the time $t^*$ for the extremum in $x_2$ at which $g_{3_1}(x)$ reaches $g_1(x)$ as

$$x_1 = u(1 - e^{-\frac{t}{\tau}}) + x_{1_0} e^{-\frac{t}{\tau}}, \quad (u, x_1) \in \{(\overline{u}, \overline{x}_1), (\underline{u}, \underline{x}_1)\} \quad . \tag{30}$$

Assuming $u \neq x_{1_0}$, the equation can be solved for $t^*$ as

$$t^* = -\tau \ln\left(\frac{x_1 - u}{x_{1_0} - u}\right), \quad (u, x_1) \in \{(\overline{u}, \overline{x}_1), (\underline{u}, \underline{x}_1)\} \quad . \tag{31}$$

Inserting $t^*$ into Eq. (26) gives the trajectory $\boldsymbol{x}(t)$ at the extremal time $t^*$ as

$$\begin{aligned} x_1(t^*) &= x_1 \\ x_2(t^*) &= x_{2_0} - \tau x_1 + \tau x_{1_0} - \tau u \ln\left(\tfrac{u - x_1}{u - x_{1_0}}\right) \\ x_3(t^*) &= x_{3_0} + \tau^2 x_1 - \tau^2 x_{1_0} + \tau^2 u \ln\left(\tfrac{u - x_1}{u - x_{1_0}}\right) \\ &\quad - \tau^2 x_{1_0} \ln\left(\tfrac{u - x_1}{u - x_{1_0}}\right) + \tfrac{\tau^2 u}{2}\left(\ln\left(\tfrac{u - x_1}{u - x_{1_0}}\right)\right)^2 \\ &\quad - \tau x_{2_0} \ln\left(\tfrac{u - x_1}{u - x_{1_0}}\right) \end{aligned} \quad , \quad (u, x_1) \in \left\{(\overline{u}, \overline{x}_1), (\underline{u}, \underline{x}_1)\right\}. \tag{32}$$

This is the extremal trajectory $\boldsymbol{x}(t^*)$ which has its extremum in $x_2$ at $g_1$. Next, to maximise the set $C_\infty$, the extremum in $x_3$ shall intersect with the boundary of the state constraints $\mathcal{X}$ after attaining $g_1$. Within $g_1$ the state $x_1$ is held constant such that the condition of $g_1$ is satisfied, i.e., $x_1(t) = \overline{x}_1$ for $\underline{g}_{3_1}$ and $x_1(t) = \underline{x}_1$ for $\overline{g}_{3_1}$. This can be done by setting $u = x_1$ in Eq. (23), which sets $\dot{x}_1 = 0$, preventing any change in $x_1$. With $\dot{x}_1 = 0$, Eq. (23) is reduced to a double integrator system:

$$\begin{aligned} \dot{x}_2(t) &= x_1(t) \\ \dot{x}_3(t) &= x_2(t) \end{aligned}, \quad x_1(t) \in \{\underline{x}_1, \overline{x}_1\} \tag{33}$$

The solution of the reduced system is given for a constant state $x_1 = \{\underline{x}_1, \overline{x}_1\}$ by

$$\begin{aligned} x_2(t) &= x_{2_0} + x_1 t \\ x_3(t) &= x_{2_0} t + x_{3_0} + x_1 \frac{t^2}{2} \end{aligned}, \quad x_1 \in \{\underline{x}_1, \overline{x}_1\}. \tag{34}$$

The extremal time $t_{23}^*$ is calculated by setting $x_2 = 0$ and is then inserted into the solution of the reduced system in Eq. (34) to obtain the extremum of $x_3(t)$:

$$x_3(t_{23}^*) = -\frac{x_{2_0}^2}{2x_1} + x_{3_0}, \quad x_1 \in \{\underline{x}_1, \overline{x}_1\} \tag{35}$$

$\boldsymbol{x}$ from Eq. (32) is inserted as initial states $\boldsymbol{x}_0$ into the solution of the reduced system in Eq. (35) to combine the extremal trajectory that has its extremum in $x_2$ at $g_1$ and commands the input $u \in \{\underline{u}, \overline{u}\}$ with the extremal trajectory that has its extremum in $x_3$ inside $g_1$ for $u = x_1$. This yields

$$\begin{aligned} x_3(t_{23}^*) =& x_{3_0} - \frac{\left(x_{2_0} - \tau x_1 + \tau x_{1_0} - \tau u \ln\left(\frac{u-x_1}{u-x_{1_0}}\right)\right)^2}{2x_{1_{\max}}} \\ & - \tau \ln\left(\frac{u-x_1}{u-x_{1_0}}\right)\left(x_{2_0} - \tau u\left(\ln\left(\frac{u-x_1}{u-x_{1_0}}\right) + \frac{u-x_{1_0}}{u-x_1} - 1\right)\right) \\ & + \tau^2 u\left(\ln\left(\frac{u-x_1}{u-x_{1_0}}\right) + \frac{u-x_{1_0}}{u-x_1} - \frac{1}{2}\ln^2\left(\frac{u-x_1}{u-x_{1_0}}\right) - 1\right) \\ & - \frac{\tau^2 x_1 (u-x_{1_0})\left(\ln\left(\frac{u-x_1}{u-x_{1_0}}\right) - \frac{u-x_1}{u-x_{1_0}} + 1\right)}{u-x_1}, \\ & (u, x_1) \in \left\{(\overline{u}, \overline{x}_1), (\underline{u}, \underline{x}_1)\right\} \end{aligned} \tag{36}$$

Using Characterization 3.1, the level functions $g_{3_1}(x) \leq 0$ can be defined as stated in Eq. (A.44). Intersecting the $g_i$ level functions gives the feasible initial states, or maximal control invariant set, $C_\infty$ for the Altitude Hold Mode:

$$C_\infty = \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \max \begin{bmatrix} \underline{g}_1(x_{1_0}, \overline{u}), \overline{g}_1(x_{1_0}, \underline{u}), \\ \underline{g}_2(x_{1_0}, x_{2_0}, \overline{u}), \overline{g}_2(x_{1_0}, x_{2_0}, \underline{u}), \\ \underline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}), \overline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}), \\ \underline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}), \overline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}) \end{bmatrix} \leq 0 \right\} \tag{37}$$

The full set of level functions for the considered system dynamics can be found in Section A.3. The maximal control invariant set $C_\infty$ is visualised in Fig. 7.

## 5.3 $\mathcal{K}$-Space - Control Law

The advantage of using the formulation of the Lambert $W$ function instead of a numerical calculation of $C_\infty$ is that the Lambert $W$ function is an analytical function and thus differentiable. This allows the calculation of gradients, which can be used to find the $\mathcal{K}$ space and to determine a valid linear control

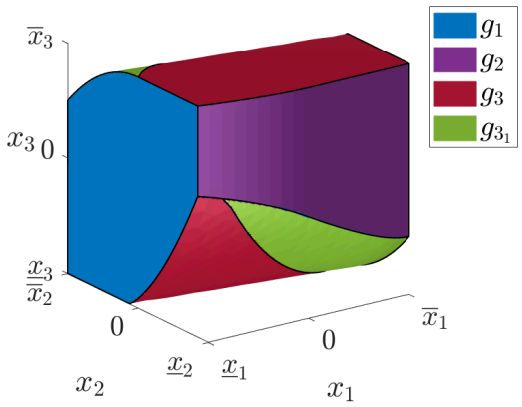



**Figure 7 Maximal control invariant set $C_\infty$ for the current system dynamics with the level functions $g_1$, $g_2$, $g_3$, and $g_{3_1}$.**

law $\boldsymbol{u} = \boldsymbol{k}^\top \boldsymbol{x}$ according to Definition 3.2. Because of the conflict between $g_1$ and $g_3$ mentioned in Section 5.2, a switching logic must be found for the control input $u$ to satisfy the newly defined level function $g_{3_1}$, which is defined to switch to $x_1 = \overline{x}_1$ for $\underline{g}_{3_1}$ and $x_1 = \underline{x}_1$ for $\overline{g}_{3_1}$ when the extremum at $g_1$ is reached (see Eq. (33)). This implies that $\dot{x}_1 = 0$ when attaining the extremum at $g_1$. With Eq. (23) and $\dot{x}_1 = 0$, the condition on $g_1$ is:

$$\begin{aligned} \dot{x}_1 &= -\frac{1}{\tau} x_1 + \frac{1}{\tau} u = 0 \\ \iff u &= x_1 \end{aligned} \tag{38}$$

This means that the control input $u$ must be equal to the state $x_1$ when the extremum at $g_1$ is reached, which can be formulated as a switching logic:

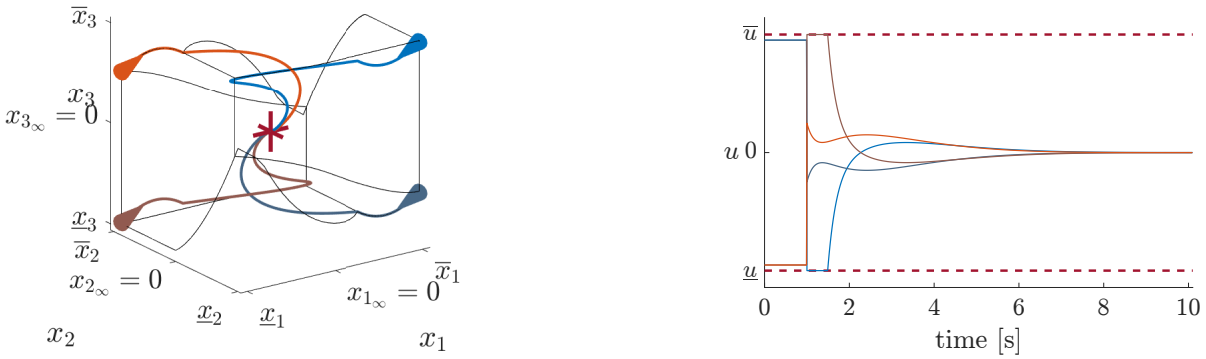
$$u_* = \begin{cases} x_1 & \text{if } x_1 = \underline{x}_1 \ \wedge \ x_2 > 0 \\ x_1 & \text{if } x_1 = \overline{x}_1 \ \wedge \ x_2 < 0 \\ \underline{u} & \text{if } u'_* < \underline{u} \\ \overline{u} & \text{if } u'_* > \overline{u} \\ u'_* & \text{otherwise} \end{cases} \tag{39}$$

with $u'_* = -\boldsymbol{k}^\top \boldsymbol{x}$ being the unsaturated linear control law. When $x_1$ is at a constraint boundary and $x_2$ has a sign that would drive $x_3$ further toward its bound, the input is set to $u_* = x_1$ (yielding $\dot{x}_1 = 0$) until $x_2$ crosses zero; otherwise the linear control law $u_* = u'_*$ is applied.

**Remark 5.1 (Chattering avoidance)** *The first-order lag in Eq.* (23) *ensures that $x_1(t)$ approaches a constraint boundary asymptotically, so the switch to $u_* = x_1$ is triggered at most once per excursion and no chattering occurs. For a digital implementation, the exact equality can be replaced by a boundary layer $|x_1 - \overline{x}_1| \leq \delta$ with $\delta \propto T_s/\tau$, analogous to the boundary-layer technique used in sliding-mode control to suppress chattering [22].*

The resulting trajectories and the control input $u_*$ are visualised in Fig. 8. Trajectories starting at the boundary of $C_\infty$ first move alongside the boundary of $C_\infty$ while $u_* = x_1$ until they attain the extremum at $g_1$, which is when $x_2 = 0$. At this point, the control input switches to $u_* = u'_*$, and the trajectory converges towards the origin.

The linear control gain $\boldsymbol{k} \in \mathcal{K}$ is defined inside the $\mathcal{K}$ space, which is defined as the intersection of the $\mathcal{K}_i$ spaces according to Definition 3.2 and formulated in Eq. (A.47). Since the switching logic covers




**(a)** **(b)**

**Figure 8 Trajectories in (a) starting at the boundary of $C_\infty$, moving on the boundary while $u_* = x_1$. Control input $u_*$ in (b).**

the regions in which it sets $u = x_1$ there exists no definition of $\mathcal{K}$ in these regions. The gradients used to calculate the $\mathcal{K}$ space can be found in Section A.3. The resulting set of linear inequalities is visualised in Fig. 9.

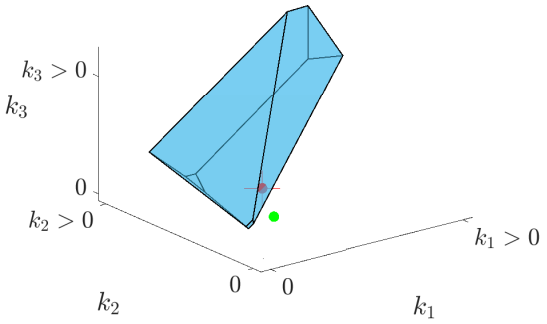



**Figure 9 Feasible $\mathcal{K}$-space of the intersected $\mathcal{K}$-space of $g_1$, $g_2$, $g_3$, and $g_{3_1}$ functions in blue with the solution of the Continuous-time Algebraic Riccati Equation (CARE) in green and the optimal linear control gain $\boldsymbol{k}$ in red.**

## 5.4 Simulation Results for the Altitude Hold Mode

In this section, a simulative study is shown in Figs. 10 and 11 which demonstrates that for a chosen $\boldsymbol{k}$ from the $\mathcal{K}$-space, the system drives always to the equilibrium if the initial states from the feasible set $C_\infty$ are taken. In the ALT mode, the vertical position $p_z$ is tracked. The input constraint and the state constraints are taken from Section 4.4. Additionally, the position $p_z = x_3$ is constrained with a value of $\overline{x}_3 = 7\ m$. Further, an actuator a time constant of $\tau = 0.5\ s$ is assumed. The linear control gain $\boldsymbol{k}$ is chosen from the $\mathcal{K}$-space of the system as $\boldsymbol{k}^\top = [\ 1.87 \quad 1.73 \quad 0.33\ ]$ It can be observed that the input- and state constraints are respected by the system and the trajectories converge towards the origin.

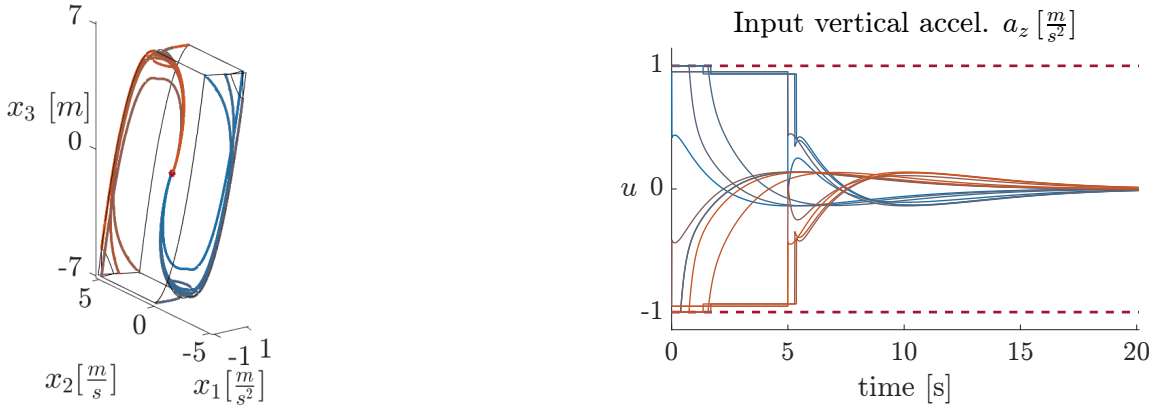



**Figure 10 Critical trajectories and control input $u$ of of the Altitude Hold Mode obtained through a linear control law.**

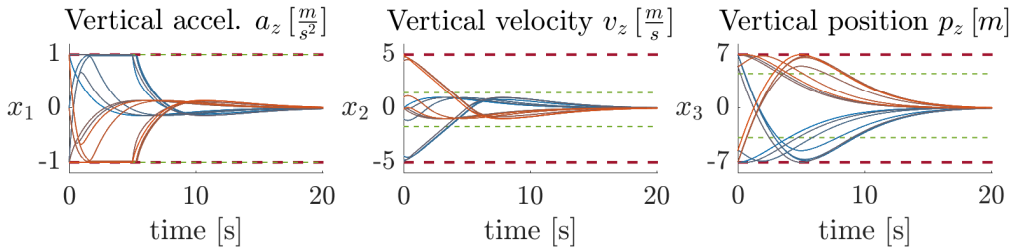



**Figure 11 System states of the Altitude Hold Mode obtained through a linear control law.**

# 6 Conclusion

In this work, the problem of input and state-constrained trajectories for helicopter upper modes is addressed. A method is developed to compute the maximal control invariant set analytically for a second-order system (VS mode) and a third-order system (ALT mode). The calculation of the invariant set provides the feasible set of initial states $C_\infty$ that guarantees the existence of a control law $u$ that respects the input and state constraints. Subsequently, a saturated linear control law $u$ is derived that steers the system to the origin. To ensure that the level functions defining the boundary of the maximal control invariant set $C_\infty$ do not conflict with each other, a switching logic was introduced for systems of order $n \geq 3$.

Since the presented method is based on the analytical solution of the invariant set, it is mathematically comprehensible and facilitates the documentation and certification of the control law. The main computational effort lies in the calculation of the maximal control invariant set $C_\infty$ and the linear control gain vector $\boldsymbol{k}$, both of which can be computed offline. Then, only the linear control gain vector $\boldsymbol{k}$ and the feasible initial states $C_\infty$ need to be retrieved online from a lookup table. This enables the implementation of the control law on embedded flight control computers or other real-time systems.

In summary, this work demonstrates that the use of the maximal control invariant set $C_\infty$ in combination with a saturated linear control input $u$ (and possible switching in $u$) ensures trajectory stability and guarantees that the input constraints $\mathcal{U}$ and state constraints $\mathcal{X}$ are respected. States outside $C_\infty$ correspond to flight conditions for which no admissible input can prevent constraint violation – a physical, not algorithmic, limitation of the vehicle – while the control law itself degrades gracefully (see Remark 3.3).

## Outlook

Several promising directions for future research emerge from this work.

### *Generalisation to other vehicle types*

Due to the general analytical formulation of the method and the use of low-order models it is possible to apply the method to other systems that exhibit similar dynamics, such as drones, rockets, satellites, or other vehicles.

### *Extension to higher-order systems*

For the helicopter, most models that directly control the actuators are of order $n \leq 3$, and the presented method is therefore sufficient. However, it would be interesting to also consider higher-order modes with more states using the method presented in this work. For integrator systems, the analytical computation of the invariant set can be extended to higher orders, e.g., the quadruple integrator system. Nevertheless, the intersection of the $g_i$ level functions becomes more complex for higher-order systems. As the system becomes four-dimensional in the example of the quadruple integrator, it becomes more difficult to visualise $C_\infty$. Additionally, if a low-pass filter is present in the model, an analytical solution of the invariant set is no longer possible for orders $n > 3$, since the differential equation becomes transcendental and can no longer be described with the Lambert $W$ function. Therefore, for higher-order systems, it may be more practical to compute the invariant set numerically. The numerical method could then be validated with the analytical solution for orders $n \leq 3$, and the approach could subsequently be extended to higher orders. The gradients on the numerically calculated set must also be computed numerically. A disadvantage of the numerical approach is that the structure of the $g$-functions is no longer known. Consequently, it is unclear where the numerical method has resolved conflicts between $g$-functions and where a switching logic in $u$ is therefore necessary. This could be addressed by developing a numerical method similar to that used by Multi-Parametric Toolbox 3 (MPT3), but with the additional feature of identifying conflicting $g$-functions.

### *Robustness under model mismatch*

The constraint guarantees established in this work rely on the perfect-model assumption ($\hat{P} = P$, see Assumption 3.1). When the plant deviates from the model, the robust controller in the full MRC architecture compensates for this mismatch. Quantifying the resulting tracking-error bound $\epsilon$ from the frozen robust controller and computing a correspondingly contracted invariant set $C_\infty$ - as outlined in Remark 3.1 - is an important direction for future work.

### *Industrial deployment*

The offline nature of the invariant-set computation and the simplicity of the online control law make the method well suited for integration into existing certified AFCS architectures, where it can serve as the trajectory-planning layer within the MRC framework for helicopters as well as for emerging urban air mobility platforms.

### *Online computation*

Due to real-time computation constraints on the helicopter, the method calculates the $C_\infty$ offline. However, with higher computational power, for example on a new AFCS generation or a drone, it may be possible to compute the invariant set online.

# Appendix

## A.1 $\mathcal{K}$-Space of the Double Integrator

$$\mathcal{K} = \left\{ \boldsymbol{k} \in \mathbb{R}^2 \ : \begin{array}{l} k_1\underline{x}_1 + k_2 x_2 \ \leq 0, \ \ \forall x_2 \in [\underline{x}_{2_0}, \overline{x}_2] \cap \\ -k_1 x_1 - k_2 x_2 \leq -\overline{u}, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \underline{g}_2(\boldsymbol{x}_{1:2}, \overline{u}) = 0 \right\} \cap \\ -k_1\overline{x}_1 - k_2 x_2 \leq 0, \ \ \forall x_2 \in [\underline{x}_2, \overline{x}_{2_0}] \cap \\ k_1 x_1 + k_2 x_2 \ \leq \underline{u}, \ \ \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \overline{g}_2(\boldsymbol{x}_{1:2}, \underline{u}) = 0 \right\} \end{array} \right\} \tag{A.40}$$

## A.2 Lambert $W$ Function

Assume a problem is formulated in the structure $we^w = z$, where the goal is to find $w$. Such transcendental equations can be solved by the Lambert $W$ function, which is defined as $w = W_k(z)$, where $k$ is the branch of the Lambert $W$ function. Assuming that $w$ and $z$ are real numbers, the equation can be reformulated as:

$$ye^y = x, \quad (x, y) \in \left\{ (x, y) \in \mathbb{R}^2 : x \geq -\frac{1}{e} \right\} \tag{A.41}$$

For $x \geq 0$, the solution is given by the $k = 0$ branch of the Lambert $W$ function:

$$y = W_0(x) \quad \text{if} \quad x \geq 0 \tag{A.42}$$

For $-\frac{1}{e} \leq x < 0$, the solution is given by both branches $k = 0$ and $k = 1$ of the Lambert $W$ function:

$$y = \{W_0(x), W_1(x)\} \quad \text{if} \quad -\frac{1}{e} \leq x < 0 \tag{A.43}$$

## A.3 Double Integrator with PT1

For the sake of completeness, the $g_i$ functions and the gradients are given for the double integrator with PT1.

$$
\begin{aligned}
\underline{g}_1(x_{1_0}) &= \underline{x}_1 - x_{1_0} \leq 0 \\
\overline{g}_1(x_{1_0}) &= x_{1_0} - \overline{x}_1 \leq 0 \\
\underline{g}_2(x_{1_0}, x_{2_0}, \overline{u}) &= \begin{cases} \underline{x}_2 - x_{2_0} - \tau\overline{u}\left(\ln\left(1 - \frac{x_{1_0}}{\overline{u}}\right) + \frac{x_{1_0}}{\overline{u}}\right) & \leq 0, \quad \text{if } x_{1_0} < 0 \\ \underline{x}_2 - x_{2_0} & \leq 0, \quad \text{if } x_{1_0} \geq 0 \end{cases} \\
\overline{g}_2(x_{1_0}, x_{2_0}, \underline{u}) &= \begin{cases} x_{2_0} - \overline{x}_2 & \leq 0, \quad \text{if } x_{1_0} \leq 0 \\ x_{2_0} - \overline{x}_2 + \tau\underline{u}\left(\ln\left(1 - \frac{x_{1_0}}{\underline{u}}\right) + \frac{x_{1_0}}{\underline{u}}\right) & \leq 0, \quad \text{if } x_{1_0} > 0 \end{cases} \\
\underline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}) &= \begin{cases} \underline{x}_3 - \frac{1}{2\overline{u}}\left[\tau^2\overline{u}^2 - \tau^2 {x_{1_0}}^2 + 2\,\overline{u}\,x_{3_0} - {x_{2_0}}^2\right. \\ \quad -2\,\tau^2\overline{u}^2\, e^{\frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}}} \\ \quad +\tau^2\overline{u}^2\, W_0(\tilde{x})^2 + 2\,\tau\,\overline{u}\,x_{2_0} - 2\,\tau\,x_{1_0}\,x_{2_0} \\ \quad \left.+2\,\tau^2\,\overline{u}\,x_{1_0}\, e^{\frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}}}\right], & \text{if } t^* \geq 0 \\ \underline{x}_3 - x_{3_0}, & \text{if } t^* < 0 \end{cases} \\
\overline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}) &= \begin{cases} -\overline{x}_3 + \frac{1}{2\underline{u}}\left[\tau^2\underline{u}^2 - \tau^2 {x_{1_0}}^2 + 2\,\underline{u}\,x_{3_0} - {x_{2_0}}^2\right. \\ \quad -2\,\tau^2\underline{u}^2\, e^{\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau\underline{u} W_0(\tilde{x})}{\tau\underline{u}}} \\ \quad +\tau^2\underline{u}^2\, W_0(\tilde{x})^2 + 2\,\tau\,\underline{u}\,x_{2_0} - 2\,\tau\,x_{1_0}\,x_{2_0} \\ \quad \left.+2\,\tau^2\,\underline{u}\,x_{1_0}\, e^{\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau\underline{u} W_0(\tilde{x})}{\tau\underline{u}}}\right], & \text{if } t^* \geq 0 \\ x_{3_0} - \overline{x}_3, & \text{if } t^* < 0 \end{cases} \\
\underline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}) &= \begin{cases} \underline{x}_3 - x_{3_0} \leq 0, & \text{if } t^* < 0 \\ \underline{x}_3 - x_{3_0} + \frac{\left(x_{2_0} - \tau\overline{x}_1 + \tau x_{1_0} - \tau\overline{u}\ln\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right)\right)^2}{2x_{1_{\max}}} \\ \quad +\tau\ln\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right)\left(x_{2_0} - \tau\overline{u}\left(\ln\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right) + \frac{\overline{u} - x_{1_0}}{\overline{u} - \overline{x}_1} - 1\right)\right) \\ \quad -\tau^2\overline{u}\left(\ln\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right) + \frac{\overline{u} - x_{1_0}}{\overline{u} - \overline{x}_1} - \frac{1}{2}\ln^2\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right) - 1\right) \\ \quad +\frac{\tau^2\overline{x}_1(\overline{u} - x_{1_0})\left(\ln\left(\frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}}\right) - \frac{\overline{u} - \overline{x}_1}{\overline{u} - x_{1_0}} + 1\right)}{\overline{u} - \overline{x}_1} \leq 0, & \text{if } t^* \geq 0 \end{cases} \\
\overline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}) &= \begin{cases} x_{3_0} - \overline{x}_3 \leq 0, & \text{if } t^* < 0 \\ x_{3_0} - \frac{\left(x_{2_0} - \tau\underline{x}_1 + \tau x_{1_0} - \tau\underline{u}\ln\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right)\right)^2}{2x_{1_{\max}}} \\ \quad -\tau\ln\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right)\left(x_{2_0} - \tau\underline{u}\left(\ln\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right) + \frac{\underline{u} - x_{1_0}}{\underline{u} - \underline{x}_1} - 1\right)\right) \\ \quad +\tau^2\underline{u}\left(\ln\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right) + \frac{\underline{u} - x_{1_0}}{\underline{u} - \underline{x}_1} - \frac{1}{2}\ln^2\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right) - 1\right) \\ \quad -\frac{\tau^2\underline{x}_1(\underline{u} - x_{1_0})\left(\ln\left(\frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}}\right) - \frac{\underline{u} - \underline{x}_1}{\underline{u} - x_{1_0}} + 1\right)}{\underline{u} - \underline{x}_1} - \overline{x}_3 \leq 0, & \text{if } t^* \geq 0 \end{cases}
\end{aligned}
\tag{A.44}
$$

with the gradients

$$
\begin{aligned}
\nabla \underline{g}_1(x_{1_0}) &= \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \\
\nabla \overline{g}_1(x_{1_0}) &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
\nabla \underline{g}_2(x_{1_0}, x_{2_0}, \overline{u}) &= \begin{cases} \begin{bmatrix} \frac{\tau x_{1_0}}{\overline{u} - x_{1_0}} \\ -1 \\ 0 \end{bmatrix}, & \text{if } x_{1_0} < 0 \\ \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, & \text{if } x_{1_0} \geq 0 \end{cases} \\
\nabla \overline{g}_2(x_{1_0}, x_{2_0}, \underline{u}) &= \begin{cases} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, & \text{if } x_{1_0} \leq 0 \\ \begin{bmatrix} -\frac{\tau x_{1_0}}{\underline{u} - x_{1_0}} \\ 1 \\ 0 \end{bmatrix}, & \text{if } x_{1_0} > 0 \end{cases} \\
\nabla \underline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}) &= \begin{cases} \begin{bmatrix} \frac{1}{\overline{u}(W_0(\tilde{x})+1)\left(\overline{u} - x_{1_0}\right)} \left[ \tau^2 \overline{u} x_{1_0} - \tau^2 x_{1_0}^2 - \tau^2 x_{1_0}^2 W_0(\tilde{x}) \right. \\ \quad + \tau^2 x_{1_0}^2 \exp\left( \frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}} \right) \\ \quad + \tau\overline{u} x_{2_0} - \tau x_{1_0} x_{2_0} + \tau^2 \overline{u} x_{1_0} W_0(\tilde{x}) \\ \quad - \tau^2 \overline{u} x_{1_0} \exp\left( \frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}} \right) \\ \left. \quad + \tau^2 \overline{u} x_{1_0} W_0(\tilde{x})^2 + \tau\overline{u} x_{2_0} W_0(\tilde{x}) - \tau x_{1_0} x_{2_0} W_0(\tilde{x}) \right] \\ \frac{1}{\overline{u}(W_0(\tilde{x})+1)} \left[ x_{2_0} - \tau\overline{u} + \tau x_{1_0} + x_{2_0} W_0(\tilde{x}) - \tau\overline{u} W_0(\tilde{x}) \right. \\ \quad + \tau x_{1_0} W_0(\tilde{x}) + \tau\overline{u} \exp\left( \frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}} \right) \\ \left. \quad - \tau x_{1_0} \exp\left( \frac{x_{2_0} - \tau\overline{u} + \tau x_{1_0} - \tau\overline{u} W_0(\tilde{x})}{\tau\overline{u}} \right) - \tau\overline{u} W_0(\tilde{x})^2 \right] \\ -1 \end{bmatrix}, & \text{if } t^* \geq 0 \\ \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, & \text{if } t^* < 0 \end{cases}
\end{aligned}
\tag{A.45}
$$

$$
\nabla \overline{g}_3(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}) = \begin{cases} \begin{bmatrix} -\frac{1}{\underline{u}(W_0(\tilde{x})+1)\left(\underline{u}-x_{1_0}\right)} \left[\tau^2 \underline{u} x_{1_0} - \tau^2 x_{1_0}^2 - \tau^2 x_{1_0}^2 W_0\left(\tilde{x}\right) \right. \\ \quad + \tau^2 x_{1_0}^2 \exp\left(\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau \underline{u} W_0(\tilde{x})}{\tau \underline{u}}\right) \\ \quad + \tau \underline{u} x_{2_0} - \tau x_{1_0} x_{2_0} + \tau^2 \underline{u} x_{1_0} W_0\left(\tilde{x}\right) \\ \quad - \tau^2 \underline{u} x_{1_0} \exp\left(\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau \underline{u} W_0(\tilde{x})}{\tau \underline{u}}\right) \\ \quad \left. + \tau^2 \underline{u} x_{1_0} W_0\left(\tilde{x}\right)^2 + \tau \underline{u} x_{2_0} W_0\left(\tilde{x}\right) - \tau x_{1_0} x_{2_0} W_0\left(\tilde{x}\right)\right] \\ -\frac{1}{\underline{u}(W_0(\tilde{x})+1)} \left[x_{2_0} - \tau \underline{u} + \tau x_{1_0} + x_{2_0} W_0\left(\tilde{x}\right) - \tau \underline{u} W_0\left(\tilde{x}\right) \right. \\ \quad + \tau x_{1_0} W_0\left(\tilde{x}\right) + \tau \underline{u} \exp\left(\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau \underline{u} W_0(\tilde{x})}{\tau \underline{u}}\right) \\ \quad \left. - \tau x_{1_0} \exp\left(\frac{x_{2_0} - \tau\underline{u} + \tau x_{1_0} - \tau \underline{u} W_0(\tilde{x})}{\tau \underline{u}}\right) - \tau \underline{u} W_0\left(\tilde{x}\right)^2\right] \\ 1 \end{bmatrix}, & \text{if } t^* \geq 0 \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, & \text{if } t^* < 0 \end{cases}
$$

$$
\nabla \underline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \overline{u}) = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, & \text{if } t^* < 0 \\ \begin{bmatrix} -\frac{\tau}{\overline{x}_1(\overline{u}-x_{1_0})} \left[x_{1_0} x_{2_0} - x_{2_0} \overline{x}_1 + \tau x_{1_0}^2 - \tau x_1 x_{1_0} \right. \\ \quad \left. - \tau \overline{u} x_{1_0} \ln\left(\frac{\overline{u}-x_1}{\overline{u}-x_{1_0}}\right) + \tau x_{1_0} \overline{x}_1 \ln\left(\frac{\overline{u}-x_1}{\overline{u}-x_{1_0}}\right)\right] \\ \frac{1}{\overline{x}_1} \left[x_{2_0} - \tau x_1 + \tau x_{1_0} - \tau \overline{u} \ln\left(\frac{\overline{u}-x_1}{\overline{u}-x_{1_0}}\right) \right. \\ \quad \left. + \tau \overline{x}_1 \ln\left(\frac{\overline{u}-x_1}{\overline{u}-x_{1_0}}\right)\right] \\ -1 \end{bmatrix}, & \text{if } t^* \geq 0 \end{cases}
$$

$$
\nabla \overline{g}_{3_1}(x_{1_0}, x_{2_0}, x_{3_0}, \underline{u}) = \begin{cases} \begin{bmatrix} \frac{\tau}{\underline{x}_1(\underline{u}-x_{1_0})} \left[x_{1_0} x_{2_0} - x_{2_0} \underline{x}_1 + \tau x_{1_0}^2 - \tau x_1 x_{1_0} \right. \\ \quad \left. - \tau \underline{u} x_{1_0} \ln\left(\frac{\underline{u}-x_1}{\underline{u}-x_{1_0}}\right) + \tau x_{1_0} \underline{x}_1 \ln\left(\frac{\underline{u}-x_1}{\underline{u}-x_{1_0}}\right)\right] \\ -\frac{1}{\underline{x}_1} \left[x_{2_0} - \tau x_1 + \tau x_{1_0} - \tau \underline{u} \ln\left(\frac{\underline{u}-x_1}{\underline{u}-x_{1_0}}\right) \right. \\ \quad \left. + \tau \underline{x}_1 \ln\left(\frac{\underline{u}-x_1}{\underline{u}-x_{1_0}}\right)\right] \\ 1 \end{bmatrix}, & \text{if } t^* \geq 0 \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, & \text{if } t^* < 0 \end{cases} \tag{A.46}
$$

The $\mathcal{K}$-space of the double integrator system with PT1 is defined as:

$$\mathcal{K} = \left\{ \boldsymbol{k} \in \mathbb{R}^3 \ : \begin{array}{l} \nabla \underline{g}_1^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \underline{g}_1(\boldsymbol{x}_1, \overline{u}) = 0 \right\} \cap \\ \nabla \underline{g}_2^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \underline{g}_2(\boldsymbol{x}_{1:2}, \overline{u}) = 0 \right\} \cap \\ \nabla \underline{g}_3^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \underline{g}_3(\boldsymbol{x}_{1:3}, \overline{u}) = 0 \right\} \cap \\ \nabla \underline{g}_{3_1}^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \underline{g}_{3_1}(\boldsymbol{x}_{1:3}, \overline{u}) = 0 \right\} \cap \\ \nabla \overline{g}_1^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \overline{g}_1(\boldsymbol{x}_1, \underline{u}) = 0 \right\} \cap \\ \nabla \overline{g}_2^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \overline{g}_2(\boldsymbol{x}_{1:2}, \underline{u}) = 0 \right\} \cap \\ \nabla \overline{g}_3^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \overline{g}_3(\boldsymbol{x}_{1:3}, \underline{u}) = 0 \right\} \cap \\ \nabla \overline{g}_{3_1}^\top \dot{\boldsymbol{x}} \leq 0, \forall \boldsymbol{x} \in \partial C_\infty \cap \left\{ \boldsymbol{x} \in \mathbb{R}^3 : \overline{g}_{3_1}(\boldsymbol{x}_{1:3}, \underline{u}) = 0 \right\} \end{array} \right\} \tag{A.47}$$

# Declaration of Use of Artificial Intelligence

Artificial intelligence was not used in the work presented.

# References


[1] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2020. ISBN: 9780975937754.

[2] E. F. Camacho and C. Bordons. *Model Predictive control*. Springer London, 2007. ISBN: 9780857293985. doi: 10.1007/978-0-85729-398-5.

[3] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, Jan. 2017. ISSN: 0005-1098. doi: 10.1016/j.automatica.2016.08.013.

[4] A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–419, Mar. 1998. ISSN: 0018-9286. doi: 10.1109/9.661611.

[5] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, Apr. 2009. ISSN: 0005-1098. doi: 10.1016/j.automatica.2008.11.017.

[6] Keng Peng Tee and Shuzhi Sam Ge. Control of nonlinear systems with full state constraint using a barrier lyapunov function. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, page 8618–8623. IEEE, Dec. 2009. doi: 10.1109/cdc.2009.5400484, http://dx.doi.org/10.1109/CDC.2009.5400484.

[7] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, Nov. 1999. ISSN: 0005-1098. doi: 10.1016/s0005-1098(99)00113-2.

[8] Ludvig Doeser, Petter Nilsson, Aaron D. Ames, and Richard M. Murray. Invariant sets for integrators and quadrotor obstacle avoidance. In *2020 American Control Conference (ACC)*, page 3814–3821. IEEE, July 2020. doi: 10.23919/acc45564.2020.9147872, http://dx.doi.org/10.23919/ACC45564.2020.9147872.

[9] W. Schumacher. Erweiterte methoden der regelungstechnik. https://www.tu-braunschweig.de/fileadmin/Redaktionsgruppen/Institute_Fakultaet_5/IFR/Dateien_RT/Erweiterte_Methoden_der_RT/Skript_EMDR.pdf, 2019. [Accessed 15-11-2024].

[10] M.B. Tischler. *Practical Methods for Aircraft and Rotorcraft Flight Control Design: An Optimization-based Approach*. AIAA education series. American Institute of Aeronautics and Astronautics, Incorporated, 2017. ISBN: 9781624104435.

[11] MIL-F-83300, Military Specification: Flying Qualities of Piloted V/STOL Aircraft. http://everyspec.com/MIL-SPECS/MIL-SPECS-MIL-F/MIL-F-83300_17560/, 1970. [Accessed 26-09-2025].

[12] Handling Qualities Requirements for Military Rotorcraft. Aeronautical Design Standard. ADS-33E, 2000. https://www.avmc.army.mil/Portals/51/Documents/TechData%20PDF/ads33.pdf. [Accessed 26-09-2025].

[13] Requirements for Rotorcraft Vibration Specifications, Modeling and Testing. Aeronautical Design Standard. ADS-27, 2006. https://www.avmc.army.mil/Portals/51/Documents/TechData%20PDF/ADS27ASP.pdf. [Accessed 23-10-2025].

[14] Peter Ortner and Luigi del Re. Predictive control of a diesel engine air path. *IEEE Transactions on Control Systems Technology*, 15(3):449–456, May 2007. ISSN: 1063-6536. doi: 10.1109/tcst.2007.894638.

[15] Elmer Gilbert and Ilya Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12):2063–2073, Dec. 2002. ISSN: 0005-1098. doi: 10.1016/s0005-1098(02)00135-8.

[16] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Springer International Publishing, 2015. ISBN: 9783319179339. doi: 10.1007/978-3-319-17933-9.

[17] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. http://control.ee.ethz.ch/~mpt.

[18] SC-205. DO-331 - Model Based Development and Verification Supplement to DO-178C and DO-278A. https://my.rtca.org/productdetails?id=a1B36000001IcfiEAC, 2011. [Accessed 28-09-2025].

[19] Ulrich Butter. *Hubschrauber Flugmechanik und -Flugregelung*. Institut für Flugmechanik und Flugregelung Universität Stuttgart, 2013.

[20] Julian Schließus. Design of a constrained controlled trajectory for a helicopter. Master's thesis, iFR, University of Stuttgart, 2025.

[21] Mitio Nagumo. Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559, 1942.

[22] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991.