

SENSITIVITY-BASED REFERENCE TRAJECTORY UPDATES FOR AIRCRAFT PATH-FOLLOWING MODEL PREDICTIVE CONTROL

Simon Fuchsgruber[1], Matthias Gerdts[2] & Florian Holzapfel[1]

[1]Institute of Flight System Dynamics, Technical University of Munich, Germany
[2]Institute of Applied Mathematics and Scientific Computing, University of the Bundeswehr Munich, Germany

Abstract

This paper deals with the post-optimal sensitivity analysis of parametric trajectory optimization problems in order to carry out sensitivity-based reference trajectory updates for path-following Model Predictive Control (MPC) of a fixed-wing aircraft. These updates are performed with low computational cost when deviations from the reference path occur. It is shown that following this updated reference trajectory is more beneficial in terms of optimality than keeping track of the old one. The proposed combination of MPC and reference trajectory updating is particularly interesting when the reference path cannot be precisely followed due to constraints such as obstacles or no-fly zones.

Keywords: parametric optimization, post-optimal sensitivity analysis, path-following model predictive control, online trajectory updates

1. Introduction

In autonomous aircraft missions, a two-level approach to plan and execute optimized trajectories is often used. In a pre-planning phase before the flight, a trajectory is computed on a potentially highly detailed level satisfying all requirements already known at this stage of the mission. Afterwards, this trajectory serves as a reference for a controller, which forms the second level of the approach.
In order to plan trajectories offline, optimal control methods can be applied [1, 2]. A wide range of goals can be considered and formulated by this kind of trajectory optimization, e.g., minimizing energy consumption or flight time. Moreover, multiple constraints, such as state and control bounds, dynamical system constraints, and mission-specific initial and final boundary conditions, can be taken into account.
Often during flight, changes to the pre-planned flight path are necessary, e.g., due to emerging no-fly zones or previously unknown obstacles. These issues can be addressed by the use of Model Predictive Control (MPC) [3, 4] as a path-following controller, since it can handle newly added constraints. An MPC-based path-following controller for a fixed-wing military aircraft, for example, was implemented in [5]. However, these constraints may cause deviations from the reference trajectory, e.g., if the pre-planned path passes through a previously unknown no-fly zone, leading to non-optimal trajectories with respect to the cost function of the pre-planning task. Hence, when a deviation occurs, it is desired to update the reference trajectory with a new optimal trajectory taking the current state as an initial boundary condition. Following this updated reference by the MPC is more beneficial in terms of optimality than keeping track of the old one as well as possible.
Updating the trajectory by solving an optimal control problem over the entire time horizon again is time-consuming and, therefore, not a good option during the online process. Instead of fully re-optimizing parametric optimization problems for a given parameter perturbation by running an optimization algorithm, approximated solutions based on post-optimal sensitivity information were introduced in [6] for small perturbations.

In the paper at hand, a path-following MPC algorithm is combined with these sensitivity-based approximations in order to update the reference trajectory. These updates can be quickly computed online, requiring only matrix vector multiplications and additions, after a post-optimal sensitivity analysis was performed as part of the pre-planning task. Moreover, the necessary computation of the sensitivity differentials for multiple trajectory tails is done in an efficient manner through an appropriate partition of the involved matrices. Furthermore, the combination with an MPC algorithm allows sensitivity-based approximations for larger perturbations in practice, as possible constraint violations in the approximated solution caused by changes in the active constraint set are prevented. The incorporated constraints of the MPC ensure that the overall resulting trajectory satisfies the constraints, even if there have been constraint violations due to the approximation.
The paper is structured as follows: In section 2., the theoretical background of optimal control, sensitivity-based approximations, and MPC is presented. The application to an aircraft path-following MPC and a discussion of the results are provided in section 3.. Finally, in section 4., some conclusions are drawn.

2. Methodology

This section first introduces the theory of trajectory optimization via optimal control methods in subsection 2.1. Then, the sensitivity theorem and the computation of the sensitivity-based trajectory updates are explained in subsection 2.2, before the general MPC algorithm and its underlying optimization problem are presented in subsection 2.3.

2.1 Trajectory Optimization via Optimal Control

Consider an optimal control problem (OCP) formulated as

$$\begin{aligned} \text{minimize} \quad & \int_{t_0}^{t_f} F(\mathbf{x}(t),\mathbf{u}(t))\, dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)), \quad t \in [t_0,t_f], \\ & \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t)) \leq \mathbf{0}, \quad t \in [t_0,t_f], \\ & \mathbf{x}(t_0) = \mathbf{x}_0, \\ & \Psi(\mathbf{x}(t_f)) = \mathbf{0} \end{aligned} \tag{1}$$

with states $\mathbf{x}$ and controls $\mathbf{u}$ on a time horizon defined by a fixed initial time t_0 and a free final time parameter t_f. Moreover, F denotes a Lagrange cost function, and the optimization is subject to dynamic constraints, path constraints, and initial and final boundary conditions, where $\mathbf{x}_0$ defines the initial state.
Furthermore, consider the following parametric OCP in discrete time and note that OCP$(0,\mathbf{x}_0)$ is a discretization of the problem in (1) using trapezoidal collocation as transcription method [2]:

$$\begin{aligned} \text{OCP}(n,\mathbf{p}): \quad & \min_{\mathbf{z}_n} \sum_{k=n}^{N-1} \frac{h}{2}(F(\mathbf{x}_{k+1},\mathbf{u}_{k+1}) + F(\mathbf{x}_k,\mathbf{u}_k)) \\ \text{subject to} \quad & \mathbf{x}_{k+1} - \mathbf{x}_k = \frac{h}{2}(\mathbf{f}(\mathbf{x}_{k+1},\mathbf{u}_{k+1}) + \mathbf{f}(\mathbf{x}_k,\mathbf{u}_k)), \quad k = n,\dots,N-1, \\ & \mathbf{g}(\mathbf{x}_k,\mathbf{u}_k) \leq \mathbf{0}, \quad k = n,\dots,N, \\ & \mathbf{x}_n = \mathbf{p}, \\ & \Psi(\mathbf{x}_N) = \mathbf{0}. \end{aligned} \tag{2}$$

Herein, $\mathbf{z}_n := (\mathbf{x}_n,\mathbf{u}_n,\mathbf{x}_{n+1},\mathbf{u}_{n+1},\dots,\mathbf{x}_N,\mathbf{u}_N,t_N)^\top$ collects the optimization variables, while $h := t_{k+1} - t_k$ is the equidistant time discretization. OCP$(n,\mathbf{p})$ are viewed as parametric optimization problems, where the parameter vector $\mathbf{p}$ denotes the initial state value, and $n \in \{0,...,N-1\}$ distinguishes between multiple trajectory tails with corresponding starting time t_n. A parametric sensitivity analysis with respect to $\mathbf{p}$ will be performed in the following.
This transcribed problem is a finite-dimensional nonlinear programming problem (NLP) and can be solved by common NLP solvers. Moreover, let $\hat{\mathbf{z}}_0$ be an optimal solution of OCP$(0,\mathbf{x}_0)$, then the

trajectory tails $\hat{\mathbf{z}}_n$ are optimal for the problems OCP$(n, \hat{\mathbf{x}}_n)$ for $n \in \{0, ..., N-1\}$ according to Bellman's optimality principle. Therefore, it is sufficient to solve only OCP$(0, \mathbf{x}_0)$ to obtain optimal solutions for OCP$(n, \hat{\mathbf{x}}_n)$. This property is exploited when computing the sensitivity differentials.

2.2 Sensitivity-based Trajectory Updates

The originally computed solution of OCP$(0, \mathbf{x}_0)$ will serve as a reference trajectory in the following. In case it is desired to update the trajectory at time t_n, results of a parametric sensitivity analysis of the trajectory tails with respect to the initial states can be applied. To do so, properties of the parameter-to-solution mapping $\mathbf{p} \mapsto \mathbf{z}_n^*(\mathbf{p})$ are examined, where $\mathbf{z}_n^*(\mathbf{p})$ denotes an optimal solution of OCP$(n, \mathbf{p})$. Under conditions where $\mathbf{z}_n^*$ is continuously differentiable, it is possible to perform a linearization

$$\mathbf{z}_n^*(\mathbf{p}) \approx \tilde{\mathbf{z}}_n(\mathbf{p}) := \mathbf{z}_n^*(\bar{\mathbf{p}}) + \frac{\mathrm{d}\mathbf{z}_n^*}{\mathrm{d}\mathbf{p}}(\bar{\mathbf{p}})(\mathbf{p} - \bar{\mathbf{p}}) \tag{3}$$

around a nominal parameter $\bar{\mathbf{p}}$ to approximate a perturbed solution $\mathbf{z}_n^*(\mathbf{p})$, see [6].
The differentiability of the parameter-to-solution mapping and the computation of the sensitivity differentials $\frac{\mathrm{d}\mathbf{z}_n^*}{\mathrm{d}\mathbf{p}}(\bar{\mathbf{p}})$ is given by the sensitivity theorem established in [7]. Let $\hat{\mathbf{z}}_n$ be a local minimum of OCP$(n, \bar{\mathbf{p}})$. Under the assumption that all cost and constraint functions are twice differentiable and that $\hat{\mathbf{z}}_n$ is a strongly regular solution (see [1, Definition 6.1.2]), the sensitivity matrix can be obtained as solution of the linear equation system

$$\begin{pmatrix} \nabla^2_{\mathbf{z}_n \mathbf{z}_n} L(\hat{\mathbf{z}}_n, \hat{\eta}_n, \bar{\mathbf{p}}) & \mathbf{G}'_{\mathbf{z}_n}(\hat{\mathbf{z}}_n, \bar{\mathbf{p}})^\top \\ \mathbf{G}'_{\mathbf{z}_n}(\hat{\mathbf{z}}_n, \bar{\mathbf{p}}) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}\mathbf{z}_n^*}{\mathrm{d}\mathbf{p}}(\bar{\mathbf{p}}) \\ \frac{\mathrm{d}\eta_n^*}{\mathrm{d}\mathbf{p}}(\bar{\mathbf{p}}) \end{pmatrix} = - \begin{pmatrix} \mathbf{0} \\ \mathbf{G}'_{\mathbf{p}}(\hat{\mathbf{z}}_n, \bar{\mathbf{p}}) \end{pmatrix}, \tag{4}$$

where L denotes the Lagrangian associated to OCP$(n, \mathbf{p})$ with corresponding multipliers η_n and

$$\mathbf{G}(\mathbf{z}_n, \mathbf{p}) = \begin{pmatrix} \frac{h}{2}(\mathbf{f}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) + \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)) - (\mathbf{x}_{k+1} - \mathbf{x}_k),\ k = n, \dots, N-1 \\ \mathbf{g}_{A(\mathbf{z}_n(k))}(\mathbf{x}_k, \mathbf{u}_k),\ k = n, \dots, N \\ \mathbf{x}_n - \mathbf{p} \\ \Psi(\mathbf{x}_N) \end{pmatrix} \tag{5}$$

with $A(\mathbf{z}_n(k))$ denoting the index set of active path constraints. A detailed version of the sensitivity theorem and a proof can be found in [7] or in [1, Theorem 6.1.4]. Moreover, all the assumptions for a strongly regular solution can be checked numerically.
In order to perform the sensitivity-based trajectory updates, execute the following steps:

- Solve OCP$(0, \mathbf{x}_0)$ to obtain the solution trajectory $\hat{\mathbf{z}}_0$ and due to Bellman's optimality principle also $\hat{\mathbf{z}}_n$ for $n \in \{0, ..., N-1\}$.
- For $n \in \{0, ..., N-1\}$ perform a sensitivity analyisis for OCP$(n, \hat{\mathbf{x}}_n)$ and compute the sensitivity differentials $\frac{\mathrm{d}\mathbf{z}_n^*}{\mathrm{d}\mathbf{p}}(\hat{\mathbf{x}}_n)$ by solving the linear equation system in (4).
- For a perturbation $\tilde{\mathbf{p}}$ of the state $\mathbf{x}_n$ at time t_n update the trajectory by the approximation $\tilde{\mathbf{z}}_n(\tilde{\mathbf{p}})$ from (3).

Note that it is sufficient to derive the matrices required in (4) only for $n = 0$ and re-use parts of them for the subsequent problems OCP$(n, \hat{\mathbf{x}}_n)$. To do so, an appropriate partition of the matrices is used. Therefore, the involved submatrices are considered one by one.
The Lagrangian $L(\mathbf{z}_n, \eta_n, \mathbf{p})$ is defined as a sum of the objective function and the constraints weighted by the multipliers η_n. Here, the individual terms of the objective and the dynamic constraints for each index $k \in \{n, ..., N-1\}$ depend on the state and control variables of exactly two timesteps, t_k and t_{k+1}, because of the discretization by trapezoidal rule. This leads to a block diagonal structure of the Lagrangian hessian $\nabla^2_{\mathbf{z}_n \mathbf{z}_n} L(\mathbf{z}_n, \eta_n, \mathbf{p})$, where it can be observed that the block of matrix entries referring to objective and constraint terms of index k overlaps with those blocks referring to $k-1$ and $k+1$. In order to prevent this overlapping, the Lagrangian is partitioned in terms referring to an even

and an odd index k respectively, denoted by $L_{\text{even}}(\mathbf{z}_n, \eta_n, \mathbf{p})$ and $L_{\text{odd}}(\mathbf{z}_n, \eta_n, \mathbf{p})$. By this, a partition of the Lagrangian hessian of the following structure is obtained:

$$\nabla^2_{\mathbf{z}_n\mathbf{z}_n} L(\mathbf{z}_n, \eta_n, \mathbf{p}) = \begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \ddots \\ & & & & & \times \\ & & & & & & \times \end{pmatrix} \tag{6}$$

$$= \underbrace{\begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \ddots \\ & & & & & \times \end{pmatrix}}_{\nabla^2_{\mathbf{z}_n\mathbf{z}_n} L_{\text{even}}(\mathbf{z}_n, \eta_n, \mathbf{p})} + \underbrace{\begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \ddots & \\ & & & & \times \\ & & & & & \times \end{pmatrix}}_{\nabla^2_{\mathbf{z}_n\mathbf{z}_n} L_{\text{odd}}(\mathbf{z}_n, \eta_n, \mathbf{p})}.$$

Here, the blue blocks indicate the matrix entries referring to terms of an even index, while the red ones indicate those referring to an odd index.
The structure of the submatrix $\mathbf{G}'_{\mathbf{z}_n}(\mathbf{z}_n, \mathbf{p})$ is given by

$$\mathbf{G}'_{\mathbf{z}_n}(\mathbf{z}_n, \mathbf{p}) = \begin{pmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & & \ddots & & \\ & & & & \times & \times \\ \times & & & & & \\ & \times & & & & \\ & & & \ddots & & \\ & & & & & \times \\ \times & & & & & \\ & & & & & \times \end{pmatrix}, \tag{7}$$

where the first rows correspond to the dynamic constraints depending on state and control variables of two timesteps, while the remaining constraints only depend on variables of one time instance. Again, the blocks referring to constraints of an even or odd index are indicated accordingly. The last two rows correspond to the initial and final boundary conditions, with the first resulting in an identity matrix block as matrix entry. For the submatrix $\mathbf{G}'_{\mathbf{p}}(\mathbf{z}_n, \mathbf{p})$, a zero matrix except for a negated identity matrix block corresponding to the initial condition is obtained.
Given this matrix structure and the provided partition of the Lagrangian hessian, the matrices in (4) for the subsequent problems OCP$(n, \hat{\mathbf{x}}_n)$ for $n \in \{1, ..., N-1\}$ can be obtained by discarding all blocks referring to indices 0 to $n-1$ from the matrices for problem OCP$(0, \hat{\mathbf{x}}_0)$. Additionally, the identity matrix block in $\mathbf{G}'_{\mathbf{z}_n}(\mathbf{z}_n, \mathbf{p})$ is shifted to the position of the new initial index. In this way, the sensitivity differentials for multiple trajectory tails can be computed efficiently.

2.3 Model Predictive Control

In a next step, the MPC scheme is briefly introduced. Compare [3, 4, 1] for a comprehensive overview. MPC is a model-based control framework which allows to take state and control constraints into account. The idea is to solve optimal control problems in a repetitive manner on moving time horizons to predict the future behavior of the model and apply only the first control value to the system. In this way, a feedback action is created.

According to the proposed use case, a nonlinear MPC algorithm of tracking-type is considered. It requires to solve instances of the following parametric OCP in discrete time:

$$\begin{aligned}
\text{MPC-OCP}(n,\mathbf{p},N_{\text{MPC}}): \qquad & \min_{\mathbf{z}_{\text{MPC},n}} \sum_{k=n}^{n+N_{\text{MPC}}} E_{\text{track}}(\mathbf{x}_k,\mathbf{x}_{\text{ref},k}) \\
\text{subject to} \quad & \mathbf{x}_{k+1}-\mathbf{x}_k = \frac{h_{\text{MPC}}}{2}(\mathbf{f}(\mathbf{x}_{k+1},\mathbf{u}_k)+\mathbf{f}(\mathbf{x}_k,\mathbf{u}_k)), \quad && k=n,\dots,n+N_{\text{MPC}}-1, \\
& \mathbf{g}_{\mathbf{x}}(\mathbf{x}_k) \leq \mathbf{0}, && k=n,\dots,n+N_{\text{MPC}}, \\
& \mathbf{g}_{\mathbf{u}}(\mathbf{u}_k) \leq \mathbf{0}, && k=n,\dots,n+N_{\text{MPC}}-1, \\
& \mathbf{x}_n = \mathbf{p}, &&
\end{aligned} \tag{8}$$

where $\mathbf{z}_{\text{MPC},n} := (\mathbf{x}_n,\mathbf{u}_n,\mathbf{x}_{n+1},\mathbf{u}_{n+1},\dots,\mathbf{x}_{n+N_{\text{MPC}}-1},\mathbf{u}_{n+N_{\text{MPC}}-1},\mathbf{x}_{n+N_{\text{MPC}}})^\top$ denotes the vector of optimization variables. The dynamics are again discretized by trapezoidal rule and $\mathbf{u}$ is modeled as zero-order hold control. A fixed time discretization $h_{\text{MPC}} := t_{k+1}-t_k$ for all subproblems and a prediction horizon of $[t_n, t_{n+N_{\text{MPC}}}]$ is chosen. Again, the problem includes state and control constraints, as well as an initial boundary condition. For the considered tracking purpose, a reference state trajectory $\mathbf{x}_{\text{ref}}$ is required. The tracking-type objective consists of the function E_{track}, which is usually formulated as a weighted quadratic error term

$$E_{\text{track}} = \|\mathbf{x}_k - \mathbf{x}_{\text{ref},k}\|_W^2 \tag{9}$$

with norm $\|\cdot\|_W$ and weight matrix W.
The basic MPC scheme with its moving horizon fashion is described in the following algorithm:

Algorithm 1 MPC

Input: prediction horizon N_{MPC}, reference state trajectory $\mathbf{x}_{\text{ref}}$. Set $n=0$.
(1) Measure state $\mathbf{x}(t_n)$ at time t_n.
(2) Solve MPC-OCP$(n,\mathbf{x}(t_n),N_{\text{MPC}})$ on time horizon $[t_n,t_{n+N_{\text{MPC}}}]$ to compute an optimal control sequence $\hat{\mathbf{u}}_n,\dots,\hat{\mathbf{u}}_{n+N_{\text{MPC}}-1}$.
(3) Apply the first control $\hat{\mathbf{u}}_n$ to the system in the time interval $[t_n,t_{n+1}]$.
(4) Set $n \leftarrow n+1$ and go to step (1).

The resulting MPC trajectory may lack in quality if there are constraints that hinder precise tracking of the reference, e.g., obstacles as part of a path-following application. In this case, it is desirable to update the reference trajectory when deviations exceed a certain tolerance. This can be done online using the presented sensitivity-based approximations.

3. Application to Aircraft Path-following MPC

Now, the sensitivity-based trajectory updates are applied to update the reference trajectory for an aircraft path-following MPC. Therefore, a trajectory pre-planning is done in subsection 3.1. Then, this trajectory serves as a reference for the path-following MPC in subsection 3.2, where the scenario is considered that a no-fly zone not known in the pre-planning phase arises during the flight. Since the reference path passes through this no-fly zone, the aircraft will deviate from the reference, and a sensitivity-based update will be performed. The software environment used and the results are presented and discussed in subsections 3.3 and 3.4, respectively.

3.1 Modeling and Trajectory Pre-planning

Consider a planning task for a fixed-wing aircraft. The mission is to fly from the waypoint P_{init} to P_{final}, leaving and arriving at the waypoints with a given speed and in a given direction. The objective of the trajectory optimization problem is to minimize the energy consumption for this task, as it is of great interest in many use cases, e.g., eVTOL flight. Therefore, the aircraft is modeled by a point-mass model with states x, y, z describing the positional coordinates, V the velocity, χ the course angle, and γ the flight path angle, and controls α describing the angle of attack, δ_T the thrust lever position, and μ the bank angle. The positions are expressed in NED frame, where the z-coordinate is modeled as being zero at P_{init}.

Table 1 – Model parameter values

Parameters	Values
m	700 kg
g	9.81 m/s^2
$T_{\max}$	2060.1 N
ρ	1.1 kg/m^3
S	9.87 m^2
C_{L_0}	0.1
C_{L_α}	5.0
C_{D_0}	0.0761
k	0.05134

The equations of motion are given by

$$\dot{x}(t) = V(t)\cos\chi(t)\cos\gamma(t), \tag{10}$$
$$\dot{y}(t) = V(t)\sin\chi(t)\cos\gamma(t), \tag{11}$$
$$\dot{z}(t) = -V(t)\sin\gamma(t), \tag{12}$$
$$\dot{V}(t) = \frac{T(t)-D(t)}{m} - g\sin\gamma(t), \tag{13}$$
$$\dot{\chi}(t) = g\frac{n_z(t)\sin\mu(t)}{V(t)\cos\gamma(t)}, \tag{14}$$
$$\dot{\gamma}(t) = g\frac{n_z(t)\cos\mu(t)-\cos\gamma(t)}{V(t)}. \tag{15}$$

Here, m describes the mass of the aircraft, g the gravitational acceleration, and n_z the load factor defined by

$$n_z(t) = \frac{L(t)}{mg}. \tag{16}$$

Furthermore, T, L and D describe the thrust, lift and drag given by

$$T(t) = \delta_T(t)T_{\max}, \tag{17}$$
$$L(t) = \frac{1}{2}\rho V(t)^2 S C_L(t), \tag{18}$$
$$D(t) = \frac{1}{2}\rho V(t)^2 S C_D(t), \tag{19}$$

where $T_{\max}$ defines the maximum thrust available, ρ the air density, and S the reference wing area. The lift coefficient C_L is modeled by a linear lift curve

$$C_L(t) = C_{L_0} + C_{L_\alpha}\alpha(t) \tag{20}$$

with slope C_{L_α} and C_{L_0} defining the lift coefficient at zero angle of attack. The drag coefficient C_D is given by

$$C_D(t) = C_{D_0} + kC_L(t)^2 \tag{21}$$

with C_{D_0} being the zero-lift drag and k the induced drag coefficient. All model parameter values are given in Table 1.
The energy consumption, which is used as cost function, is modeled by

$$\int_{t_0}^{t_f} T(t)V(t)\,dt. \tag{22}$$

Additionally, the control rates are penalized by soft constraints as part of the objective function to avoid highly oscillating control inputs. Furthermore, there are upper and lower bounds on the states V and

Table 2 – Constraints

	Symobol	Unit	Lower Bound	Upper Bound	Initial Value	Final Value
States	x	m	-	-	0	2000
	y	m	-	-	0	2000
	z	m	-	-	0	0
	V	m/s	20	80	40	40
	χ	deg	-	-	0	0
	γ	deg	-30	30	0	0
Controls	α	deg	0	34.3	-	-
	δ_T	-	0	1	-	-
	μ	deg	-30	30	-	-
Outputs	n_z	-	0.8	1.2	-	-

γ, the controls α, δ_T and μ and the output n_z. These bounds, as well as the initial and final boundary conditions at the waypoints, are listed in Table 2. The state and control vector are summarized as

$$\mathbf{x} = (x, y, z, V, \chi, \gamma)^\top, \tag{23}$$

$$\mathbf{u} = (\alpha, \delta_T, \mu)^\top. \tag{24}$$

The pre-planning process consists of solving the described problem and computing the sensitivity differentials with respect to the parameter defining the initial boundary condition, as presented above, for all trajectory tails.

3.2 Path-following MPC with Reference Trajectory Update

The state trajectory of the pre-planning phase now serves as a reference for the path-following MPC. Therefore, the solution trajectory, which is given in discrete time, is interpolated and parametrized by arc-length using spline interpolation. Following the approach in [8, 5], the reference trajectory is evaluated by use of the dynamic equations of the arclength s and lateral deviation r, which are modeled as

$$\dot{s}(t) = V(t)\cos\gamma(t)\frac{\cos\left(\chi(t) - \chi_{\text{ref}}(s(t))\right)}{1 - r(t)\chi'_{\text{ref}}(s(t))}, \tag{25}$$

$$\dot{r}(t) = V(t)\cos\gamma(t)\sin\left(\chi(t) - \chi_{\text{ref}}(s(t))\right). \tag{26}$$

This approach is valid because, in the scenario under consideration, the horizontal movement of the aircraft occurs on a significantly larger scale than the vertical movement. The state and control vector for the path-following MPC problem are summarized as

$$\mathbf{x} = (x, y, z, V, \chi, \gamma, s, r)^\top, \tag{27}$$

$$\mathbf{u} = (\alpha, \delta_T, \mu)^\top. \tag{28}$$

The MPC objective is to minimize the tracking error in the positional coordinates and, with a small weight, the tracking error in the velocity. Additionally, the control effort is minimized. The objective function reads as

$$\sum_{k=n}^{n+N_{\text{MPC}}} \|\mathbf{x}_k - \mathbf{x}_{\text{ref}}(s_k)\|^2_{W_{\text{track}}} + \|\mathbf{u}_k\|^2_{W_{\mathbf{u}}}. \tag{29}$$

The dynamical system, state and control constraints are like in the pre-planning problem. The time discretization is given by $h_{\text{MPC}} = 0.5\text{s}$ and the prediction horizon is defined by $N_{\text{MPC}} = 12$, which results in a preview time horizon of 6s.

During the flight, a no-fly zone arises which is added to the constraints of MPC's optimization problem. The no-fly zone is modeled as a cylinder with centerpoint (c_x, c_y) in the horizontal plane, radius R, and without specified limits in the vertical direction. The constraint is given by

$$(x_k - c_x)^2 + (y_k - c_y)^2 \geq R^2, \qquad k = n, \ldots, n + N_{\mathrm{MPC}}. \tag{30}$$

When the reference trajectory passes through this no-fly zone, the MPC avoids it, and the flight path will deviate from the reference. At the point of maximal lateral deviation, which is determined by exploiting the prediction on MPC's preview horizon, a sensitivity-based reference trajectory update is performed. In order to perform this update, at first, the reference trajectory tail is chosen, which has its starting point closest to the current position. Then the corresponding pre-computed sensitivity differentials are used to approximate the new reference trajectory according to (3) with the current state as the new parameter value. This approximation requires only a matrix vector multiplication and two vector additions, enabling rapid online computation.

3.3 Software Environment

The discretized OCP of the pre-planning phase, as well as the MPC algorithm, are implemented using the open-source software CasADi [9] in MATLAB. CasADi is an efficient tool for nonlinear optimization and algorithmic differentiation, especially for optimal control and MPC. Within this framework, the well-known solver IPOPT [10] is used to solve the underlying optimization problems.

3.4 Results and Discussion

The results of the path-following MPC with sensitivity-based reference trajectory update are presented and discussed in the following. Consider a no-fly zone with centerpoint $(c_x, c_y) = (1000\mathrm{m}, 800\mathrm{m})$ and radius $R = 400\mathrm{m}$. The MPC is supposed to follow the reference trajectory computed in the pre-planning phase in section 3.1 and to avoid the no-fly zone when it appears in the preview horizon. Figure 1 shows the resulting trajectory of the path-following MPC in case the sensitivity-based reference trajectory update is performed.

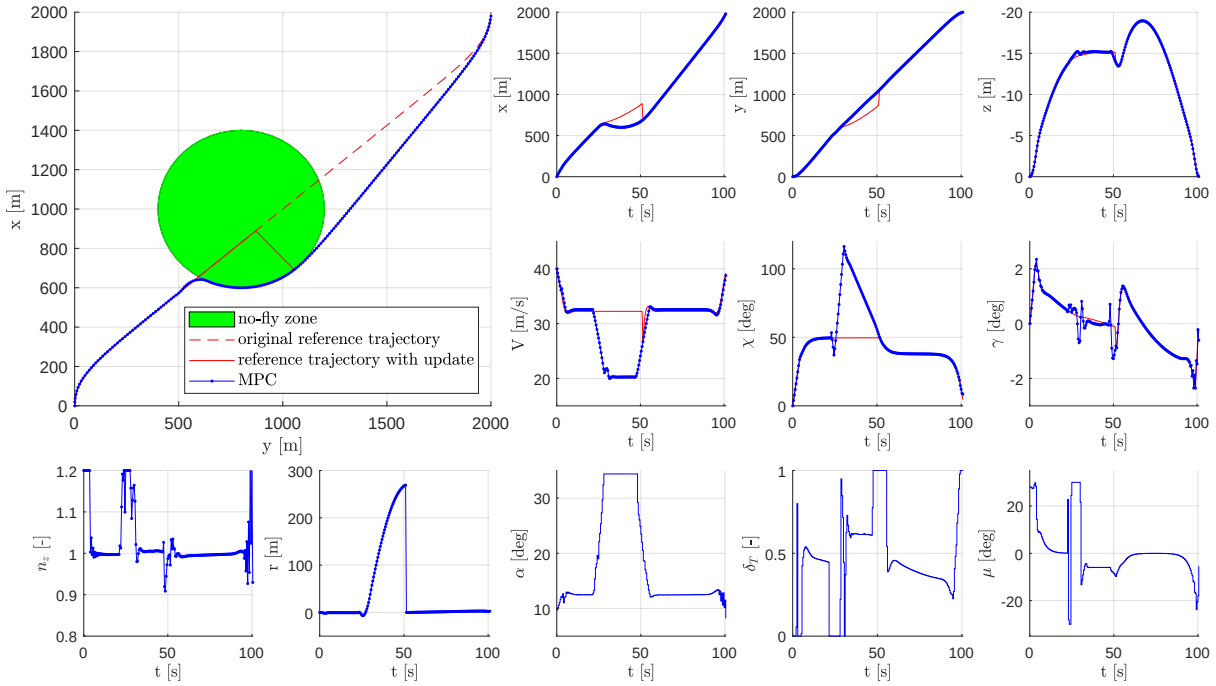


Figure 1 – Path-following MPC with sensitivity-based reference trajectory update

The result is compared with the case when no update of the reference trajectory is performed, see Figure 2. In this case, the MPC leads the aircraft all the way back to the old reference path after passing by the no-fly zone. Moreover, during this flight phase, the aircraft is accelerated to quickly reduce the tracking error in position.

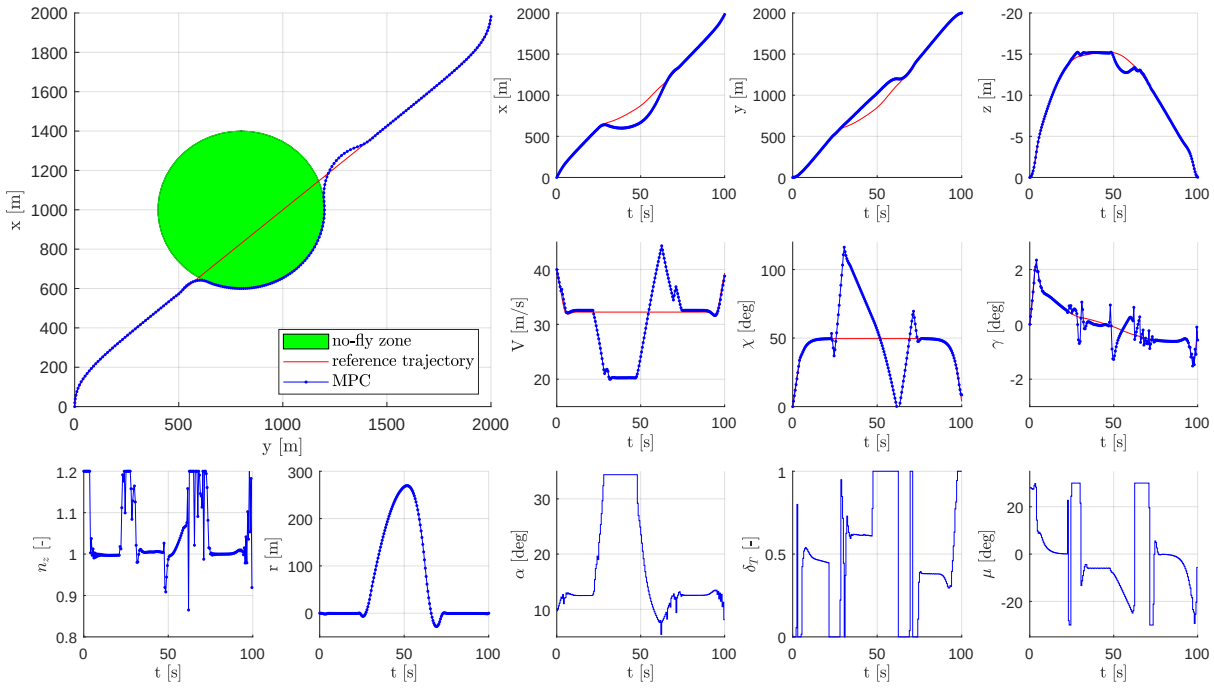


Figure 2 – Path-following MPC without reference trajectory update

Evaluating the energy consumption for both cases, which was the cost function of the trajectory optimization problem in the pre-planning task, it can be observed that the flight resulting from the MPC with reference update consumed more than 6% less energy than the flight resulting from the case without update and even more than 10% less if only the part of the flight after the update was performed is considered, see Figure 3. This shows that it is beneficial to update the reference trajectory, in case it cannot be precisely followed due to constraints. The update can be computed online at low computational cost by sensitivity-based approximations.

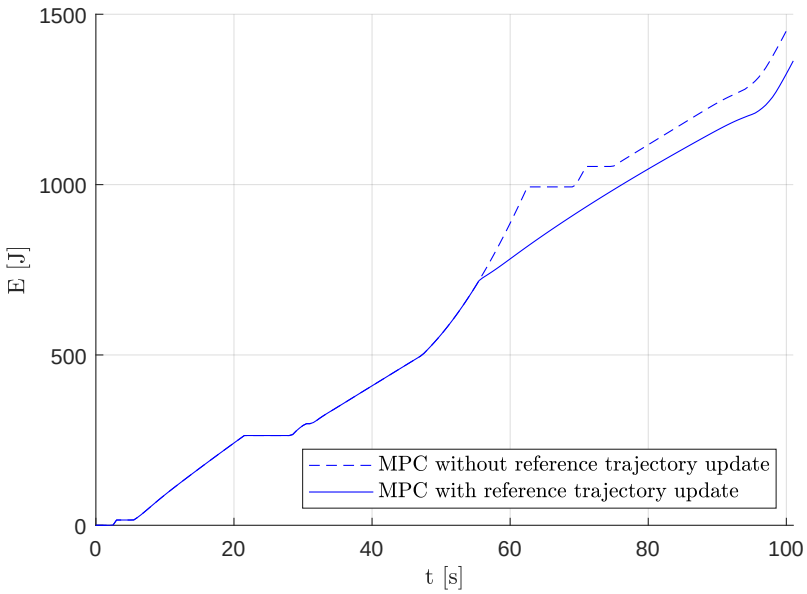


Figure 3 – Energy consumption

A downside of the approximation in (3) is that the linearization is only justified in some neighborhood of the nominal parameter. Problems occur if the set of active constraints changes due to large perturbations. Then, in general, the sensitivity theorem is not applicable anymore, and the computed

approximation of the perturbed solution may cause constraint violations where an inactive constraint is entering the set of active constraints. Figure 4 compares the approximated and the re-optimized trajectory for the considered perturbed initial boundary condition when the sensitivity-based update was performed. Here, it can be observed that the approximation caused a constraint violation at the upper bound of δ_T. But since in the presented approach this approximation serves only as a reference trajectory, and the MPC takes these constraints into account as well, the overall resulting flight path is feasible again. This is an advantage of the presented combination of MPC with sensitivity-based reference trajectory updates.

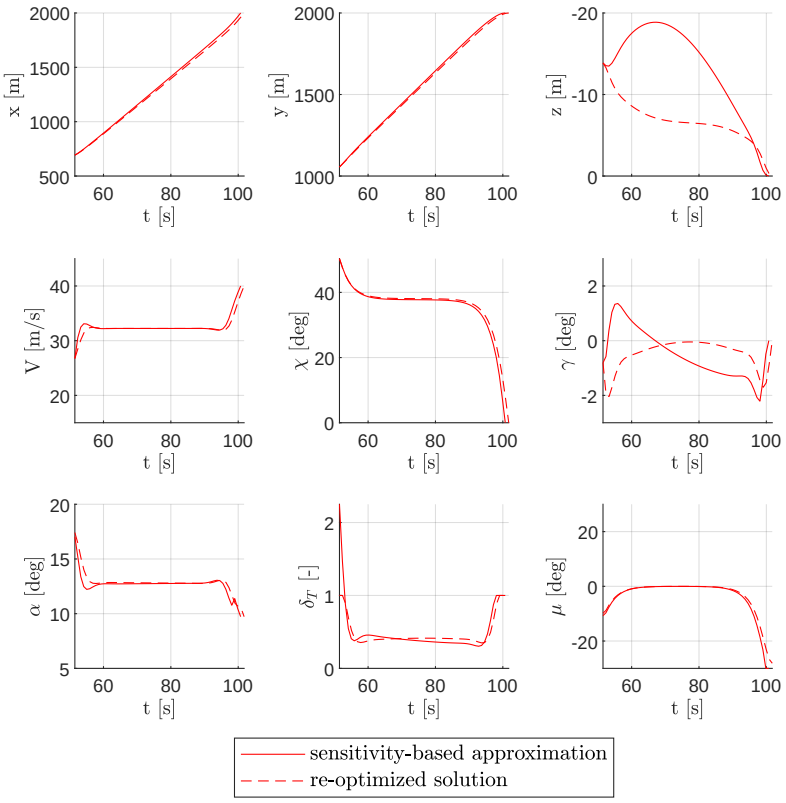


Figure 4 – Comparison of approximated and re-optimized trajectory

4. Conclusion and Outlook

This paper presents a combination of an MPC algorithm with reference trajectory updating. The updates are performed online at low computational cost via sensitivity-based approximations. The approach is applied to a path-following MPC of a fixed-wing aircraft in a no-fly zone avoidance scenario and shows its benefit in terms of optimality when the original reference trajectory cannot be precisely followed due to constraints, such as the considered no-fly zone. Moreover, an efficient way of computing the sensitivity-differentials by an appropriate matrix partitioning is provided. It is worth noting that the theory presented here is not limited to sensitivity-based updates for perturbed initial boundary conditions, but it can also be applied to other parameters of parametric OCPs. Integrating this into the MPC framework will be the scope of future work.

5. Contact Author Email Address

simon.fuchsgruber@tum.de

6. Acknowledgement

This research is supported by Munich Aerospace e.V. within the research group "Robust and Efficient Real-Time Flight Path Optimization".

7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Gerdts M. *Optimal Control of ODEs and DAEs*. De Gruyter Oldenbourg, Berlin and Boston, 2024.

[2] Betts JT. *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 2nd edition, 2010.

[3] Rawlings JB, Diehl M, and Mayne D. *Model predictive control: Theory, computation and design*. Nob Hill, Madison, 2nd edition, 2017.

[4] Grüne L and Pannek J. *Nonlinear Model Predictive Control*. Springer International Publishing, Cham, 2017.

[5] Rani S and Gerdts M. Nonlinear optimal guidance and control of a fixed-wing aircraft using NMPC with alternate dynamic modes. In *AIAA SCITECH 2023 Forum*, Reston, Virginia, 2023. American Institute of Aeronautics and Astronautics.

[6] Büskens C and Maurer H. Sensitivity analysis and real-time optimization of parametric nonlinear programming problems. *Online optimization of large scale systems*, pages 3–16, 2001.

[7] Fiacco AV. Introduction to sensitivity and stability analysis in nonlinear programming. *Mathematics in science and engineering*, 1983.

[8] Burger M and Gerdts M. DAE aspects in vehicle dynamics and mobile robotics. *Applications of Differential-Algebraic Equations: Examples and Benchmarks*, pages 37–80, 2019.

[9] Andersson JAE, Gillis J, Horn G, Rawlings JB, and Diehl M. CasADi – a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[10] Wächter A and Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.