Chinese Journal of Aeronautics, (2021), **xxx**(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics

Patrick PIPREK [a], **Haichao HONG** [a,*], **Florian HOLZAPFEL** [a]

[a] *Institute of Flight System Dynamics, Technical University of Munich, Garching D-85748, Germany*

Received 28 April 2021; revised 28 September 2021; accepted 28 October 2021

KEYWORDS

LTV error dynamics;
LTV stability;
Optimal control-based LTV stabilization;
Path-following error controller;
Trajectory generation;
Trajectory optimization

Abstract This study is dedicated to the development of a direct optimal control-based algorithm for trajectory optimization problems that accounts for the closed-loop stability of the trajectory tracking error dynamics already during the optimization. Consequently, the trajectory is designed such that the Linear Time-Varying (LTV) dynamic system, describing the controller's error dynamics, is stable, while additionally the desired optimality criterion is optimized and all enforced constraints on the trajectory are fulfilled. This is achieved by means of a Lyapunov stability analysis of the LTV dynamics within the optimization problem using a time-dependent, quadratic Lyapunov function candidate. Special care is taken with regard to ensuring the correct definiteness of the ensuing matrices within the Lyapunov stability analysis, specifically considering a numerically stable formulation of these in the numerical optimization. The developed algorithm is applied to a trajectory design problem for which the LTV system is part of the path-following error dynamics, which is required to be stable. The main benefit of the proposed scheme in this context is that the designed trajectory trades-off the required stability and robustness properties of the LTV dynamics with the optimality of the trajectory already at the design phase and thus, does not produce unstable optimal trajectories the system must follow in the real application.

© 2021 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The optimal control-based design of trajectories is widely used in engineering applications.[1–3] This paper investigates an efficient trajectory design by optimization that additionally accounts for the closed-loop stability and robustness of a trajectory tracking error controller, described by Linear Time-Varying (LTV) dynamics, required in real application scenarios.[4,5] Often, the trajectory and the feedback controller are designed separately. This is generally not ideal, as it may require the optimal trajectory to be conservative, i.e., in this context a "sub-optimal" trajectory resulting e.g., by reducing the allowed value-range of the constraints, for the controller to follow it. On the other hand, if the trajectory is very aggressive, i.e., at its theoretical optimum obtained from an optimization with the least amount of constraints and limitations, the controller may be required to be conservatively designed to control the system in a stable manner. Thus, compared to designing trajectory and controller independent of each other, designing them in a single setup facilitates the performance of the designed optimal trajectory and increases the

* Corresponding author.
E-mail address: haichao.hong@tum.de (H. HONG).

https://doi.org/10.1016/j.cja.2021.10.031
1000-9361 © 2021 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

Uncorrected Proof

availability of trajectories in the real application. This is based on the fact that the optimizer only returns a solution if the setup can be solved with all its constraints, which contain both the physical model (like acceleration limits) as well the controller constraints (like stability and gain magnitude) in the single setup. In contrast, if the controller is designed independent of the trajectory, further (on-board) testing by e.g., simulations, may be required to ensure that only trajectories that do not de-stabilize the controller are commanded. Thus, to formulate the single setup and ensure feasible trajectories, the error dynamics used in designing the feedback controller, which are often of a LTV system type [4,5], must be stabilized using suitable constraints already inside the trajectory optimization problem.

Thus, this study is dedicated to the development of a direct optimal control-based algorithm (i.e., gradient-based) for trajectory optimization problems that directly accounts for the stability of a dependent LTV dynamic system by means of Lyapunov stability. Consequently, the trajectory is designed such that the incorporated LTV dynamics are ensured to be stable and fulfill specified robustness features, based on the convergence rate of the Lyapunov function, by designing feedback gains in the optimization, while additionally the desired optimality criterion is optimized and all enforced constraints on the trajectory are fulfilled. This automatically provides a trade-off between the optimality of the trajectory and the required stability of the LTV system by designing both parts dependent on each other.

In general, assessing the stability of LTV systems was always of paramount importance in Refs. [6–8]. Compared to well-known Linear Time-Invariant (LTI) controller design, it is no longer given by simply checking the eigenvalues of the state matrix.[6] Therefore, methods often avoid the design of controllers for LTV systems, but rather design the controller by gain scheduling methods, i.e., designing the gains for LTI systems at a sufficiently dense grid of parameter sets.[9] Although this approach generally yields good results, a drawback is that it only produces stable controllers if the time-dependent changes are sufficiently slow, i.e., the problem is in a quasi-steady state.[9] As trajectory optimization often results in aggressive maneuvers and thus, fast time-dependent changes, this assumption may no longer be valid.

To cope with this issue, researchr investigated optimal control and especially convex trajectory optimization for LTV systems. A review of common methods is given in Ref. [10]. Here, specifically model predictive control algorithms play an important role: For instance, Ref. [11] introduces an algorithm to represent a nonlinear dynamic model using sequential linear parameter-varying representations that are then used within the convex optimization. By this, feasibility and stability of the controller can be ensured. In this context, Ref. [12] introduces further methodologies to cope with constrained optimization of linear parameter-varying systems. Finally, Ref. [13] introduces a method for output feedback controller design based on parameter-dependent Lyapunov functions.

Considering the literature review, it is clear that stabilizing LTV systems, especially for error controllers as well as designing feedback controller gains for these, is of high importance. Still, studying the impact of LTV systems already in the trajectory design phase has not yet been thoroughly researched. Therefore, this study aims at bridging this gap by considering the stability of LTV systems already in the trajectory design phase. In addition to basic stability, it is also possible to enforce desired convergence and robustness requirements on the Lyapunov function, and thus the controller, as constraints in the optimization problem. Here, it is important to stress that this study does not try to invent new methods to stabilize LTV systems, but uses well-established stability concepts as a baseline to develop a design procedure for trajectory optimization accounting for the stabilization of a dependent LTV system. Generally, this goal is achieved by introducing the LTV dynamics within the trajectory optimization problem, which is used to calculate the desired, feedforward trajectory that the dynamic system should follow. By incorporating these LTV dynamics and imposing constraints to address stability, it is ensured that the trajectory design is already ensuring stable operation by itself. Thus, the feedback controller, which can be designed simultaneously, is not required in the real application to stabilize an unstable designed trajectory but can focus on disturbance rejection and coping with non-modeled dynamics. Furthermore, by incorporating the controller design within the trajectory generation by means of Lyapunov stability analysis, it is also possible to generalize the results of feedback controller gain design from the open left complex plane, as e.g., done in gain scheduling applications,[9] to the full complex plane. This is due to the fact that LTV systems do not have such a limiting requirement. Still, stability of the system within assured convergence bounds is ensured. Even further, these can be used to improve the robustness of the solution by introducing them as constraints in the optimization problem. By this, it can be assured that the Lyapunov function and its derivative follow e.g., a minimal convergence rate and thus, the error reduces with at least this rate. In this context, special care is taken in the proposed method with regard to ensuring the correct definiteness of the matrices within the Lyapunov stability analysis, specifically considering a numerically stable formulation for the gradient-based optimization.

The contributions of the presented work can therefore be summarized as:

(1) An optimal control-based trajectory design method is proposed, which accounts for the stability of a tracking controller described by LTV dynamics already within the optimization problem. This provides a trade-off between the required stability of the controller and the optimality of the trajectory.
(2) The closed-loop stability of the trajectory tracking error controller, i.e., the LTV-type error dynamics, is already guaranteed in the design phase by appropriately designing a trajectory and the associated feedback gains automatically in an optimal manner.
(3) The sufficient conditions of the desired stability are derived based on the mathematical properties of the Lyapunov method, which are implemented as simple inequality constraints in the optimal control problem. This reduces the overall design complexity, leading to a very pragmatic and effective design process.
(4) The closed-loop robustness, in the sense of e.g., a minimal convergence rate of the Lyapunov function and its derivative, and by this consequently the tracking error, is ensured using constraints in the optimization problem. By this, a desired performance of the controller can be

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

achieved in connection with the designed optimal trajectory.

To introduce the mentioned concepts, this study is organized as follows: Section 2 introduces the general idea of stability analysis for LTV systems. These principles are used in Section 3 to formulate suitable constraints within a gradient-based trajectory optimization problem. Following, Section 4 gives an application example: Here, Section 4.1 gives an overview on the LTV error dynamics used in this study to illustrate the introduced stability concept within trajectory optimization. Building on these error dynamics, Section 4.2 gives an overview of methods to stabilize them by means of feedback control. Furthermore, the dynamic model as well as the trajectory optimization problem formulation with stabilization of the feedback controller is specified in Section 4.3. The results of the trajectory optimization problem as well as an analysis of the controller performance is shown in Sections 4.4 and 4.5, respectively. Concluding, Section 5 gives some remarks and an outlook on future work.

2. Stability analysis of linear time-varying systems

Generally, a LTV system, with state vector $\boldsymbol{e}(t) \in \mathbb{R}^n$, is defined in this study as follows:

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{A}(\boldsymbol{\theta}(t)) \cdot \boldsymbol{e}(t) \tag{1}$$

Here, the piecewise continuous system matrix $\boldsymbol{A}(\boldsymbol{\theta}(t)) \in \mathbb{R}^{n\times n}$ is depending on time-varying (symbolized by (t)) parameters $\boldsymbol{\theta}(t) \in \Theta \subseteq \mathbb{R}^{\theta}$, where $\Theta$is the set of all feasible parameter values. Take into account that for stability analysis only the homogeneous system, i.e., without control inputs must be considered. This is reasonable as long as the inputs are bounded, which can be assumed specifically in the context of constrained optimization as applied in this study.

As the state matrix in Eq. (1) is time-varying due to the parameter dependence, a solution based on a time-independent Lyapunov equation,[14] ensuring stability, is normally difficult to find and, if found, very conservative. Therefore, it is common to extend the analysis to a formulation based on a quadratic, time-dependent Lyapunov function candidate as follows:[14]

$$V(t) = \frac{1}{2} \cdot \boldsymbol{e}(t)^{\mathsf{T}} \cdot \boldsymbol{P}(t) \cdot \boldsymbol{e}(t) > 0, \quad \forall \boldsymbol{e}(t) \setminus \{\boldsymbol{0}\}, \quad \boldsymbol{P}(t) \succ 0 \tag{2}$$

Here, $\boldsymbol{P}(t) \in \mathbb{P} \subseteq \mathbb{R}^{n\times n} \succ 0$ is a positive definite (symbolized in this study by $\succ 0$), real design matrix specified by the user, where $\mathbb{P}$denotes its feasible set. For the sake of simplicity, it is assumed in this study that the matrix is symmetric, i.e., $\boldsymbol{P}(t) = \boldsymbol{P}(t)^{\mathsf{T}}$.

The stability is then analyzed by considering the derivative of the Lyapunov function candidate, incorporating the homogeneous LTV system in Eq. (2), which must be strictly negative:

$$\begin{aligned} \dot{V}(t) &= \dot{\boldsymbol{e}}(t)^{\mathsf{T}} \cdot \boldsymbol{P}(t) + \boldsymbol{e}(t)^{\mathsf{T}} \dot{\boldsymbol{P}}(t) \boldsymbol{e}(t) + \boldsymbol{P}(t) \dot{\boldsymbol{e}}(t) \\ &= \boldsymbol{e}(t)^{\mathsf{T}} \Big(\boldsymbol{A}(\boldsymbol{\theta}(t))^{\mathsf{T}} \cdot \boldsymbol{P}(t) + \dot{\boldsymbol{P}}(t) + \boldsymbol{P}(t) \cdot \boldsymbol{A}(\boldsymbol{\theta}(t)) \Big) \boldsymbol{e}(t) < 0 \end{aligned} \tag{3}$$

Thus, the LTV system in Eq. (1) is quadratically stable if the following time-dependent Lyapunov equation is satisfied:[14]

$$\underbrace{\boldsymbol{A}(\boldsymbol{\theta}(t))^{\mathsf{T}} \cdot \boldsymbol{P}(t) + \dot{\boldsymbol{P}}(t) + \boldsymbol{P}(t) \cdot \boldsymbol{A}(\boldsymbol{\theta}(t))}_{=:-\boldsymbol{Q}(t)} \prec 0 \tag{4}$$

Here, the matrix $\boldsymbol{Q}(t) \in \mathbb{Q} \subseteq \mathbb{R}^{n\times n} \succ 0$ must be positive definite, where $\mathbb{Q}$defines its feasible set. Because of the requirement in Eq. (4), the well-known result for LTI systems, i.e., that the eigenvalues of the matrix $\boldsymbol{A}(\boldsymbol{\theta}(t))$ must be located in the open-left complex plane, is no longer a sufficient criterion for stability. Indeed, it is not even necessary as there are LTV systems with eigenvalues of the dynamic matrix in the right complex plane that are stable, while there are also systems with only eigenvalues of the dynamic matrix in the left complex plane that are unstable.[6,7] Removing this eigenvalue constraint, which may e.g., limit the possible gain value combinations, of classical gain scheduling approaches operating on LTI systems is also one of the contributions of this paper within the connected view on the trajectory optimization as well as LTV error dynamics stabilization, as it allows an improved exploitation of the system capabilities within numerical optimization. It should be noted in this context that Eq. (2) is still a conservative formulation for an LTV system as the Lyapunov matrix $\boldsymbol{P}(t)$ is only time-dependent rather than parameter-dependent and thus, the equation must be fulfilled for all parameters and their time derivatives and not just the ones that are encountered and physical.[15] Still, it is common to use this time-dependent approach to remain valid for generic LTV system formulations. However, it is important to stress that the proposed algorithm is not limited to the quadratic, time-dependent Lyapunov function candidate in Eq. (2), but can also be evaluated using any other suitable candidate function.

In the context of analyzing the convergence and stability properties of the LTV system based on the Lyapunov method, it is important to note that the time-varying Lyapunov function candidate in Eq. (2) can be bounded as follows:[16]

$$\begin{aligned} 0 < \frac{1}{2} \cdot c_1 \cdot \|\boldsymbol{e}(t)\|_2^2 < V(t) &= \frac{1}{2} \cdot \boldsymbol{e}(t)^{\mathsf{T}} \cdot \boldsymbol{P}(t) \cdot \boldsymbol{e}(t) \\ &< \frac{1}{2} \cdot c_2 \cdot \|\boldsymbol{e}(t)\|_2^2, \qquad \forall t \end{aligned} \tag{5}$$

Furthermore, the derivative of the Lyapunov function candidate in Eq. (3) can be bounded similarly defining:[16]

$$-c_4 \cdot \|\boldsymbol{e}(t)\|_2^2 < \dot{V}(t) = \boldsymbol{e}(t)^{\mathsf{T}} \cdot \boldsymbol{Q}(t) \cdot \boldsymbol{e}(t) < -c_3 \cdot \|\boldsymbol{e}(t)\|_2^2 < 0, \qquad \forall t$$

The coefficients in Eqs. (5) and (6) are given as the extreme eigenvalues of the respective matrices along the time interval:[16]

$$\begin{cases} c_1 = \min\limits_k \Big(\inf\limits_t \lambda_k(\boldsymbol{P}(t)) \Big), \quad c_2 = \max\limits_k \Big(\sup\limits_t \lambda_k(\boldsymbol{P}(t)) \Big) \\ c_3 = \min\limits_k \Big(\inf\limits_t \lambda_k(\boldsymbol{Q}(t)) \Big), \quad c_4 = \max\limits_k \Big(\sup\limits_t \lambda_k(\boldsymbol{Q}(t)) \Big) \end{cases} \tag{7}$$

Here, $\lambda_k(\cdot)$ is the k-th eigenvalue of the respective matrix evaluated at each time instance independently by solving a classic eigenvalue problem.[16] Thus, the time-varying Lyapunov function candidate and its derivative may be bounded by a static, time-independent analysis of the properties, and especially the eigenvalues, of the corresponding matrix. This property is used in Section 4.5 to validate the performance of the controller in the application example. Furthermore, these bounds are used to provide constraints in the optimiza-

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

tion problem to e.g., enforce a lower bound on the minimal error convergence rate and thus, to ensure robustness and a desired performance of the designed feedback controller.

3. Linear time-varying stability analysis within trajectory optimization

Section 2 introduced the basic principles of analyzing the stability of an LTV system using quadratic, time-dependent Lyapunov functions. The introduced methods must now be connected within a trajectory optimization problem as the stability of the LTV system should already be ensured when calculating the optimal trajectory. By this, an optimal trajectory that automatically accounts for the stability requirements of the dependent LTV systems, which may be the error dynamics of a path-following controller, is calculated. To stabilize the LTV system inside numerical gradient-based trajectory optimization, the required formulation of the Lyapunov stability analysis for trajectory optimization is introduced in the following, such that it can be solved numerically stable and efficient.

At first, Eq. (4) is a central equation required to verify the stability of the LTV dynamics in Eq. (1) within trajectory optimization. Consequently, the following matrix Equation of Motion (EoM) is added to the trajectory optimization problem:

$$\dot{\boldsymbol{P}}(t) = -\boldsymbol{Q}(t) - \boldsymbol{A}(\boldsymbol{\theta}(t))^{\mathsf{T}} \cdot \boldsymbol{P}(t) - \boldsymbol{P}(t) \cdot \boldsymbol{A}(\boldsymbol{\theta}(t)) \tag{8}$$

Thus, the entries of the matrix $\boldsymbol{P}(t)$ become states within the optimization problem, while the matrix entries of $\boldsymbol{Q}(t)$ are considered as free variables within the optimization.

Furthermore, it is important to note that the matrices $\boldsymbol{P}(t)$ and $\boldsymbol{Q}(t)$ in Eq. (8) must be ensured to be positive definite within the trajectory optimization problem. One option to do this, comprises the methods of "determinant-based" analysis of the matrices based on e.g., the Routh-Hurwitz criterion[17] or Sylvester's criterion.[17] A drawback of these is that they require the calculation of considerably large determinants including their derivatives in each iteration of the optimization. Additionally, the magnitude of the determinant values (especially when using the Sylvester's criterion) varies significantly (i.e., the magnitude of the determinant of the full matrix is normally much larger than the magnitude of only one entry), which generally leads to a numerically ill-conditioned optimization problem. Therefore, this study employs a constraint based on the idea of the Cholesky factorization as an alternative to create positive definite matrices. First of all, the Cholesky factorization for the matrix $\boldsymbol{Q}(t)$ is defined as follows:[18]

$$\boldsymbol{Q}(t) = \boldsymbol{L}_Q(t) \cdot \boldsymbol{L}_Q(t)^{\mathsf{T}} \tag{9}$$

Here, $\boldsymbol{L}_Q(t) \in \mathbb{L}_Q \subseteq \mathbb{R}^{n\times n}(t)$ (with $\mathbb{L}_Q$ being the set of all feasible values for the matrix) is a lower triangular matrix specified as follows:

$$\boldsymbol{L}_Q(t) = \begin{bmatrix} l_{11,Q}(t) & 0 & \cdots & 0 \\ l_{21,Q}(t) & l_{22,Q}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1,Q}(t) & l_{n2,Q}(t) & \dots & l_{nn,Q}(t) \end{bmatrix} \tag{10}$$

Now, $\boldsymbol{Q}(t)$ is positive definite if, and only if, the diagonal entries of $\boldsymbol{L}_Q(t)$ are real and have the same sign. Thus, the positive definiteness of $\boldsymbol{Q}(t)$ may be ensured by enforcing all signs of the diagonal entries of $\boldsymbol{L}_Q(t)$ to be strictly positive with a user-defined threshold ε_{L_Q}, which requires the following constraint within the optimization problem:

$$l_{ii,Q}(t) \geqslant \varepsilon_{L_Q} > 0, \quad i = 1, 2, \dots, n \tag{11}$$

Thus, the positive definiteness of $\boldsymbol{Q}(t)$ in Eq. (8) can be ensured in constrained optimization by specifying the matrix entries of $\boldsymbol{L}_Q(t)$ as real optimization variables, including the constraint on the diagonal elements in Eq. (11), rather than by using the matrix $\boldsymbol{Q}(t)$ directly. This matrix can, however, be directly calculated based on Cholesky factorization definition in Eq. (9). This procedure is generally numerically efficient, stable, and well-conditioned in the context of optimization. Here, the threshold ε_{L_Q} plays a vital role as it ensures that the constraint values are sufficiently large to result in a numerically stable problem and that the problem is posed in a standard form with inequality constraints for a nonlinear optimizer.[19] From experience gathered through this study, this threshold should be at least one order of magnitude larger than the optimizer tolerances.

A similar procedure must be applied to ensure that $\boldsymbol{P}(t)$ is positive definite. However, as this matrix is specified using the matrix equation of motion in Eq. (8), the procedure is not as straightforward as the entries cannot be considered as free variables. Indeed, it is necessary to calculate the diagonal entries of the lower triangular Cholesky factorization matrix based on $\boldsymbol{P}(t) = \boldsymbol{L}_P(t) \cdot \boldsymbol{L}_P(t)^{\mathsf{T}}$ as a constraint as opposed to specifying the entries directly as optimization variables. Therefore, the diagonal elements may be calculated recursively and included as constraints in the trajectory optimization problem as follows:[18]

$$\begin{cases} l_{11,P}(t) \propto p_{11}(t) \geqslant \varepsilon_P > 0 \\ l_{22,P}(t) \propto p_{22}(t) - \left(\frac{p_{21}(t)}{l_{11,P}(t)}\right)^2 \geqslant \varepsilon_P > 0 \\ \vdots \\ l_{nn,P}(t) \propto p_{nn}(t) - \sum_{k=1}^{n-1} l_{nk,P}^2(t) \geqslant \varepsilon_P > 0 \end{cases} \tag{12}$$

Take into account that the square root, which is normally required to exactly calculate the diagonal entries for the Cholesky factorization using Eq. (12), is left out in this formulation on purpose and only the proportionality (symbolized by $\propto$) with the radicand is specified and used in the optimization. This is first of all due to the fact that a real, positive radicand ensures the existence of a Cholesky factorization (i.e., that the analyzed matrix is positive definite) and second of all that the square root generally creates numerically ill-conditioned derivatives close to zero. Furthermore, the initial guess for the optimization variables might not ensure that the radicand is greater than zero and thus, the optimization may abort due to taking the square root of a negative value. Thus, only the radicand is used in the constraint and must be greater than a user-specified, strictly positive threshold ε_P (once more, this threshold ensures numerical stability and a proper formulation for the nonlinear optimizer), which is a sufficient condition for the purpose of ensuring that the matrix $\boldsymbol{P}(t)$ itself is positive definite.

Overall, Eqs. (11) and (12) provide the basic means to ensure stability of the LTV system inside optimization and are therefore already sufficient to ensure the system's stability.

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

In addition, robustness related to desired convergence properties of the error based on the Lyapunov function may be ensured by using the convergence bounds specified in Eq. (7). Here, specifically the coefficient c_3 is of interest as it enforces a lower bound on the convergence rate of the error to zero. Thus, the minimal eigenvalue of the $\boldsymbol{Q}(t)$ should be larger than a user-defined threshold ε_{λ_Q} (again, the threshold ensures numerical stability and a proper formulation of the nonlinear optimization problem):

$$\lambda_{\min}(\boldsymbol{Q}(t)) \geqslant \varepsilon_{\lambda_Q} > 0 \tag{13}$$

Compared to the previous values ε_{L_Q} and ε_P, the threshold value ε_{λ_Q} can be interpreted in a more physical manner. It describes a lower bound on the minimum error convergence rate. Thus, it is similar to well-known certification and handling qualities requirements in LTI theory.[20] Consequently, the threshold may be defined based on such requirements.

Because the direct calculation of eigenvalues inside optimization is generally problematic regarding smooth gradient formulations, the eigenvalue is instead calculated based on the fact that the squared, minimal singular value of the matrix $\boldsymbol{L}_Q(t)$ is the minimal eigenvalue of the matrix $\boldsymbol{Q}(t)$.[21] Thus, Eq. (13) results in:

$$\lambda_{\min}(\boldsymbol{Q}(t)) = \sigma^2_{\min}\left(\boldsymbol{L}_Q(t)\right) \geqslant \varepsilon_{\lambda_Q} > 0 \tag{14}$$

Here, $\sigma_{\min}(\cdot)$ is symbolizing the minimal singular value.

To avoid the calculation of the singular value in gradient-based optimization (due to the same reasons as for the eigenvalue), a method proposed in Ref. [22] is used to bound the minimal singular value based on the determinant, $\det(\cdot)$, and Frobenius norm, $||\cdot||_{\mathrm{F}}$, of the matrix $\boldsymbol{L}_Q(t)$:

$$\begin{aligned}\lambda_{\min}(\boldsymbol{Q}(t)) &= \sigma^2_{\min}\left(\boldsymbol{L}_Q(t)\right)\\ &\geqslant \left[\det\left(\boldsymbol{L}_Q(t)\right)\cdot\left(\frac{\sqrt{n-1}}{||\boldsymbol{L}_Q(t)||_{\mathrm{F}}}\right)^{n-1}\right]^2 \geqslant \varepsilon_{\lambda_Q}\\ &> 0\end{aligned} \tag{15}$$

Consequently, Eq. (15) provides a safe bound that can be easily and numerically efficiently incorporated within the trajectory optimization problem formulation.

Overall, the designed trajectory does therefore not only feature a stable LTV system, which is ensured by applying Eqs. (11) and (12) as constraints in the optimization problem, but also desired convergence, i.e., robustness, properties of the Lyapunov function, and thus the error, by using the methodology in Eq. (15).

4. Application example

This section gives an application examples for the developed optimal control-based design of trajectories accounting for the stability of error dynamics LTV systems. Therefore, the LTV system for a path-following controller, used as the example in this study, are introduced in Section 4.1, while their stabilization by a feedback error controller is described in Section 4.2. Then, the dynamic model used in the trajectory optimization is given in Section 4.3. Following, Section 4.4 shows the statement of the trajectory optimization problem including the stabilization of the LTV error dynamics. Finally, Section 4.5 shows an assessment of the controller's performance in simulation.

4.1. Linear time-varying trajectory-/path-following error dynamics

In this study, a LTV system for the error dynamics of a nonlinear trajectory deviation controller is used to analyze the proposed trajectory optimization problem formulation accounting for the LTV dynamics stabilization. It is formulated based on Ref. [4]. The considered path-following problem is visualized in Fig. 1 with the path deviation state vector defined as follows:

$$\boldsymbol{e}(t) = [\Delta x_T(t)\ \ \Delta y_T(t)\ \ \Delta z_T(t)\ \ \Delta u_T(t)\ \ \Delta v_T(t)\ \ \Delta w_T(t)]^{\mathsf{T}} \in \mathbb{R}^6 \tag{16}$$

Here, $\Delta x_T(t), \Delta y_T(t)$, and $\Delta z_T(t)$ are the position errors between the trajectory foot-point F and the real aircraft position R defined as follows:

$$[\Delta x_T(t)\ \ \Delta y_T(t)\ \ \Delta z_T(t)]^{\mathsf{T}} = \left(\boldsymbol{r}^{FR}\right)_T(t) = \left(\boldsymbol{r}^{R}\right)_T(t) - \left(\boldsymbol{r}^{F}\right)_T(t) \in \mathbb{R}^3 \tag{17}$$

The equivalent definition applies for the velocity errors $\Delta u_T, \Delta v_T$, and Δw_T.

Using this definition, the error dynamics are formulated as follows:[4]

$$\begin{aligned}\dot{\boldsymbol{e}}(t) = &\underbrace{\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ \dot{\boldsymbol{\Omega}}_T^{TE}(t) & \boldsymbol{\Omega}_T^{TE}(t)\end{bmatrix}}_{=:\boldsymbol{A}(\boldsymbol{\theta}(t))\in\mathbb{R}^{6\times 6}}\cdot\boldsymbol{e}(t) + \underbrace{\begin{bmatrix}\boldsymbol{0}\\ -1\\ 0\\ 0\end{bmatrix}}_{\in\mathbb{R}^6}\cdot\underbrace{\left(a^F_{x,K,\mathrm{req}}\right)^{ET}_T(t)}_{\in\mathbb{R}}\\ &+\underbrace{\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix}}_{\in\mathbb{R}^{6\times 3}}\cdot\underbrace{\underbrace{\left(\boldsymbol{f}^R\right)_T(t) + \left(\Delta\boldsymbol{f}^R\right)_T(t)}_{\in\mathbb{R}^3} + \left(\boldsymbol{\omega}_K^{TN}\right)_T\times\left(\boldsymbol{v}_K^R\right)^N_T}_{=:\boldsymbol{g}_{\mathrm{ac}}(\cdot)\in\mathbb{R}^3}\end{aligned} \tag{18}$$

Here, $\boldsymbol{\Omega}_T^{TE}(t) \in \mathbb{R}^{3\times 3}$ is the three-dimensional skew-symmetric matrix for the rotation of the Earth-Centered, Earth-Fixed (ECEF) frame E with respect to the trajectory frame T (see Fig. 1), i.e., the frame that follows the desired trajectory. Consequently, $\dot{\boldsymbol{\Omega}}_T^{TE}(t) \in \mathbb{R}^{3\times 3}$ is the corresponding rotational acceleration skew-symmetric matrix. Additionally, $\left(a^F_{x,K,\mathrm{req}}\right)^{ET}_T(t)$ is the required (index: req) kinematic acceleration of the foot point F, i.e., a design parameter, in the x direction of the trajectory (i.e., the T frame), which is derived with respect to the E and subsequently the T frame. This quantity is calculated by the optimization algorithm. Furthermore, $\boldsymbol{I}$ and $\boldsymbol{0}$ are an identity and a zero matrix of appropriate size, respectively. Finally, there are nonlinear influences due to the aircraft-related terms summarized in $\boldsymbol{g}_{\mathrm{ac}}(\cdot)$, which comprise the point mass aircraft equations of motion (symbolized by the specific forces $(\boldsymbol{f}^R)_T(t)$ and $(\Delta\boldsymbol{f}^R)_T(t)$ used for control purposes) and rotational transport terms when following the trajectory.[4]

The skew-symmetric matrices used in Eq. (18) are defined as follows:

Uncorrected Proof

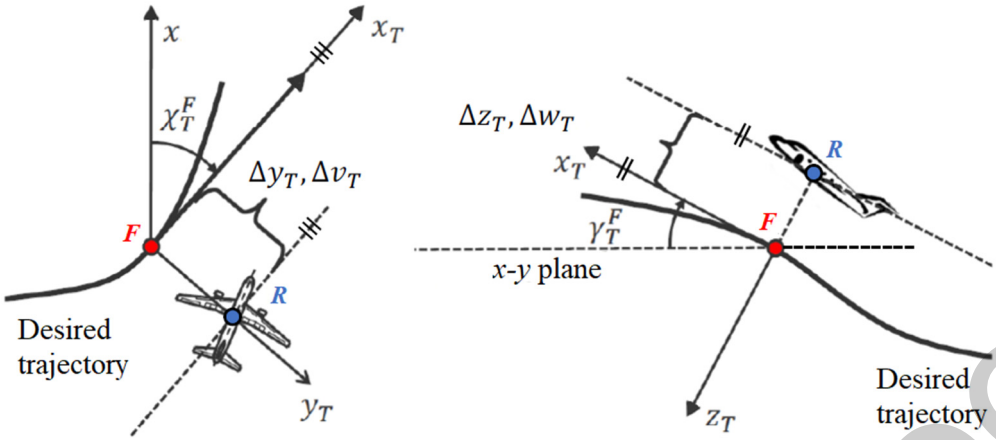

Fig. 1 Visualization of basic idea for nonlinear trajectory/path-following controller based on trajectory reference point F, i.e., the foot point along the desired trajectory, as well as the definition of the path deviations and their derivatives (adapted from Ref. [4]).

$$\boldsymbol{\Omega} = \boldsymbol{\omega}_\times = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{19}$$

Here, $\boldsymbol{\omega}$ is the three-dimensional vector of the entries used in the skew-symmetric matrix. Due to their definition, the error dynamics in Eq. (18) are generally unstable in an LTI sense, because eigenvalues symmetric to the imaginary axis result from it. Finally, the error dynamics in Eq. (18) depend on the time-varying trajectory parameters $\boldsymbol{\theta}(t) = \left[\dot{\boldsymbol{\omega}}_T^{TE}(t)^\mathsf{T} \quad \boldsymbol{\omega}_T^{TE}(t)^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^6$, which are related to the desired trajectory. Thus, the error dynamics are of a LTV type and the homogeneous system may be analyzed for stability using the methods introduced in Section 2.

4.2. Stabilizing linear time-varying error dynamics by feedback controller design

As already noted, the matrix $\mathsf{A}(\boldsymbol{\theta}(t))$ in the path-following error dynamics of Eq.(18) is generally indefinite by construction due to the skew-symmetric matrices in Eq. (19) and thus Eq. (4) may not have a stabilizing solution for this kind of problem. Consequently, we can apply a feedback control law for stabilization purposes.

Using this setup, a stabilizing controller for the specific forces is introduced that depends on trajectory deviations and controller gain matrices as follows:[4]

$$\left(\Delta \mathsf{f}^R\right)_T =: -\underbrace{\mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t))}_{\mathbb{R}^{3\times 3}} \cdot \begin{bmatrix} \Delta u_T(t) \\ \Delta v_T(t) \\ \Delta w_T(t) \end{bmatrix} - \underbrace{\mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t))}_{\mathbb{R}^{3\times 3}} \cdot \begin{bmatrix} \Delta x_T(t) \\ \Delta y_T(t) \\ \Delta z_T(t) \end{bmatrix} \tag{20}$$

Thus, the feedback path deviation error controller is of a standard proportional-derivative type. Here, the parameter-dependent matrices $\mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t))$ and $\mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t))$ are controller gains that are designed to stabilize the dynamic system.

It should be noted that the controller gain matrices in Eq. (20) can generally be chosen freely by the user to stabilize the system. Thus, they can also be designed during trajectory optimization to stabilize the error dynamics matrix, which is an important aspect of the proposed method. Hence, the LTV matrix, which is required to be stable, is defined as follows:

$$\mathsf{A}\left(\boldsymbol{\theta}(t), \mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t)), \mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t))\right) = \begin{bmatrix} 0 & \mathsf{I} \\ \dot{\boldsymbol{\Omega}}_T^{TE}(t) - \mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t)) & \boldsymbol{\Omega}_T^{TE}(t) - \mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t)) \end{bmatrix} \tag{21}$$

When using nonlinear dynamic inversion for the controller, the gain matrices are often chosen to be diagonal to keep the system decoupled, although this is not mandatory. Further take into account that $\left(a_{x,K,\mathrm{req}}^F\right)_T^{ET}(t)$ in Eq. (18) may in addition be used for controlling the system. This is explicitly not considered here, as it only has authority in x_T-direction and is calculated directly by the optimization algorithm as a feed-forward term.

4.3. Model and trajectory optimization formulation

As an application example, the following planar optimization problem (no consideration of earth rotation and earth ellipsoid shape), which represents a vehicle (e.g., car, airplane, or missile) moving only in the horizontal plane with initial and terminal directional constraints, is considered. Defining the state vector as $\mathsf{x}(t) = \left[x_E(t) \quad y_E(t) \quad V_T(t) \quad \chi_T(t) \quad \omega_{z,T}^{ET}(t)\right]^\mathsf{T} \in \mathbb{R}^5$ and command vector as $\mathsf{u}(t) = \left[\dot{V}_{T,\mathrm{cmd}}(t) \quad \dot{\omega}_{z,T,\mathrm{cmd}}^{ET}(t)\right]^\mathsf{T} \in \mathbb{R}^2$ the model is given as follows:

$$\begin{cases} \dot{x}_E(t) = V_T(t) \cdot \cos\left(\chi_T(t)\right) \\ \dot{y}_E(t) = V_T(t) \cdot \sin\left(\chi_T(t)\right) \\ \dot{V}_T(t) = \dot{V}_{T,\mathrm{cmd}}(t) \\ \dot{\chi}_T(t) = \omega_{z,T}^{ET}(t) \\ \dot{\omega}_{z,T}^{ET}(t) = \dot{\omega}_{z,T,\mathrm{cmd}}^{ET}(t) \end{cases} \tag{22}$$

This formulation is often used to model simple trajectories, where the dynamic system should only move in one plane. In aviation, this is often required when following different waypoints on a constant flight level.[23,4] The states and controls in the EoMs are defined and limited as well as scaled for the optimization problem as shown in Table 1.

Uncorrected Proof

Table 1 States and controls in optimization problem.

Definition	Symbol	Lower Bound	Upper Bound	Scaling	Unit
Position in North direction	x_E	0	+100	0.01	m
Position in East direction	y_E	0	+100	0.01	m
Kinematic velocity	V_T	0	+5	1	m/s
Kinematic course angle	χ_T	$-\infty$	$+\infty$	1	rad
Kinematic turn rate (T w.r.t. E)	$\omega_{z,T}^{ET}$	$-\pi/8$	$+\pi/8$	1	rad/s
Kinematic acceleration command (equivalent to $\left(a_{x,K,\mathrm{req}}^{F}\right)_{T}^{ET}(t)$ in Eq. (18))	$\dot{V}_{T,\mathrm{cmd}}$	−0.1	+0.1	1	m/s^2
Kinematic turn acceleration (T w.r.t. E)	$\dot{\omega}_{z,T,\mathrm{cmd}}^{ET}$	$-\pi/16$	$+\pi/16$	1	rad/s^2

As this model can only move in the horizontal plane and has the yaw motion as a single degree of rotational freedom, the error dynamics of Eq. (18) simplify to:

$$\dot{\mathsf{e}}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dot{\omega}_{z,T}^{ET}(t) & 0 & \omega_{z,T}^{ET}(t) \\ -\dot{\omega}_{z,T}^{ET}(t) & 0 & -\omega_{z,T}^{ET}(t) & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x_T(t) \\ \Delta y_T(t) \\ \Delta u_T(t) \\ \Delta v_T(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \left(a_{x,K,\mathrm{ctrl}}^{F}\right)_{T}^{ET}(t) + \mathsf{g}_{\mathrm{ac}}(\cdot) \tag{23}$$

Here, it was used that the rotational states are anti-symmetric,[20] i.e., $\dot{\omega}_{z,T}^{ET}(t) = -\dot{\omega}_{z,T}^{TE}(t)$ and $\omega_{z,T}^{ET}(t) = -\omega_{z,T}^{TE}(t)$. Thus, the matrix entries have inverted signs compared to the skew-symmetric matrix definition in Eq. (19).

To stabilize Eq. (23), the approach introduced in Eq. (21) with diagonal feedback matrices is applied and thus the stabilization problem is defined as follows:

$$\mathsf{A}\big(\boldsymbol{\theta}(t), \mathsf{K}_{\mathrm{p}}(\boldsymbol{\theta}(t)), \mathsf{K}_{\mathrm{d}}(\boldsymbol{\theta}(t))\big) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_x(\boldsymbol{\theta}(t)) & \dot{\omega}_{z,T}^{ET}(t) & -k_u(\boldsymbol{\theta}(t)) & \omega_{z,T}^{ET}(t) \\ -\dot{\omega}_{z,T}^{ET}(t) & -k_y(\boldsymbol{\theta}(t)) & -\omega_{z,T}^{ET}(t) & -k_v(\boldsymbol{\theta}(t)) \end{bmatrix} \tag{24}$$

Then, Eq. (24) is directly used in the equation of motion for the Lyapunov matrix from Eq. (8):

$$\dot{\mathsf{P}}(t) = -\mathsf{Q}(t) - \mathsf{A}\big(\boldsymbol{\theta}(t), \mathsf{K}_{\mathrm{p}}(\boldsymbol{\theta}(t)), \mathsf{K}_{\mathrm{d}}(\boldsymbol{\theta}(t))\big)^{\mathsf{T}} \cdot \mathsf{P}(t) - \mathsf{P}(t) \cdot \mathsf{A}\big(\boldsymbol{\theta}(t), \mathsf{K}_{\mathrm{p}}(\boldsymbol{\theta}(t)), \mathsf{K}_{\mathrm{d}}(\boldsymbol{\theta}(t))\big) \tag{25}$$

Here, the positive definiteness of the matrices $\mathsf{Q}(t) \succ 0 \in \mathbb{R}^{4\times4}$ and $\mathsf{P}(t) \succ 0 \in \mathbb{R}^{4\times4}$ is ensured as described in Eq. (11) and Eq. (12), respectively. For these inequalities, threshold values of $\varepsilon_{L_Q} = 10^{-3}$ and $\varepsilon_P = 10^{-3}$ are used respectively to be certain of the positive definiteness of the matrices. Generally, a well-conditioned optimization problem is also created by this choice and optimality is only reduced marginally. In addition, the threshold for the minimal eigenvalue of the matrix $\mathsf{Q}(t)$ in Eq. (15), i.e., the indicator for the minimal convergence rate of the Lyapunov function derivative in Eq. (6), is defined to be $\varepsilon_{\lambda_Q} = 0.1$. This has proven to be a good trade-off between convergence properties and optimality, while also considering the conservative nature of the approximation in Eq. (15).

Consequently, the state and control vectors for the planar EoMs in Eq. (22) are augmented by the matrix entries required in the Lyapunov matrix equation of motion as well as the controller gains as follows:

$$\begin{cases} \tilde{\mathsf{x}}(t) = [\mathsf{x}^{\mathsf{T}}(t)\ \ p_{11}(t)\ \ p_{21}(t)\ \ p_{31}(t)\ \ p_{41}(t)\ \ p_{22}(t)\ \ p_{32}(t)\ \ p_{42}(t) \\ \qquad p_{33}(t)\ \ p_{43}(t)\ \ p_{44}(t)]^{\mathsf{T}} \in \mathbb{R}^{15} \\ \tilde{\mathsf{u}}(t) = [\mathsf{u}^{\mathsf{T}}(t)\ \ l_{11,Q}(t)\ \ l_{21,Q}(t)\ \ l_{31,Q}(t)\ \ l_{41,Q}(t)\ \ l_{22,Q}(t)\ \ l_{32,Q}(t)\ \ l_{42,Q}(t) \\ \qquad l_{33,Q}(t)\ \ l_{43,Q}(t)\ \ l_{44,Q}(t)\ \ k_x(\boldsymbol{\theta}(t))\ \ k_y(\boldsymbol{\theta}(t))\ \ k_u(\boldsymbol{\theta}(t))\ \ k_v(\boldsymbol{\theta}(t))]^{\mathsf{T}} \in \mathbb{R}^{16} \end{cases} \tag{26}$$

Finally, the time-optimal trajectory optimization problem (i.e., minimizing the final time t_{f}), connecting the optimal trajectory design for the model in Eq. (22) with the stabilization of the LTV path deviation error dynamics (by calculating suitable feedback controller gains) in Eq. (23) and Eq. (24), is formulated as follows:

$$\begin{cases} \min\limits_{\tilde{\mathsf{x}}(t),\tilde{\mathsf{u}}(t)} & \text{Cost function}: \quad t_{\mathrm{f}} \\ \text{s.t.} & \text{Cholesky constraint for } \mathsf{Q}(t): \quad \text{Eq.(11)} \\ & \text{Cholesky constraint for } \mathsf{P}(t): \quad \text{Eq.(12)} \\ & \text{Convergence rate constraint}: \quad \text{Eq.(15)} \\ & \text{Lyapunov matrix EoM}: \quad \text{Eq.(25)} \\ & \text{Dynamic model EoMs}: \quad \text{Eq.(22)} \\ & \text{Rate constraint}: \quad -\frac{1}{2} \leqslant V_T(t) \cdot \omega_{z,T}^{ET}(t) \leqslant \frac{1}{2} \\ & \text{State/Controls bounds}: \quad \text{Table 1} \\ & \text{Initial/Final conditions}: \quad \text{Table 2} \end{cases} \tag{27}$$

The proposed design procedure is demonstrated through this example. Here, the rate constraint specifies a limitation of the maximal allowable turn acceleration, which is introduced to improve the model's fidelity. The state/controls bounds and the boundary constraints are given in Tables 1 and 2, respectively (unspecified values are assumed to be unbounded). Generally, the aforementioned constraints together with the cost function and the dynamic model EoMs formulate a classic time-optimal nonlinear trajectory optimization problem with state and control constraints. In parallel, the trajectory optimization problem in Eq. (27) designs the feedback gains in an optimal manner for the path controller to stabilize the LTV system of the path deviation error dynamics using the constraints and EoMs based on a time-dependent

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

Table 2 Boundary conditions of states and controls in trajectory optimization problem.

Initial condition	Final condition	"Unbounded"	"Bounded"	"Tightly bounded"
$[0\,\mathrm{m}\quad 0\,\mathrm{m}\quad 5\,\mathrm{m/s}\quad 0\,\mathrm{rad}\quad 0\,\mathrm{rad/s}]^{\mathsf{T}}$	$[100\,\mathrm{m}\quad 100\,\mathrm{m}\quad 5\,\mathrm{m/s}\quad 0\,\mathrm{rad}\quad 0\,\mathrm{rad/s}]^{\mathsf{T}}$	$k_x, k_y \in [-50, 50]1/s^2$ $k_u, k_v \in [-50, 50]1/s$	$k_x, k_y \in [-1, 1]1/s^2$ $k_u, k_v \in [-1, 1]1/s$	$k_x, k_y \in [-0.5, 0.5]1/s^2$ $k_u, k_v \in [-0.5, 0.5]1/s$

Lyapunov stability analysis. Conventionally, these two parts are done separately, while the proposed method integrates them into a single trajectory optimization problem as illustrated in Eq. (27). Therefore, the proposed methodology enables a better tuned optimal design of trajectory and controller for the desired task.

The trajectory optimization problem in Eq. (27) is solved using the free-of-charge MATLAB® toolbox FALCON.m,[24] which applies a trapezoidal collocation discretization to solve this continuous trajectory optimization problem in Eq. (27) by means of nonlinear programming. For this purpose, the interior-point optimizer IPOPT[19] with linear solver `ma97`[25] is used. The problem was discretized in time on a linearly-spaced grid with 101 points with optimality and feasibility thresholds of 10^{-5}. As the initial guess for the states, a linear interpolation between the values at the initial and final boundaries was chosen. The controls were initialized by the mean value of their bounds. Generally, the Cholesky matrix entries were initialized with identity matrices. This setup yielded smooth and reasonably fast convergence without numerical difficulties.

4.4. Optimal control-based trajectory design

The optimal trajectories, obtained for the reference optimal trajectory without a constraint on the error dynamics stability ("Reference"; solid blue), the results with "unbounded" gains (i.e., gains that can be chosen by the optimizer to be very large; "Unbounded"; dashed red), the results with limits on the gains ("Bounded"; dash-dotted green; values defined in Table 2), and the results with more tightly bounded gains ("Tightly bounded"; dotted black; values defined in Table 2) are shown in Fig. 2: It is clear that the reference and the unbounded case match very well and are equal from the viewpoint of the numerical tolerances of an optimizer. On the other hand, the results with bounded gains clearly show that the optimal trajectory differs when the gains cannot be chosen large enough by the optimizer to exploit the full physical capabilities of the dynamic model. Contrary to often used approaches in optimization by introducing a Lagrange function penalty (e.g., for the control effort), achieving these trajectories is automatically given in this setup by the stability constraint. This avoid the cumbersome procedure of appropriately solving the resulting Pareto problem using appropriate scaling factors, but gives a clear and unqiue connection between the reduced optimality and the enforced stability constraints. This consequently means that a less aggressive, less optimal, and smoother trajectory is chosen as the optimal one. This is also clear when looking at the final times obtained from the optimization: For the reference case, we have 31.8114 s, while the cases with stabilizing the LTV system result in 31.8114 s, 33.0422 s, and 34.5679 s for the unbounded, bounded, and tightly bounded case, respectively. Thus, as already discussed, the unbounded case results in the same optimal time, while the other cases result in a relative optimality reduction of 3.8689 % (bounded) and 8.6650 % (tightly bounded), respectively, explaining their smoother, less aggressive trajectory design.

This reduced aggressiveness and increased smoothness is also depicted in Fig. 3 showing the trajectory states. First of all, the turn velocity is reduced, which is one factor for an increased optimal time. Furthermore, the turn angle changes smoother, which is a consequence of the smoother rate com-

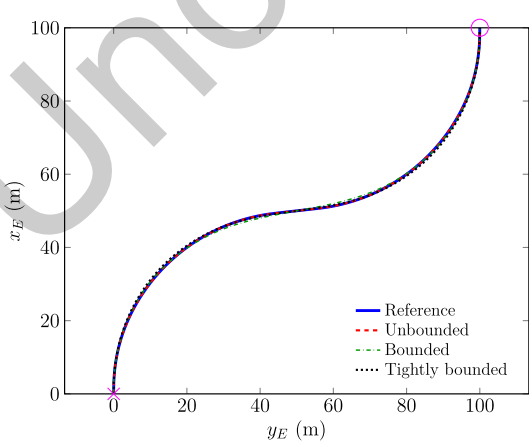

Fig. 2 Comparison of optimal trajectories.

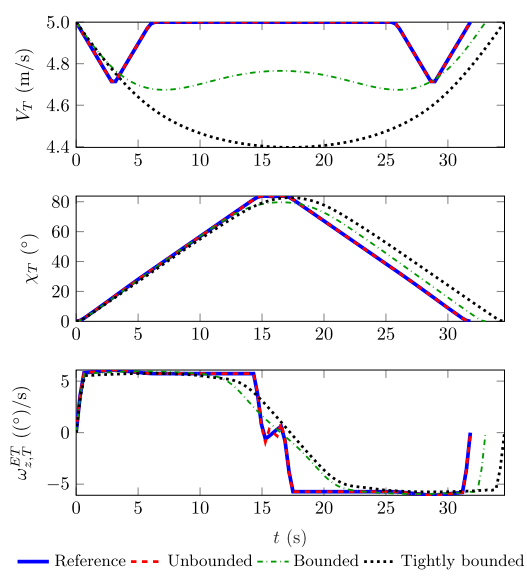

Fig. 3 Comparison of optimal state histories.

Uncorrected Proof

mand, especially in the beginning, the middle, and the end of the trajectory. Still, the maximal magnitude of the turn rate is similar in all cases, suggesting that, although the stability of the error dynamics must be ensured, the physical capabilities of the vehicle are still reasonably exploited.

Continuing, Figs. 4 and 5 show the optimal gain history and the corresponding eigenvalues of the matrix $\mathsf{Q}(t)$ used in the Lyapunov function candidate derivative in Eq. (3). It can be seen in Fig. 4 that the unbounded case has large feedback gain values, but still does not reach its limits (Table 1), which shows that it can be considered as "unbounded". The gains are significantly varying over the time interval, which shows that the optimizer tries to find an optimal, fast adapting controller gain for each state in the LTV system it encounters. For the bounded cases the gains are actually significantly smoother and to some extinct almost constant, which suggests a reduced optimality but also smoother operation. This is also specifically seen when looking at the results in Fig. 5 showing the real part ($\mathrm{Re}\{\cdot\}$) of the eigenvalues of the matrix $\mathsf{Q}(t)$, which is an indicator of how fast errors in the dynamics are reduced by the error controller. Here, the unbounded case has stable (negative real part), but very large eigenvalues. This suggests fast reduction of the error in theory, but may not be reasonable in practical applications due to modeling or sensor errors as well as disturbances. On the other hand, the bounded cases also stabilize the system, but with much smaller magnitude. This relates to a more reasonable behavior of the error controller and also its practical applicability. In addition, Fig. 5 shows that the desired constraint on the minimal eigenvalue (displayed in dashed-dotted magenta as labeled as $\lambda_{\min}(t)$) defined in Eq. (15) is always fulfilled even with additional margins. The reason for these safety margins is illustrated in Fig. 6: Here, the approximation (dashed red line) for the minimal singular value of the matrix $\mathsf{L}_Q(t)$ for the three gain cases, which is used in Eq. (15), is compared to the real singular value of the matrix (solid blue line), which is calculated in a post-processing step after the optimization. It is clear that the approximation is indeed

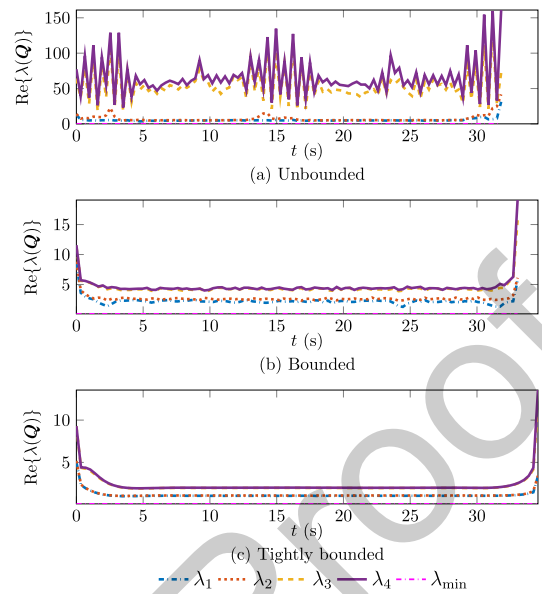

Fig. 5 Comparison of real part of optimal eigenvalue histories for unbounded and bounded cases for the matrix $\mathsf{Q}(t)$ in the Lyapunov function candidate derivative.

a conservative lower bound and thus, the optimal solution does not reach the lower bound as it would be expected for the solution of an optimization problem. Still, an increased conservatism and thus robustness may be desired in multiple applications. Additionally, less conservative approximations for the singular value in Eq. (15) may be used to improve optimality.

Furthermore, Fig. 7 depicts the real parts of the eigenvalue histories for the matrix $\mathsf{P}(t)$ used in the definition of the Lyapunov function candidate in Eq. (2). Here, it is first of all clear that the required positive definiteness is ensured as all real

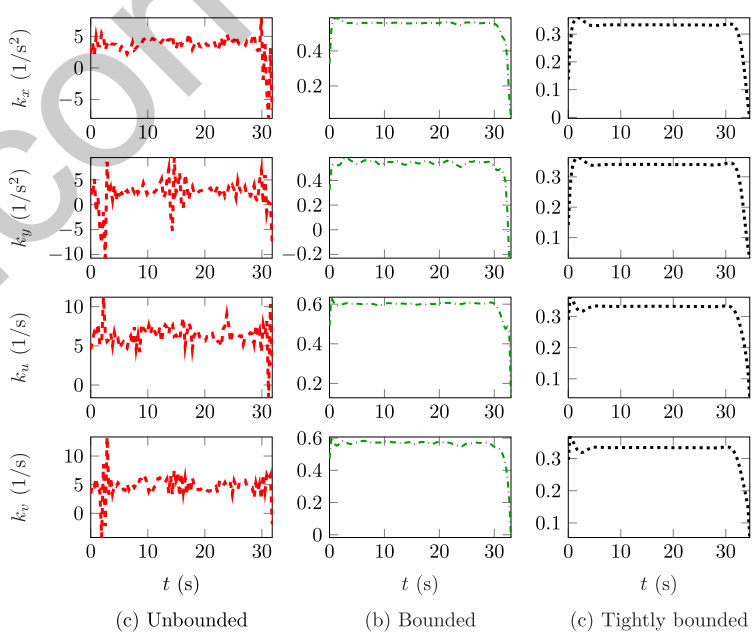

Fig. 4 Comparison of optimal gain histories for unbounded and bounded cases.

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

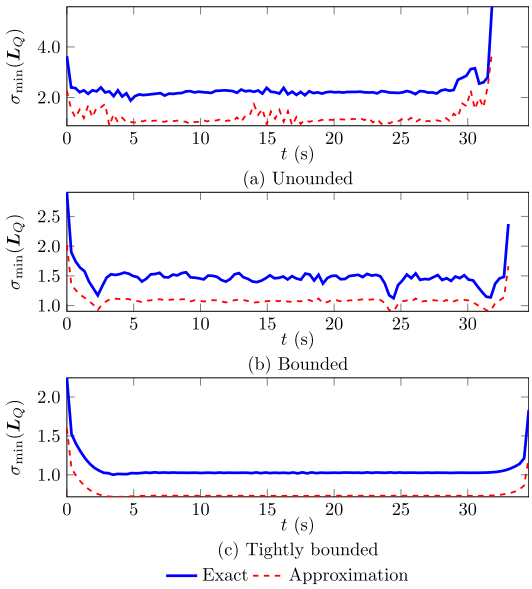

Fig. 6 Comparison of approximation for and exact minimal singular value for unbounded and bounded cases for the matrix $\mathsf{L}_Q(t)$ used in the convergence rate constraint for the minimal eigenvalue of the matrix $\mathsf{Q}(t)$.

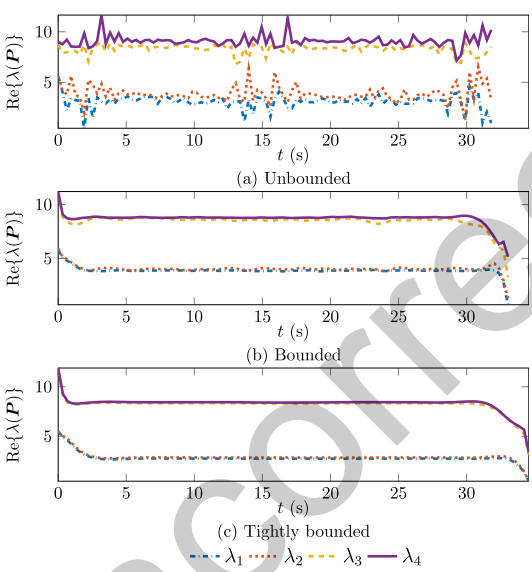

Fig. 7 Comparison of real part of optimal eigenvalue histories for unbounded and bounded cases for the matrix $\mathsf{P}(t)$ in the Lyapunov function candidate.

parts of the eigenvalues are positive. Furthermore, a similar behavior as in Fig. 5 can be seen, i.e., that by decreasing the gains the eigenvalues become smaller, which results in a feedback controller that is better suited in practice.

Concluding, Fig. 8 displays the eigenvalues of the stabilized system matrix $\mathsf{A}\big(\boldsymbol{\theta}(t), \mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t)), \mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t))\big)$: Here, it is clear that the system matrix has poles in the complete complex plane, i.e., also in the open right half plane. This displays the mentioned capacity of the proposed trajectory optimization algorithm that accounts for the stability of the feedback

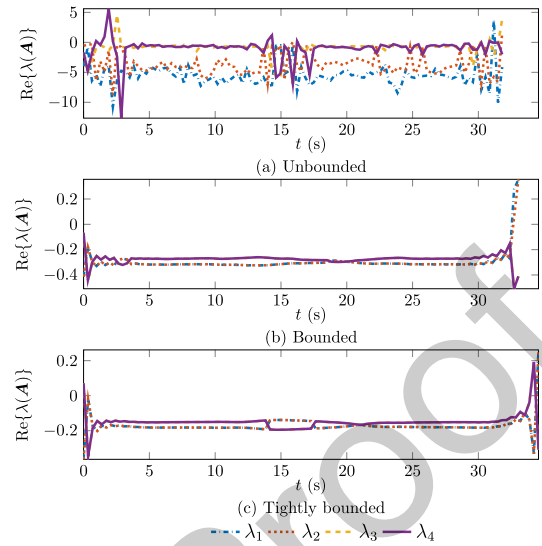

Fig. 8 Comparison of real part of optimal eigenvalue histories for unbounded and bounded cases for the system matrix $\mathsf{A}\big(\boldsymbol{\theta}(t), \mathsf{K}_\mathrm{p}(\boldsymbol{\theta}(t)), \mathsf{K}_\mathrm{d}(\boldsymbol{\theta}(t))\big)$ including the optimal gains.

controller described by an LTV error dynamics, which not only allows to design systems with poles in the open left complex plane as generally done with gain scheduling and LTI systems. This behavior and capability is one of the main benefits of the algorithm introduced in this study, as it allows for the full exploitation of the system dynamics.

4.5. Evaluation of controller performance

In this section, a simulation assessment of the controller's performance is carried out, in which specifically the stability and robustness properties are examined and verified. This is done to validate the theoretical stability characteristics obtained from the optimal control-based design in Section 4.4 within a practical application scenario. Additionally, the resulting feedback gains performance should be assessed. For this purpose, the model's initial position is offset by 10 m to the left ("west" direction). Therefore, the controller has an initial tra-

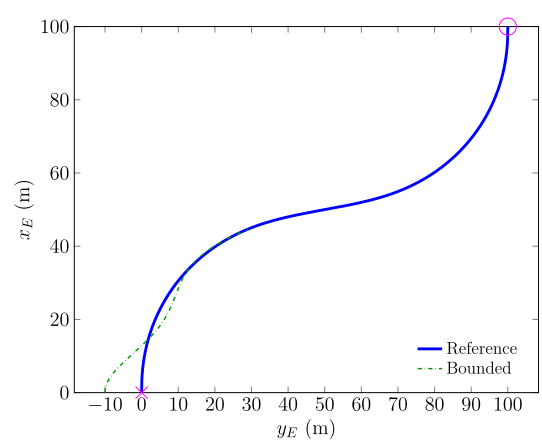

Fig. 9 Comparison of planned optimal trajectory and simulated trajectory including initial offset with bounded gains.

Uncorrected Proof

jectory deviation that needs to compensate to follow the originally desired path properly. For the controller, the results with the bounded gains is chosen.

At first, Fig. 9 visualizes the planned optimal trajectory ("Planned Trajectory"; solid blue) and the simulated trajectory using the bounded gains ("Bounded Controller"; dashed green) to specify the pseudocontrol within the nonlinear dynamic inversion control law that is derived from the LTV error dynamics.[4] It is clear that the controller achieves the desired task to follow the reference trajectory properly and smoothly. This also shows that the bounded gain case is a trade-off between tracking accuracy and speed. Consequently, different performance criteria can be fulfilled by adapting the bounds on the gains suitably. There is an overshoot when coming close to the trajectory which suggests that the gains may yet be still too aggressive for perfect, smooth convergence.

Continuing, Fig. 10 shows the Lyapunov function and its derivative until a simulation time of 12 s is reached and the error between real and desired trajectory becomes negligible. It is clear that the Lyapunov function is always positive, while its derivative is always negative as required for stability and enforced within the optimization. It is furthermore clear that after around 8 s the error becomes very small between real and desired trajectory and we achieve a good convergence. To visualize this behavior further, Fig. 10 contains the theoretical bounds on the decay of the Lyapunov function as well as its derivative as formulated in Ref. [16] and specified in Eq. (5) (denoted by "c_1-decay" and "c_2-decay", respectively) and Eq. (6) (denoted by "c_3-decay" and "c_4-decay", respectively). These show that the actual decay of the Lyapunov function and its derivative is contained within the theoretically predicted extreme bounds, i.e., the coefficient bounding the matrix norm multiplied by the norm of the error. In addition, the decay of the Lyapunov function derivative is also always at least given by the bound enforced using the constraint on the minimal eigenvalue ($\lambda_{\min}$-decay; dotted cyan line) in Eq. (15). Thus, the constraint enforced in the optimization also has an influence in the real scenario and accurately predicts a conservative, robust convergence bound as desired. This is a further strong argument on the applicability of the proposed trajectory optimization framework accounting for the LTV stabilization in application scenarios.

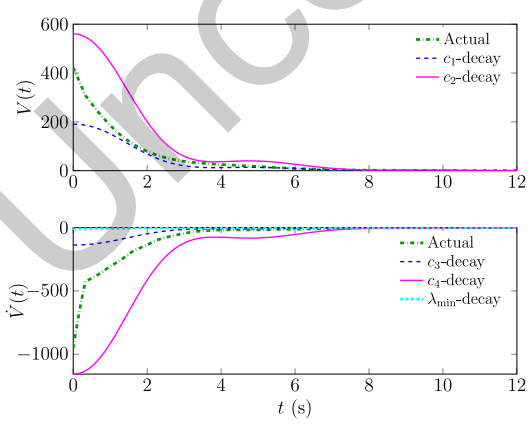

Fig. 10 Development of Lyapunov function and its derivative including theoretical decay bounds when trying to follow the optimal trajectory with an initial offset and bounded gains.

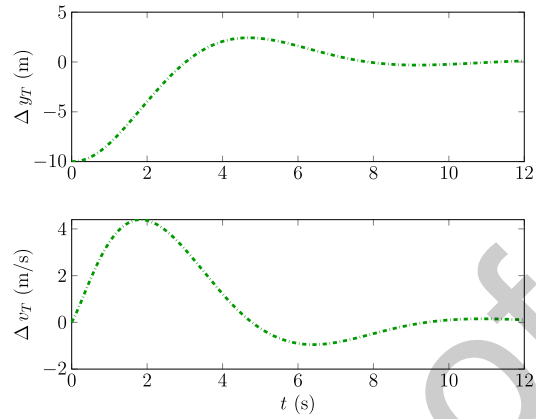

Fig. 11 Development of lateral path deviation and its derivative when trying to follow the optimal trajectory with an initial offset and bounded gains.

This behavior is further displayed in Fig. 11, which shows the lateral path deviation and its derivative. As already noted the lateral path deviation is often the most interesting aspect of the trajectory controller as it is most often the most critical. Thus, this figure is displayed here to show the achieved tracking accuracy by the controller even when only small and bounded gains are allowed. As already clearly indicated by the previous figures, there is one overshoot and one small undershoot but after 8 s the trajectory tracking is very good especially considering that no additional error integrators are used in the setup.

5. Conclusions

This study introduced the connection of optimal control-based trajectory design with stabilization of Linear Time-Varying (LTV) dynamics, including the design of adequate gains, which showed significant application potential. The novelty of this approach is the connected design of both the trajectory (as part of the feedforward control) and the feedback controller part of a trajectory tracking error dynamics that is described by stabilizing an LTV system. To this end, a trajectory optimization problem was formulated that contained both the standard dynamic model and its constraints as well as the Lyapunov analysis-based constraints accounting for the stability and robustness of the LTV system. These include constraints on the positive definiteness of the involved matrices in the Lyapunov matrix equation of motion that have been numerically efficient and stable incorporated using the idea of the Cholesky factorization. Additionally, constraints on the minimal decay rate of the Lyapunov function derivative, and thus, the error, are enforced to account for robustness and performance requirements. Consequently, this view allows for a coupled optimal trajectory and stable feedback controller design, which ensures stability and efficiency of the designed trajectory in real application, thus facilitating performance and availability.

Concluding, this study has formed a reliable theoretical foundation for the practical application case of designing optimal trajectories considering the stability and robustness of dependent LTV systems in a single framework to exploit the full capacity of the dynamic system. This theoretical founda-

Uncorrected Proof

Please cite this article in press as: PIPREK P et al. Optimal trajectory design accounting for the stabilization of linear time-varying error dynamics, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.10.031

tion enables the exploration of more sophisticated gain design strategies for the LTV dynamics, including system response shaping or robust design considering uncertainties, within the optimization as additional cost or constraints. Furthermore, it also enables handling of objectives dealing with different practical needs, such as a required level of smoothness. Finally, it is extendable to specifically tailored (parameter-dependent) Lyapunov functions to increase the optimality and reduce conservatism.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported in part by the TUM University Foundation Fellowship and in part by the German Federal Ministry for Economic Affairs and Energy (BMWi) within the Federal Aeronautical Research Program LuFo V-3 through Project "HOTRUN" (No. 20E1720A).

References

[1]. Claussmann L, Revilloud M, Gruyer D, et al. A review of motion planning for highway autonomous driving. *IEEE Trans Intell Transp Syst* 2020;**21**(5):1826–48.
[2]. Yang Y, Pan J, Wan W. Survey of optimal motion planning. *IET Cyber-Syst Robot* 2019;**1**(1):13–9.
[3]. He SM, Shin HS, Tsourdos A. Trajectory optimization for target localization with bearing-only measurement. *IEEE Trans Rob* 2019;**35**(3):653–68.
[4]. Schatz SP, Holzapfel F. Modular trajectory/ path following controller using nonlinear error dynamics*2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*. Yogyakarta, Indonesia. Piscataway: IEEE Press; 2014. p. 157–63.
[5]. Piprek P, Marb MM, Bhardwaj P, et al. Trajectory/path-following controller based on nonlinear jerk-level error dynamics. *Appl Sci* 2020;**10**(23):8760.
[6]. Tseng CH, Fong IK. New examples of stable LTV systems with eigenvalues in the ORHP*1999 European Control Conference (ECC)*. Karlsruhe, Germany. Piscataway: IEEE Press; 1999. p. 443–5.
[7]. Zhou B. On asymptotic stability of linear time-varying systems. *Automatica* 2016;**68**:266–76.
[8]. Agulhari CM, Lacerda MJ. Observer-based state-feedback control design for LPV periodic discrete-time systems. *Eur J Control* 2019;**49**:1–14.
[9]. Hoffmann C. Werner H.A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Trans Control Syst Technol* 2015;**23**(2):416–33.
[10]. López-Estrada FR, Rotondo D, Valencia-Palomo G. A review of convex approaches for control, observation and safety of linear parameter varying and takagi-sugeno systems. *Processes* 2019;**7**(11):814.
[11]. Hanema J, Toth R, Lazar M. Stabilizing non-linear MPC using linear parameter-varying representations*2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. Melbourne, Australia. Piscataway: IEEE Press; 2017. p. 3582–7.
[12]. White A, Zhu G, Choi J. Optimal LPV control with hard constraints. *Int J Control Autom Syst* 2016;**14**(1):148–62.
[13]. Han X, Liu Z, Li H, et al. Output feedback controller design for polynomial linear parameter varying system via parameter-dependent Lyapunov functions. *Adv Mech Eng* 2017;**9**(2), 168781401769032.
[14]. Briat C. *Stability of LPV systems. Linear parameter-varying and time-delay systems*. Heidelberg: Springer; 2015. p. 37–92.
[15]. Bruzelius F, Pettersson S, Breitholtz C. Linear parameter-varying descriptions of nonlinear systems. *Proceedings of the 2004 American control conference*, vol 2. Boston, USA. Piscataway: IEEE Press; 2004. p. 1374–9.
[16]. Zhu JJ, Liu Y, Hang R. A spectral Lyapunov function for exponentially stable LTV systems*2009 American Control Conference*. St. Louis, USA. Piscataway: IEEE Press; 2009. p. 1146–53.
[17]. Lunze J. *Regelungstechnik*. 8th ed. Heidelberg: Springer; 2010.
[18]. Davis TA. *Direct methods for sparse linear systems*. Philadelphia: SIAM; 2006.
[19]. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program* 2006;**106**(1):25–57.
[20]. Stevens BL, Lewis FL, Johnson EN. *Aircraft control and simulation: Dynamics, controls design, and autonomous systems*. 3rd ed. Hoboken: Wiley; 2016.
[21]. Arens T, Hettlich F, Karpfinger C, et al. *Mathematik*. Heidelberg: Springer; 2015.
[22]. Li HB, Huang TZ, Li H. Some new results on determinantal inequalities and applications. *J Inequal Appl* 2010;**2010**:1–16.
[23]. Schneider V, Piprek P, Schatz SP, et al. Online trajectory generation using clothoid Segments*14th International Conference on Control, Automation, Robotics & Vision*. Phuket, Thailand. Piscataway: IEEE Press; 2016. p. 1–6.
[24]. Rieck M, Bittner M, Grüter B, et al. FALCON.m user guide. Institute of Flight System Dynamics, Technical University of Munich, 2020. Available from: URL:www.falcon-m.com. .
[25]. Science & Technology Facilities Council. HSL: A collection of Fortran codes for large scale scientific computation, 2021. Available from: URL:www.hsl.rl.ac.uk. .

Uncorrected Proof