Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Inexact Adjoint-based SQP Algorithm for Real-Time Stochastic Nonlinear MPC

Xuhui Feng [*,**] **Stefano Di Cairano** [*] **Rien Quirynen** [*]

[*] *Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.*
[**] *ShanghaiTech University, Pudong, Shanghai, China.*

Abstract: This paper presents a real-time algorithm for stochastic nonlinear model predictive control (NMPC). The optimal control problem (OCP) involves a linearization based covariance matrix propagation to formulate the probabilistic chance constraints. Our proposed solution approach uses a tailored Jacobian approximation in combination with an adjoint-based sequential quadratic programming (SQP) method. The resulting algorithm allows the numerical elimination of the covariance matrices from the SQP subproblem, while ensuring Newton-type local convergence properties and preserving the block-sparse problem structure. It allows a considerable reduction of the computational complexity and preserves the positive definiteness of the covariance matrices at each iteration, unlike an exact Jacobian-based implementation. The real-time feasibility and closed-loop control performance of the proposed algorithm are illustrated on a case study of an autonomous driving application subject to external disturbances.

Keywords: Optimization algorithms, Stochastic model predictive control

1. INTRODUCTION

Nonlinear model predictive control (NMPC) has grown mature and shown its capability of handling relatively complex constrained processes (Rawlings et al., 2017). Although NMPC exhibits an inherent robustness due to feedback, such controllers do not take uncertainties directly into account and, consequently, the satisfaction of safety-critical constraints cannot be guaranteed in the presence of model uncertainties or external disturbances. One alternative approach is robust NMPC that relies on the optimization of control policies under worst-case scenarios in the presence of bounded uncertainty (Bemporad and Morari, 1999). However, robust NMPC can lead to a conservative control performance, due to the worst-case scenarios occurring with an extremely small probability.

Stochastic NMPC aims at reducing the conservativeness of robust NMPC by directly incorporating the probabilistic description of uncertainties into the optimal control problem (OCP) formulation (Mesbah, 2016). It requires constraints to be satisfied with a certain probability, i.e., by formulating so-called chance constraints that allow for a specified, yet non-zero, probability of constraint violation. In addition, stochastic NMPC is advantageous in settings where high performance in closed-loop operation is achieved near the boundaries of the plant's feasible region (Nagy and Braatz, 2007). In the general case, chance constraints are computationally intractable and typically require an approximate formulation (Mesbah et al., 2019).

Sampling techniques (Maciejowski et al., 2007) characterize the stochastic system dynamics using a finite set of random realizations of uncertainties, which may lead to a considerable computational cost. Scenario-based methods exploit an adequate representation of the probability distributions (Campi et al., 2009), but determining the number of scenarios leads to a trade off between robustness and efficiency (Calafiore and Fagiano, 2013). Gaussian-mixture approximations can be used to describe the transition probability distributions of states (Weissel et al., 2009), but the adaptation of the weights is often computationally expensive. Another approach relies on the use of polynomial chaos (PC) (Fagiano and Khammash, 2012), which replaces the implicit mappings with expansions of orthogonal polynomial basis functions but, for time-varying uncertainties, PC-based stochastic NMPC requires a large number of expansion terms (Mesbah, 2016).

We rely on a formulation of the chance constraints that uses a linearization-based covariance propagation, similar to (Gillis and Diehl, 2013; Telen et al., 2015). For ensuring computational tractability, we do not include nonlinearity bounders (Villanueva et al., 2017). Our main contribution is an inexact adjoint-based sequential quadratic programming (SQP) algorithm that allows the numerical elimination of the covariance matrices from the SQP subproblem while preserving the block-sparse problem structure, resulting in a considerable reduction of the computational complexity. The proposed optimization algorithm enjoys Newton-type convergence properties for local minimizers of the large-scale stochastic NMPC problem and it preserves the positive definiteness of the covariance matrices at each SQP iteration, unlike existing approaches. We present a tailored software implementation and illustrate its performance on a case study of real-time stochastic NMPC for an autonomous vehicle control system.

The paper is organized as follows. Section 2 introduces the stochastic NMPC formulation. Based on the description of SQP in Section 3, the inexact adjoint-based SQP algorithm is presented in Section 4. Section 5 discusses the efficient software implementation. Results of the case study are presented in Section 6 and Section 7 concludes the paper.

2. STOCHASTIC MODEL PREDICTIVE CONTROL

We consider nonlinear systems of the form

$$x_{k+1} = f(x_k, u_k, w_k), \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ denotes the state, $u_k \in \mathbb{R}^{n_u}$ the control inputs, $w_k \in \mathbb{R}^{n_w}$ the process noise, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ the right-hand side function. The disturbance $w_k \sim \mathcal{N}(0, \Sigma)$ is assumed to be a normally distributed signal with zero mean and variance Σ. In certainty-equivalent NMPC, the disturbances w_k are predicted to be zero. At each sampling time, nominal NMPC solves

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \sum_{k=0}^{N-1} l(x_k, u_k) + m(x_N) \\ \text{s.t.} \quad & \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ 0 = x_{k+1} - f(x_k, u_k, 0), \qquad x_0 = \hat{x}_t, \\ h(x_k, u_k) \le 0, u_{\min} \le u_k \le u_{\max}, \end{cases} \end{aligned} \tag{2}$$

based on the current state estimate $\hat{x}_t$. The function $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$ denotes the path constraints, while $u_{\min}$ and $u_{\max}$ are, respectively, the lower and upper bounds of admissible control values and $\mathbf{x} = (x_0, \dots, x_N)$, $\mathbf{u} = (u_0, \dots, u_{N-1})$. For simplicity, given an output function $\phi_k(x_k, u_k)$, we consider the stage and terminal cost to be least squares functions

$$l(\cdot) = \frac{1}{2}\|\phi_k(\cdot) - \phi_k^{\text{ref}}\|_{W_k}^2, \; m(\cdot) = \frac{1}{2}\|\phi_N(\cdot) - \phi_N^{\text{ref}}\|_{W_N}^2. \tag{3}$$

2.1 Linearization based Covariance Propagation

To achieve computational tractability, we use a linearization based approximate propagation of the state covariance matrix similar to (Gillis and Diehl, 2013; Telen et al., 2015). For the discrete-time system dynamics in (1), this results in the discrete-time Lyapunov equations

$$P_{k+1} = A_k P_k A_k^\top + B_k \Sigma B_k^\top, \quad P_0 = \hat{P}_t, \tag{4}$$

where $P_k \in \mathbb{R}^{n_x \times n_x}$ is the covariance matrix for the predicted state value x_k, the matrix $\hat{P}_t$ denotes the uncertainty of the current state estimate, and the Jacobian matrices A_k and B_k are computed as

$$A_k = \frac{\partial f}{\partial x}(x_k, u_k, 0), \quad B_k = \frac{\partial f}{\partial w}(x_k, u_k, 0). \tag{5}$$

Throughout this paper, we adopt the discrete-time Lyapunov equation instead of a continuous time formulation (Telen et al., 2015), to reduce the computational cost and to preserve the positive definiteness of the covariance matrix (Gillis and Diehl, 2013).

2.2 Probabilistic Chance Constraints

Taking uncertainty into account in the OCP formulation, we introduce individual chance constraints to ensure that the probability of violating each of the path constraints $h_i(x_k, u_k) \le 0$ is below a certain probability level ϵ_i, i.e.,

$$\Pr\left(h_i(x_k, u_k) \le 0\right) \ge 1 - \epsilon_i, \tag{6}$$

for each chance constraint $i = 1, \dots, n_h$ and at each instant $k = 0, \dots, N$ along the prediction horizon. Based on the state covariance propagation in (4), each chance constraint can be approximated by

$$h_i(x_k, u_k) + \alpha_i \sqrt{C_{k,i} P_k C_{k,i}^\top} \le 0, \tag{7}$$

where $C_k = \frac{\partial h}{\partial x}(x_k, u_k)$ is the constraint Jacobian matrix and $C_{k,i}$ is the i^{th} row of C_k. The back-off coefficient value α_i is computed to ensure the probability level ϵ_i in the chance constraint (6). One option is to use the Cantelli-Chebyshev inequality, $\alpha_i = \sqrt{\frac{1-\epsilon_i}{\epsilon_i}}$, which holds regardless of the underlying probability distribution, but may lead to relatively conservative bounds (Telen et al., 2015). An alternative approach is based on an approximation, assuming normally distributed state trajectories, such that the coefficient α_i can be chosen as

$$\alpha_i = \sqrt{2}\,\text{erf}^{-1}(1 - 2\epsilon_i), \tag{8}$$

where $\text{erf}^{-1}(\cdot)$ is the inverse error function.

2.3 Prestabilizing Feedback Control

The feedback control action should be taken into account in stochastic NMPC, for which different approaches have been proposed (Goulart et al., 2006; Mesbah, 2016). For simplicity, we rely on prestabilizing the nonlinear system dynamics based on an affine feedback law. Given the reference steady state and input $\left(x^{\text{ref}}, u^{\text{ref}}\right)$, we apply the infinite-horizon linear-quadratic regulator $u_k = K\,x_k$ for the linearized dynamics at the steady state, $A_r = \frac{\partial f}{\partial x}(x^{\text{ref}}, u^{\text{ref}}, 0)$ and $B_r = \frac{\partial f}{\partial u}(x^{\text{ref}}, u^{\text{ref}}, 0)$ and a quadratic stage cost of $x_k^\top Q x_k + u_k^\top R u_k$,

$$K = -\left(R + B_r^\top X B_r\right)^{-1} B_r^\top X A_r, \tag{9}$$

where the matrix X is computed by solving the discrete-time algebraic Riccati equation,

$$X = A_r^\top X A_r - A_r^\top X B_r \left(R + B_r^\top X B_r\right)^{-1} B_r^\top X A_r + Q.$$

As we consider a fixed linearization, K is a time-invariant feedback gain, but one could use a time-varying sequence of affine feedback laws as well.

2.4 Stochastic NMPC Problem Formulation

As a result, for the stochastic NMPC, we aim at solving at each sampling instant the nonlinear OCP

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u},\mathbf{P}} \quad & \sum_{k=0}^{N-1} l(x_k, u_k + K\,x_k) + m(x_N) \\ \text{s.t.} \quad & \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ 0 = x_{k+1} - f(x_k, u_k + K\,x_k, 0), \qquad x_0 = \hat{x}_t, \\ 0 = P_{k+1} - \left(\tilde{A}_k P_k \tilde{A}_k^\top + \tilde{B}_k \Sigma \tilde{B}_k^\top\right), \quad P_0 = \hat{P}_t, \\ u_{\min} \le u_k + K\,x_k \le u_{\max}, \\ 0 \ge h_i(x_k, u_k + K\,x_k) + \alpha_i \sqrt{C_{k,i} P_k C_{k,i}^\top}, \; \forall i, \end{cases} \end{aligned} \tag{10}$$

where the overall control action is in the feedforward-feedback form $u_k + K\,x_k$ due to the prestabilizing controller, and the Jacobian matrices $\tilde{A}_k = \frac{\partial f}{\partial x}(x_k, u_k + K\,x_k, 0)$ and $\tilde{B}_k = \frac{\partial f}{\partial w}(x_k, u_k + K\,x_k, 0)$.

Remark 1. The input bounds are imposed on the nominal control action $u_k + K\,x_k$ in (10). A more accurate problem formulation relies on probabilistic chance constraints for each control variable $j = 1, \dots, n_u$ as follows

$$u_{k,j} + K_j\,x_k + \alpha_j \sqrt{K_j P_k K_j^\top} \le u_{\max,j}, \tag{11}$$

where K_j denotes the j^{th} row of the matrix K. □

3. SEQUENTIAL QUADRATIC PROGRAMMING

In this section, we briefly introduce a compact notation of the nonlinear program (NLP) in (10) and we summarize the basics of sequential quadratic programming (SQP).

3.1 Compact Nonlinear Program Formulation

First, we introduce the optimization variables

$$\begin{aligned} y &= \left[x_0^\top, u_0^\top, \dots, x_{N-1}^\top, u_{N-1}^\top, x_N^\top\right]^\top, \\ z &= \left[\mathrm{vec}(P_0)^\top, \dots, \mathrm{vec}(P_{N-1})^\top, \mathrm{vec}(P_N)^\top\right]^\top, \end{aligned} \tag{12}$$

where P_k is symmetric such that $\mathrm{vec}(P_k) \in \mathbb{R}^{\frac{n_x(n_x+1)}{2}}$. Then, we introduce the shorthand notation

$$F(y) := \begin{bmatrix} x_0 - \hat{x}_t \\ \vdots \\ x_N - f(x_{N-1}, u_{N-1} + K x_{N-1}, 0) \end{bmatrix},$$

$$E(y,z) := \begin{bmatrix} P_0 - \hat{P}_t \\ \vdots \\ P_N - \left(\tilde{A}_{N-1} P_{N-1} \tilde{A}_{N-1}^\top + \tilde{B}_{N-1} \Sigma \tilde{B}_{N-1}^\top\right) \end{bmatrix}, \tag{13}$$

and $I(y,z)$ denotes the inequality constraints and $L(y)$ the function which is used as the least squares cost in (3). Therefore, the stochastic nonlinear OCP (10) can be compactly written as the NLP

$$\begin{aligned} \min_{y,z} \quad & \frac{1}{2}\|L(y)\|_2^2 \\ \text{s.t.} \quad & 0 = F(y), \quad 0 = E(y,z), \quad 0 \geq I(y,z), \end{aligned} \tag{14}$$

for which the Lagrangian function is

$$\Lambda(\cdot) := \frac{1}{2}\|L(y)\|_2^2 + \lambda^\top F(y) + \mu^\top E(y,z) + \kappa^\top I(y,z), \tag{15}$$

where λ and μ are the Lagrange multipliers for the equality and κ the ones for the inequality constraints.

Remark 2. Note that $E(\cdot)$ is linear in z but nonlinear in y and $\frac{\partial E}{\partial z}$ is invertible, i.e., z can be computed easily given y. Thus, we also introduce a compact notation to rewrite $0 = E(y,z)$ in an explicit form $z = E_z(y)$. □

Remark 3. The cost function in (14) only depends on the variables in y, which simplifies our notation. However, the proposed algorithms can be readily applied if the cost depends also on the covariance matrices in z. □

3.2 Exact Jacobian based SQP Algorithm (EX-SQP)

In a standard SQP for solving the NLP (14), each iteration i, given the solution guess (y^i, z^i), solves the quadratic subproblem

$$\begin{aligned} \min_{\Delta y, \Delta z} \quad & \frac{1}{2}\left(\Delta y^i\right)^\top H^i \Delta y^i + \left(g^i\right)^\top \Delta y^i \\ \text{s.t.} \quad & \begin{cases} \begin{array}{c|l} \sigma_F^i \\ \sigma_E^i \end{array} \; 0 = \begin{bmatrix} F(y^i) \\ E(y^i, z^i) \end{bmatrix} + \begin{bmatrix} \frac{\partial F}{\partial y}(\cdot) & 0 \\ \frac{\partial E}{\partial y}(\cdot) & \frac{\partial E}{\partial z}(\cdot) \end{bmatrix} \begin{bmatrix} \Delta y^i \\ \Delta z^i \end{bmatrix}, \\ \sigma_I^i \;\Big|\; 0 \geq I(y^i, z^i) + \begin{bmatrix} \frac{\partial I}{\partial y}(\cdot) & \frac{\partial I}{\partial z}(\cdot) \end{bmatrix} \begin{bmatrix} \Delta y^i \\ \Delta z^i \end{bmatrix}, \end{cases} \end{aligned} \tag{16}$$

to compute the new search direction $(\Delta y^i, \Delta z^i)$. Then, a full-step implementation of the SQP method updates the iterates as $y^{i+1} \leftarrow y^i + \Delta y^i$ and $z^{i+1} \leftarrow z^i + \Delta z^i$. Similarly, the Lagrange multipliers can be updated from (16) as $\lambda^{i+1} \leftarrow \sigma_F^i$, $\mu^{i+1} \leftarrow \sigma_E^i$, and $\kappa^{i+1} \leftarrow \sigma_I^i$.

Since the objective is of the least squares form, it is common to solve the NLP in (14) by the generalized Gauss-Newton (GGN) variant of the SQP algorithm (Bock, 1983). In this case, the Hessian of the Lagrangian $\Lambda(\cdot)$ can be approximated as

$$H^i := \frac{\partial L}{\partial y}(y^i)^\top \frac{\partial L}{\partial y}(y^i) \approx \nabla^2 \Lambda\left(y^i, z^i, \lambda^i, \mu^i, \kappa^i\right), \tag{17}$$

and the gradient is computed as $g^i := \frac{\partial L}{\partial y}(y^i)^\top L(y^i)$. In contrast to SQP applied to nominal NMPC, each QP subproblem (16) additionally involves the covariance matrices in z for the stochastic NMPC formulation. This leads to a considerable increase in the computational cost of each SQP iteration. To remedy this, we propose tailored inexact Newton-type implementations of SQP for stochastic NMPC, aimed at achieving a computational cost comparable to that of nominal NMPC.

4. INEXACT ADJOINT-BASED SQP ALGORITHM TAILORED TO STOCHASTIC NMPC

Next, we introduce our inexact adjoint-based SQP algorithm that is tailored to stochastic NMPC.

4.1 Inexact Adjoint-based SQP Algorithm (ADJ-SQP)

Based on Remark 2, first note that function $E(\cdot)$ is linear in z but nonlinear in y, the matrix $\frac{\partial E}{\partial z}$ is invertible while both $\frac{\partial E}{\partial z}(\cdot)$ and $\frac{\partial E}{\partial y}(\cdot)$ depend on the linearization point (y^i, z^i). Applying numerical elimination to Δz^i in the QP subproblem (16) would destroy the block-sparse problem structure that allows the efficient implementation of SQP for NMPC (Gros et al., 2016). Instead, we propose a tailored approximation for the Jacobian of the equality constraints $\tilde{J}_{\mathrm{eq}} \approx J_{\mathrm{eq}}$, which allows the elimination of the variables Δz^i while preserving the block-sparse problem structure. We present an inexact adjoint-based SQP algorithm that solves the QP subproblem (16) with the following Jacobian approximation

$$\tilde{J}_{\mathrm{eq}}^i = \begin{bmatrix} \frac{\partial F}{\partial y}(\cdot) & 0 \\ 0 & \frac{\partial E}{\partial z}(\cdot) \end{bmatrix} \approx \begin{bmatrix} \frac{\partial F}{\partial y}(\cdot) & 0 \\ \frac{\partial E}{\partial y}(\cdot) & \frac{\partial E}{\partial z}(\cdot) \end{bmatrix} = J_{\mathrm{eq}}^i. \tag{18}$$

Due to the inexact Jacobian matrix $\tilde{J}_{\mathrm{eq}}$, a correction to the gradient vector g^i is needed in each Newton-type iteration (Wirsching et al., 2006; Quirynen et al., 2018). This adjoint-based gradient correction is

$$g_{\mathrm{a}}^i := g^i + \left(J_{\mathrm{eq}} - \tilde{J}_{\mathrm{eq}}\right)^\top \begin{bmatrix} \lambda^i \\ \mu^i \end{bmatrix} = g^i + \frac{\partial E}{\partial y}(y^i, z^i)^\top \mu^i, \tag{19}$$

where $g^i := \frac{\partial L}{\partial y}(y^i)^\top L(y^i)$ denotes the objective gradient.

Because of the particular Jacobian approximation in (18), in which the derivative information $\frac{\partial E}{\partial y}(\cdot)$ is set to zero, the update of the z-variables simplifies to

$$\Delta z^i = -\frac{\partial E}{\partial z}(y^i, z^i)^{-1} E(y^i, z^i). \tag{20}$$

By inserting the inexact update (20) for Δz^i into the QP (16) with the Jacobian approximation (18), we obtain the equivalent reduced QP

$$\begin{aligned} \min_{\Delta y} \quad & \frac{1}{2}\left(\Delta y^i\right)^\top H^i \Delta y^i + \left(g_\mathrm{a}^i\right)^\top \Delta y^i \\ \text{s.t.} \quad & \begin{cases} \sigma_F^i \;\Big|\; 0 = F(y^i) + \dfrac{\partial F}{\partial y}(y^i)\Delta y^i, \\ \sigma_I^i \;\Big|\; 0 \geq \tilde{I}(y^i, z^i) + \dfrac{\partial I}{\partial y}(y^i, z^i)\Delta y^i, \end{cases} \end{aligned} \tag{21}$$

in which the condensed evaluation $\tilde{I}(\cdot)$ of the inequality constraints is

$$\tilde{I}(y^i, z^i) = I(y^i, z^i) - \frac{\partial I}{\partial z}(\cdot)\frac{\partial E}{\partial z}(\cdot)^{-1} E(y^i, z^i). \tag{22}$$

In each iteration of our inexact adjoint-based SQP method, based on the QP solution in (21), the updates for the variables are $y^{i+1} \leftarrow y^i + \Delta y^i$, $\lambda^{i+1} \leftarrow \sigma_F^i$, and $\kappa^{i+1} \leftarrow \sigma_I^i$. In addition, $z^{i+1} \leftarrow z^i + \Delta z^i$ and $\mu^{i+1} \leftarrow \sigma_E^i$ where Δz^i is computed by (20) and the Lagrange multipliers by

$$\sigma_E^i = -\frac{\partial E}{\partial z}(y^i, z^i)^{-\top} \frac{\partial I}{\partial z}(y^i, z^i)^\top \sigma_I^i. \tag{23}$$

Remark 4. The computations of $\tilde{I}(\cdot)$ and σ_E^i can be performed efficiently by exploiting the block-structured sparsity of the matrices. For example, $\tilde{I} = I - \frac{\partial I}{\partial z}\frac{\partial E}{\partial z}^{-1} E(y^i, z^i)$ can be computed sequentially as

$$\begin{aligned} \tilde{I}_0 &= I_0 - \frac{\partial I_0}{\partial z_0} E_0, \quad \tilde{E}_0 = E_0, \\ \tilde{E}_k &= E_k + (A_{k-1} \otimes A_{k-1})\, \tilde{E}_{k-1}, \\ \tilde{I}_k &= I_k - \frac{\partial I_k}{\partial z_k}\, \tilde{E}_k, \quad k = 1, \dots, N, \end{aligned} \tag{24}$$

based on matrix-vector multiplications for each block, where E_k and I_k denote the equality and inequality constraints at stage k, respectively, and $\tilde{E}_k$ denotes the intermediate result from the back substitution with $\frac{\partial E}{\partial z}$. □

4.2 Inexact SQP with Nonlinear Covariance Propagation

Next, we propose an alternative implementation of the inexact adjoint-based SQP algorithm, in which we aim at performing a more accurate update of the covariance matrices from one SQP iteration to the next. Instead of the inexact SQP update step in (20) to recover $z^{i+1} \leftarrow z^i + \Delta z^i$, we could apply the exact Jacobian step

$$\Delta z^i = -\frac{\partial E}{\partial z}(y^i, z^i)^{-1} \left(E(y^i, z^i) + \frac{\partial E}{\partial y}(y^i, z^i)\Delta y^i \right), \tag{25}$$

which can be implemented efficiently based on the structure exploitation as detailed in Remark 4. Yet another alternative is to use the nonlinear covariance propagation dynamics (4) to evaluate z^{i+1} directly as

$$z^{i+1} = E_z(y^{i+1}), \tag{26}$$

where the notation $E_z(\cdot)$ was introduced in Remark 2. Unlike the inexact SQP update in (20) or the exact Jacobian based update in (25), it is important to note that Eq. (26) intrinsically preserves the positive definiteness of the covariance matrices from one iteration to the next. In addition, note that $E(y^{i+1}, E_z(y^{i+1})) = 0$ such that $\tilde{I}(y^{i+1}, z^{i+1}) = I(y^{i+1}, z^{i+1})$ and the additional computations in Eq. (22) become unnecessary.

4.3 Adjoint-free Variant of Inexact SQP Algorithm

An alternative approach to implement the inexact SQP method is to remove the covariance propagation dynamics. This amounts to removing the constraints in $E(\cdot)$ and approximating the Jacobian $\frac{\partial I}{\partial z}(\cdot)$ by zero. Each iteration i of the resulting algorithm consists of two steps:

(1) Obtain $\bar{z}^i$ by propagating the covariance matrices
$$\bar{z}^i = E_z(y^i).$$
(2) Compute $y^{i+1} = y^i + \Delta y^i$ by solving the QP

$$\begin{aligned} \min_{\Delta y} \quad & \frac{1}{2}\left(\Delta y^i\right)^\top H^i \Delta y^i + \left(g^i\right)^\top \Delta y^i \\ \text{s.t.} \quad & \begin{cases} \sigma_F^i \;\Big|\; 0 = F(y^i) + \dfrac{\partial F}{\partial y}(y^i)\Delta y^i, \\ \sigma_I^i \;\Big|\; 0 \geq I(y^i, \bar{z}^i) + \dfrac{\partial I}{\partial y}(y^i, \bar{z}^i)\Delta y^i. \end{cases} \end{aligned} \tag{27}$$

The above inexact SQP approach is rather intuitive and was adopted, e.g., by (Hewing et al., 2018). Similar to our inexact adjoint-based SQP method, the above approach solves QP subproblems of the same dimensions as those of nominal NMPC, i.e., with states and controls as variables. But unlike our proposed method, this alternative approach performs an additional approximation of the linearized inequality constraints and it omits the computation of an adjoint-based gradient correction in (27).

The lack of a gradient correction means that the above inexact SQP optimization algorithm cannot converge to a solution of the original stochastic NMPC problem in (10). More specifically, due to the use of inexact Jacobian information, a fixed point to the above adjoint-free SQP iterations is generally not a local minimizer for the NLP in (10), as discussed in (Wirsching et al., 2006; Quirynen et al., 2018) and references therein.

5. EFFICIENT SOFTWARE IMPLEMENTATION FOR REAL-TIME STOCHASTIC NMPC

Following the standard real-time iteration (RTI) algorithm, as described for nominal NMPC in (Gros et al., 2016), we propose our tailored adjoint-based algorithm for real-time stochastic NMPC.

5.1 Adjoint-based Inexact RTI for Stochastic NMPC

To ensure real-time feasibility and to achieve fast feedback to the system, Algorithm 1 performs only one inexact adjoint-based SQP iteration per control time step. The approach consists of three main steps, including the preparation step that computes the Jacobian matrices and directional derivatives, followed by the solution of the block-sparse QP in (21) and the expansion of the covariance matrices and corresponding Lagrange multiplier values. Note that a shifting procedure should be performed when using the updated values for the primal and dual variables from Algorithm 1 as the solution guess to the same algorithm, from one control time step to the next.

5.2 Efficient Preparation of Block-Sparse QP Subproblem

One of the most computationally expensive tasks in preparing the QP subproblem (16) in an exact SQP algo-

Algorithm 1 Real-time Adjoint-based Stochastic NMPC

1: **Input:** Guess $(y^i, z^i, \lambda^i, \mu^i, \kappa^i)$, and feedback gain K.
`Preparation step` of QP subproblem:
2: **for** $k = 0, \dots, N$ **do**
3: Compute Jacobians $\frac{\partial F}{\partial y_k}(y_k^i)$ and $\frac{\partial I}{\partial y_k}(y_k^i, z_k^i)$.
4: Evaluate $\tilde{I}_k(y_k^i, z_k^i)$ in (22) using forward AD.
5: Evaluate gradient $g_{\mathrm{a},k}^i$ in (19) using adjoint AD.
6: **end for**
`Solution step` of block-sparse QP:
7: Receive current state estimate $\hat{x}_t$.
8: Solve QP in (21) to obtain Δy^i, σ_F^i and σ_I^i.
9: $y^{i+1} \leftarrow y^i + \Delta y^i$, $\lambda^{i+1} \leftarrow \sigma_F^i$, and $\kappa^{i+1} \leftarrow \sigma_I^i$.
10: `Feedback`: send control $u^* = u_0^{i+1} + K\hat{x}_t$ to process.
`Expansion step` for variables:
11: Compute $\mu^{i+1} \leftarrow \sigma_E^i$ in (23) using adjoint AD.
12: Compute $z^{i+1} \leftarrow z^i + \Delta z^i$ using Eq. (20) or (26).
13: **Output:** New values $(y^{i+1}, z^{i+1}, \lambda^{i+1}, \mu^{i+1}, \kappa^{i+1})$.

rithm is the evaluation of the Jacobian matrix J_{eq}^i for the equality constraints. Especially the Jacobian evaluation for the covariance propagation dynamics is often computationally expensive, consisting of N blocks of $\frac{n_{\mathrm{x}}(n_{\mathrm{x}}+1)}{2}$ equations in $E(\cdot)$. None of the proposed inexact SQP methods require the explicit evaluation of the complete Jacobian matrix $\left[\frac{\partial E}{\partial y}(\cdot)\ \frac{\partial E}{\partial z}(\cdot)\right]$. Instead, only directional derivatives are required for computing the gradient correction in (19), the condensed constraint evaluation in (22) and the expansion steps in (20) and (23).

The above mentioned directional derivatives can be evaluated efficiently using either the forward or the adjoint mode of algorithmic differentiation (AD), which require a computational cost that is a small multiple of the cost for the corresponding function evaluation (Griewank and Walther, 2008). The latter observation results in a considerable computational speedup for the QP preparation step in our proposed algorithm, in comparison with a standard exact Jacobian based implementation. In our preliminary C code implementation of Algorithm 1, we use the open-source software package `CasADi` (Andersson et al., 2018) and its code generation capabilities for the efficient evaluation of the directional derivatives.

5.3 Tailored Solution of Convex QP Subproblem

For our proposed inexact adjoint-based Algorithm 1, the QP in (21) has only $n_{\mathrm{x}} + n_{\mathrm{u}}$ variables per interval instead of $n_{\mathrm{x}} + n_{\mathrm{u}} + \frac{n_{\mathrm{x}}(n_{\mathrm{x}}+1)}{2}$ for the exact Jacobian-based subproblem (16). The QP subproblem in (21) is of the same form as the convex subproblem solved in the standard RTI approach to nominal NMPC. In our preliminary C code implementation, we use the recently proposed block-structured primal active-set method in `PRESAS` (Quirynen and Cairano, 2019). Block-sparse QP solvers tailored to MPC, such as `PRESAS`, typically have a computational cost that scales linearly with the horizon length N, but cubically in the number of optimization variables per interval (Quirynen and Cairano, 2019). Therefore, the solution by the inexact SQP algorithm has a computational complexity of $\mathcal{O}(n_{\mathrm{x}}^3)$ instead of $\mathcal{O}(n_{\mathrm{x}}^6)$ for the case of the exact SQP.

6. NUMERICAL CASE STUDY: AUTONOMOUS VEHICLE CONTROL UNDER UNCERTAINTY

Our simulation case study is based on a stochastic formulation of the NMPC trajectory tracking controller that was used in the autonomous driving system on the test platform of small-scale vehicles in (Berntorp et al., 2018).

6.1 Vehicle Model and Problem Formulation

We use the kinematic single-track vehicle model

$$\begin{bmatrix} \dot{p}_{\mathrm{X}}(t) \\ \dot{p}_{\mathrm{Y}}(t) \\ \dot{\psi}(t) \\ \dot{\delta}_{\mathrm{f}}(t) \end{bmatrix} = \begin{bmatrix} v_{\mathrm{x}} \cos(\psi + \beta) \\ v_{\mathrm{x}} \sin(\psi + \beta) \\ \frac{v_{\mathrm{x}}}{L} \tan(\delta_{\mathrm{f}}) \cos(\beta) \\ \frac{1}{\tau}(\delta + \delta_0 - \delta_{\mathrm{f}}) \end{bmatrix}, \tag{28}$$

where $\beta = \arctan\left(l_{\mathrm{r}} \tan(\delta_{\mathrm{f}})/L\right)$ is the body-slip angle and $L = l_{\mathrm{f}} + l_{\mathrm{r}}$ denotes the wheel base. The state vector is $x = [p_{\mathrm{X}}\ p_{\mathrm{Y}}\ \psi\ \delta_{\mathrm{f}}]^\top \in \mathbb{R}^4$, in which p_{X} and p_{Y} denote the longitudinal and lateral position in the world frame, ψ is the heading angle and δ_{f} the front wheel steering angle. The input vector is $u = [v_{\mathrm{x}}\ \delta]^\top \in \mathbb{R}^2$ where v_{x} is the longitudinal velocity and δ is the commanded steering angle. In (28), δ_0 denotes an input disturbance that groups model errors and actuator offsets and is modeled as a normally distributed disturbance $\delta_0 \sim \mathcal{N}(0, \Sigma)$. The nonlinear system dynamics (28) are discretized using the explicit Runge-Kutta method of order 4.

The goal of the stochastic NMPC controller is to track a reference motion that corresponds to the sharp lane change maneuver shown in Figure 2, while considering safety-critical constraints under the uncertainty. More specifically, the objective function is of the form in Eq. (3) based on a time trajectory of reference position values $\left(p_{\mathrm{X}}^{\mathrm{ref}}(t_i), p_{\mathrm{Y}}^{\mathrm{ref}}(t_i)\right)_{i=0,\dots}$ and a constant reference velocity $v_{\mathrm{x}}^{\mathrm{ref}}$. We introduce simple bounds $u_{\min} \leq u_k + K x_k \leq u_{\max}$ on the control inputs and, for our simplified setting, the road boundary constraints can be imposed directly with respect to p_{Y} position by time-varying lower and upper bounds (see Figure 2). We approximate the feasible region on the road by the discrete-time chance constraints

$$\begin{aligned} p_{\mathrm{Y,k}} - p_{\mathrm{Y,k}}^{\max} + \alpha\sqrt{P_{2,k}} - s_k &\leq 0, \\ p_{\mathrm{Y,k}}^{\min} - p_{\mathrm{Y,k}} + \alpha\sqrt{P_{2,k}} - s_k &\leq 0, \end{aligned} \tag{29}$$

where $P_{2,k}$ denotes the variance for $p_{\mathrm{Y,k}}$ and the back-off value α is computed based on the maximum allowed violation rate $\epsilon = 0.05$ in Eq. (8). We additionally introduced the slack variables $s_k \geq 0$, with an exact penalty in the cost function, such that a feasible solution to the constrained optimization problem always exists. The NMPC controller uses a prediction horizon of $T = 2.0$ s, with 20 control intervals and 40 Runge-Kutta integration steps.

6.2 Numerical Results: Local Convergence Analysis

Let us first illustrate the Newton-type convergence properties of our proposed inexact adjoint-based SQP algorithm by comparing the following variants:

(`EX`) exact Jacobian SQP method in Section 3.2.
(`ADJ`) inexact adjoint-based SQP method in Section 4.1.
(`IN`) inexact adjoint-free SQP method in Section 4.3.

We introduce an additional SQP algorithm (ADJ-IN) that corresponds to the inexact method in Section 4.3, including the Jacobian approximation in the inequality constraints, but with an adjoint-based gradient correction. Figure 1 shows the convergence for each of these SQP algorithms, compared to a high-accuracy solution that was obtained by IPOPT for the stochastic NMPC problem.

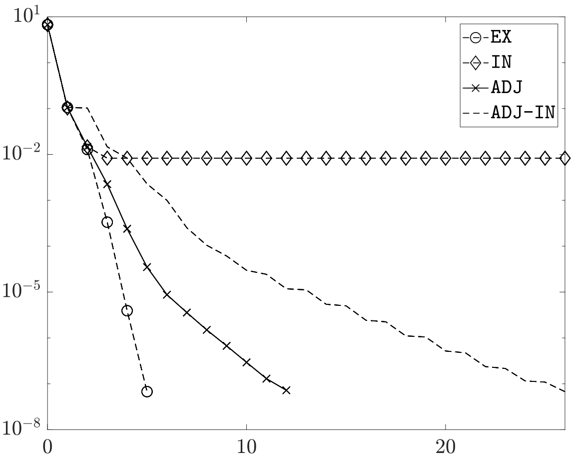

Fig. 1. Convergence results on the stochastic NMPC case study for 4 SQP variants: EX, ADJ, ADJ-IN and IN.

From the results in Figure 1, one can immediately observe that IN does not converge to the same solution due to the lack of a gradient correction in (27). More specifically, IN would not converge to the same solution when initialized arbitrarily close to this minimizer. Instead, the adjoint-based variants do converge to the local solution of the stochastic NMPC problem and it can be observed that ADJ leads to a local convergence rate that is faster than ADJ-IN but slower than EX, for this case study.

6.3 Closed-loop Performance for Stochastic NMPC

From now on, we further refer to our proposed stochastic MPC (SMPC) approach in Algorithm 1 as ADJ-SMPC. Similarly, we refer to the same stochastic NMPC controller using either of the alternative SQP variants in the RTI framework as EX-SMPC and IN-SMPC. In addition, we used IPOPT to implement the SMPC in IPOPT-SMPC and the exact Jacobian based RTI algorithm for certainty-equivalent MPC, EX-MPC. All Newton-type variants (EX, ADJ and IN) perform only one SQP iteration per control time step, while IPOPT-SMPC solves each OCP to a high accuracy. Our aim is to compare the closed-loop performance of these NMPC controllers for the case study in Figure 2, given the normally distributed external disturbance $\delta_0 \sim \mathcal{N}(0, \Sigma)$.

Figure 3 presents a comparison of the closed-loop performance for the five implementations of the NMPC controller for the vehicle control case study in Figure 2, under 1000 random realizations of the trajectory of external disturbances. Figure 3 shows the total area of violation for the vehicle position outside the road boundaries in blue and the total closed-loop tracking cost in green, as well as the relative difference for each algorithm compared to IPOPT-SMPC. It can be observed that EX-MPC achieves the best tracking performance since no robustness information is taken into account, but it simultaneously leads

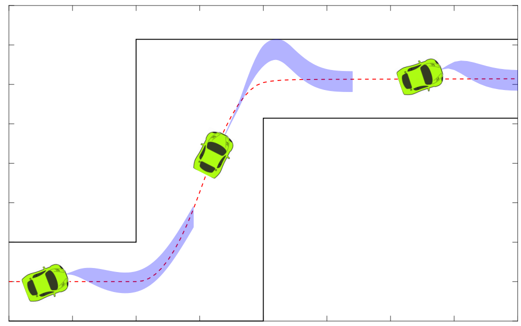

Fig. 2. Illustration of the predicted uncertainty propagation in the stochastic NMPC formulation for a sharp lane change maneuver of an autonomous vehicle.

to the largest amount of constraint violations. On the other hand, IPOPT-SMPC reduces considerably the total area of violations at the cost of a relatively small increase in the tracking cost. The trade off between robustness and tracking performance can be clearly observed.

Focusing on the online Newton-type SQP algorithms for stochastic NMPC in Figure 3, it can be observed that our ADJ-SMPC implementation outperforms both IN-SMPC and the certainty-equivalent EX-MPC. In addition, the inexact ADJ-SMPC exhibits a closed-loop performance that is very close to that of the EX-SMPC method, in terms of both violations and tracking cost, even though the ADJ-SMPC implementation in Algorithm 1 has a much lower computational complexity, as discussed next.

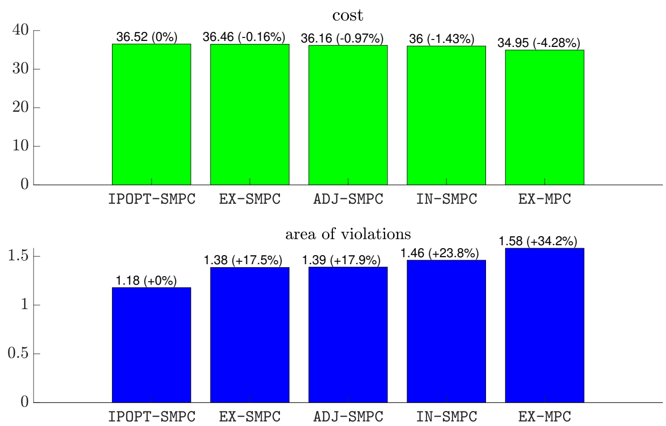

Fig. 3. Closed-loop performance of five different NMPC implementations for the vehicle control case study for 1000 random realizations of the external disturbances: tracking cost in green and violations in blue (relative difference compared to IPOPT-SMPC in parentheses).

6.4 Computational Timing Results

Table 1 summarizes the average and worst-case timing results per call to the different Newton-type implementations of NMPC. The required computational effort for the QP preparation step is deterministic, corresponding to the evaluation of nonlinear functions and derivatives. On the other hand, since the number of iterations can be different for each control time step, Table 1 shows both the average and worst-case timings for the entire QP solution by PRESAS. In addition, the timing results for `IPOPT-SMPC` have been included as a reference, even though this implementation can be considered unsuited for a real-time application in our case study.

Table 1. Average and worst-case computation times per iteration of the NMPC algorithms. [1]

	IPOPT SMPC	EX SMPC	ADJ SMPC	IN SMPC	EX MPC
QP preparation	–	3.69 ms	1.28 ms	0.82 ms	0.57 ms
QP solution (A)	–	1.83 ms	0.32 ms	0.32 ms	0.32 ms
Total time	534 ms	5.52 ms	1.60 ms	1.14 ms	0.89 ms
QP solution (W)	–	8.74 ms	1.72 ms	1.71 ms	1.71 ms
Total time	973 ms	12.43 ms	3.00 ms	2.53 ms	2.28 ms

Each QP subproblem in `EX-SMPC` has 14 state variables (10 for the covariance) and 2 control inputs, versus the original 4 states and 2 inputs in the algorithms `ADJ-SMPC`, `IN-SMPC` and `EX-MPC`. Therefore, the QP solution in the latter inexact methods is about 5 times faster than in `EX-SMPC`, resulting in an overall speedup of factor $4-5$. The use of the reverse mode of AD in calculating the gradient correction (19) leads to a computationally efficient preparation phase in `ADJ-SMPC`, which is about 50 % more expensive than that of `IN-SMPC`. An important observation to be made is that `ADJ-SMPC` leads to an increase of the total computation time of only 30 % compared to `EX-MPC` and 20 % compared to the `IN-SMPC` controller, even though `ADJ-SMPC` leads to convergence in Figure 1 and considerably improved robustness in Figure 3.

7. CONCLUSIONS

We proposed a real-time stochastic NMPC algorithm using an inexact adjoint-based Newton-type method that exploits the particular problem structure. The method enjoys standard Newton-type convergence results and preserves positive definiteness of the covariance matrices at each iteration. We presented a software implementation and illustrated its convergence properties, its computational efficiency and closed-loop control performance on a stochastic NMPC implementation for a vehicle control case study. Our adjoint-based stochastic NMPC algorithm was shown to require only 30 % more computations than nominal NMPC and resulted in a speedup of factor 4 compared to an exact Jacobian based implementation.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl., M. (2018). Casadi–a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation.*

Bemporad, A. and Morari, M. (1999). Robust model predictive control: A survey. In *Robustness in identification and control*, 207–226. Springer.

Berntorp, K., Hoang, T., Quirynen, R., and Di Cairano, S. (2018). Control architecture design for autonomous vehicles. In *IEEE Conf. Contr. Tech. Appl.*

Bock, H.G. (1983). Recent advances in parameter identification techniques for ODE. In *Treatment of Inverse Problems in Differential and Integral Equations*, 95–121.

[1] Computation times were obtained on a powerful computer that is equipped with an Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz.

Calafiore, G.C. and Fagiano, L. (2013). Stochastic model predictive control of LPV systems via scenario optimization. *Automatica*, 49(6), 1861–1866.

Campi, M.C., Garatti, S., and Prandini, M. (2009). The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2), 149–157.

Fagiano, L. and Khammash, M. (2012). Nonlinear stochastic model predictive control via regularized polynomial chaos expansions. In *IEEE Conference on Decision and Control*, 142–147.

Gillis, J. and Diehl, M. (2013). A positive definiteness preserving discretization method for nonlinear Lyapunov differential equations. In *Proc. IEEE Conf. Decision and Control (CDC).*

Goulart, P.J., Kerrigan, E.C., and Maciejowski, J.M. (2006). Optimization over state feedback policies for robust control with constraints. *Autom.*, 42(4), 523–533.

Griewank, A. and Walther, A. (2008). *Evaluating Derivatives.* SIAM, 2 edition.

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2016). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control.*

Hewing, L., Liniger, A., and Zeilinger, M.N. (2018). Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars. In *European Contr. Conf.*

Maciejowski, J.M., Visintini, A.L., and Lygeros, J. (2007). NMPC for complex stochastic systems using a Markov chain Monte Carlo approach. In *Assessment and Future Directions of NMPC*, 269–281. Springer.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6), 30–44.

Mesbah, A., Kolmanovsky, I.V., and Di Cairano, S. (2019). Stochastic model predictive control. In *Handbook of Model Predictive Control*, 75–97. Springer Int. Publ.

Nagy, Z. and Braatz, R.D. (2007). Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17(3), 229–240.

Quirynen, R., Gros, S., and Diehl, M. (2018). Inexact Newton-type optimization with iterated sensitivities. *SIAM Journal on Optimization*, 28(1), 74–95.

Quirynen, R. and Cairano, S.D. (2019). PRESAS: Block-structured preconditioning of iterative solvers within a primal active-set method for fast MPC. *arXiv preprint arXiv:1912.02122.*

Rawlings, J.B., Mayne, D.Q., and Diehl, M.M. (2017). *Model Predictive Control: Theory, Computation, and Design.* Nob Hill, 2nd edition edition.

Telen, D., Vallerio, M., Cabianca, L., Houska, B., Impe, J.V., and Logist, F. (2015). Approximate robust optimization of nonlinear systems under parametric uncertainty and process noise. *J. Proc. Contr.*, 33, 140 – 154.

Villanueva, M.E., Quirynen, R., Diehl, M., Chachuat, B., and Houska, B. (2017). Robust MPC via min-max differential inequalities. *Automatica*, 77, 311–321.

Weissel, F., Huber, M.F., and Hanebeck, U.D. (2009). Stochastic nonlinear model predictive control based on gaussian mixture approximations. In *Informatics in control, automation and robotics*, 239–252. Springer.

Wirsching, L., Bock, H.G., and Diehl, M. (2006). Fast NMPC of a chain of masses connected by springs. In *IEEE Inter. Conf. Contr. Appl.*, 591–596.