doi: 10.82124/CEAS-GNC-2026-100

EuroGNC
Madrid, Spain May 5th-7th 2026

Online Trajectory Generation with Adaptive Set Point Filters for Target Trajectory Tracking

Alexander Funke Research Associate, German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, Germany. alexander.funke@dlr.de

Philipp Schitz Research Associate, German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, Germany. philipp.schitz@dlr.de

ABSTRACT

The problem of online trajectory generation is a common challenge when dealing with moving target tracking. Offline planning is often impractical when the target's intentions are not clear in advance. When using model predictive control for the application of trajectory tracking control, the trajectory towards the target can be generated as part of the optimal control problem. This, however, often requires large control horizons and can lead to numerical divergence of the solver. To counteract these effects, simple first-order lag elements are commonly used in industry to create reference trajectories, which steadily approach the target set point. However, they typically do not take system limitations into account, which can lead to infeasible trajectories. We present an approach utilizing higher-order lag elements with adaptive pole placement based on the tracking error to keep the generated trajectories smooth and feasible. We demonstrate the simplicity of this approach using a three dimensional double integrator example, which serves as a representative model for many feedback linearizable mechanical systems. The adaptive lag elements can be used to quickly generate smooth and feasible trajectories while enforcing constraints on its derivatives. We show that the tracking performance of the presented approach corresponds to or even improves upon the direct use of the unfiltered target trajectory while at the same time improving the smoothness of the actuator commands.

Keywords: Trajectory Generation, Set Point Filter, Model Predictive Control, Reference Governor, Rendezvous

1 Introduction

Target trajectory tracking is a common and broad-ranging problem in control. A particular case of this problem is rendezvous control, which concerns the coordinated motion of two or more vehicles. Rendezvous has been studied in a variety of contexts like automated aerial refueling or in-air capturing of aircraft and reentry vehicles [1], heterogeneous scenarios involving aircraft and ground vehicles [2, 3] or in space [4]. To accurately follow the target, a trajectory must first bring the follower to the target to then be able to track the target's subsequent motion. Model predictive control (MPC) is a widely employed technique for generating such trajectories while ensuring that the follower meanwhile obeys input and system constraints. Because infeasible target trajectories can provoke instabilities in MPC, it is often accompanied by a filter that smooths the desired target trajectory given a reference model [5].

The general concept of set point modification using reference models can be roughly broken down into two categories. Set point filters are used to smooth the set point progression over time. They have their origins in classical PID-control and are used to dampen sudden set point changes to stabilize the system, limit actuator aggressiveness, and avoid overshoot [6]. Set point filters can also take constraints of the system into account to ensure feasibility of the generated trajectory. This is usually done by saturating the derivatives in the filter dynamics. They are also often used in MPC to improve feasibility of the optimal control problem (OCP) [5, 7]. Reference governors additionally take the current and possibly also future system states into account to adapt the reference only when necessary to preserve system performance. They can also incorporate future constraints but usually rely on solving an optimization problem [8]. Reference governors are often implemented as low-pass filters with adjustable bandwidth parameter which is determined online as in [9]. MPC itself can also be used as an outer supervisory layer to act as a reference governor for lower level controllers. The generated optimal trajectories ensure safety of the system [10]. Furthermore, artificial references can directly be incorporated in the optimization problem as decision variables when dealing with tracking problems [11].

Reference models are also widely used in flatness-based control techniques which involve system model inversion and in model reference adaptive control (MRAC). In [12], the reference model used in MRAC is modified by feeding back the tracking error to prevent the system from initially attempting to maneuver aggressively towards it. The reference model approaches its original form as the tracking error approaches zero. In this work, we follow a similar idea for the use case of reference trajectory generation. We use a higher-order variable lag element as a set point filter which depends on the tracking error, i.e., the distance to the target, to quickly generate new reference trajectories in each time step. For the application of target trajectory tracking, we use the most recent target state estimate to repeatedly generate updated target trajectories which are fed into the filter. The closer the system gets to the target position, the faster the filter dynamics become until they match with a static unit gain and thus make the reference trajectory match the predicted target trajectory. The generated trajectory can then be used as reference in trajectory tracking control like traditional PID control, MPC, or, since higher-order derivatives are now available, even as a full-state set point to a differential-flatness-based controller as for example in [13]. For many mechanical systems in trajectory tracking applications, there exist equivalent differentiable flat system descriptions (e.g. quadrotor [14], Dubins airplane [15], kinematic bicycle [16]). If continuous differentiability is required for the chosen control architecture, the order of the variable set point filter must be chosen based on the system's relative degree.

Our method is a pragmatic approach designed to quickly generate feasible reference trajectories in receding horizon fashion via step response shaping of a higher-order lag element for tracking of time-varying target trajectories. The generated trajectories account for constraints in the derivatives of the controlled variable. This allows us to simplify the optimization problem in MPC by separating state constraint satisfaction from closed-loop stabilization. In comparison to traditional reference governors, our method does not require constrained online optimization, yet still respects the constraints on the derivatives while preserving smoothness of the generated trajectories. Although demonstrated here for generating reference trajectories for MPC, this approach can also be applied to other trajectory tracking methods, thereby complementing a broad range of controllers to satisfy state constraints. When approaching the set point, the time constant of the lag element converges to zero such that the typically observed phase delay of a lag element diminishes. The adaptive set point filter alters the target trajectory increasingly as the tracking error grows which allows for more aggressive controller tuning without the need for gain scheduling.

Depending on the order of the set point filter used, the presented method requires the availability of the target states and derivatives to generate a sufficiently differentiable target trajectory. If, for example, the velocity shall be constrained, a second-order filter is being used, which requires observation of target velocity and acceleration. Naturally, the target derivatives cannot be larger than the constraints imposed on the reference trajectory as otherwise the filter may become unstable. Our method can only be used to encode explicit constraints of any absolute or relative states of feedback linearized differentially flat systems in Isidori normal form (e.g. the velocity if the position is a flat output). It is not suitable for constraining combinations thereof that cannot be directly deduced from the derivatives of the flat output.

By using this method, which is described in detail in Section 3, we can easily improve the closed loop command response and numerical stability of the OCP without adding a lot of computational overhead. We explore the applicability of our method in Section 4 based on a simple MPC-controlled three dimensional double integrator example, which serves as a representative model for the class of feedback linearizable mechanical systems.

2 Problem Statement

Consider a feedback linearizable system in Isidori normal form (cf. [17]) with relative degree n

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f(x) + g(x)u \end{aligned} \tag{1}$$

and output $y = x_1$ which shall follow a trajectory generated by propagating the system equation of a target system with the same form as in Eq. (1). Both systems are initialized with distinct initial states. Moreover, the target's input can change arbitrarily and is not known in advance. The target's states are observable and an estimate is available.

Our objective is to find an n-times differentiable reference trajectory that guides the follower system smoothly to align with the target trajectory while adhering to constraints on the derivatives of the output $(\dot{x}_1 \ldots \dot{x}_{n-1})$. To account for changes in the input of the target system, we regenerate the reference trajectory regularly in receding horizon fashion as depicted in Fig. 1.

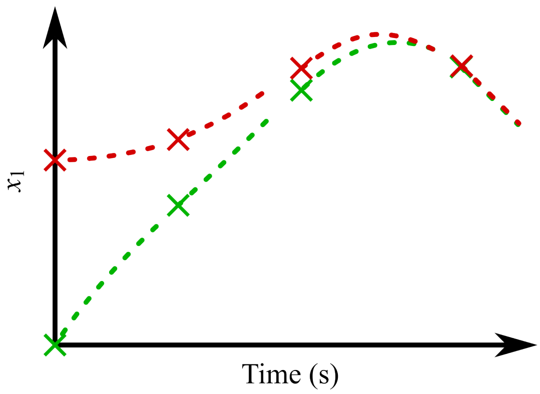

Fig. 1 Follower (green) approaching target (red) by following generated reference trajectories with derivative constraint based on set point filtered predicted target trajectories in receding horizon fashion.

3 Trajectory Generation with Adaptive Set Point Filters

When a set point is given, smooth trajectories towards this set point can be easily generated with n^{th}-order lag elements, which are also commonly referred to as PTn-elements or n^{th}-order low-pass filters. They account for sudden set point changes and smooth out noise at the cost of phase delay and diminished tracking accuracy. Without modifications, these trajectories usually do not satisfy constraints of the mechanical system that has to perform the given task. A commonly applied technique in set point filtering involves saturating the derivatives to account for the capabilities of the mechanical system. This, however, will inevitably cause discontinuities in the higher-order derivatives when running into the specified limits. In this work, we instead shape the step response of the lag element by adaptive pole placement such that it does not exceed given constraints. The poles are determined by maximizing the step response derivatives in the time-domain. For trajectory generation, we discretize the transfer function, add initial conditions for the filter states and consider target movement as feed-forward term in the resulting difference equations. In the following sections, we derive the poles based on the example of a feedback linearized mechanical system which features the position as a flat output.

Adaptive Set Point Filters

By adapting the time constant based on the distance to the target, we can constrain the trajectory's derivatives without saturation and therefore generate a smooth trajectory. Depending on the constraint set, various strategies are possible. For simple velocity constrained systems, a first-order lag element can often be sufficient. The transfer function and the continuous time step response of a first-order lag used for trajectory generation are given by

$$f_1(t) = \mathscr{L}^{-1}\left\{\frac{d}{s}\,G_1(s)\right\} = d\left(1 - e^{-t/T}\right) \quad \text{with } G_1(s) = \frac{1}{Ts+1} \tag{2}$$

where $t \geq 0$ is the time, d is the distance to target and T is the time constant. The velocity on the trajectory is given by the first time derivative of Eq. (2)

$$f_1'(t) = \frac{d}{T}e^{-t/T}. \tag{3}$$

The maximum velocity $\overline{v} \in \mathbb{R}^+$ of the trajectory generated with the first-order lag element can be observed right at the start ($t = 0$) because the absolute function value of Eq. (3) is strictly monotonously decreasing. By rearranging, we obtain the value for the distance-dependent time constant

$$\overline{v} = f_1'(0) = \frac{d}{T} \quad \Rightarrow \quad T = \frac{d}{\overline{v}}. \tag{4}$$

When dealing with second-order constraints or when a smooth velocity transition is required, a second-order lag has to be used. We compose the second-order lag with the transfer function

$$G_2(s) = \frac{1}{(T_1 s + 1)(T_2 s + 1)} \tag{5}$$

from two first-order elements with their respective time constants T_1 and T_2. For this system and subsequent higher-order systems, a closed solution, which provides the time constants given the absolute velocity $\overline{v}$ and acceleration $\overline{a}$ maxima, cannot be derived algebraically because the maximized time derivatives are transcendental equations. For those cases where a solution exists, it is still possible to solve the resulting problem numerically, which we describe in Appendix A. Instead, we choose to make the simplifying assumption of critical damping $T_1 = T_2$ and only actively limit one of the two derivatives at a time by introducing a piecewise affine function for the remaining time constant T. By applying the inverse Laplace transformation, the simplified transfer function leads to the step response

$$f_2(t) = \mathscr{L}^{-1}\left\{\frac{d}{s}\,G_2(s)\right\} = d\left(1 - e^{-t/T}(1 + t/T)\right) \quad \text{with } G_2(s) = \frac{1}{(Ts+1)^2} \tag{6}$$

The first and second time derivatives of the step response are then

$$f_2'(t) = \frac{d}{T^2}\,t\,e^{-t/T}, \quad f_2''(t) = \frac{d}{T^2}\,e^{-t/T}\,(1 - t/T)\,. \tag{7}$$

Setting the second derivative to zero yields the time $t_{\overline{v}}$ at which the first derivative reaches its maximum

$$\frac{d}{T^2}\, e^{-t/T}\,(1 - t/T) = 0 \quad \Rightarrow \quad t_{\overline{v}} = T. \tag{8}$$

The respective time constants for either a velocity constrained system or for an acceleration constrained system are then given as

$$\overline{v} = f_2'(t_{\overline{v}}) = \frac{d}{Te} \quad \Rightarrow \; T_{\overline{v}} = \frac{d}{\overline{v}e} \qquad \text{for } |v(t)| < \overline{v} \in \mathbb{R}^+,\; a(t) \in \mathbb{R}^+, \tag{9}$$

$$\overline{a} = f_2''(0) = \frac{d}{T^2} \quad \Rightarrow \; T_{\overline{a}} = \sqrt{\frac{d}{\overline{a}}} \qquad \text{for } |a(t)| < \overline{a} \in \mathbb{R}^+,\; v(t) \in \mathbb{R}^+. \tag{10}$$

The maximum acceleration reached on the horizon in the velocity constrained case and the maximum velocity reached in the acceleration constrained case can be formulated as a function of the distance to the target by substituting the time constant in Eq. (9) from Eq. (10) and vice versa

$$\overline{a}(d) = \frac{(\overline{v}e)^2}{d} \text{ for } |v(t)| < \overline{v} \in \mathbb{R}^+, \quad \overline{v}(d) = \frac{\sqrt{\overline{a}d}}{e} \text{ for } |a(t)| < \overline{a} \in \mathbb{R}^+. \tag{11}$$

If both derivatives shall be constrained, we can now define a piecewise affine function for the variable time constant by rearranging Eq. (11) and isolating $d \in \mathbb{R}^+$ to determine the intersection point

$$d^\star = \frac{(\overline{v}e)^2}{\overline{a}} \quad \Rightarrow \quad T = \max\bigl(T_{\overline{v}}, T_{\overline{a}}\bigr) = \begin{cases} \sqrt{\frac{d}{\overline{a}}} & \text{if } d \le d^\star \\ \frac{d}{\overline{v}e} & \text{if } d > d^\star \end{cases}, \tag{12}$$

which will satisfy both constraints. The variable time constant in Eq. (12) is a continuous non-smooth function with its intersection point $d^\star$ and value $T^\star = \overline{v}/\overline{a}\, e$, at which both constraints become active at the same time.

For higher-order lag elements, maximizing the derivatives of the step response grows increasingly complex. In this work we present the method to up to the third-order. Extending it for higher-order filters is possible by following the same steps as before. For a third-order system, the step response is

$$f_3(t) = \mathscr{L}^{-1}\left\{\frac{d}{s}\, G_3(s)\right\} = d\left(1 - e^{-t/T}(1 + t/T + t^2/2T^2)\right) \quad \text{with } G_3(s) = \frac{1}{(Ts+1)^3} \tag{13}$$

The first three time derivatives of the step response are then

$$f_3'(t) = \frac{d}{2T^3}\, t^2\, e^{-t/T}, \quad f_3''(t) = \frac{d}{T^3}\, t e^{-t/T}\,(1 - t/2T), \quad f_3'''(t) = \frac{d}{T^3}\, e^{-t/T}\left(1 - 2t/T + t^2/2T^2\right). \tag{14}$$

Maximizing the derivatives yields

$$\overline{v} = f_3'(t_{\overline{v}}) \; = \frac{2d}{Te^2} \qquad \Rightarrow T_{\overline{v}} = \frac{2d}{\overline{v}e^2} \qquad \text{where } t_{\overline{v}} = 2T, \tag{15}$$

$$\overline{a} = f_3''(t_{\overline{a}}) \; = \frac{d}{T^2}\left(\sqrt{2} - 1\right) e^{\sqrt{2}-2} \quad \Rightarrow T_{\overline{a}} = \sqrt{\frac{d}{\overline{a}}\left(\sqrt{2}-1\right) e^{\sqrt{2}-2}} \quad \text{where } t_{\overline{a}} = 2T - \sqrt{2}T, \tag{16}$$

$$\overline{j} = f_3'''(t_{\overline{j}}) = \frac{d}{T^3} \qquad \Rightarrow T_{\overline{j}} = \sqrt[3]{\frac{d}{\overline{j}}} \qquad \text{where } t_{\overline{j}} = 0 \tag{17}$$

by respectively setting the higher-order derivatives to zero and solving for t. Finally, to adhere to all constraints, the time constants are recalculated at each time step, and the largest – and thus slowest – time constant is selected and retained on the whole trajectory generation horizon.

$$T = \max\left(T_{\overline{v}}, T_{\overline{a}}, T_{\overline{j}}\right). \tag{18}$$

The safe region to choose a time constant from, using a third-order lag element, is illustrated in Fig. 2 for exemplary constraints. It can be observed that with decreasing distance to the target, the time constant vanishes, which effectively transforms the transfer function of the adaptive set point filter into a unit gain. The target trajectory is then directly tracked.

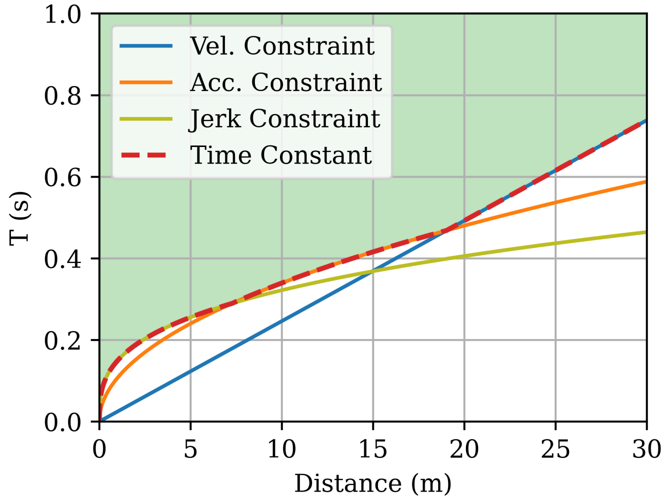

Fig. 2 Distance dependent variable time constants with exemplary velocity ($\overline{v}$ = 11 m/s), acceleration ($\overline{a}$ = 20 m/s^2), and jerk ($\overline{j}$ = 300 m/s^3) constraints. Choosing a time constant from the green area guarantees constraint adherence of the generated trajectory.

Initial Conditions

Our proposed method requires recomputing the trajectories during execution to account for changes of the target trajectory. Therefore, we must take care of the initial conditions of the generated trajectory, which namely are the $n-1$ derivatives of the mechanical system for a n^{th}-order lag element. To shape the step response considering the initial conditions, we add additional zeros to the transfer functions. We present the derivation of the adjusted step response using the example of the second-order lag

$$G_2^{\circ}(s) = \frac{mTs+1}{(Ts+1)^2} \tag{19}$$

with the design parameter m. By equating the initial values of the step response and first derivative

$$f_2^{\circ}(t) = \mathscr{L}^{-1}\left\{\frac{d}{s}G_2^{\circ}(s)\right\} = d\left(1 - e^{-t/T}\left(1 + {}^{t}\!/\!{}_{T}(1-m)\right)\right), \tag{20}$$

$$f_2^{\circ\prime}(t) = \frac{d}{T^2}e^{-t/T}\left(t(1-m) + Tm\right) \tag{21}$$

with the initial condition

$$v_0 = f_2^{\circ\prime}(0) = \frac{d}{T}m \quad \Rightarrow \quad m = \frac{T}{d}v_0 \tag{22}$$

where v_0 is the initial velocity, we receive the transfer function

$$G_2^{\circ}(s) = \frac{{}^{v_0}\!/\!{}_{d}\,T^2 s + 1}{(Ts+1)^2} \tag{23}$$

by rearranging Eq. (22) for m and inserting it in the original transfer function in Eq. (19). Equally, by substituting m in the step response (20), we receive

$$f_2^{\circ}(t) = d - e^{-t/T}\big(d + t/T(d - Tv_0)\big). \tag{24}$$

Altering the transfer function by adding zeros will also change the maximum values of the derivatives. In particular, it may happen that the maximum velocity is already reached at the start. Hence, it is necessary to adjust the time constant accordingly. Similar to before, we receive the function extremum in the interval $t \in (0, \infty)$ by setting the second derivative to zero, solving for $t_{\overline{v}} > 0$ and inserting into the first derivative

$$f_2^{\circ\prime}(t_{\overline{v}}) = (d/T - v_0)\, e^{-\frac{d-2Tv_0}{d-Tv_0}} \quad \text{with } t_{\overline{v}} = \frac{d - 2Tv_0}{d - Tv_0} T. \tag{25}$$

It is necessary to check if the velocity at time $t = 0$ is already larger than the function extremum at time $t_{\overline{v}}$ to obtain the global maximum. By equating Eq. (25) with the initial velocity v_0 at time $t = 0$, we derive the intervals

$$\max_{t \in [0,\infty)} |v(t)| = \begin{cases} v_0 & \text{if } v_0 \geq d/2T \\ -v_0 & \text{if } v_0 \leq -0.386\, d/T \\ f_2^{\circ\prime}(t_{\overline{v}}) & \text{if } -0.386\, d/T < v_0 < d/2T \end{cases} \qquad \text{for } |v(t)| \leq \overline{v} \in \mathbb{R}^+. \tag{26}$$

Depending on the magnitude of the initial velocity, the maximum velocity may become independent of the time constant, which then can be chosen to shape the second derivative instead. Eq. (25) is a transcendental equation and therefore cannot be solved algebraically for T. We can use the Newton-Raphson method presented in Algorithm 1 to find the root(s) for the velocity constrained case. We use the function $g(T) = f_2^{\circ\prime}(t_{\overline{v}})$ from Eq. (25) and its first derivative

$$g'(T) = -d/T^2\, \frac{d-2Tv_0}{d-Tv_0}\, e^{-\frac{d-2Tv_0}{d-Tv_0}}. \tag{27}$$

In the very first time step, we initialize the algorithm with the time constant from Eq. (9) and thereafter with the Newton-Raphson solution of the previous time step, which in most cases results in a small number of iterations – often only one for sufficiently small sampling intervals – to solve the root finding problem with an acceptable tolerance ε. The tolerance can be chosen moderately based on sample time T_s and maximum velocity $\overline{v}$. Typical values we used range from $\varepsilon = 1\text{e}{-4}\,\text{s}$ to $1\text{e}{-2}\,\text{s}$, but always depend on the specific use case. The maximum number of iterations $N_{\max}$ is hardware dependent and should be chosen such that the computation time does not exceed a fraction of the sample time.

Algorithm 1 Newton-Raphson method to determine time constant $T_{\overline{v}}$ in a second-order lag element

Require: Function $g(T)$, its derivative $g'(T)$, velocity constraint $\overline{v}$, solution from previous time step T_{k-1}, tolerance ε, maximum number of iterations $N_{\max}$
Ensure: Approximate root $T_{\overline{v}}$ of $g(T) - \overline{v} = 0$

$T_k \leftarrow T_{k-1}$
for $n \leftarrow 0$ to $N_{\max}$ **do**
  $T_k^* \leftarrow T_k - \frac{g(T_k) - \overline{v}}{g'(T_k)}$
  if $|T_k^* - T_k| < \varepsilon$ **then**
    return T_k^*
  end if
  $T_k \leftarrow T_k^*$
end for

As with velocity, when considering the acceleration constraint, we have to evaluate the second derivative at time $t_{\overline{a}}$ which is obtained by setting the third derivative equal to zero and solving for $t_{\overline{a}} > 0$

$$f_2^{\circ\prime\prime}(t_{\overline{a}}) = (v_0/T - d/T^2)\, e^{-\frac{2d-3Tv_0}{d-Tv_0}} \quad \text{with } t_{\overline{a}} = \frac{2d - 3Tv_0}{d - Tv_0} T \tag{28}$$

Similar to before, Eq. (28) has to be solved iteratively for $T_{\overline{a}}$, e.g. with the Newton-Raphson method. By again setting the local acceleration maximum in the interval of $t \in (0, \infty)$ from Eq. (28) to the initial acceleration a_0 at $t = 0$, we obtain the intervals for the global acceleration maximum

$$\max_{t \in [0,\infty)} |a(t)| = \begin{cases} -a_0 & \text{if } v_0 \geq {}^{2d}/_{3T} \\ a_0 & \text{if } v_0 \leq 0.419\, {}^{d}/_{T} \\ -f_2^{\circ\prime\prime}(t_{\overline{a}}) & \text{if } 0.419\, {}^{d}/_{T} < v_0 < {}^{2d}/_{3T} \end{cases} \qquad \text{for } |a(t)| \leq \overline{a} \in \mathbb{R}^+ \tag{29}$$

$$\text{where } a_0 = f_2^{\circ\prime\prime}(0) = \frac{d - 2Tv_0}{T^2} \quad \Rightarrow \quad T = \begin{cases} {}^{d}/_{2v_0} & \text{if } a_0 = 0 \\ \sqrt{{}^{d}/_{a_0} + {}^{v_0^2}/_{a_0^2}} - {}^{v_0}/_{a_0} & \text{if } a_0 > {}^{-v_0^2}/_{d} \end{cases} \tag{30}$$

This case distinction is necessary because, depending on the initial velocity and the current distance, it is possible that the maximum acceleration no longer occurs at time $t = 0$ compared to the critically damped second-order lag element without zeros (cf. Eq. (6)). This is especially the case for $d \to 2Tv_0$ and when the initial acceleration a_0 in Eq. (30) approaches zero.

Instead of constraining velocity and acceleration, it is also possible to force initial acceleration by explicitly using the time constant from Eq. (30) at all times. In this case, it is not necessary to numerically solve a root finding problem. This time constant can also be used when the initial velocity is larger than the proposed velocity constraint to push the velocity back within the feasible range.

When acceleration constraints are desired, while at the same time providing the initial velocity and acceleration (e.g. in a system of relative degree three), a third-order lag element must be used. The transfer function and step response

$$G_3^{\circ}(s) = \frac{{}^{v_0}/_{d}(Ts+3)T^2 s + {}^{a_0}/_{d}T^3 s + 1}{(Ts+1)^3}, \tag{31}$$

$$f_3^{\circ}(t) = d - e^{-t/T}\left(d + {}^{t}/_{T}(d - Tv_0) + {}^{t^2}/_{2T^2}(d - 2Tv_0 - T^2 a_0)\right) \tag{32}$$

are derived based on the equations of the second-order lag element (23) and (24) and the previously derived third-order step response from Eq. (13). The equations to maximize the derivatives are given in Appendix B.

Trajectory Generation

To iteratively generate a reference trajectory towards a moving target, we first estimate the target trajectory for a given horizon length N based on a simplified target model by observing the states of the target and then propagating them via the system equations. In each time step, the time constant will be reevaluated based on the current states of the target and the follower. Then, the target trajectory is fed into the adaptive set point filter to generate the reference trajectory relative to the tracking system. In order to apply the presented algorithm, the transfer function of the adaptive set point filter is discretized using zero-order hold discretization. In the discrete time domain, the initial states are specified by careful initialization of the filter states. The time constant remains the same as in the time continuous domain. It is then possible to either initialize the filter based on current position, velocity and acceleration or based on a sequence of past position measurements. For the latter, a smooth position estimate is essential. When the tracked target is moving, we also have to take the target derivatives into account to eliminate steady state errors. Additionally to the target position p_t, the target velocity v_t, acceleration a_t, and jerk j_t are added as another input to the difference equations. In continuous time domain, this essentially relates to feeding forward the target derivatives into the internal filter states. The difference equations for first- ①, second- ②, and third- ③ order lag elements are then given by

$$① \; y_{k+1} = \left(1 - e^{-T_s/T}\right) \; (p_{t_k} + T v_{t_k}) + e^{-T_s/T} y_k \quad \text{with} \;\; y_0 = p_0, \tag{33}$$

$$② \; y_{k+2} = \left(1 - e^{-T_s/T}\right)^2 \; (p_{t_k} + 2T v_{t_k} + T^2 a_{t_k}) + 2e^{-T_s/T} y_{k+1} - e^{-2T_s/T} y_k \\ \text{with} \;\; y_0 = p_0 - T_s v_0, \;\; y_1 = p_0, \tag{34}$$

$$③ \; y_{k+3} = \left(1 - e^{-T_s/T}\right)^3 \; (p_{t_k} + 3T v_{t_k} + 3T^2 a_{t_k} + T^3 j_{t_k}) + 3e^{-T_s/T} y_{k+2} - 3e^{-2T_s/T} y_{k+1} + e^{-3T_s/T} y_k \\ \text{with} \;\; y_0 = p_0 - 2T_s v_0 - 2T_s^2 a_0, \;\; y_1 = p_0 - T_s v_0 - 0.5 T_s^2 a_0, \;\; y_2 = p_0 \tag{35}$$

to recursively generate the reference trajectory for each time step $k \in \{0, \ldots, N\}$ with horizon length N.

The previously presented method to determine the time constants in the continuous time domain is used to generate a trajectory relative to the states of the follower, position p_p, velocity v_p and acceleration a_p. Therefore, the initial conditions also refer to relative derivative values between tracked target and follower (e.g. relative velocity) and must be initialized accordingly

$$v_0 = v_{p_0} - v_{t_0}, \qquad a_0 = a_{p_0} - a_{t_0}. \tag{36}$$

Additionally, if the absolute derivatives of the follower shall be constrained (e.g. the follower jerk j_p), the constraints used in the derivation of the time constants also need to be adapted as shown in Table 1.

Table 1 Adaption of constraints given absolute constraints of follower derivatives

Ord.	Vel. Constraint $\overline{v}$	Acc. Constraint $\overline{a}$	Jerk Constraint $\overline{j}$
1^{st}	$\overline{v}_p - v_{t_0}$		
2^{nd}	$\overline{v}_p - \left(v_{t_0} + t_{\overline{v}} a_{t_0}\right)$	$\overline{a}_p - a_{t_0}$	
3^{rd}	$\overline{v}_p - \left(v_{t_0} + t_{\overline{v}} a_{t_0} + {}^1\!/_2\, t_{\overline{v}}^2 j_{t_0}\right)$	$\overline{a}_p - \left(a_{t_0} + t_{\overline{a}} j_{t_0}\right)$	$\overline{j}_p - j_{t_0}$

Table 2 Summary of online trajectory generation methods with n^{th}-order lag elements

Ord.	Unit Step Response	Difference Equation	Constraint	Initial Time Constant
1^{st}	$1 - e^{-t/T}$	$y_{k+1} = \left(1 - e^{-T_s/T}\right)(p_{t_k} + T v_{t_k}) + e^{-T_s/T} y_k$	Velocity	$\frac{d}{\overline{v}}$ (4)
2^{nd}	$1 - e^{-t/T}(1 + {}^t\!/_T)$	$y_{k+2} = \left(1 - e^{-T_s/T}\right)^2 (p_{t_k} + 2T v_{t_k} + T^2 a_{t_k}) + 2e^{-T_s/T} y_{k+1} - e^{-2T_s/T} y_k$	Velocity	$\frac{d}{\overline{v}e}$ (9)
			Acceleration	$\sqrt{\frac{d}{\overline{a}}}$ (10)
3^{rd}	$1 - e^{-t/T}(1 + {}^t\!/_T + {}^{t^2}\!/_{2T^2})$	$y_{k+3} = \left(1 - e^{-T_s/T}\right)^3 (p_{t_k} + 3T v_{t_k} + 3T^2 a_{t_k} + T^3 j_{t_k}) + 3e^{-T_s/T} y_{k+2} - 3e^{-2T_s/T} y_{k+1} + e^{-3T_s/T} y_k$	Velocity	$\frac{2d}{\overline{v}e^2}$ (15)
			Acceleration	$\sqrt{\frac{d}{\overline{a}}} \cdot 0.48$ (16)
			Jerk	$\sqrt[3]{\frac{d}{\overline{j}}}$ (17)
n^{th}	$1 - e^{-t/T} \sum_{k=0}^{n-1} ({}^t\!/_T)^k \frac{1}{k!}$	$y_{k+n} = \left(1 - e^{-T_s/T}\right)^n \sum_{m=0}^{n} \binom{n}{m} T^m f_t^m(k) + \sum_{m=1}^{n} (-1)^{m+1} \binom{n}{m} e^{-T_s/T\, m} y_{k+n-m}$	n^{th} Derivative	$\sqrt[n]{\frac{d}{\overline{f^n(t)}}}$

For targets that are faster than the velocity constraint of the follower, trajectory generation is not possible as the time constant becomes negative, which translates to unstable poles in the right half-plane. It is therefore important to ensure that the resulting constraints are always positive or alternatively to saturate the time constant. For very large distances to the target, it can be useful to ignore the velocity constraint at first and use a time constant based on the initial acceleration from Eq. (30) to get up to speed and prevent slow excitation with large time constants. However, it must be ensured that the commanded initial acceleration will not cause velocity constraint violation within the horizon.

Table 2 gives an overview of the here-presented higher-order set point filter trajectory generation methods with their respective time constants depending on the chosen constraints.

4 Results

In this section, we first verify our approach using a one-dimensional toy example assuming perfect model following with first- and second-order set point filters with active velocity constraints. We compare the generated trajectories and time constant profiles of the two different filters and verify constraint adherence. In the second part, we apply our algorithm with an adaptive second-order set point filter to a three-dimensional trajectory tracking problem utilizing model predictive control. There, we use a simple linear second-order integrator chain for simulation, which also often results from feedback linearizing differentially flat mechanical systems.

Toy Example

For illustrative purposes and to verify the presented approach, we generate trajectories towards a target using Eqs. (33) to (35) and assume perfect model following. This means, the position of the follower is always set to the first generated trajectory point, which implies no target prediction ($N = 1$). When using this approach, it is important to introduce a lower bound for the time constant, such that for small distances the filter dynamics will not grow infinitely fast and the follower slowly adapts to the target trajectory. We use a sigmoid-like convergence function

$$T \leftarrow \frac{T}{\tanh(\underline{T}^{-1}T)} \tag{37}$$

with the lower bound $\underline{T}$ to gradually limit the time constant. Introducing this scaling mechanism will not cause constraint violation as the time constant remains within the safe range when it is increased compared to the originally calculated value (c.f. Fig. 2). This adaption is only necessary when the horizon length is chosen small and the system states and inputs are generated directly from the reference trajectory. When using the generated trajectories as input for MPC on the other hand, adapting the time constant is not required.

In Fig. 3, the simulated trajectories of a target and a follower for a one dimensional problem under the assumption of perfect model following are depicted. We apply first- and second-order lag elements as set point filters and regenerate trajectories in every time step with a sample time of 0.01 s. We perform two independent simulation runs for each filter, constraining the absolute velocity of the follower in one and the relative velocity $v_p - v_t$ in the other. The target's movement is based on a decaying sinusoidal wave with a short resting phase at the end.

① Constraining the velocity of the follower will lead to linearly approaching the target with the maximum velocity. The initial velocity of the follower cannot be taken into account because in a first-order system the velocity is not continuous. The velocity constraints are properly adhered to in both relative and absolute constrained cases. The time constants are linearly decreasing until limited by the soft distance saturation. This trajectory generation method has only limited use for

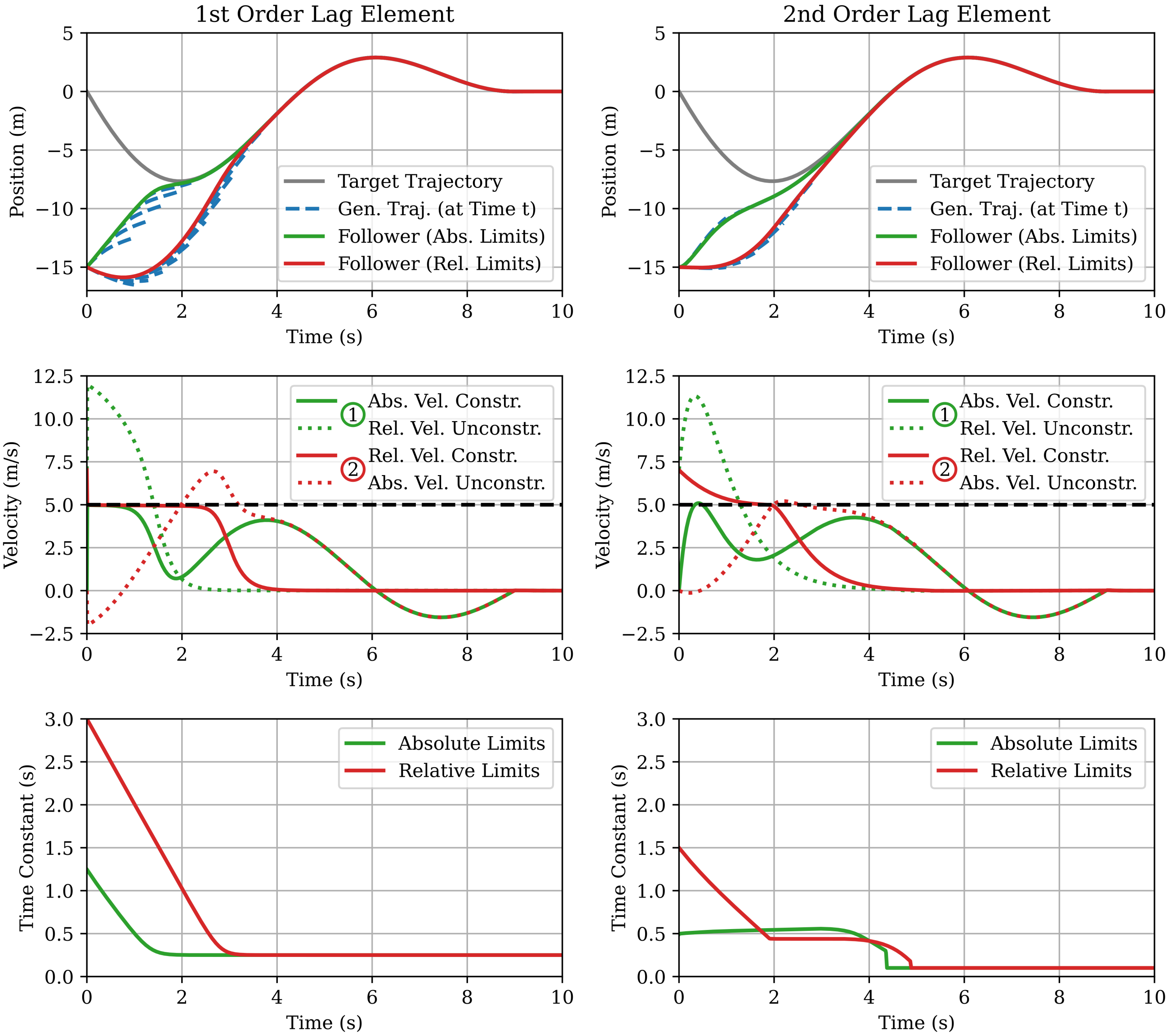

Fig. 3 Generated trajectories with absolute (1) and relative velocity (2) constraints ($\overline{v} = 5\,\mathrm{m/s}$) with first- and second-order reference models. The upper graphs illustrate the simulated positions of the target and the follower. Below, the absolute and relative velocities are depicted. At the bottom, the distance dependent time constants are shown for both cases.

smooth trajectory generation as most mechanical systems require continuous velocity progression (esp. at $t = 0$). It is still suitable for trajectory generation for trajectory tracking MPC and easily implemented without need for numerical root-finding.

② In contrast to the first-order system, the velocity progression is continuous when using a second-order lag element as set point filter. In case of the absolute velocity constraint, the velocity smoothly approaches the constraint and then slowly converges to the target velocity. In case of the relative velocity constraint, the absolute value of the initial relative velocity $v_{t_0} - v_{p_0}$ is already larger than allowed. The target is moving towards the follower, so the time constant is chosen such that the follower position is nearly constant until the target velocity falls below the constraint. From this point onward, the relative velocity converges to zero as the distance decreases. In comparison to the first-order case, the time constants attenuate following a square root-like trend with respect to distance.

Double Integrator MPC Example

We apply the presented set point filtering approach to a three-dimensional trajectory tracking problem. As underlying system model, we use a simple second-order integrator chain with position and velocity as states $\boldsymbol{x}$, position as controlled variable $\boldsymbol{y}$, and acceleration as control variable $\boldsymbol{u}$. For closed loop control, we implemented linear MPC using the open source framework acados [18]. The horizon length is set to $N = 25$ with a total prediction time of $t_f = 0.5$ s. As before, we evaluate the set point filters and the controller with a sample time of $T_s = 0.1$ s. The linear least-squares weights of the cost function

$$\boldsymbol{J} = 0.5\,||\boldsymbol{y} - \boldsymbol{y}_{\text{ref}}||^2_{\boldsymbol{Q}} + 0.5\,||\boldsymbol{u}||^2_{\boldsymbol{R}} \tag{38}$$

are choosen as $\boldsymbol{Q} = \text{diag}(1000, 1000, 1000)$ and $\boldsymbol{R} = \text{diag}(1, 1, 1)$ to accurately follow the given reference. We constraint the controls within the OCP to $\pm 20\,\text{m/s}^2$ to prevent infinite excitation.

Two different implementations are compared in Fig. 4, where a moving target in 10 m altitude shall be tracked. The target is moving with a constant velocity of 5 m/s in x-direction and with the same decaying sinusoidal motion as before in y-direction. The first implementation uses the target trajectory directly as reference $\boldsymbol{y}_{\text{ref}}$. It is generated by propagating the target states with the currently observed velocity and acceleration into the future. The second one, referred to as adaptive set-point-filtered MPC (ASF-MPC), feeds the target trajectory into a second-order adaptive set point filter with active velocity constraint $\overline{v} = 10\,\text{m/s}$ and then uses the result as position reference. Both controllers are tuned with the same weights and parameters. The only difference is the reference trajectory handling.

In Fig. 4, we can observe that the adaptive set point filter is effective in constraining the follower velocity to the given value of 10 m/s in comparison to the unconstrained MPC implementation. Although we only use a second-order set point filter, the resulting trajectory is smooth. We also tested first-order set point filters and the same behavior can be observed. The control action is more smooth in comparison to an explicitly state-constrained MPC. Smoothing out the control action can also be achieved in traditional MPC by increasing the weights on the control inputs R but this comes at the cost of tracking performance and remaining steady-state errors. In the given example, the resulting follower trajectory of the unmodified MPC is overshooting by roughly 7 %. Due to the faster maximum velocity, the target initially is reached earlier, but it takes 0.8 s for the overshoot to subside. This results in 0.5 s slower settling time in comparison to the here-presented set point filtering approach, with both implementations only showing minimal steady-state errors of ~3 cm. Overshoot could be reduced by tweaking down control aggressiveness via the weighting matrices, extending the prediction horizon, or by introducing velocity constraints in the OCP. However, the first measure will degrade agility and tracking performance if used without distance dependent gain scheduling while the latter two come at the cost of expanding the OCP with a significant amount of additional decision variables e.g. state constraints and corresponding slack variables. Furthermore, we have observed that, depending on the weights in the cost function, the solver of the unmodified MPC does not converge for very large distances to the target due to numerical issues. This is particularly the case with a condensed OCP formulation and long shooting intervals, but can also be observed with a sparse OCP formulation.

The time constants of the adaptive set point filter gradually decrease until reaching their lower limit which we have set to the sample time $\underline{T} = 0.01$ s. This helps with arising numerical issues when $T \rightarrow 0$ s. In most cases, the number of Newton-Raphson iterations required is less than three for a specified tolerance of $\varepsilon = 1\text{e}{-4}$ s, with the exception of the first step, where it reaches four in the case of the x- and y-axes and seven in the case of the z-axis. These numbers can be easily improved by using larger tolerances. In our example, we observed that increasing the tolerances to up to $1\text{e}{-2}$ s does not evidently change the behavior of the filters and limits the number of maximum iterations to one except for the initial case.

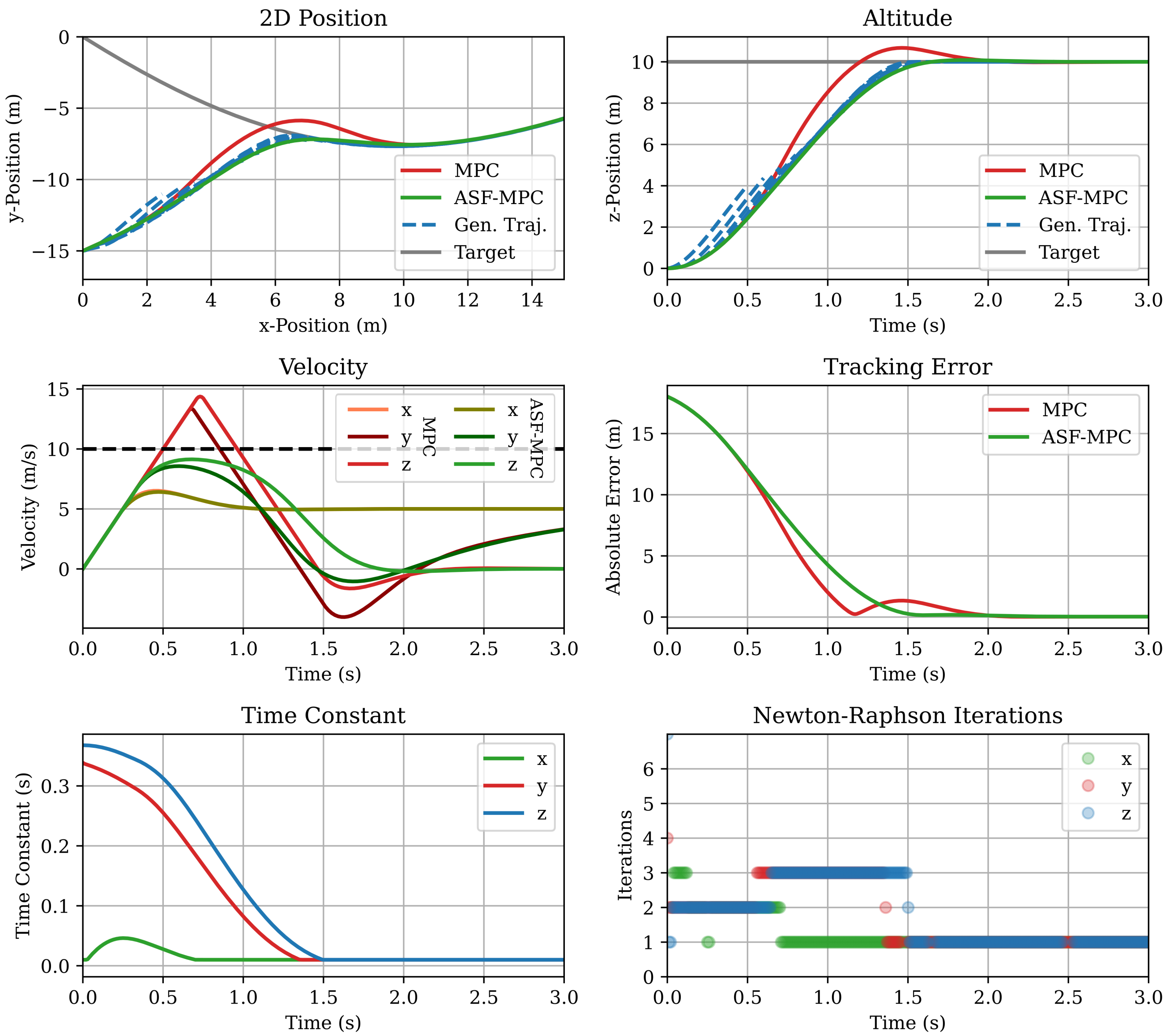

Fig. 4 Comparison of classical MPC without set point filter and adaptive second-order set-point-filtered MPC (ASF-MPC) with velocity constraint $(10\,\mathrm{m/s})$**. The upper graphs show the simulated positions of the target and the follower. Below, the velocities and control errors are depicted. At the bottom, the distance dependent time constants are shown and the Newton-Raphson iterations necessary to solve the root-finding problem.**

5 Discussion

We have shown that by introducing adaptive higher-order set point filters for reference trajectory generation, we can smoothly limit the derivatives of the mechanical system inherently while preserving and in some cases even improving the control performance. When using MPC for trajectory tracking, adaptive filtering of the target trajectory additionally can help with infeasibility problems. The presented approach does not require additional state estimation in comparison to the typical requirements of state space controllers. However, for accurate target prediction, fast and accurate target state estimation is essential to properly track highly dynamic targets. An up-to-date estimate of at least the second derivative should be available in order to accurately predict the target movement in the period of the control horizon.

The presented first- and second-order set point filters are simple to implement and can easily extend already existing controller architectures for trajectory tracking. Additional computational overhead by solving a root-finding problem is kept to a minimum by tuning the acceptance tolerance of the root-finding algorithm. In comparison to traditional set point filters, the generated trajectories are both feasible and smooth at the same time. We can only partially recommend selecting adaptive lag elements of order three and strongly advise against using any higher-order for set point filtering with the goal of strict constraint satisfaction. Although this is theoretically possible, the implementation effort and the computational cost are not negligible because of disproportionately complex equations as in Eqs. (45) to (47).

Systems with nonholonomic constraints like fixed-wings or most ground vehicles with wheels need to be taken special care of when applying the variable set point filtering approach. One idea to explore in future work is to calculate the trajectories in body-fixed frame of the follower and to impose stricter derivative constraints in lateral than in longitudinal direction according to the system specification. Nonetheless, this approach still requires that the follower is already roughly pointing in target direction.

6 Conclusion

In this work we presented a method for iteratively generating feasible reference trajectories based on higher-order adaptive reference models that act as time-variable set point filters for trajectory tracking control. We derived the variable pole placement for filters up to third order, using the tracking error, current target states, and the follower's initial conditions. The method was applied to a three-dimensional control problem with a double-integrator model, demonstrating that control performance was not degraded despite smoothing of the controls. We additionally observed that, for large distances, numerical conditioning of the optimal control problem improves substantially. The presented approach has no significant impact on computing time and is capable to be easily integrated into existing controller architectures. In future work, we will analyze the application of this method to rendezvous control and evaluate its performance in flight tests.

Appendix A

The step response with magnitude d and the corresponding derivatives of a second-order lag are

$$f_2(t) = \mathscr{L}^{-1}\left\{\frac{d}{s}\,G_2(s)\right\} = \frac{d}{T_1 - T_2}\left(T_1\left(1 - e^{-t/T_1}\right) - T_2\left(1 - e^{-t/T_2}\right)\right), \tag{39}$$

$$f_2'(t) = \frac{d}{T_1 - T_2}\left(e^{-t/T_1} - e^{-t/T_2}\right), \tag{40}$$

$$f_2''(t) = \frac{d}{T_1 - T_2}\left({}^1\!/\!{}_{T_2}e^{-t/T_2} - {}^1\!/\!{}_{T_1}e^{-t/T_1}\right). \tag{41}$$

Setting the second derivative to zero yields the time at which the first derivative reaches its maximum

$$0 = \frac{d}{T_1 - T_2}\left({}^1\!/\!{}_{T_2}e^{-t/T_2} - {}^1\!/\!{}_{T_1}e^{-t/T_1}\right) \;\Rightarrow\; t_{\overline{v}} = \frac{T_1 T_2}{T_1 - T_2}\log({}^{T_1}\!/\!{}_{T_2}). \tag{42}$$

Inserting Eq. (42) into Eq. (40) yields the first, inserting $t = 0$, the time of maximum acceleration, into Eq. (41) the second condition in the system of equations

$$\overline{v} = f_2'(t_{\overline{v}}) = \frac{d}{T_1 - T_2}\left(e^{-\frac{T_2}{T_1 - T_2}log({}^{T_1}\!/\!{}_{T_2})} - e^{-\frac{T_1}{T_1 - T_2}log({}^{T_1}\!/\!{}_{T_2})}\right) = d\,T_1^{-\frac{T_1}{T_1-T_2}}\,T_2^{\frac{T_2}{T_1-T_2}} \tag{43}$$

$$\overline{a} = f_2''(0) = \frac{d}{T_1 - T_2}\left(\frac{1}{T_2} - \frac{1}{T_1}\right) = \frac{d}{T_1 T_2} \tag{44}$$

to solve for the time constants T_1 and T_2, which shape the step response according to the maximum absolute velocity $\overline{v}$ and acceleration $\overline{a}$.

Appendix B

The velocity, acceleration and jerk constraints in a third-order lag element are given by

$$\begin{aligned}
&\overline{v} = \max(|f_3^{\circ\prime}(0)|, |f_3^{\circ\prime}(t_{\overline{v}})|) \quad \text{where} \quad f_3^{\circ\prime}(0) = v_0 \quad \text{and} \\
&f_3^{\circ\prime}(t_{\overline{v}}) = \frac{\left(T^3 a_0 t_{\overline{v}} + T^3 v_0 - \frac{T^2 a_0 t_{\overline{v}}^2}{2} + T^2 t_{\overline{v}} v_0 - T t_{\overline{v}}^2 v_0 + \frac{d t_{\overline{v}}^2}{2}\right) e^{-\frac{t_{\overline{v}}}{T}}}{T^3} \\
&\text{with } t_{\overline{v}} = \frac{T\left(2T^2 a_0 + 3T v_0 - d + \sqrt{2T^4 a_0^2 + 8T^3 a_0 v_0 - 2T^2 a_0 d + 9T^2 v_0^2 - 6T d v_0 + d^2}\right)}{T^2 a_0 + 2T v_0 - d},
\end{aligned} \tag{45}$$

$$\begin{aligned}
&\overline{a} = \max(|f_3^{\circ\prime\prime}(0)|, |f_3^{\circ\prime\prime}(t_{\overline{a}})|) \quad \text{where} \quad f_3^{\circ\prime\prime}(0) = a_0 \quad \text{and} \\
&f_3^{\circ\prime\prime}(t_{\overline{a}}) = \frac{\left(T^4 a_0 - 2T^3 a_0 t_{\overline{a}} + \frac{T^2 a_0 t_{\overline{a}}^2}{2} - 3T^2 t_{\overline{a}} v_0 + T d t_{\overline{a}} + T t_{\overline{a}}^2 v_0 - \frac{d t_{\overline{a}}^2}{2}\right) e^{-\frac{t_{\overline{a}}}{T}}}{T^4} \\
&\text{with } t_{\overline{a}} = \frac{T\left(3T^2 a_0 + 5T v_0 - 2d + \sqrt{3T^4 a_0^2 + 12T^3 a_0 v_0 - 4T^2 a_0 d + 13T^2 v_0^2 - 10T d v_0 + 2d^2}\right)}{T^2 a_0 + 2T v_0 - d},
\end{aligned} \tag{46}$$

$$\begin{aligned}
&\overline{j} = \max(|f_3^{\circ\prime\prime\prime}(0)|, |f_3^{\circ\prime\prime\prime}(t_{\overline{j}})|) \quad \text{where} \quad f_3^{\circ\prime\prime\prime}(0) = \frac{d - 3T(v_0 + T a_0)}{T^3} \quad \text{and} \\
&f_3^{\circ\prime\prime\prime}(t_{\overline{j}}) = \frac{\left(-3T^4 a_0 + 3T^3 a_0 t - 3T^3 v_0 - \frac{T^2 a_0 t^2}{2} + T^2 d + 5T^2 t v_0 - 2T d t - T t^2 v_0 + \frac{d t^2}{2}\right) e^{-\frac{t}{T}}}{T^5} \\
&\text{with } t_{\overline{j}} = \frac{T\left(4T^2 a_0 + 7T v_0 - 3d + \sqrt{4T^4 a_0^2 + 16T^3 a_0 v_0 - 6T^2 a_0 d + 17T^2 v_0^2 - 14T d v_0 + 3d^2}\right)}{T^2 a_0 + 2T v_0 - d}.
\end{aligned} \tag{47}$$

Declaration of Use of Artificial Intelligence

In this work, artificial intelligence was partially used to proofread and refine language and grammar. The relevant passages have been carefully reviewed to ensure that the underlying meaning has not been changed. The authorship of the work, the underlying ideas, and their presentation remain solely those of the authors of the paper.

References

[1] Sven Stappert, Sunayna Singh, Alexander Funke, and Martin Sippel. Developing an innovative and high-performance method for recovering reusable launcher stages: the in-air capturing method. *CEAS Space Journal*, 16(4):411–426, 2024. doi: 10.1007/s12567-023-00512-0.

[2] Alessandro Rucco, PB Sujit, A Pedro Aguiar, Joao Borges De Sousa, and F Lobo Pereira. Optimal rendezvous trajectory for unmanned aerial-ground vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 54(2):834–847, 2017. doi: 10.1109/TAES.2017.2767958.

[3] Christoph Hebisch and Dirk Abel. Model predictive trajectory tracking for a ground vehicle in a heterogeneous rendezvous with a fixed-wing aircraft. *IFAC-PapersOnLine*, 53(2):15693–15698, 2020. doi: 10.1016/j.ifacol.2020.12.2563.

[4] Avishai Weiss, Morgan Baldwin, Richard Scott Erwin, and Ilya Kolmanovsky. Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. *IEEE Transactions on Control Systems Technology*, 23(4):1638–1647, 2015. doi: 10.1109/TCST.2014.2379639.

[5] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, 2021. doi: 10.1007/s00170-021-07682-3.

[6] Karl Johan Åström and Tore Hägglund. *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, 2006. ISBN: 978-1-55617-942-6.

[7] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226, London, 1999. Springer London. ISBN: 978-1-84628-538-7. doi: 10.1007/BFb0109870.

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

[9] Elmer G. Gilbert and Ilya Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, 9(15):1117–1141, 1999. doi: 10.1002/(SICI)1099-1239(19991230)9:15<1117::AID-RNC447>3.0.CO;2-I.

[10] Martin Klaučo and Michal Kvasnica. *MPC-Based Reference Governors*. Springer Cham, 2019. ISBN: 978-3-030-17405-7. doi: 10.1007/978-3-030-17405-7.

[11] P. Krupa, J. Köhler, A. Ferramosca, I. Alvarado, M.N. Zeilinger, T. Alamo, and D. Limon. Model predictive control for tracking using artificial references: Fundamentals, recent results and practical implementation. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 2977–2991, 2024. doi: 10.1109/CDC56724.2024.10886854.

[12] Vahram Stepanyan and Kalmanje Krishnakumar. Adaptive control with reference model modification. *Journal of Guidance, Control, and Dynamics*, 35(4):1370–1374, 2012. doi: 10.2514/1.55756.

[13] Michiel J. Van Nieuwstadt and Richard M. Murray. Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11):995–1020, 1998. doi: 10.1002/(SICI)1099-1239(199809)8:11<995::AID-RNC373>3.0.CO;2-W.

[14] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011. doi: 10.1109/ICRA.2011.5980409.

[15] Hamidreza Chitsaz and Steven M. LaValle. Time-optimal paths for a dubins airplane. In *2007 46th IEEE Conference on Decision and Control*, pages 2379–2384, 2007. doi: 10.1109/CDC.2007.4434966.

[16] S. Fuchshumer, K. Schlacher, and T. Rittenschober. Nonlinear vehicle dynamics control - a flatness based approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6492–6497, 2005. doi: 10.1109/CDC.2005.1583203.

[17] Alberto Isidori. *Elementary Theory of Nonlinear Feedback for Single-Input Single-Output Systems*, pages 137–217. Springer London, London, 1995. ISBN: 978-1-84628-615-5. doi: 10.1007/978-1-84628-615-5_4.

[18] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021. doi: 10.1007/s12532-021-00208-8.