



# Transition trajectory generation with time-scaled trigonometric series control parameterization

Haichao Hong<sup>1</sup> · Patrick Piprek<sup>1</sup> · Shiqiang Hu<sup>2</sup> · Florian Holzapfel<sup>1</sup>

Received: 23 March 2022 / Revised: 22 May 2022 / Accepted: 27 May 2022 / Published online: 29 June 2022  
© The Author(s) 2022

## Abstract

This paper is concerned with shaping the time behavior of controls for generating transition trajectories. Parameterizing controls with trigonometric series enables infinite differentiability with respect to time, which is very appealing in real operations. In practice, one may desire the controls to have a specific distribution over time. However, to achieve this usually relies on judicious selection of the cost function. To this end, this paper introduces a time-scaling transformation for the trigonometric series. It yields an intuitive and flexible way to design the controls tailored to the desired shape of a transition maneuver. The process does not rely on additional design effort for the cost function. The proposed method is applied to generating a transition trajectory that converges to the desired condition earlier in comparison to the unscaled formulation. Numerical results demonstrate the effectiveness of the proposed method.

**Keywords** Trajectory generation · Control parameterization · Trigonometric series · Time scaling

## 1 Introduction

Trajectory generation is an essential practice in aerospace operations [2,7,19]. While discontinuous controls are common and possibly welcome in some applications [4,16], smoothness is very often considered to be an important feature of a reliable reference trajectory, especially in aeronautical operations [6,17,18]. For this purpose, there are dedicated trajectory generation approaches using fundamental geometric elements, such as lines, arcs, and clothoids, capable of generating trajectories with only discontinuities in high-order derivatives at the connecting points [3,20,21]. When dynamics are taken into consideration, trajectories

are generated using numerical optimization. For aerospace applications, one can refer to [1,5]. However, many existing methods nowadays based on numerical approaches may not be able to guarantee the continuity in higher-order derivatives. This is because the controls at every discrete point are set to be independent decision variables.

A trigonometric series-based trajectory generation approach is developed in [10] and has been applied in various applications [11–13]. It parameterizes the controls as trigonometric series and uses numerical optimization methods including convex optimization to determine the unknown series coefficients. The controls being a time-dependent trigonometric series are of  $C^\infty$ , i.e., infinitely differentiable. Therefore, smoothness of trajectories can be achieved. By imposing equality constraints, the boundary time behavior, namely, the continuity to the adjacent flight segments can be ensured as required. However, except for the boundaries, to acquire certain time behavior of controls relies on a judicious choice of the cost function. In fact, the cost function traditionally plays the sole role in deciding the control behavior in trajectory optimization [19]. Therefore, we seek for an innovative means to approach the goal within the trigonometric series-based trajectory generation.

In this paper, we develop a simple but effective way to shape the time behavior of controls parameterized by trigonometric series through time-scaling. A user-defined

---

✉ Haichao Hong  
haichao.hong@tum.de  
Patrick Piprek  
patrick.piprek@tum.de  
Shiqiang Hu  
sqhu@sjtu.edu.cn  
Florian Holzapfel  
florian.holzapfel@tum.de

<sup>1</sup> Institute of Flight System Dynamics, Technical University of Munich, 85748 Garching, Germany

<sup>2</sup> School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China

time scaling function is introduced to replace the original argument of the series basis function, namely, time, with a new independent variable. This variable is designed to have the desired time behavior and will pass the characteristic to the parameterized controls. Hence, having a specific time behavior of controls is decoupled from the design process of the cost function. Still, the cost function may affect the control behavior. However, the proposed method shapes the behavior in an external manner without modifying on the cost function. This philosophy is a unique advancement to previous works on the trigonometric series and thus a contribution to the literature.

The implementation of the newly proposed method is demonstrated in the convex-optimization framework developed in [8]. In this paper, it is shown that the proposed method only requires minimal modification of the optimization framework in its original form in [10], leading to a direct port. We show an application of transition trajectory generation, where a transition trajectory connects two steady flight conditions. A smooth connection is desired, which corresponds well to the capability of the trigonometric series parameterization method. For the transition maneuver, the application in this paper demands the aircraft to be more stable, i.e., having slow-varying control commands, in the later phase of the transition. This time behavior will better prepare the aircraft for the following flight segment. To this end, a time-scaling function based on a logistic function is selected to achieve this feature within the proposed method. Numerical simulations are carried out to verify the design.

The rest of this paper is organized as follows: Section 2 introduces the trigonometric series parameterization and the time scaling that shapes the time behavior of it prior to the determination of the series coefficients. Section 3 shows the details of the implementation of the proposed approach to a convex optimization framework. Section 4 shows the simulation results of the proposed method in combination with the convex optimization being applied to a transition trajectory generation task. Section 5 presents the conclusion of this paper.

## 2 Time-scaled trigonometric series control parameterization

In [10], a control parameterization method is introduced. A time-varying control variable  $u$  is described as an  $N$ th-order trigonometric series as

$$u(t) = a_0 + \sum_{n=1}^N \left( a_n \cos\left(\frac{n\pi}{T_f}t\right) + b_n \sin\left(\frac{n\pi}{T_f}t\right) \right) \quad (1)$$

$$= s_N(t) c_N,$$

where

$$s_N(t) = \begin{bmatrix} 1, \cos\left(\frac{\pi}{T_f}t\right), \dots, \cos\left(\frac{N\pi}{T_f}t\right), \\ \sin\left(\frac{\pi}{T_f}t\right), \dots, \sin\left(\frac{N\pi}{T_f}t\right) \end{bmatrix} \quad (2)$$

$$c_N = [a_0, \dots, a_N, b_1, \dots, b_N]^T. \quad (3)$$

Here,  $s_N(t)$  is a function of time  $t \in [0, T_f]$ . Naturally,  $u(t)$  is of  $C^\infty$  after the parameterization. Moreover, the parameterization specifies  $u$  as a linear function of the series coefficients  $c_N$ . It is intuitive that for Eq. (1), the time behavior of  $u$  over the entire horizon  $[0, T_f]$  is solely decided by  $c_N$  which is to be determined by optimization. This is generally not a disadvantage because of its simple structure. To achieve a particular time behavior of the control variable, one often relies on designing the cost function which affects the selection of  $c_N$ . This is not always favorable as such an indirect process may lead to a Pareto objective and thus possibly causes the cost function to lose advantageous properties, e.g., convexity. In addition, it is not effortless to relate certain behaviors to mathematical expressions in a cost function.

To mitigate this, we introduce a time-scaling transformation to Eq. (1). The time-scaling transformation originates in [15] and yields a new optimization problem formulation over a new horizon. Meanwhile in this paper, the proposed time-scaling transformation only influences the trigonometric series parameterization in Eq. (1). It introduces a new variable that monotonically increases with respect to time:

$$\frac{dp}{dt} = v(t), \quad (4)$$

$$p(0) = 0, \quad (5)$$

$$p(T_f) = P, \quad (6)$$

where  $p \in [0, P]$ . Here,  $v(t) > 0, \forall t \in [0, T_f]$  and it is to be designed by user in order to achieve desired behavior. It shall be noted that  $P$  does not need to be specified because the definite integral of  $v(t)$  over  $t \in [0, T_f]$  gives  $P$ .

Next, we reformulate the control parameterization in (1) to be a function of  $p$ :

$$u(p) = s_N(p) c_N. \quad (7)$$

The time behavior of the control  $u$  is no longer only affected by  $c_N$  but preserves characteristics of  $p(t)$  which are defined already in the designing process of  $v(t)$ . This can be inferred using the chain rule as

$$\frac{du}{dt} = \frac{du}{dp} \frac{dp}{dt} \quad (8)$$

It is a common measure for numerical optimization approaches that a time grid is given over the entire time horizon. The discrete time grid specifies the total number of the discrete points. Provided that  $T_f$  is fixed, the value of  $s_N$  at each discrete point is known a priori. That is to say,  $s_N$  does not change during the optimization algorithm execution. Therefore, it can be evaluated only once before executing the optimization program. For some optimization approaches including the one utilized in this paper, this feature saves some computational effort and streamlines the formulation.

The only remaining issue is the design of  $v(t)$ . Ideally, it should be rather flexible. In Sect. 4, the design is shown along with the actual need of the application. As a remark,  $p$  must be smooth in order to ensure that  $u$  is smooth.

### 3 Implementation in model predictive convex programming

In general, the proposed time-scaled trigonometric series parameterization works with various optimization approaches, such as successive convex optimization approaches [11,13], because it is only a linear constraint. In this paper, to show the difference (or the similarity in a positive sense) from the formulation in [10], we again consider the model predictive convex programming (MPCP) [8]. Only the most salient steps of the MPCP in [8] are presented in this section.

First, we write the trigonometric series parameterization in a discrete time form for the control vector

$$u_k = [S_N]_k \bar{c}_N, \quad k = 1, 2, \dots, N_t - 1, \tag{9}$$

where

$$[S_N]_k = \begin{bmatrix} [s_N^{(1)}]_k & 0 & \dots & 0 \\ 0 & [s_N^{(2)}]_k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & [s_N^{(m)}]_k \end{bmatrix}, \tag{10}$$

$$\bar{c}_N = \left[ (c_N^{(1)})^\top, (c_N^{(2)})^\top, \dots, (c_N^{(m)})^\top \right]^\top, \tag{11}$$

and the superscript indicates the index of the components. The total number of discrete points is denoted by  $N_t$ .

Because of the linearity, the incremental change of  $u_k$ , denoted as  $du_k$  can be expressed by

$$du_k = [S_N]_k d\bar{c}_N, \tag{12}$$

where  $d\bar{c}_N$  is the incremental change of the coefficient vector  $c_N$ . It is important to notice that, as mentioned in the

previous section, regardless of whether  $S_N$  is a function of  $t$  as in Eq. (1) or  $p$  as in Eq. (7), as long as  $v$  and  $T_f$  are specified,  $[S_N]_k$  is fully defined for all  $k = 1, 2, \dots, N_t - 1$ . Hence, the time-scaling of the parameterization and the functional dependency make no difference to Eq. (12) and thus have no impact on the following derivations.

For introducing the MPCP, a discrete-time form of the system dynamics is given as

$$x_{k+1} = F_k(x_k, u_k), \quad k = 1, 2, \dots, N_t - 1, \tag{13}$$

where  $F$  is continuously differentiable, and  $x$  and  $u$  are the state and the control vector, respectively. The MPCP is an iterative approach that updates the state and control histories as  $x_k = x_k^p + dx_k$  and  $u_k = u_k^p + du_k$ , where the superscript  $p$  denotes the variables in the previous iteration. Using a Taylor series expansion and neglecting higher-order terms, Eq. (13) can be expressed as

$$x_{k+1} = F_k(x_k^p + dx_k, u_k^p + du_k) \approx x_{k+1}^p + \left[ \frac{\partial F_k}{\partial x_k} \right] dx_k + \left[ \frac{\partial F_k}{\partial u_k} \right] du_k, \tag{14}$$

which yields

$$dx_{k+1} = \left[ \frac{\partial F_k}{\partial x_k} \right] dx_k + \left[ \frac{\partial F_k}{\partial u_k} \right] du_k. \tag{15}$$

Eq. (15) can be further expanded:

$$\begin{aligned} dx_{k+1} &= \left[ \frac{\partial F_k}{\partial x_k} \right] dx_k + \left[ \frac{\partial F_k}{\partial u_k} \right] du_k \\ &= \left[ \frac{\partial F_k}{\partial x_k} \right] \left[ \frac{\partial F_{k-1}}{\partial x_{k-1}} \right] dx_{k-1} \\ &\quad + \left[ \frac{\partial F_k}{\partial x_k} \right] \left[ \frac{\partial F_{k-1}}{\partial u_{k-1}} \right] du_{k-1} + \left[ \frac{\partial F_k}{\partial u_k} \right] du_k \\ &\quad \vdots \\ &= A_k dx_1 + B_{k,1} du_1 + B_{k,2} du_2 + \dots + B_{k,k} du_k, \end{aligned} \tag{16}$$

where

$$A_k = \left[ \frac{\partial F_k}{\partial x_k} \right] \left[ \frac{\partial F_{k-1}}{\partial x_{k-1}} \right] \dots \left[ \frac{\partial F_1}{\partial x_1} \right], \tag{17}$$

$$B_{k,j} = \left[ \frac{\partial F_k}{\partial x_k} \right] \left[ \frac{\partial F_{k-1}}{\partial x_{k-1}} \right] \dots \left[ \frac{\partial F_{j+1}}{\partial x_{j+1}} \right] \left[ \frac{\partial F_j}{\partial u_j} \right], \tag{18}$$

for  $j = 1, 2, \dots, k - 2$

$$B_{k,k-1} = \left[ \frac{\partial F_k}{\partial x_k} \right] \left[ \frac{\partial F_{k-1}}{\partial u_{k-1}} \right], \tag{19}$$

$$B_{k,k} = \frac{\partial F_k}{\partial u_k}. \tag{20}$$

Here, the initial condition  $\mathbf{x}_1$  is considered to be fixed, i.e.,  $d\mathbf{x}_1 = 0$ . Hence, Eq. (16) can be written as

$$d\mathbf{x}_{k+1} = \sum_{j=1}^k \mathbf{B}_{k,j} d\mathbf{u}_j, \quad (21)$$

where  $k = 1, 2, \dots, N_t - 1$ . This equality is derived in [14], connecting the state increments  $d\mathbf{x}$  to the control corrections  $d\mathbf{u}$ . Using a static Lagrangian multiplier, Eq. (21) can be used to solve a class of optimization problems and it is referred to as the model predictive static programming. For incorporating more constraints, especially path constraints, a convex optimization problem can be formulated. Substituting Eq. (12) into Eq. (21) yields

$$\begin{aligned} d\mathbf{x}_{k+1} &= \sum_{j=1}^k \mathbf{B}_{k,j} [S_N]_j d\bar{\mathbf{c}}_N \\ &= \sum_{j=1}^k \mathbf{D}_{k,j} d\bar{\mathbf{c}}_N. \end{aligned} \quad (22)$$

While  $S_N$  is different due to the time scaling, Eq. (22) is identical to the one in [10]. This shows that the proposed method is compatible with the previous works, and a very intuitive port can be realized.

To formulate a convex optimization problem, we select the cost function as

$$J = (d\bar{\mathbf{c}})^T (d\bar{\mathbf{c}}). \quad (23)$$

As one might notice, this quadratic cost function simply minimizes the incremental changes of the coefficients. It can be interpreted as minimizing the change of controls between iterations. It does not address the time behavior of the controls.

Constraints can be reformulated in the following way

$$\begin{aligned} \mathbf{A} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} &\leq \mathbf{b}, \\ \mathbf{A} \begin{bmatrix} \mathbf{x}^p + d\mathbf{x} \\ \mathbf{u}^p + d\mathbf{u} \end{bmatrix} &\leq \mathbf{b}, \\ \mathbf{A} \begin{bmatrix} \mathbf{x}^p + \mathbf{B}d\mathbf{u} \\ \mathbf{u}^p + d\mathbf{u} \end{bmatrix} &\leq \mathbf{b}, \end{aligned} \quad (24)$$

where the elements of the lower triangular matrix  $\mathbf{B}$  are  $\mathbf{B}_{k,j}$  in Eq. (21). Notice that  $d\mathbf{u}$  is the only unknown variable in Eq. (24) and it can be related to  $d\bar{\mathbf{c}}$  using Eq. (12). Therefore, constraints are ultimately functions of  $d\bar{\mathbf{c}}$ . A convex optimization problem can be formulated to iteratively search for suitable  $d\bar{\mathbf{c}}$  that satisfies all constraints. More details can be found in [10].

With a dedicated solver, the order of the trigonometric series does not significantly influence the computational efficiency for solving the convex optimization problems. Moreover, for a maneuver that is not complex, a low order suffices. In this work, we select  $N = 3$ . On the other hand, the total number of the discrete points,  $N_t$ , does affect the computational efficiency. However, this setting is decided by the required accuracy of the trajectory rather than the proposed control parameterization. Over a 20 s time horizon,  $N_t = 101$  is considered to give a sufficiently accurate result for the application introduced in the next section.

## 4 Transition flight trajectory generation

In this section, we adopt the application considered in [9] to demonstrate the effectiveness of the proposed method and in Sect. 3, while also comparing the results to the original form in [10].

The scenario is to generate a transition flight trajectory connecting two steady flight conditions. The objective is to make the maneuver relatively milder at the later phase of the transition, which prepares the aircraft for the next flight segment.

### 4.1 Problem formulation

A fixed-wing unmanned aerial vehicle point-mass model is considered:

$$\begin{aligned} \dot{x}(t) &= V(t) \cos \chi(t) \cos \gamma(t), \\ \dot{y}(t) &= V(t) \sin \chi(t) \cos \gamma(t), \\ \dot{h}(t) &= V(t) \sin \gamma(t), \\ \dot{\chi}(t) &= \frac{L(t) \sin \mu(t)}{mV(t) \cos \gamma(t)}, \\ \dot{\gamma}(t) &= \frac{L(t) \cos \mu(t) - mg \cos \gamma(t)}{mV(t)}, \\ \dot{V}(t) &= \frac{T(t) - D(t)}{m} - g \sin \gamma(t), \end{aligned} \quad (25)$$

where  $x, y, h$  are coordinates of the aerial vehicle, and  $\chi, \gamma$ , and  $V$  are the flight path course angle, the flight path climb angle, and the speed, respectively.  $\mu$  is the bank angle.  $m$  is the mass of aerial vehicle.  $g$  is the gravitational acceleration. The thrust and aerodynamic forces in Eq. (25) are expressed by

$$T(t) = T_{\max} \delta_T(t), \quad (26)$$

$$L(t) = \frac{1}{2} \rho (V(t))^2 S C_L(t), \quad (27)$$

$$D(t) = \frac{1}{2} \rho (V(t))^2 S \left( C_{D0} + k_i (C_L(t))^2 \right), \quad (28)$$

where  $T_{\max}$  is the maximum thrust,  $\delta_T$  is the thrust lever position,  $\rho$  is the air density,  $S$  is the reference area, and  $k_i$  is the induced drag factor.  $C_L$  and  $C_{D_0}$  are the lift coefficient and the zero-lift drag coefficient, respectively. The state, control, and output vectors of the dynamics in Eq. (25) are given as

$$\mathbf{x}(t) = [x(t), y(t), h(t), \chi(t), \gamma(t), V(t)]^T, \quad (29)$$

$$\mathbf{u}(t) = [C_L(t), \mu(t), \delta_T(t)]^T, \quad (30)$$

$$\mathbf{y}(t) = [y(t), h(t), \chi(t), \gamma(t), V(t)]^T. \quad (31)$$

The simulation scenario is to generate a transition trajectory connecting a coordinated turn with a steady descent. The equality constraints are given as

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0, \\ \mathbf{y}(T_f) &= \mathbf{y}_f, \\ \mathbf{u}(0) &= \mathbf{u}_0, \\ \mathbf{u}(T_f) &= \mathbf{u}_f, \\ \dot{\mathbf{u}}(0) &= 0, \\ \dot{\mathbf{u}}(T_f) &= 0. \end{aligned} \quad (32)$$

On the right-hand side of the equality constraints in Eq. (32) are all constants.

The path constraints are specified by

$$\begin{aligned} \mathbf{u}_{\min} &\leq \mathbf{u} \leq \mathbf{u}_{\max}, \\ \dot{\mathbf{u}}_{\min} &\leq \dot{\mathbf{u}} \leq \dot{\mathbf{u}}_{\max}. \end{aligned} \quad (33)$$

### 4.2 Time-scaling function

The goal of designing the time-scaling function is to properly distribute the control authority for the transition maneuver. In accordance with the need of flying more stably control-wise in the later phase of the transition, we select the following function:

$$v(t) = \frac{1}{1 + e^{(t-10)/8}}. \quad (34)$$

It is a logistic function and  $p$  can be obtained using Eqs. (4) and (5). The time histories of  $v$  and  $p$  are shown in Fig. 1. Because the problem at hand is a fixed-final-time trajectory optimization problem, the variable  $p$  and the series basis functions are fully known and they do not change with the optimization runs.

As can be seen from Fig. 1,  $v$  is monotonically decreasing as time evolves, where  $v|_{t=0} = 0.77$  and  $v|_{t=20} = 0.22$ . Here, the slope is gentle, as an aggressive maneuver during the earlier stage, or technically any stage, is not desired. Referring to Eq. (8), the control derivatives are relatively more slowly varying during the later phase of the transition.

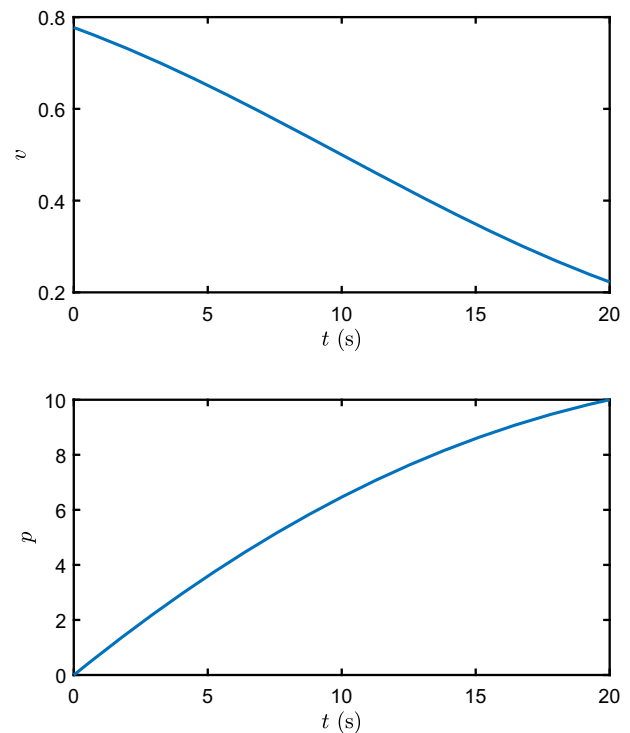


Fig. 1 Time-scaling functions

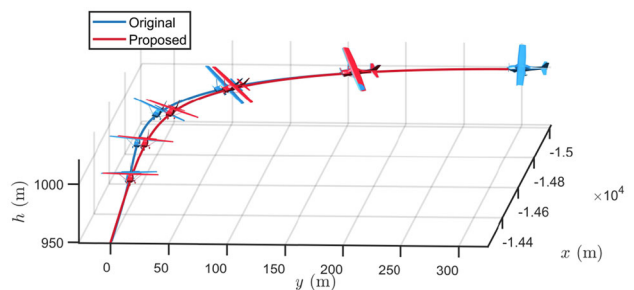


Fig. 2 Transition trajectories

### 4.3 Simulation results

The simulation results are presented in this section. The quadratic programming problems are solved with a dedicated solver—OSQP [22]. The formulation in [10] is referred to as the original method, while the time-scaled trigonometric series is called the proposed method.

The flight trajectories generated by the original method and the proposed method are shown in Fig. 2. The trajectories are similar but the maneuver is less aggressive for the proposed method in the later phase. This can be confirmed by the state histories given in Fig. 3. The trajectories generated by both methods lead the aircraft to zero course angle and  $-5$  deg climb angle. The speed is maintained at 50 m/s as desired. The main occurrence of the state variations has been shifted to the earlier part of the maneuver. It is also visible that

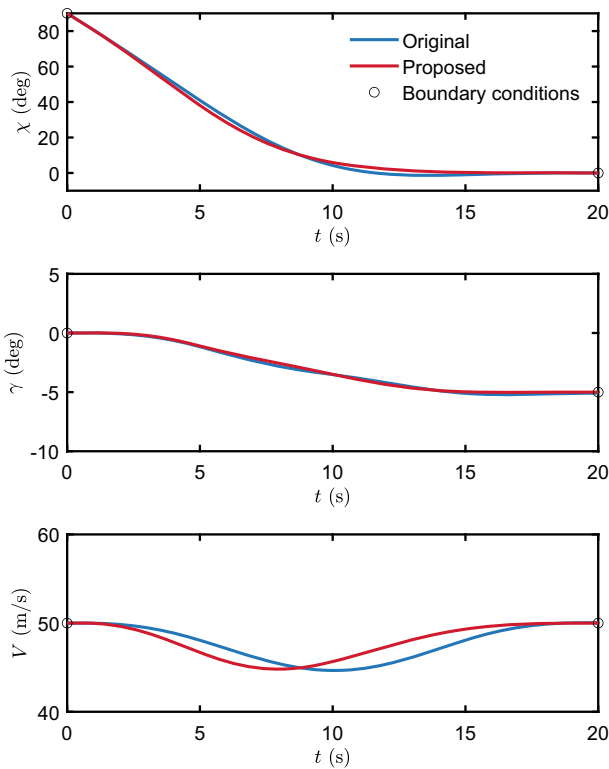


Fig. 3 Time histories of the states

the aircraft enters a quasi-steady condition relatively earlier, while the excursions of the states remain similar. This corresponds to the objective of implementing the time-scaling function. Although the proposed method parameterizes the controls, the state trajectories are the real system responses to be manipulated.

Speaking of the controls, their histories can be seen in Fig. 4. The state histories are all smooth and satisfy every constraint. It is worth noting that boundary conditions, i.e., the trim conditions, can be easily enforced for both the original and the proposed methods. These are important conditions that ensure continuous connections to the steady flights. For the proposed method, slightly more control authorities are used in the earlier phase of the transition maneuver. The maximum utilization increases by 6.6%, 5.0%, and 0%, for  $C_L$ ,  $\mu$ , and  $\delta_T$ , respectively. On the other hand, the controls approach faster to being steady-state and the desired terminal conditions in the later phase. This means that the aircraft at a relatively earlier time flies practically at a trim condition using the proposed method. Inspecting the time histories of the control derivatives in Fig. 5 reveals more details. The trade-off for getting steadier controls at the later phase is committing marginal but still higher change rate of the controls in the earlier phase. The incremental changes of the maximum absolute control rates are  $0.01 \text{ s}^{-1}$ ,  $-0.05 \text{ deg/s}$ , and  $0.02 \text{ s}^{-1}$ , for  $\dot{C}_L$ ,  $\dot{\mu}$ , and  $\dot{\delta}_T$ , respectively. Neverthe-

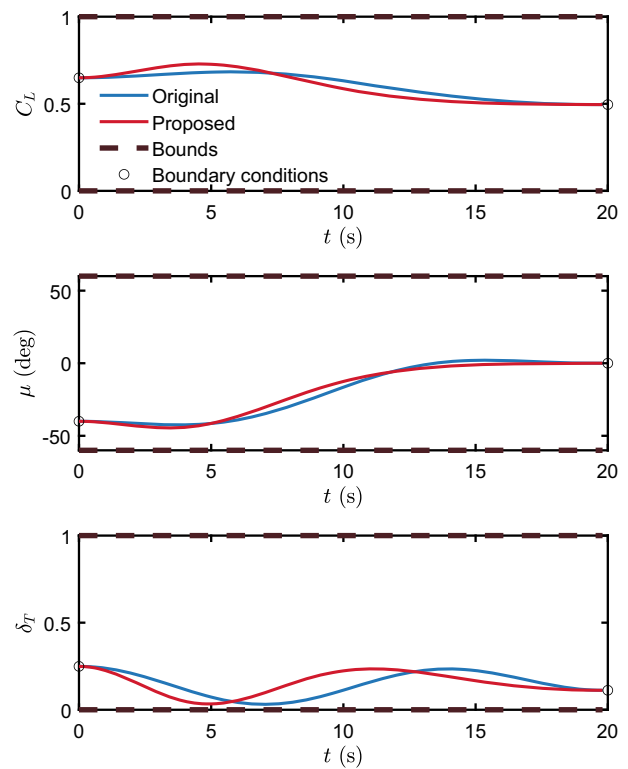


Fig. 4 Time histories of the controls

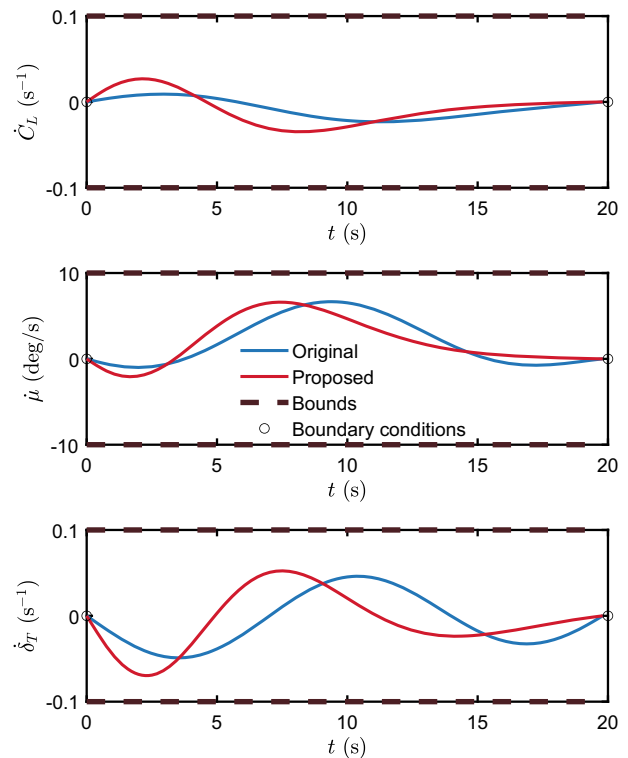
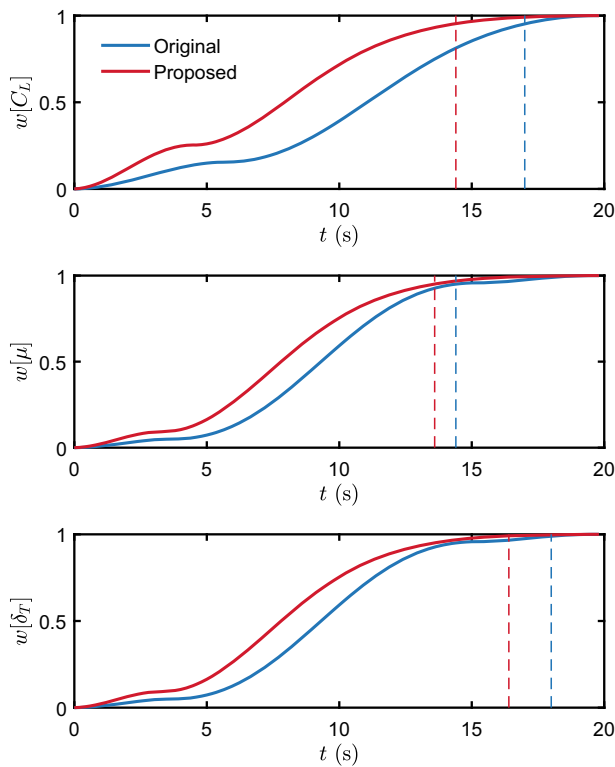


Fig. 5 Time histories of the control derivatives



**Fig. 6** Time histories of the normalized cumulative sums of the control derivatives

less, the control derivatives are smooth and well within the bounds. In fact, the mathematical property of the trigonometric functions guarantees the existence and continuity of arbitrary derivatives. The derivatives are set to be zero at both ends of the transition trajectory, which is achieved by both methods.

Next, we introduce an indicator to quantify completeness of the transition over time. This gives a very intuitive view of the shifted distribution of the control derivatives using the proposed method. The indicator is expressed by

$$w_k = \frac{\sum_{j=1}^k (\dot{u}_j)^2}{\sum_{j=1}^{N_t-1} (\dot{u}_j)^2}, \quad k = 1, 2, \dots, N_t - 1. \quad (35)$$

The indicator stands for the normalized cumulative sum of the control derivative over time, which is depicted in Fig. 6. The vertical lines identify where the indicator reaches 95%. The indicators of the proposed method approach the condition earlier than those of the original method. Specifically for  $C_L$ , the indicator reaches 95% 2.6 s earlier for the proposed method, which is already 13% of the 20 s time horizon for the transition maneuver. The differences of  $\mu$  and  $\delta_T$  are found to be 4% and 8%, respectively. As the new independent variable  $v$  has been designed to be of a slow-varying nature, the results of the proposed method and the original method are not vastly different, and yet we can observe the effectiveness of the

proposed method. In fact, the difference can be more visible if a very aggressive time-scaling function had been selected. However, it is beyond the purpose of this application that requires an overall smooth transition.

It is to be noted that there can be different ways of designing the time-scaling function and the time-scaled trigonometric series. The shown application only represents one specific concern that has been satisfactorily addressed.

### 5 Conclusions

For shaping the time behavior of the controls in generating trajectories, this paper develops a new time-scaled trigonometric series control parameterization method. The proposed method can incorporate a user-defined time-scaling function that alters the fundamental time behavior of the basis functions such that the controls inherit the desired characteristic. It has been shown that in the model predictive convex optimization framework, the modification required for using the newly developed method is minor. For the transition maneuver scenario in this paper, the aircraft is required to maneuver more gently in the later stage of the transition trajectory, for which the proposed method introduces a smooth, decreasing time-scaling function. Simulations are carried out to show the effectiveness of the proposed method and the comparison to the original method demonstrates that the proposed method accomplished the task well.

**Funding** This work is supported 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). Open Access funding enabled and organized by Projekt DEAL.

### Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### References

1. Ben-Asher JZ (2010) Optimal control theory with aerospace applications. Am Inst Aeronaut Astron

2. Chai R, Savvaris A, Tsourdos A, Chai S, Xia Y (2019) A review of optimization techniques in spacecraft flight trajectory design. *Progr Aerosp Sci* 109(100):543
3. Chen Z (2020) On dubins paths to a circle. *Automatica* 117(108):996
4. Chi Z, Li H, Jiang F, Li J (2018) Power-limited low-thrust trajectory optimization with operation point detection. *Astrophys Space Sci* 363(6):1–12
5. Gerdts M (2011) Optimal control of ODEs and DAEs. Walter de Gruyter
6. Gierszewski DM, Schneider V, Lauffs PJ, Peter L, Holzapfel F (2018) Clothoid-augmented online trajectory generation for radius to fix turns. *IFAC-Pap OnLine* 51(9):174–179
7. He S, Shin HS, Tsourdos A (2020) Trajectory optimization for multitarget tracking using joint probabilistic data association filter. *J Guid Control Dyn* 43(1):170–178
8. Hong H, Maity A, Holzapfel F, Tang S (2019) Model predictive convex programming for constrained vehicle guidance. *IEEE Trans Aerosp Electron Syst* 55(5):2487–2500
9. Hong H, Piprek P, Afonso RJM, Holzapfel F (2021) Hierarchical parameterization for seamless transition trajectory generation. *IEEE Trans Aerosp Electron Syst* 57(5):2878–2888
10. Hong H, Piprek P, Afonso RJM, Holzapfel F (2021) Trigonometric series-based smooth flight trajectory generation. *IEEE Trans Aerosp Electron Syst* 57(1):721–728
11. Hong H, Piprek P, Gerdts M, Holzapfel F (2021) Computationally efficient trajectory generation for smooth aircraft flight level changes. *J Guid Control Dyn* 44(8):1532–1540
12. Hong H, Tekin R, Holzapfel F (2021) Guaranteed smooth trajectory generation for field-of-view constrained impact-time control. *J Guid Control Dyn* 44(4):898–904
13. Hong H, Grüter B, Piprek P, Holzapfel F (2022) Smooth free-cycle dynamic soaring in unspecified shear wind via quadratic programming. *Chin J Aeronaut* 35(7):19–29
14. Kumar P, Padhi R (2014) Extension of model predictive static programming for reference command tracking. *IFAC Proc Vol* 47(1):855–861
15. Lee H, Teo K, Rehbock V, Jennings L (1997) Control parametrization enhancing technique for time optimal control problems. *Dynam Syst Appl* 6:243–262
16. Li H, Chen S, Baoyin H (2018) J2-perturbed multitarget rendezvous optimization with low thrust. *J Guid Control Dyn* 41(3):802–808
17. Niermeyer P, Akkinapalli VS, Pak M, Holzapfel F, Lohmann B (2016) Geometric path following control for multirotor vehicles using nonlinear model predictive control and 3d spline paths. In: 2016 International conference on unmanned aircraft systems (ICUAS), IEEE, pp 126–134
18. Piprek P, Marb MM, Bhardwaj P, Holzapfel F (2020) Trajectory/path-following controller based on nonlinear jerk-level error dynamics. *Appl Sci* 10(23):8760
19. Rao AV (2014) Trajectory optimization: a survey. In: Optimization and optimal control in automotive systems, Springer, New York, pp 3–21
20. Schneider V, Mumm NC, Holzapfel F (2015) Trajectory generation for an integrated mission management system. In: 2015 IEEE international conference on aerospace electronics and remote sensing technology (ICARES), IEEE, pp 1–7
21. Shanmugavel M, Tsourdos A, White B, Żbikowski R (2010) Co-operative path planning of multiple uavs using dubins paths with clothoid arcs. *Control Eng Pract* 18(9):1084–1092
22. Stellato B, Banjac G, Goulart P, Bemporad A, Boyd S (2020) OSQP: an operator splitting solver for quadratic programs. *Math Program Comput*. <https://doi.org/10.1007/s12532-020-00179-2>