PAPER • OPEN ACCESS

Sensitivity updates for linear-quadratic optimization problems in multi-step model predictive control

To cite this article: M Emam and M Gerdts 2023 *J. Phys.: Conf. Ser.* **2514** 012008

Sensitivity updates for linear-quadratic optimization problems in multi-step model predictive control

M Emam, M Gerdts*

Department of Aerospace Engineering, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

E-mail: * matthias.gerdts@unibw.de

Abstract. The paper discusses how parametric sensitivity analysis can be used in certain model predictive control (MPC) schemes. The sensitivity analysis will be performed with regard to the initial state measurement and update schemes will be derived that speed-up the computations. Throughout we restrict the discussion to linear-quadratic optimal control problems in discrete time, which frequently arise in tracking tasks with MPC. The derived tools from sensitivity analysis can be embedded into MPC schemes with a prediction step and multi-step MPC schemes with re-optimization and prediction step. Numerical experiments illustrating the sensitivity analysis are presented.

1. Introduction

Model predictive control (MPC) is a powerful model-based control framework, which allows to consider control and state constraints, see [1, 2] for a comprehensive overview. The main idea is to solve optimal control problems (often in discrete time) repeatedly on moving time horizons and to apply only the first few control values of the control trajectory. Its practical performance relies to a large extent on the ability to solve the optimal control problems in real-time. This is usually the computational bottleneck of the MPC control scheme and efficient solution techniques are necessary, see [3]. The numerical solution of the respective optimal control problems require some time until the solution is available. This leads to some latency in the application of the control. In order to reduce this latency, one could predict the future state using the model and use the predicted state as initial value for the optimal control problem. Another idea is to use multi-step MPC schemes, where the first $M \in \mathbb{N}$ control values of the computed control trajectory are implemented. As a consequence the system is in open-loop mode for a longer time period and deviations in the states are not taken into account. In order to overcome this issue, a re-optimization on shrinking time horizons can be performed for intermediate state measurements. Instead of performing a potentially costly re-optimization, a parametric sensitivity analysis can be used to approximate the solution. A thorough theoretical investigation of several multi-step MPC schemes can be found in [4]. Sensitivity analysis in the MPC context was also used in [5].

This paper aims to explore numerical methods for sensitivity updates in MPC schemes and it is organized as follows. The MPC schemes will be introduced in Section 2. The most costly component in the schemes is the solution of optimal control problems in discrete time. An efficient semi-smooth Newton method is summarized in Section 3. The structure of the linear systems in the semi-smooth Newton method can be exploited to compute sensitivities of the optimal solution w.r.t. the initial state. The sensitivities allow to derive update schemes for approximate optimal solutions of certain optimal control problems in the MPC schemes. These update schemes are explored in Section 4. Numerical results are provided in Section 5.

2. MPC schemes

The main motivation for the sensitivity analysis in this work are MPC schemes. We briefly summarize the relevant MPC schemes but emphasize that the main focus of this study is on the sensitivity analysis and the numerical techniques to solve the optimal control subproblems in the MPC schemes efficiently. Of course there are more aspects related to the MPC control scheme itself that should be investigated more carefully, such as stability and robustness. For a rigorous mathematical stability analysis of different M-multistep MPC schemes we refer the reader to [4].

Throughout this work we consider implicit linear control systems in discrete time $k \in \mathbb{N}_0$ subject to control or state constraints of the following type:

$$A_x(k)x(k) + A_u(k)u(k) + B_x(k+1)x(k+1) + B_u(k+1)u(k+1) = r(k), \tag{1}$$

$$(x(k), u(k)) \in Z(k), \tag{2}$$

$$x(0) = x_0. \tag{3}$$

Herein, $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ denote the state and control, respectively, at time $k \in \mathbb{N}_0$, $x_0 \in \mathbb{R}^n$ is a given initial state vector, and $A_x(k) \in \mathbb{R}^{n\times n}$, $A_u(k) \in \mathbb{R}^{n\times m}$, $B_x(k) \in \mathbb{R}^{n\times n}$, $B_u(k) \in \mathbb{R}^{n\times m}$ are given time dependent matrices, and $r(k) \in \mathbb{R}^n$ is a given time dependent vector. The non-empty sets $Z(k) \subset \mathbb{R}^n \times \mathbb{R}^m$, $k \in \mathbb{N}_0$, are defined by

$$Z(k) := \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^m \mid G_x(k)x + G_u(k)u \le g(k)\} \tag{4}$$

with matrices $G_x(k) \in \mathbb{R}^{\ell\times n}$ and $G_u(k) \in \mathbb{R}^{\ell\times m}$ and vectors $g(k) \in \mathbb{R}^\ell$ for $k \in \mathbb{N}_0$.

Remark 1 *(a) The time index $k \in \mathbb{N}_0$ is often associated with some time point $t_k = k \cdot h$ with step-size h. In the following we will identify k and t_k and use k to simplify the notation.*

(b) Implicit systems of type (1) often arise from discretizations of the linear differential equation

$$x'(t) = A(t)x(t) + B(t)u(t) + d(t).$$

E.g. the trapezoidal rule with grid points $t_k = k \cdot h$, $k \in \mathbb{N}_0$, and step-size $h > 0$ yields the following scheme, which fits into the class in (1):

$$\left(I - \frac{h}{2}A(t_{k+1})\right)x(t_{k+1}) - \left(I + \frac{h}{2}A(t_k)\right)x(t_k) - \frac{h}{2}B(t_k)u(t_k) - \frac{h}{2}B(t_{k+1})u(t_{k+1}) = \frac{h}{2}\left(d(t_k) + d(t_{k+1})\right),$$

Please note that the trapezoidal rule is just one possible and frequently used option. Many other discretization schemes also fit into the class (1).

The aim of this study is to investigate certain classes of model predictive control (MPC) schemes for (1)-(3). The MPC schemes require to solve instances of the following parametric optimal control problem in discrete time.

Journal of Physics: Conference Series **2514** (2023) 012008 doi:10.1088/1742-6596/2514/1/012008

Problem 1 (DOCP(p, n, N)) *Minimize*

$$\frac{1}{2}\sum_{k=n}^{n+N}\Big(x(k)^\top Q(k)x(k) + u(k)^\top R(k)u(k)\Big) \tag{5}$$

with respect to $(x(n)^\top, u(n)^\top, \ldots, x(n+N)^\top, u(n+N)^\top)^\top \in \mathbb{R}^{(N+1)(n+m)}$ *and subject to*

- *(1) for* $k = n, \ldots, n+N-1$*,*
- *(2) for* $k = n, \ldots, n+N$*, and*
- $x(n) = p$*.*

In DOCP(p, n, N), n denotes the current time, $N \in \mathbb{N}$ is the *prediction horizon*, and p is the initial state at time n. For given (p, n, N) let $x_{n,N}(k; p)$ and $u_{n,N}(k; p)$, $k = n, \ldots, n+N$, denote a solution of DOCP(p, n, N). The basic MPC scheme yields a feedback control law $\mu_N : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R}^m$ by setting

$$\mu_N(n, p) := u_{n,N}(n; p) \qquad (n \in \mathbb{N}_0, p \in \mathbb{R}^n)$$

whenever a solution of DOCP(p, n, N) exists. The basic MPC scheme works as follows:

Algorithm 1 (Basic MPC algorithm)

(0) Initialization: Set $n = 0$*, choose prediction horizon* $N \in \mathbb{N}$*.*
(1) Measure (or predict) state x_n *at time* n*.*
(2) Solve DOCP(x_n, n, N*) and set* $\mu_N(n, x_n) = u_{n,N}(n; x_n)$*.*
(3) Apply control $\mu_N(n, x_n)$ *at time* n *to the control system.*
(4) Set $n \leftarrow n+1$ *and go to (1).*

The basic MPC algorithm has to be seen as an idealized control scheme because it is implicitly assumed that a solution of DOCP(p, n, N) in step (2) is available instantaneously at time n, which in practice is not the case. Solving DOCP(p, n, N) requires some computational time and thus the control input $\mu_N(n, x_n)$ can only be implemented with a delay. In order to take this delay into account we will investigate two modifications of Algorithm 1:

(a) MPC with a prediction step.
(b) Multi-step MPC with re-optimization and prediction.

To this end, let $M \in \mathbb{N}$ denote the (maximum) solution time for DOCP(p, n, N). Of course, this number can only be estimated in practice. The MPC scheme with prediction step works as follows:

Algorithm 2 (MPC with prediction step)

(0) Initialization: Set $n = 0$*, choose prediction horizon* $N \in \mathbb{N}$ *and solver budget* M*.*
(1) Measure state x_n *at time* n *and predict state* x_{n+M} *at time* $n+M$ *using the model (1) for* $k = n, \ldots, n+M-1$ *with initial value* $x(n) = x_n$*.*
(2) Solve DOCP($x_{n+M}, n+M, N$*) and set* $\mu_N(n+M, x_{n+M}) = u_{n+M,N}(n+M; x_{n+M})$*.*
(3) Apply control $\mu_N(n+M, x_{n+M})$ *at time* $n+M$ *to the control system.*
(4) Set $n \leftarrow n+M$ *and go to (1).*

Please note that solving the optimal control problem in step (2) of Algorithm 2 takes M time units by assumption and thus its solution is available only at time $n+M$, if the optimization is started at time n. In the time period from n to $n+M$ the process runs with the previous control $u_{n,N}(n;x_n)$ (control is kept constant for simplicity) or with $u_{n,N}(n+i;x_n)$ for $i=0,\ldots,M-1$ in an open-loop mode without feedback.

The accuracy of the MPC scheme with prediction step depends on the accuracy of the prediction, that is, the accuracy of the model. If the actual state at $n+M$ differs from the predicted state x_{n+M} in step (1), then a procedure to quickly update the solution of DOCP$(x_{n+M}, n+M, N)$ in step (2) would be desirable. We will achieve this by sensitivity updates as described in Section 4.

A further improvement to the MPC scheme with prediction step can be obtained by the following multi-step MPC scheme with re-optimization and prediction step. It is assumed that a solution for the first time range from 0 to M is known.

Algorithm 3 (Multi-step MPC with re-optimization and prediction step)

(0) Initialization: Set $n=0$, choose prediction horizon $N \in \mathbb{N}$ and solver budget M. Measure state x_0.

(1) Predict state x_{n+M} at time $n+M$ using the model (1) for $k=n,\ldots,n+M-1$ with initial value $x(n)=x_n$.

(2) Compute future solution for predicted state: Solve DOCP$(x_{n+M}, n+M, N)$.

(3) In parallel to (2) perform the following steps for $k=0,\ldots,M$:

- *(3.1) Measure state x_{n+k} at time $n+k$ for $k>0$.*
- *(3.2) Solve DOCP$(x_{n+k}, n+k, N-k)$ and define $\mu_{N,M}(n+k, x_{n+k}) = u_{n+k,N-k}(n+k; x_{n+k})$.*
- *(3.3) Apply control $\mu_{N,M}(n+k, x_{n+k})$ at time $n+k$ to the control system.*

(4) Set $n \leftarrow n+M$ and go to (1).

In step (3.2) optimal control problems are solved on a shrinking time horizon from $n+k$ to $n+N$. For these optimal control problems, a potentially very good initial guess is available at time n from the solution of DOCP(x_n, n, N) in the preceding step of (2). Please notice that the trajectory tail of the solution of DOCP(x_n, n, N) is feasible in (3.2) on the shrinkíng time horizon owing to Bellman's optimality principle. This preceding solution, however, was computed for predicted states based on the model and not for the actual measured states in (3.1). Again, the accuracy of the initial guess depends on the accuracy of the model predictions. Although the optimal control problems in (3.2) can usually be solved quicker than the problem in (2), step (3.2) still requires time and it would be desirable to have a quick way to simply update the preceding solution for the actual state measurements and to avoid the explicit solution of DOCP$(x_{n+k}, n+k, N-k)$ in (3.2). Again, we will achieve this by sensitivity updates as described in Section 4.

3. Numerical solution of DOCP(p, n, N)

The efficient numerical solution of DOCP(p, n, N) is a key ingredient for the MPC schemes. To this end we propose a semi-smooth Newton method for the numerical solution of the first order necessary Karush-Kuhn-Tucker (KKT) conditions, see [6, 7]. It is convenient to use the abbreviations

$$z_k = \begin{pmatrix} x(k) \\ u(k) \end{pmatrix}, \quad H_k = \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix}, \quad \Psi = (I|0), \quad r_k = r(k), \quad g_k = g(k),$$

and

$$A_k = (A_x(k)|A_u(k)), \quad B_k = (B_x(k)|B_u(k)), \quad G_k = (G_x(k)|G_u(k))$$

for $k \in \mathbb{N}_0$. With these abbreviations DOCP(p, n, N) reads as follows:

Problem 2 *Minimize*

$$\frac{1}{2}\sum_{k=n}^{n+N} z_k^\top H_k z_k \tag{6}$$

with respect to $z = (z_n^\top, \ldots, z_{n+N}^\top)^\top$ *and subject to the constraints*

$$A_k z_k + B_{k+1} z_{k+1} = r_k, \qquad k = n, \ldots, n+N-1, \tag{7}$$

$$G_k z_k \leq g_k, \qquad k = n, \ldots, n+N, \tag{8}$$

$$\Psi z_n = p. \tag{9}$$

Let $\lambda = (\lambda_n^\top, \ldots, \lambda_{n+N-1}^\top)^\top$, $\mu = (\mu_n^\top, \ldots, \mu_{n+N}^\top)^\top$, and σ denote the vectors of Lagrange multipliers for the constraints (7), (8), and (9), respectively. The Lagrange function of DOCP(p, n, N) is defined by

$$\begin{aligned} L(z, \lambda, \mu, \sigma, p) \;:=\; & \frac{1}{2}\sum_{k=n}^{n+N} z_k^\top H_k z_k + \sigma^\top(\Psi z_n - p) \\ & + \sum_{k=n}^{n+N-1} \lambda_k^\top (A_k z_k + B_{k+1} z_{k+1} - r_k) + \sum_{k=n}^{n+N} \mu_k^\top (G_k z_k - g_k). \end{aligned}$$

Let H_k be positive semidefinite and symmetric for all $k \in \mathbb{N}_0$. Then DOCP(p, n, N) is convex and, since the Abadie constraint qualification is satisfied for the linear constraints, the following Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for a global minimum of DOCP(p, n, N):

$$\begin{aligned} 0 &= \nabla_{z_n} L = H_n z_n + A_n^\top \lambda_n + G_n^\top \mu_n + \Psi^\top \sigma, \\ 0 &= \nabla_{z_k} L = H_k z_k + A_k^\top \lambda_k + B_k^\top \lambda_{k-1} + G_k^\top \mu_k, \qquad k = n+1, \ldots, n+N-1, \\ 0 &= \nabla_{z_{n+N}} L = H_{n+N} z_{n+N} + B_{n+N}^\top \lambda_{n+N-1} + G_{n+N}^\top \mu_{n+N}, \end{aligned}$$

and for $k = n, \ldots, n+N$ the complementarity conditions

$$0 \leq \mu_k \perp g_k - G_k z_k \geq 0$$

are satisfied. The Fischer-Burmeister function $\varphi : \mathbb{R}^2 \to \mathbb{R}$,

$$\varphi(a, b) := \sqrt{a^2 + b^2} - a - b,$$

see [6], allows to equivalently re-write the complementarity conditions as the non-smooth equation

$$\varphi(g_k - G_k z_k, \mu_k) = 0,$$

where the latter equation is understood as a component-wise application of φ to the components of the vectors $g_k - G_k z_k$ and μ_k.

The necessary and sufficient conditions together with the constraints yield the nonlinear equation

$$\mathcal{F}(\eta) = 0, \qquad \eta = (\sigma^\top, z_n^\top, \mu_n^\top, \lambda_n^\top, \ldots, z_{n+N-1}^\top, \mu_{n+N-1}^\top, \lambda_{n+N-1}^\top, z_{n+N}^\top, \mu_{n+N}^\top)^\top \tag{10}$$

Journal of Physics: Conference Series **2514** (2023) 012008 doi:10.1088/1742-6596/2514/1/012008

with

$$\mathcal{F}(\eta) = \begin{pmatrix} \Psi z_n - p \\ \beta_n(\eta) \\ \vdots \\ \beta_{n+N}(\eta) \end{pmatrix},$$

where for $k = n, \dots, n+N-1$,

$$\beta_k(\eta) \;=\; \begin{pmatrix} \nabla_{z_k} L(z, \lambda, \mu^{(j)}, \sigma, p) \\ \varphi(g_k - G_k z_k, \mu_k) \\ A_k z_k + B_{k+1} z_{k+1} - r_k \end{pmatrix}, \quad \beta_N = \begin{pmatrix} \nabla_{z_{n+N}} L(z, \lambda, \mu, \sigma, p) \\ \varphi(g_{n+N} - G_{n+N} z_{n+N}, \mu_{n+N}) \end{pmatrix}.$$

Application of a semi-smooth Newton method to the non-smooth equation (10) to this nonlinear and non-smooth equation yields the iteration

$$\begin{aligned} V_j d^{(j)} &= -\mathcal{F}(\eta^{(j)}), \\ \eta^{(j+1)} &= \eta^{(j)} + d^{(j)}, \qquad j = 0, 1, 2, \dots \end{aligned} \tag{11}$$

Herein $V_j \in \partial\mathcal{F}(\eta^{(j)})$, where $\partial\mathcal{F}(\eta^{(j)})$ is the generalized Jacobian of $\mathcal{F}$ owing to Clarke [8]. The linear system (11) possesses a particular structure with a block banded matrix and reads as follows:

$$\begin{pmatrix} & \Omega_0 & & & \\ \Omega_0^\top & \Gamma_n & \Omega_{n+1} & & \\ & \Omega_{n+1}^\top & \Gamma_{n+1} & \ddots & \\ & & \ddots & \ddots & \Omega_{n+N} \\ & & & \Omega_{n+N}^\top & \Gamma_{n+N} \end{pmatrix} \begin{pmatrix} d_\sigma \\ d_n \\ d_{n+1} \\ \vdots \\ d_{n+N} \end{pmatrix} = - \begin{pmatrix} \Psi z_n^{(j)} - p \\ \beta_n \\ \beta_{n+1} \\ \vdots \\ \beta_{n+N} \end{pmatrix}$$

Herein, for $k = n, \dots, n+N-1$ we have $d_k = (d_{z_k}^\top, d_{\mu_k}^\top, d_{\lambda_k}^\top)^\top$, $d_{n+N} = (d_{z_{n+N}}^\top, d_{\mu_{n+N}}^\top)^\top$,

$$\Gamma_k = \begin{pmatrix} H_k & G_k^\top & A_k^\top \\ -S_k G_K & T_k & \\ A_k & & \end{pmatrix}, \quad \Gamma_{n+N} = \begin{pmatrix} H_{n+N} & G_{n+N}^\top \\ -S_{n+N} G_{n+N} & T_{n+N} \end{pmatrix},$$

and

$$(S_k, T_k) \in \partial\varphi(g_k - G_k z_k^{(j)}, \mu_k^{(j)}),$$

and for $k = n+1, \dots, n+N-1$,

$$\Omega_0 = \begin{pmatrix} \Psi & 0 & 0 \end{pmatrix}, \quad \Omega_k = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_k & 0 & 0 \end{pmatrix}, \quad \Omega_{n+N} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ B_{n+N} & 0 \end{pmatrix}.$$

The semi-smooth Newton method possesses fast local convergence properties under appropriate assumptions, i.e. a superlinear and even a quadratic convergence rate, see [9, 10, 6, 11, 12]. Extensions to infinite spaces can be found in, e.g., [9, 10, 13, 14]. It can be globalized using an Armijo-type line-search, see [12, 15]. Within an MPC method it is particularly useful owing to its excellent warm start properties. The generalized Jacobian has a block-banded structure and can be factorized using a banded LU decomposition. In our implementation we used the LAPACK routine `DGBTRF`.

4. Sensitivity analysis and sensitivity updates

DOCP(p, n, N) is a parametric optimization problem. We are particularly interested in the dependence of its solution on the parameter p. For given (p, n, N) let $z_{n,N}(p)$ denote a solution of DOCP(p, n, N) and $\lambda_{n,N}(p)$, $\mu_{n,N}(p)$, and $\sigma_{n,N}(p)$ the corresponding Lagrange multipliers.

We are interested in properties of the *solution mapping* or *parameter-to-solution mapping* $p \mapsto z_{n,N}(p)$. Conditions under which the solution mapping depends locally in a continuously differentiable way on the parameter p are of particular interest, since then a linearization

$$z_{n,N}(p) = z_{n,N}(\hat{p}) + z'_{n,N}(\hat{p})(p - \hat{p}) + o(\|p - \hat{p}\|)$$

around a nominal parameter $\hat{p}$ is permitted and

$$\tilde{z}_{n,N}(p) := z_{n,N}(\hat{p}) + z'_{n,N}(\hat{p})(p - \hat{p})$$

serves as an approximate solution to DOCP(p, n, N), see [16]. The solution differentiability of the solution maping is established in [17] for parametric nonlinear programs. The following theorem is an adapted version and tailored to DOCP(p, n, N).

Theorem 1 (Sensitivity theorem) *Let $\hat{p}$ be a nominal parameter and let the following assumptions be satisfied:*

(i) Let $\hat{\eta}$ with components $\hat{z}_{n,N}$, $\hat{\lambda}_{n,N}$, $\hat{\mu}_{n,N}$, and $\hat{\sigma}_{n,N}$ be a KKT point with (10).

(ii) Let $\hat{z}_{n,N}$ satisfy the Linear Independence Constraint Qualification (LICQ).

(iii) Let strict complementarity hold for the multiplier $\hat{\mu}_{n,N}$.

(iv) Let the Hessian matrix of the Lagrange function L w.r.t. to z be positive definite on the critical cone of DOCP$(\hat{p}, n, N)$.

Then:

(a) There exist neighborhoods $B_\epsilon(\hat{p})$ and $B_\delta(\hat{\eta})$, such that DOCP(p, n, N) has a unique local minimum

$$\eta(p) \in B_\delta(\hat{\eta})$$

for each $p \in B_\epsilon(\hat{p})$.

(b) The active sets of DOCP(p, n, N) and DOCP$(\hat{p}, n, N)$ coincide for $p \in B_\epsilon(\hat{p})$.

(c) The mapping $\eta(\cdot)$ is continuously differentiable with respect to p and the sensitivities

$$\begin{aligned} \zeta'_{n,N,k}(\hat{p}) &= (z'_{n,N,k}(\hat{p})^\top, \mu'_{n,N,k}(\hat{p})^\top, \lambda'_{n,N,k}(\hat{p})^\top)^\top \qquad (k = n, \ldots, n+N-1), \\ \zeta'_{n,N,n+N}(\hat{p}) &= (z'_{n,N,n+N}(\hat{p})^\top, \mu'_{n,N,n+N}(\hat{p})^\top)^\top, \\ \sigma'_{n,N}(\hat{p}) & \end{aligned}$$

in $\eta'(\hat{p}) = \left(\sigma'_{n,N}(\hat{p})^\top, \zeta'_{n,N,n}(\hat{p})^\top, \ldots, \zeta'_{n,N,n+N}(\hat{p})^\top\right)^\top$ satisfy the linear equation

$$\begin{pmatrix} & \Omega_0 & & & \\ \Omega_0^\top & \Gamma_n & \Omega_n & & \\ & \Omega_n^\top & \Gamma_{n+1} & \ddots & \\ & & \ddots & \ddots & \Omega_{n+N} \\ & & & \Omega_{n+N}^\top & \Gamma_{n+N} \end{pmatrix} \begin{pmatrix} \sigma'_{n,N}(\hat{p}) \\ \zeta'_{n,N,n}(\hat{p}) \\ \zeta'_{n,N,n+1}(\hat{p}) \\ \vdots \\ \zeta'_{n,N,n+N}(\hat{p}) \end{pmatrix} = \begin{pmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad (12)$$

where the matrices Γ_k, $k = n, \ldots, n+N$, are evaluated at $\hat{\eta}$.

A proof of the theorem follows the lines of the proofs in [17] or [18, Theorem 6.1.4] for more general nonlinear programs. The Hessian matrix of L w.r.t. z is given by the block diagonal matrix $diag(H_n, \dots, H_{n+N})$. In order to show (12) it is exploited that either S_k or T_k in Γ_k is zero while the respective other value is non-zero, if strict complementarity holds. Sensitivity results without strict complementarity can be found in [19]. The conditions (i)-(iv) are sufficient for $\hat{z}$ being a minimum of DOCP$(\hat{p}, n, N)$. In particular, the matrix in (12) is non-singular at $\hat{z}$ under these conditions. The estimation of the neighborhoods in the theorem is difficult in general. A numerical approach can be found in [20]. For linear-quadratic problems it is known, see [21], that the solution depends piecewise linearly on the initial state. Hence, it is theoretically possible to identify the corresponding polytopic regions in the state space, where the active set changes. This would allow to obtain globally exact parametric solutions to the linear-quadratic problems. However, this becomes numerically intractable for high dimensional state spaces and we prefer the local analysis provided by the sensitivity theorem, which is valid for nonlinear problems as well. Nevertheless part (b) indicates that the boundaries of the polytopic regions are connected to a change in the active set.

4.1. Sensitivity update for MPC with prediction step

We exploit the sensitivity result in Theorem 1 and use it in step (2) of Algorithm 2 to improve the solution of the prediction. Recall that x_{n+M} in DOCP$(x_{n+M}, n+M, N)$ is a prediction of the state at time $n+M$. This prediction relies on the measurement x_n at time n and the model (1). In general, however, $\hat{p} = x_{n+M}$ will deviate from the actual state at time $n+M$. Let $p = \tilde{x}_{n+M}$ denote the measurement of the state at time $n+M$. Then the solution of DOCP$(\hat{p}, n+M, N)$ can be improved using a sensitivity update. To this end, let $u'_{n+M,N,k}(\hat{p})$, $k = n+M, \dots, n+M+N$, denote the sensitivities of the optimal control sequence of DOCP$(\hat{p}, n+M, N)$, which is determined by (12) with n replaced by $n+M$.

For $p \approx \hat{p}$ and $k = n+M, \dots, n+M+N$ the nominal optimal control can be updated by the linear Taylor approximation

$$u_{n+M,N,k}(p) \approx \tilde{u}_{n+M,N,k}(p) := u_{n+M,N,k}(\hat{p}) + u'_{n+M,N,k}(\hat{p})(p - \hat{p}). \tag{13}$$

The approximations for $k = n+M$ and $k = n+M+1$ can then be used to update the solution of DOCP$(x_{n+M}, n+M, N)$ in step (2) of Algorithm 2 and we obtain the following algorithm:

Algorithm 4 (MPC with prediction step and sensitivity update)

(0) Initialization: Set $n = 0$, choose prediction horizon $N \in \mathbb{N}$ and solution budget $M \leq N$.

(1) Measure state x_n at time n and predict state x_{n+M} at time $n+M$ using the model (1) for $k = n, \dots, N+M-1$ with initial value $x(n) = x_n$.

(2) Solve DOCP$(x_{n+M}, n+M, N)$ and perform a sensitivity analysis with respect to x_{n+M} by solving (12) with n replaced by $n+M$.

(3) Measure the state $\tilde{x}_{n+M}$ at $n+M$ and set $\mu_N(n+M, x_{n+M}) = \tilde{u}_{n+M,N,n+M}(\tilde{x}_{n+M})$.

(4) Apply control $\mu_N(n+M, x_{n+M})$ at time $n+M$ to the control system.

(5) Set $n \leftarrow n+M$ and go to (1).

Please note that the sensitivity analysis is only justified under the assumptions of Theorem 1 for sufficiently small perturbations p from $\hat{p}$. As a consequence the linearization might not provide good approximations for large deviations. In the latter situation, it is recommended to fully re-solve the optimal control problem for the measured state $\tilde{x}_{n+M}$. The updated solution may serve as an initial guess, though.

4.2. Sensitivity update for multi-step MPC with prediction step and re-optimization

In this section we aim to avoid the explicit solution of DOCP$(x_{n+k}, n+k, N-k)$ for $k = 0, \ldots, M$, in step (3.2) of Algorithm 3. Instead we approximate the optimal solution of DOCP$(x_{n+k}, n+k, N-k)$ with suitable sensitivity updates. We require the following auxiliary result.

Theorem 2 *Let $n \in \mathbb{N}_0$, $N \in \mathbb{N}$, and $\hat{p}, \hat{q} \in \mathbb{R}^n$ be given such that (1) is satisfied, i.e.*

$$A_x(n)\hat{p} + A_u(n)u_{n,N,n}(\hat{p}) + B_x(n+1)\hat{q} + B_u(n+1)u_{n,N,n+1}(\hat{p}) = r_n, \tag{14}$$

where $u_{n,N,k}(\hat{p})$, $k = n, \ldots, n+N$, denotes the optimal control of DOCP($\hat{p}, n, N$). Let the assumptions of Theorem 1 be fulfilled for DOCP($\hat{p}, n, N$) and let $u'_{n,N,k}(\hat{p})$ denote the sensitivities of $u_{n,N,k}(\hat{p})$ for $k = n, \ldots, n+N$. Let the matrix

$$P(n) := A_x(n) + A_u(n)u'_{n,N,n}(\hat{p}) + B_u(n+1)u'_{n,N,n+1}(\hat{p})$$

be non-singular. Then there exist $\delta > 0$ and $\Delta > 0$ and a function $\rho_n : B_\delta(\hat{q}) \to B_\Delta(\hat{p})$ with $\hat{p} = \rho_n(\hat{q})$ and

$$A_x(n)\rho_n(q) + A_u(n)u_{n,N,n}(\rho_n(q)) + B_x(n+1)q + B_u(n+1)u_{n,N,n+1}(\rho_n(q)) = r_n \tag{15}$$

for all $q \in B_\delta(\hat{q})$. Moreover, ρ_n is differentiable in $B_\delta(\hat{q})$ with

$$\rho'_n(\hat{q}) = -P(n)^{-1}B_x(n+1). \tag{16}$$

Proof. According to Theorem 1 there exists a neighborhood $B_\epsilon(\hat{p})$ such that $p \mapsto u_{n,N,k}(p)$ is differentiable for all $p \in B_\epsilon(\hat{p})$ and all $k = n, \ldots, n+N$. Since $P(n)$ is non-singular and $\hat{p}$ and $\hat{q}$ satisfy the equation (14), the implicit function theorem yields the existence of neighborhoods $B_\delta(\hat{q})$ and $B_\Delta(\hat{p})$ with $\Delta \leq \epsilon$, and a differentiable function $\rho_n : B_\delta(\hat{q}) \to B_\Delta(\hat{p})$ with (15). Differentiation of (15) w.r.t. q and evaluation at $q = \hat{q}$ yields (16). ∎

We are now interested in the problem DOCP$(\hat{q}_{n+1}, n+1, N-1)$ with $\hat{q}_{n+1} = x_{n,N,n+1}(\hat{p})$, where $x_{n,N,n+1}(\hat{p})$ refers to the optimal solution of DOCP$(\hat{p}, n, N)$ with $\hat{p} = \hat{x}_n$. We assume that an optimal solution of DOCP$(\hat{x}_n, n, N)$ is available and a sensitivity analysis has been performed for the parameter $\hat{p}$.

The optimal solution of DOCP$(\hat{q}_{n+1}, n+1, N-1)$ is then immediately available according to Bellman's optimality principle, since the trajectory tail

$$z_{n,N,n+1}(\hat{p}), \ldots, z_{n,N,n+N}(\hat{p})$$

is optimal for DOCP$(\hat{q}_{n+1}, n+1, N-1)$ with $\hat{q}_{n+1} = x_{n,N,n+1}(\hat{p})$. We are now interested in investigating perturbations in $\hat{q}_{n+1}$.

This is essentially the situation of step (3.2) in Algorithm 3. For $k = 0$ we can use the sensitivities of DOCP$(\hat{x}_n, n, N)$ to update its solution in the presence of a perturbation x_n of $\hat{x}_n$ as outlined in the previous sections. Recall that this solution is already available in the MPC algorithm from the previous prediction step. For $k = 1$ and a perturbation $q_{n+1} = x_{n+1}$ of $\hat{q}_{n+1}$ the sensitivities of DOCP$(\hat{q}_{n+1}, n+1, N-1)$ could be used to update the solution of DOCP$(\hat{q}_{n+1}, n+1, N-1)$. One way to obtain these sensitivities is to solve linear equation (12) with n replaced by $n+1$. Once the linear system is solved, an approximate solution to DOCP$(q_{n+1}, n+1, N-1)$ can be computed by the sensitivity update

$$u_{n+1,N-1,k}(q_{n+1}) \approx u_{n+1,N-1,k}(\hat{q}_{n+1}) + u'_{n+1,N-1,k}(\hat{q}_{n+1})(q_{n+1} - \hat{q}_{n+1})$$

Journal of Physics: Conference Series **2514** (2023) 012008 doi:10.1088/1742-6596/2514/1/012008

for $k = n+1, \ldots, n+N$, where $u'_{n+1,N-1,k}(\hat{q}_{n+1})$ denotes the sensitivity of the optimal control of DOCP$(\hat{q}_{n+1}, n+1, N-1)$. In view of MPC only the update for $k = n+1$ is needed for implementation on the controlled system. Solving (12), however, requires time and causes a delay.

Hence, we are seeking to answer the following question:

Can we compute an approximate solution for DOCP$(q_{n+1}, n+1, N-1)$ for a perturbation q_{n+1} of $\hat{q}_{n+1}$ using only the solution and senstivities of DOCP$(\hat{p}, n, N)$ without re-solving (12) for DOCP$(\hat{q}_{n+1}, n+1, N-1)$?

The answer is given in the following theorem.

Theorem 3 *Let the assumptions of Theorem 2 be fulfilled. Then the sensitivity of the optimal control $u_{n+1,N-1,n+1}(\hat{q}_{n+1})$ of DOCP$(\hat{q}_{n+1}, n+1, N-1)$ is given by*

$$\begin{aligned} u'_{n+1,N-1,n+1}(\hat{q}_{n+1}) &= u'_{n,N,n+1}(\hat{p})\rho'_n(\hat{q}_{n+1}) \\ &= -u'_{n,N,n+1}(\hat{p})P(n)^{-1}B_x(n+1). \end{aligned}$$

Proof. For $q \in B_\delta(\hat{q}_{n+1})$, see Theorem 2, we have the relation

$$u_{n+1,N-1,n+1}(q) = u_{n,N,n+1}(\rho_n(q))$$

and by differentiation

$$\begin{aligned} u'_{n+1,N-1,n+1}(\hat{q}_{n+1}) &= u'_{n,N,n+1}(\hat{p})\rho'_n(\hat{q}_{n+1}) \\ &= -u'_{n,N,n+1}(\hat{p})P(n)^{-1}B_x(n+1). \end{aligned}$$

■

Theorem 3 provides an update rule, which relies on the already computed sensitivities of DOCP$(\hat{p}, n, N)$.

The above procedure can be extended inductively for a number $k \in \{1, \ldots, M\}$, since the trajectory tail

$$z_{n,N,n+k}(\hat{p}), \ldots, z_{n,N,n+N}(\hat{p})$$

is optimal for DOCP$(\hat{q}_{n+k}, n+k, N-k)$ with $\hat{q}_{n+k} = z_{n,N,n+k}(\hat{p})$. To this end we exploit the relation

$$u_{n+k,N-k,n+k}(q_{n+k}) = u_{n,N,n+k}(\rho_n \circ \rho_{n+1} \circ \cdots \circ \rho_{n+k-1}(q_{n+k}))$$

for $k = 1, \ldots, M$ and q_{n+k} sufficiently close to $\hat{q}_{n+k}$ (such that $\rho_n \circ \rho_{n+1} \circ \cdots \circ \rho_{n+k-1}(q_{n+k}) \in B_\epsilon(\hat{p})$). By the chain rule we obtain

$$u'_{n+k,N-k,n+k}(\hat{q}_{n+k}) = (-1)^{k-1} u'_{n,N,n+k}(\hat{p}) \prod_{j=0}^{k-1} P(n+j) B_x(n+1+j)$$

with

$$P(n+j) = \left(A_x(n+j) + A_u(n+j)u'_{n,N,n+j}(\hat{p}) + B_u(n+1+j)u_{n,N,n+1+j}(\hat{p})\right)^{-1}.$$

This proves the following theorem.

Theorem 4 *Let the assumptions of Theorem 2 be fulfilled for $n, n+1, \ldots, n+M$ with $M \leq N$. Then the sensitivity of the optimal control $u_{n+k,N-k,n+k}(\hat{q}_{n+k})$ of DOCP$(\hat{q}_{n+k}, n+k, N-k)$, $k \in \{1, \ldots, M\}$, is given by*

$$u'_{n+k,N-k,n+k}(\hat{q}_{n+k}) = (-1)^{k-1} u'_{n,N,n+k}(\hat{p}) \prod_{j=0}^{k-1} P(n+j) B_x(n+1+j).$$

5. Numerical experiments

All computations were performed on a laptop with 8 GB RAM and Intel®Core™i5-5200U @ 2.20 GHz processor to signify the minimal required computational time and the fast system convergence even under limited resources.

Here we investigate the path tracking problem for a kinematic vehicle, in which we employ and compare the performance of the basic MPC Algorithm 1, the MPC Algorithm 2 with prediction step, and the MPC Algorithm 4 with prediction step and sensitivity update. To avoid redundancy, these algorithms are denoted in the sequel as *Alg.1*, *Alg.2*, and *Alg.4* respectively. The motion of the vehicle is formulated in curvilinear coordinates relative to a reference path [22] as

$$s'(t) = \frac{v(t)\cos(\psi(t)-\psi_r(t))}{1-r(t)\cdot\kappa_r(s(t))}, \tag{17a}$$

$$r'(t) = v(t)\sin(\psi(t)-\psi_r(t)), \tag{17b}$$

$$\psi'(t) = v(t)\kappa(t), \tag{17c}$$

$$\kappa'(t) = u(t), \tag{17d}$$

$$\psi_r'(t) = s'(t)\kappa_r(s(t)). \tag{17e}$$

Where s is the arclength of the projected vehicle position unto the reference path, r denotes the lateral offset to this path, ψ is the vehicle's yaw angle, κ is the curvature of the vehicle's driven path, ψ_r is the reference path angle, and u controls the vehicle's lateral jerk. This non-linear model is discretized and used in coordination with a Runge-Kutta fourth-order method to reach an accurate estimation of the states x_{n+M} at time $n+M$ for the prediction phase in Algorithms 2 and 4. Henceforth, the vehicle is assumed to drive with the constant velocity $V = 15$ [m/s].

Linearization of (17) for small deviations from the reference path yields the linear differential equation system

$$\underbrace{\begin{pmatrix} s(t)\\ r(t)\\ \psi(t)\\ \kappa(t)\\ \psi_r(t)\end{pmatrix}'}_{=x'(t)} = \underbrace{\begin{pmatrix} 0&0&0&0&0\\ 0&0&V&0&-V\\ 0&0&0&V&0\\ 0&0&0&0&0\\ 0&0&0&0&0\end{pmatrix}}_{=A}\underbrace{\begin{pmatrix} s(t)\\ r(t)\\ \psi(t)\\ \kappa(t)\\ \psi_r(t)\end{pmatrix}}_{=x(t)} + \underbrace{\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix}}_{=B} u(t) + \underbrace{\begin{pmatrix} V\\0\\0\\0\\ V\kappa_r(s(t))\end{pmatrix}}_{=d}. \tag{18}$$

Thus, a tracking optimal control problem on the time horizon $[0, t_f]$ reads as

Minimize

$$\frac{1}{2}\int_0^{t_f} x(t)^\top Q x(t) + u(t)^\top R u(t)\ dt$$

subject to (18), initial value $x(0) = \hat{p}$, control constraints $u(t) \in [u_{min}, u_{max}]$, and state constraints $r(t) \in [r_{min}, r_{max}]$, $\kappa(t) \in [\kappa_{min}, \kappa_{max}]$.

Herein, $R > 0$ and

$$Q = \begin{pmatrix} 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&-1\\ 0&0&0&0&0\\ 0&0&-1&0&1\end{pmatrix}.$$

Discretization by the implicit trapezoidal rule yields the following optimal control problem in discrete time:

Minimize

$$\frac{h}{2}\left(\frac{1}{2}\left(x_0^\top Qx_0+u_0^\top Ru_0\right)+\sum_{k=1}^{N-1}\left(x_k^\top Qx_k+u_k^\top Ru_k\right)+\frac{1}{2}\left(x_N^\top Qx_N+u_N^\top Ru_N\right)\right)$$

subject to the constraints

$$\begin{aligned} x_{k+1} &= x_k+\frac{h}{2}\left(Ax_k+Bu_k+d_k+Ax_{k+1}+Bu_{k+1}+d_{k+1}\right), && k=0,\dots,N-1,\\ g &\geq G_x x_k+G_u u_k, && k=0,\dots,N,\\ x_0 &= \hat{p}. \end{aligned}$$

This problem fits into the class DOCP$(\hat{p},0,N)$ with the settings

$$A_x=-(I+\frac{h}{2}A),\quad A_u=B_u=-\frac{h}{2}B,\quad B_x=(I-\frac{h}{2}A),\quad r_k=\frac{h}{2}(d_k+d_{k+1})$$

and

$$G_x=\begin{pmatrix}0&1&0&0&0\\0&-1&0&0&0\\0&0&0&1&0\\0&0&0&-1&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix},\quad G_u=\begin{pmatrix}0\\0\\0\\0\\1\\-1\end{pmatrix},\quad g=\begin{pmatrix}r_{max}\\-r_{min}\\\kappa_{max}\\-\kappa_{min}\\u_{max}\\-u_{min}\end{pmatrix}.$$

For numerical evaluation, we choose the following test parameters: $\hat{p}=(0.0,3,0.1,0,0)^\top$, $t_f=10$ [s], $N=100$, $M=10$, $u_{max}=-u_{min}=0.3$ [1/(ms)], $\kappa_{max}=-\kappa_{min}=0.1$ [1/m], $r_{max}=-r_{min}=4$ [m]. Additionally, we recreate sensor measurement noise on $\tilde{x}_{n+M}$ for the sensitivity updates by imposing random perturbations on $\tilde{r}-\hat{r}\in[-0.1,0.1]$ and on $\tilde{\kappa}-\hat{\kappa}\in[-0.002,0.002]$. For the tracking problem, we use the test track depicted in figure 1 as the reference path, which is a 1:1 model of the test track located in campus of the *Universität der Bundeswehr München*.

In the sequel we will vary the weight R for the control effort in the objective function and discuss the achieved results by the proposed MPC algorithms. Herein, the initial guess was set to zero for the entire vector η. The CPU time for solving (12), i.e. for computing the sensitivities, contains the set-up of equation (12), an LU decomposition of the matrix, and n (=state dimension) forward-backward substitutions to solve the equation. In fact, in a more efficient implementation one could even spare the LU decomposition, since it can be re-used from the final iterate of the semi-smooth Newton method. This would further reduce the CPU time for the sensitivity analysis. However, we will see in the following subsections that the CPU time for the sensitivities is very low.

5.1. The case $R=100$

We choose $R=100$ as the weight for the control effort in the objective function, which yields the trajectories displayed in figure 2 for the different MPC algorithms. The deviation in the vehicle's trajectory from the reference path is represented by the state r and the error in heading $\psi-\psi_r$, which are illustrated in figures 3 and 4 respectively. The scenario begins with the initial deviation $r=0.3$ [m] and $\psi-\psi_r=0.1$ [rad] and, as expected, the idealized Alg.1 converges the fastest since it operates instantaneously at 0 [s] as shown in figure 6. However, the other two algorithms operate more realistically and only yield a control action after the first time step has passed, which explains the greater overshoot in system states before stabilization. Nevertheless,

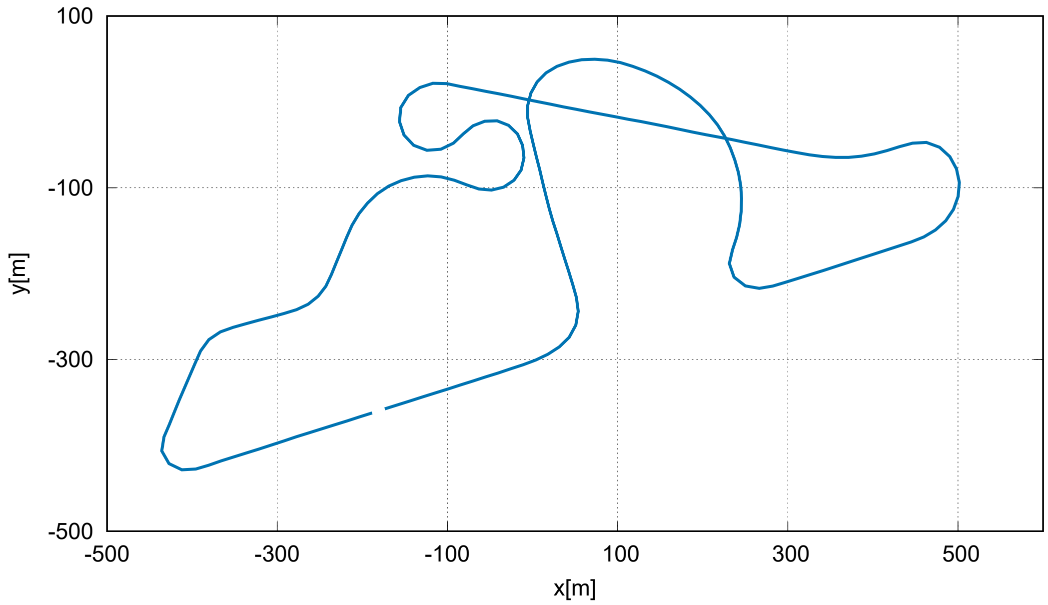

Figure 1. Reference test track for the path tracking problem

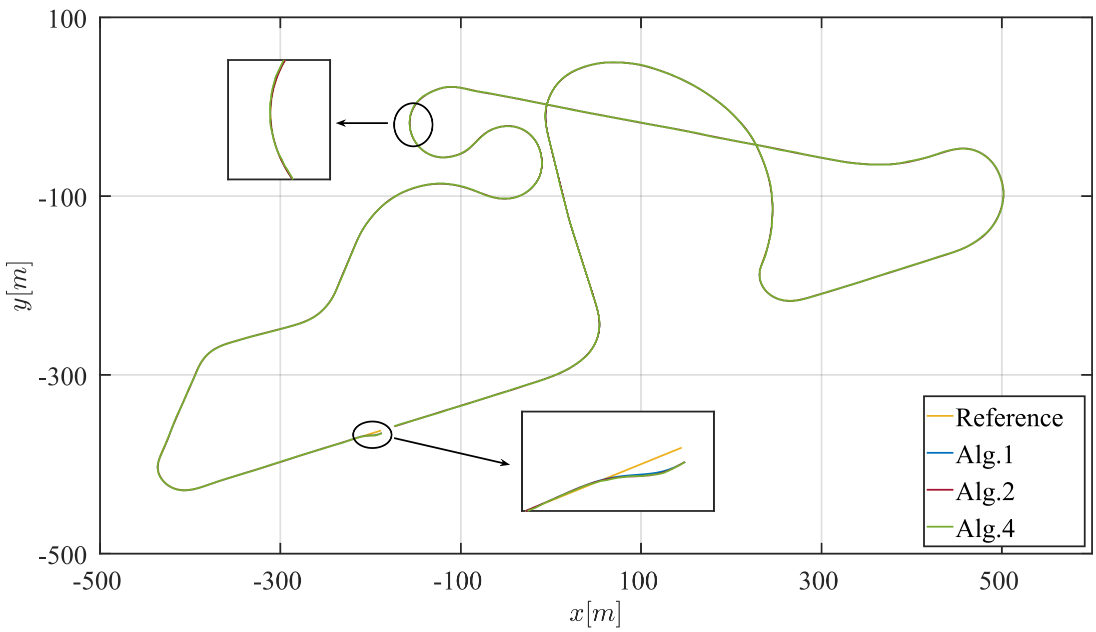

Figure 2. Generated vehicle trajectories as achieved by the MPC algorithms with $R = 100$.

all algorithms are able to eliminate the initial deviation within the first 4 seconds (around $s = 60$ [m]) and the state constraints are not active during the entire solution space.

To properly assess the path tracking behavior, we omit data entries prior to the initial error stabilization and summarize the algorithms' results in table 1. For compactness, we also include

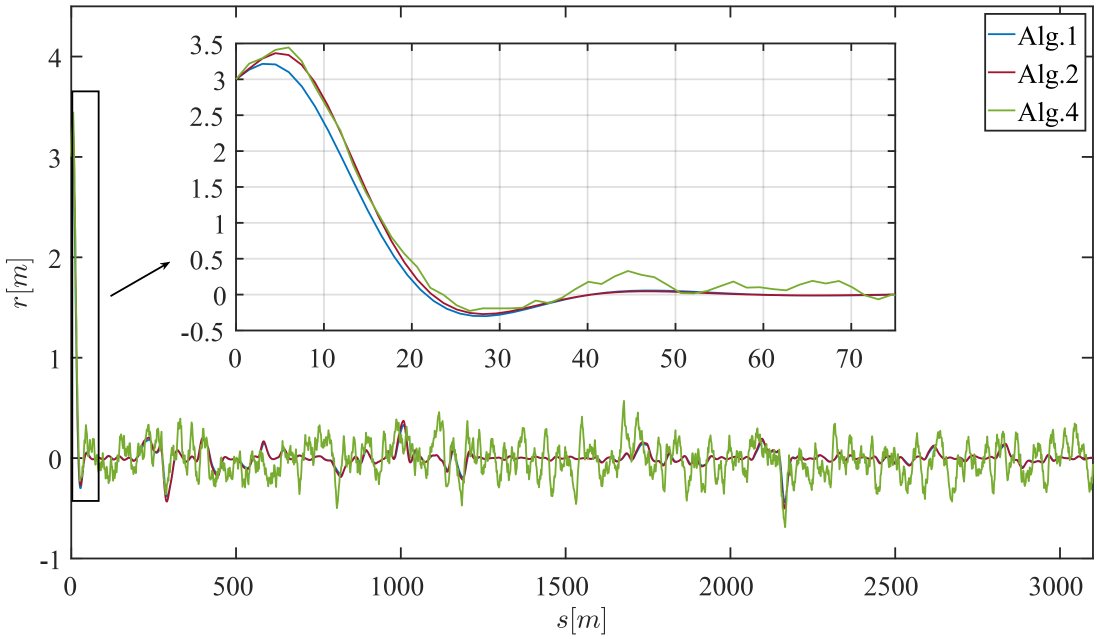

Figure 3. Vehicle's lateral offset r across the reference path with $R = 100$.

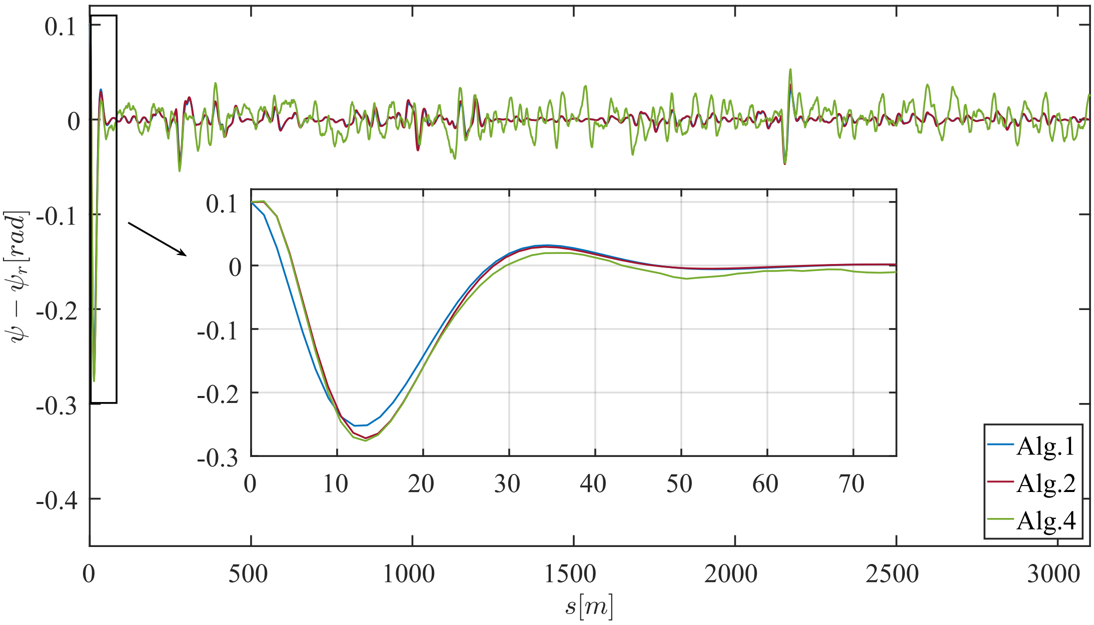

Figure 4. Deviation of the vehicle heading to the reference path $\psi - \psi_r$ with $R = 100$.

the CPU time t_{CPU} for solving the DOCP$(\hat{p}, 0, 100)$ using the semi-smooth Newton method in Section 3 and the CPU time $t_{CPU,s}$ for solving (12), i.e. for computing the sensitivities, for all data entries. We notice that Alg.1 and Alg.2 exhibit similar behaviors with minimal deviation from the reference path, while Alg.4 yields comparatively larger deviations. These values are however coherent with the magnitude of the perturbations we introduced to the system states, which highlights the controller's ability to consistently stabilize the system despite the imposed noise. Additionally, this approach is suitable for real-time applications

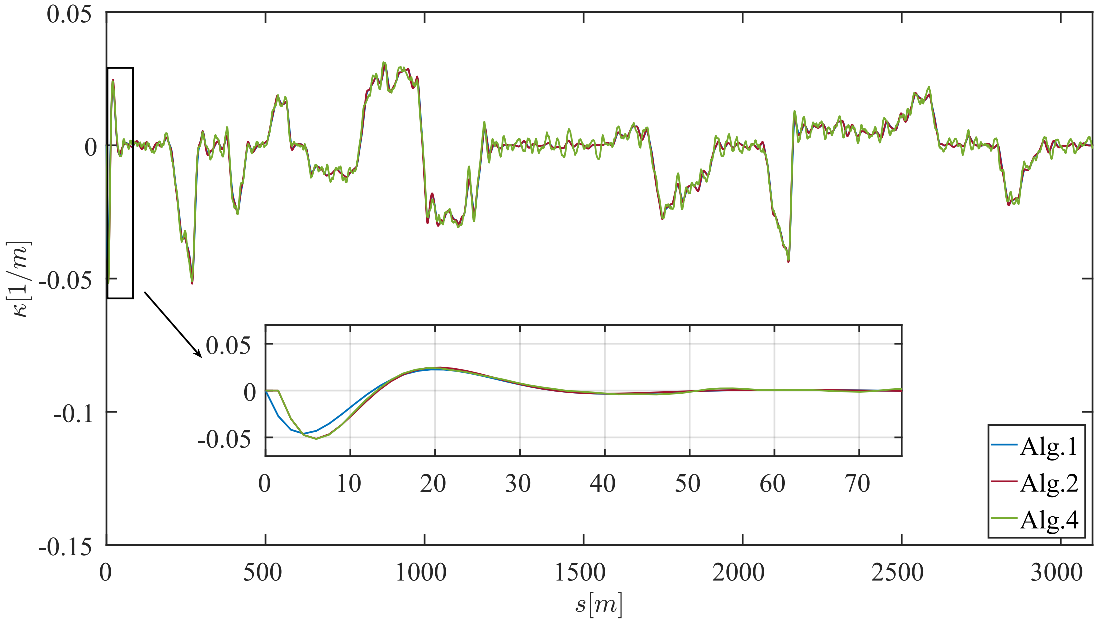

Figure 5. The constraints for κ are not activated with $R = 100$.

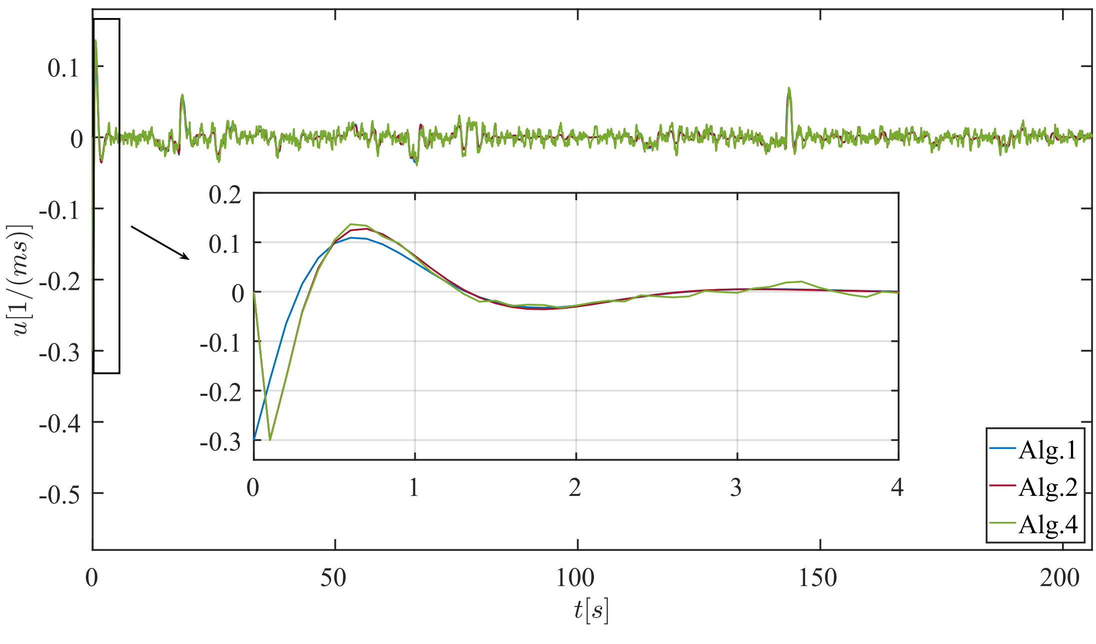

Figure 6. Numerical solution of DOCP with $R = 100$: Control u. The constraint $u \geq u_{min}$ is only active momentarily at the first solution step.

as $t_{CPU} < t_{k+1} - t_k, k \in \mathbb{N}_0$, i.e. the DOCP solution is always available at the requested time. Finally, the time for computing the sensitivities is negligible compared to solving the DOCP, as $t_{CPU,s} \leq 0.003731$ [s].

To better illustrate the control and its sensitivities w.r.t. an initial state p, we shift our focus to the numerical solution of the DOCP for a single step, i.e. the first solution step at time 0.1 [s] with $\hat{p} = (1.4925, 3.2187, 0.1012, 0.0, 1.0e-06)^\top$. The control and sensitivities are displayed in figure 7, where the plots have different scales for the control value and the respective

Table 1. Path tracking results achieved by the MPC algorithms with $R = 100$

	$\|\|r\|\|$ [m]		$\|\|\psi - \psi_r\|\|$ [rad]		t_{CPU} [s]		$max(t_{CPU,s})$ [s]
	mean	*max*	*mean*	*max*	*mean*	*max*	
Alg.1	0.038275	0.440323	0.003680	0.040423	0.005587	0.021811	-
Alg.2	0.043051	0.502377	0.004178	0.047116	0.005524	0.018018	-
Alg.4	0.136355	0.688770	0.010950	0.054566	0.008287	0.021578	0.003731

sensitivities, e.g., the sensitivity w.r.t. the initial curvature $\kappa(0)$ is high whereas the sensitivity w.r.t. to the initial offset $r(0)$ is comparatively low. As stated by Theorem 1, the sensitivities on the bang-bang arcs of the control are zero.

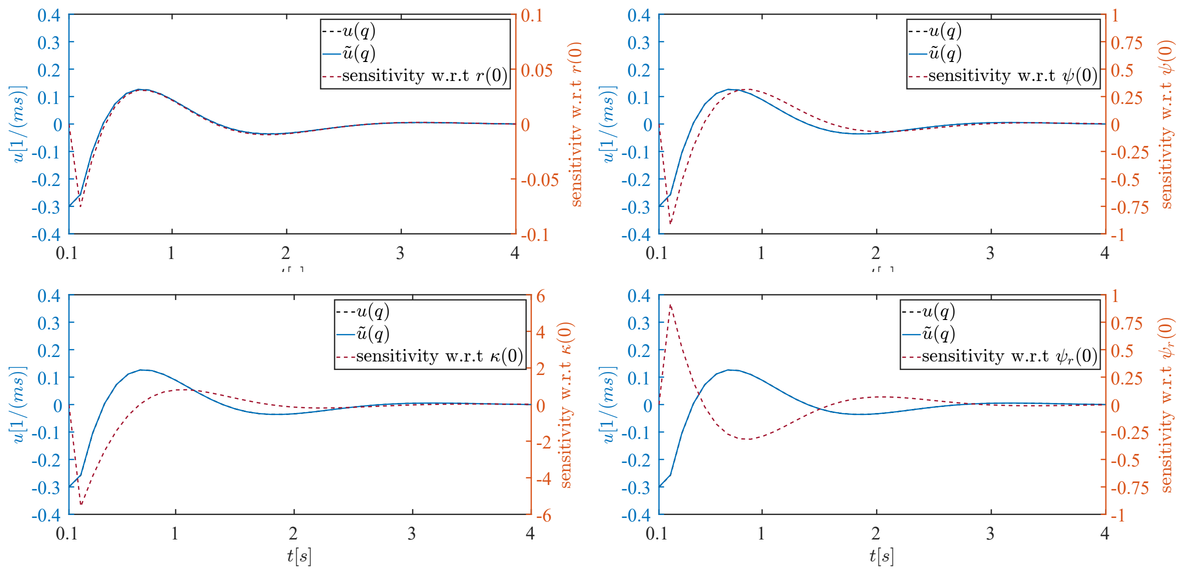

Figure 7. Comparison of control $\tilde{u}(q)$ from combined solution of DOCP$(\hat{p}, 1, 100)$ and sensitivity analysis against the control solution $u(q)$ from re-optimization with $R = 100$. Moreover, the sensitivities w.r.t. the relevant initial states $r(0), \psi(0), \kappa(0), \psi_r(0)$ are displayed (sensitivity w.r.t. $s(0)$ is $= 0$). Please note the different scales of control values (left y-axis) and sensitivity values (right y-axis). In all cases the sensitivities and control values beyond 4 [s] are close to zero and are not depicted.

The sensitivities allow to update the control values using the linear Taylor approximation in (13) and, as guaranteed by Theorem 1, the approximation will be very accurate for small perturbations in p in $\hat{p}$. We omit numerical results as they simply confirm the theory. Instead we investigate the use of sensitivity updates in Theorem 3 in Section 4.2.

For a perturbation of -0.1 [m] in the component r and 0.002 [rad] in the component ψ of $\hat{q}_1$ at time $k = 1$ (resp. time $t_1 = h$) we obtain

$$u'_{1,N-1,1}(\hat{q}_1) = (0.0, -7.5413e - 02, -9.1921e - 01, -5.5644e + 00, 9.1921e - 01)^\top,$$
$$u'_{1,N-1,2}(\hat{q}_1) = (0.0, -3.3130e - 02, -5.1694e - 01, -3.9082e + 00, 5.1694e - 01)^\top.$$

These updates allow to compute approximate control values

$$\begin{aligned}\tilde{u}_{1,N-1,1}(q) &= u_{1,N-1,1}(\hat{q}_1) + u'_{1,N-1,1}(\hat{q}_1)(q_1 - \hat{q}_1),\\ \tilde{u}_{1,N-1,2}(q) &= u_{1,N-1,2}(\hat{q}_1) + u'_{1,N-1,2}(\hat{q}_1)(q_1 - \hat{q}_1).\end{aligned}$$

For comparison, the re-optimization of DOCP$(q_1, 1, N-1)$ in this example required 0.01092 [s], which is approximately six times the solution time for the sensitivity analysis 0.00174 [s]. The errors in the control values are $|u_{1,N-1,1}(q_1)-\tilde{u}_{1,N-1,1}(q_1)| = 0$ and $|u_{1,N-1,2}(q_1)-\tilde{u}_{1,N-1,2}(q_1)| \approx 2.068e-03$, which yields a sufficiently good approximation.

5.2. The case $R = 5$

We choose $R = 5$ as the weight for the control effort in the objective function, as it offers a good compromise between system aggressiveness and minimizing the required computational effort for solving the DOCP, see Section 3. This yields the trajectories displayed in figure 8 for the different MPC algorithms, as well as the system states illustrated in figures 9, 10, and 11. The control output depicted in figure 12 starts with a bang-bang arc, then the control remains on a free arc for the rest of the time interval and approaches zero quickly, such that the initial deviation is eliminated by all algorithms within the first 2.8 seconds (around $s = 42$ [m]).

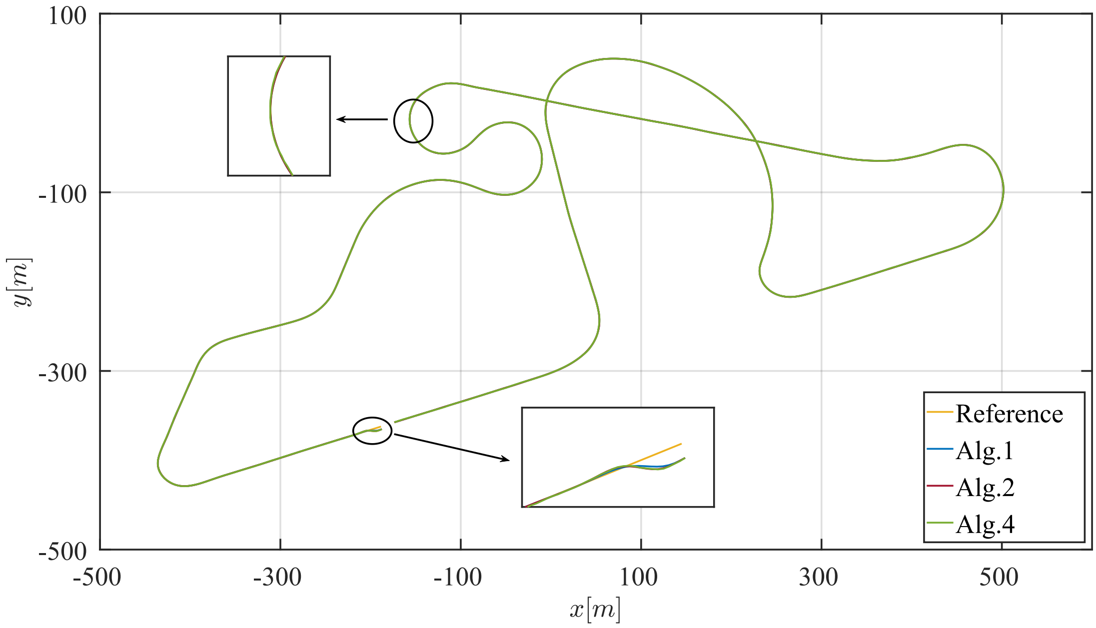

Figure 8. Generated vehicle trajectories as achieved by the MPC algorithms with $R = 5$.

Due to the aggressive controls, the state κ approaches both its upper and lower constraints during the bang-bang control arc, yet it does not activate the state constraints due to the rapid system stabilization. Hence, similar to $R = 100$, no state constraints are activated during the entire solution space.

As shown in table 2, all algorithms display significant improvement in all path tracking parameters when compared to $R = 100$, with the exception of a slight increase in $mean(||\psi-\psi_r||)$ for Alg.4. This can be explained by the higher sensitivity to ψ, which, combined with the random perturbations and the more aggressive controls, makes it difficult for the solver to correct minor deviations in the vehicle's heading. Furthermore, there is a noticeable increase in t_{CPU}, which corresponds to 150% increase in the required computational time in

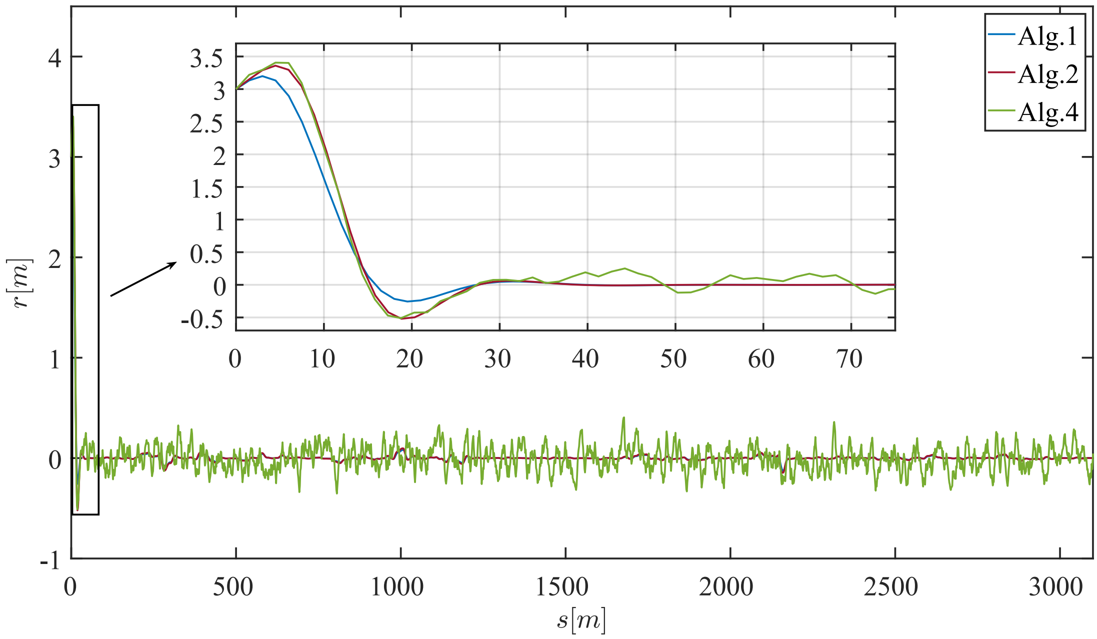

Figure 9. Vehicle's lateral offset r across the reference path with $R = 5$.

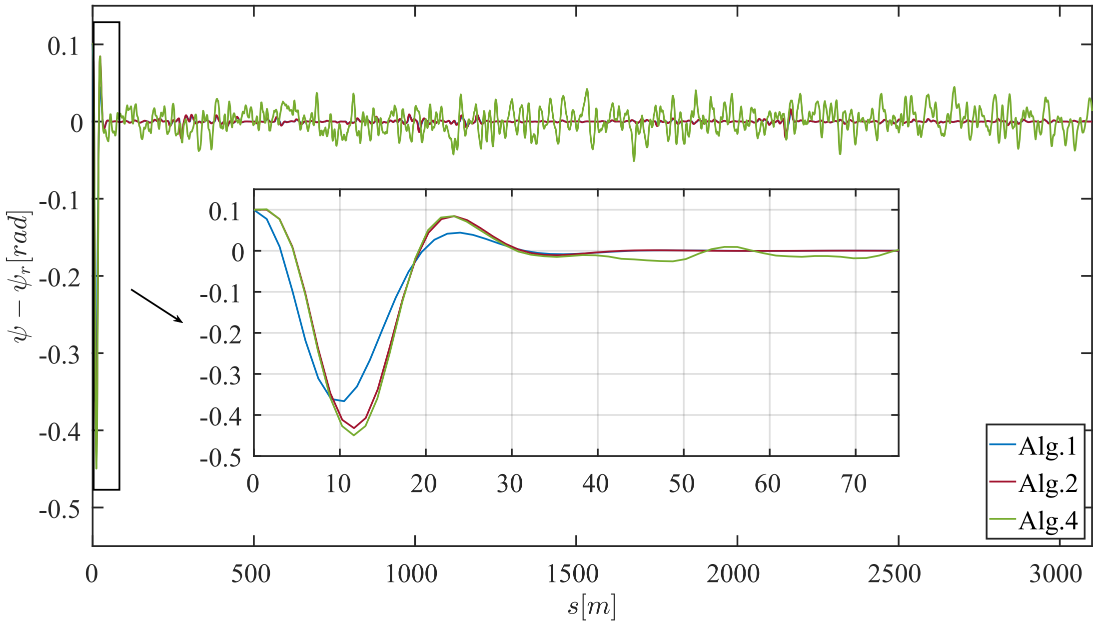

Figure 10. Deviation of the vehicle heading to the reference path $\psi - \psi_r$ with $R = 5$.

the worst case for Alg.4. However, the approach is still valid for real-time applications as $max(t_{CPU}) = 0.035966 < t_{k+1} - t_k = 0.1, k \in \mathbb{N}_0$. Finally, the time for solving (12), i.e. for computing the sensitivities, is $t_{CPU,s} \leq 0.004153$ [s] and it is in the same range as for $R = 100$ and again it is negligible.

Figure 13 shows the control and its sensitivities w.r.t. the initial state $\hat{p}$. The plots have different scales for the control value and the respective sensitivities. Especially the sensitivity w.r.t. the initial curvature $\kappa(0)$ is high whereas the sensitivity w.r.t. to the initial offset $r(0)$ is

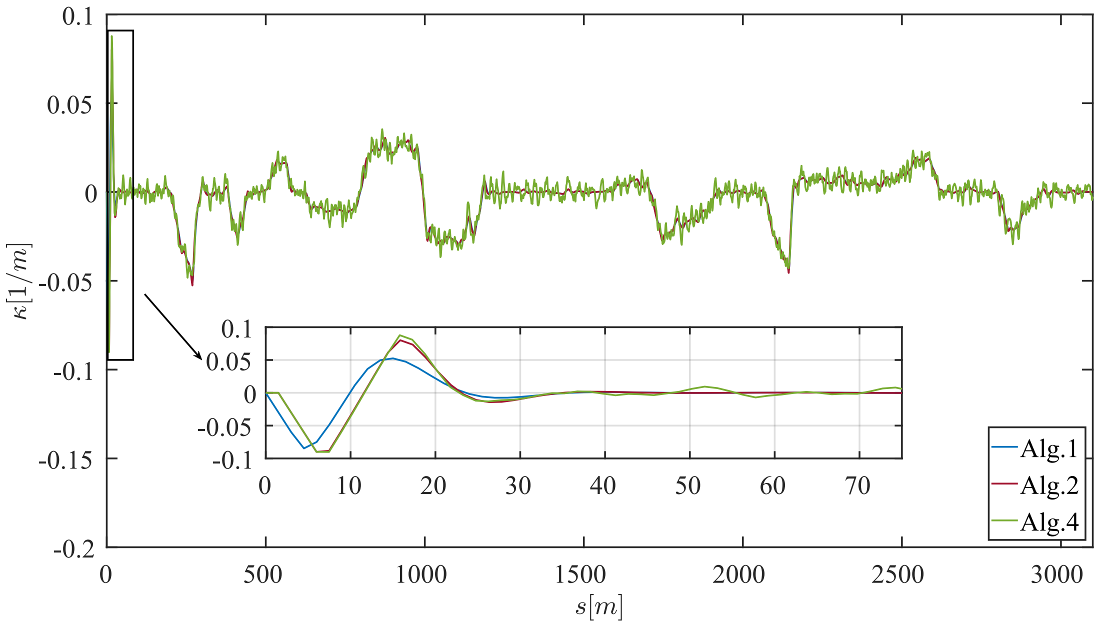

Figure 11. Despite the more aggressive response, the constraints for κ are not activated with $R = 5$.

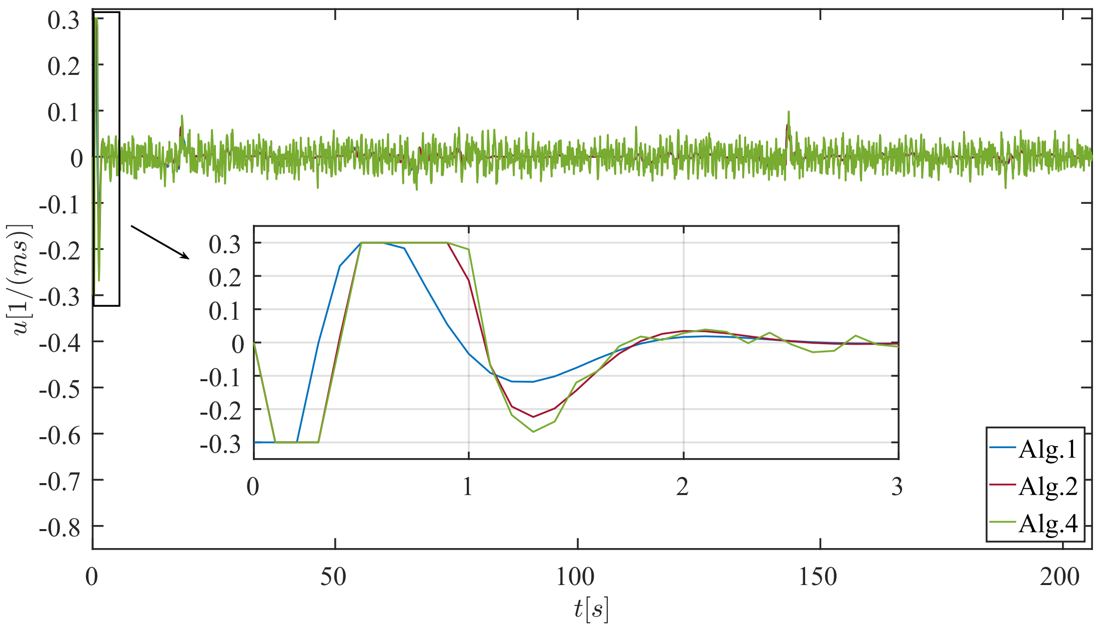

Figure 12. Numerical solution of DOCP with $R = 5$: Control u. Note the bang-bang arc at $t \in [0, 1.0]$ before rapid system stabilization.

comparatively low.

As stated by Theorem 1, the sensitivities of the control on the bang-bang arcs are zero. As a consequence, if the sensitivity update is used within the two MPC schemes, the active controls at the beginning of the time interval will remain unchanged under perturbations. Hence, the MPC schemes proceed with the control values on the boundary of the control constraints until the control leaves the box constraints. On the free arcs, the sensitivity updates kick in similar

Table 2. Path tracking results achieved by the MPC algorithms with $R = 5$

	$\|\|r\|\|$ [m]		$\|\|\psi - \psi_r\|\|$ [rad]		t_{CPU} [s]		$max(t_{CPU,s})$ [s]
	mean	*max*	*mean*	*max*	*mean*	*max*	
Alg.1	0.009891	0.125330	0.001108	0.017017	0.005627	0.029300	-
Alg.2	0.011739	0.146783	0.001301	0.021475	0.005524	0.030496	-
Alg.4	0.098705	0.405953	0.012630	0.051335	0.008387	0.035966	0.004153

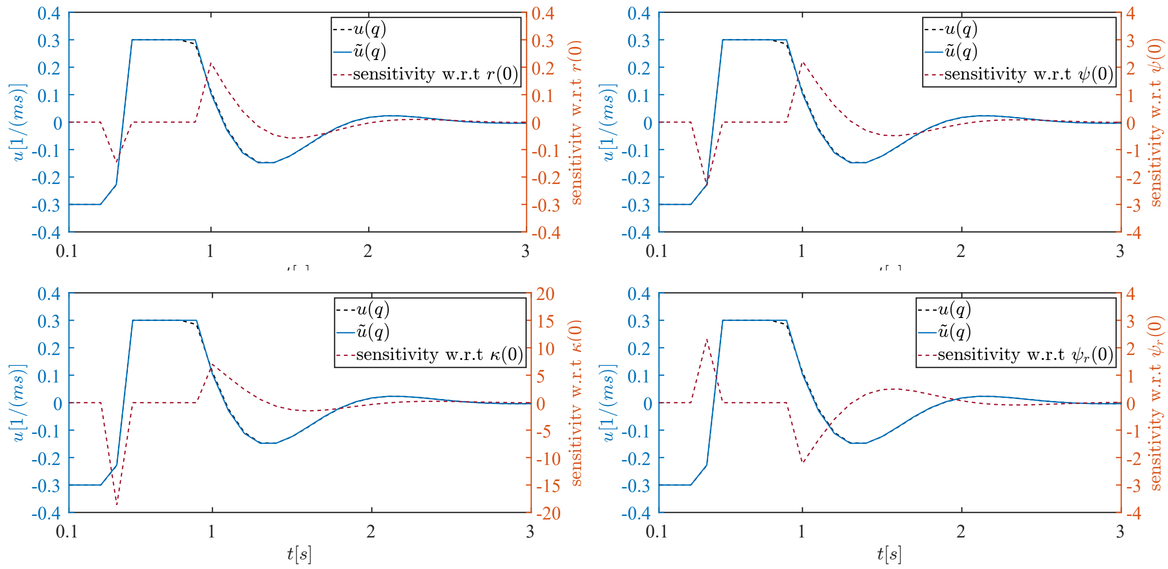

Figure 13. Comparison of control $\tilde{u}(q)$ from combined solution of DOCP$(\hat{p}, 1, 100)$ and sensitivity analysis against the control solution $u(q)$ from re-optimization with $R = 5$. Additionally, the sensitivities w.r.t. the relevant initial states $r(0), \psi(0), \kappa(0), \psi_r(0)$ are displayed (sensitivity w.r.t. $s(0)$ is $= 0$). Please note the different scales of control values (left y-axis) and sensitivity values (right y-axis). In all cases the sensitivities and control values beyond 3 [s] are close to zero and are not depicted.

to the case $R = 100$, where control bounds were inactive. Please note that this is not a mistake since the optimal solution in this case starts with an active control constraint.

6. Conclusion

We explored how parametric sensitivity analysis can be used in MPC schemes with prediction step and multi-step MPC schemes with re-optimization and prediction and showed that sensitivity updates can reduce the computational load while they deliver good approximations to perturbed optimal solutions. It was also shown that the sensitivity update has no effect on active control arcs, if the control starts with an active arc on the prediction horizon of the MPC step. While the numerical studies in this paper focused on a single optimal control problem in order to evaluate the sensitivity updates, further numerical studies are necessary for the full MPC control loop and the overall performance including stability investigations. Such investigations are beyond the scope of this study and we leave this open for future investigations.

Acknowledgments
This work is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr [projects EMERGENCY-VRD, MORE, SeRANIS, MissionLab]. dtec.bw is funded by the European Union – NextGenerationEU.

References

[1] Rawlings J B and Mayne D Q 2009 *Model Predictive Control: Theory and Design* (California, USA: Nob Hill Pub, LLC) ISBN 978-0975937709
[2] Grüne L and Pannek J 2011 *Nonlinear model predictive control: Theory and Algorithms* Communications and Control Engineering (London, England: Springer) ISBN 978-0-85729-500-2
[3] Diehl M, Bock H G and Schlöder J P 2005 *SIAM Journal on Control and Optimization* **43** 1714–1736
[4] Palma V G 2015 *Robust Updated MPC Schemes* Ph.D. thesis Fakultät für Mathematik, Physik und Informatik, Universität Bayreuth URL `https://epub.uni-bayreuth.de/id/eprint/2056`
[5] Zavala V M, Laird C D and Biegler L T 2008 *Journal of Process Control* **18** 876–884
[6] Fischer A 1992 *Optimization* **24** 269–284
[7] Gerdts M and Kunkel M 2008 *Journal of Industrial and Management Optimization* **4** 247–270
[8] Clarke F H 1990 *Optimization and Nonsmooth Analysis* (Philadelphia, USA: Society for Industrial and Applied Mathematics (SIAM))
[9] Kummer B 1988 Newton's method for non-differentiable functions *Advances in Mathematical Optimization* (De Gruyter) pp 114–125
[10] Kummer B 1992 Newton's method based on generalized derivatives for nonsmooth functions: Convergence analysis *Lecture Notes in Economics and Mathematical Systems* (Heidelberg, Germany: Springer Berlin Heidelberg) pp 171–194
[11] Qi L 1993 *Mathematics of Operations Research* **18** 227–244
[12] Qi L and Sun J 1993 *Mathematical Programming* **58** 353–367
[13] Chen X, Nashed Z and Qi L 2000 *SIAM Journal on Numerical Analysis* **38** 1200–1216
[14] Ulbrich M 2002 *SIAM Journal on Optimization* **13** 805–841
[15] Jiang H 1999 *Mathematics of Operations Research* **24** 529–543
[16] Büskens C and Maurer H 2001 Sensitivity analysis and real-time optimization of parametric nonlinear programming problems *Online Optimization of Large Scale Systems* (Heidelberg, Germany: Springer Berlin Heidelberg) pp 3–16
[17] Fiacco A V 1983 *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming* (Amsterdam, Boston: Academic Press) ISBN 978-0-080-95671-8
[18] Gerdts M 2012 *Optimal Control of ODEs and DAEs* (Berlin, Germany: De Gruyter)
[19] Jittorntrum K 1984 Solution point differentiability without strict complementarity in nonlinear programming *Sensitivity, Stability and Parametric Analysis* (Springer Berlin Heidelberg) pp 127–138
[20] Specht C, Gerdts M and Lampariello R 2020 Neighborhood estimation in sensitivity-based update rules for real-time optimal control *2020 European Control Conference (ECC)* (IEEE) pp 1999–2006
[21] Bemporad A, Bozinis N A, Dua V, Morari M and Pistikopoulos E N 2000 Model predictive control: A multi-parametric programming approach *Computer Aided Chemical Engineering* (Elsevier) pp 301–306
[22] Britzelmeier A and Gerdts M 2020 *IEEE Control Systems Letters* **4** 886–891