Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation

Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke

Optimization in Engineering Center (OPTEC) and ESAT-SCD,
K.U. Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
{moritz.diehl,joachim.ferreau,niels.haverbeke}@esat.kuleuven.be

Keywords : real-time optimization, numerical optimal control, Newton type methods, structure exploitation

This overview paper reviews numerical methods for solution of optimal control problems in real-time, as they arise in nonlinear model predictive control (NMPC) as well as in moving horizon estimation (MHE). In the first part, we review numerical optimal control solution methods, focussing exclusively on a discrete time setting. We discuss several algorithmic "building blocks" that can be combined to a multitude of algorithms. We start by discussing the sequential and simultaneous approaches, the first leading to smaller, the second to more structured optimization problems. The two big families of Newton type optimization methods, Sequential Quadratic Programming (SQP) and Interior Point (IP) methods, are presented, and we discuss how to exploit the optimal control structure in the solution of the linear-quadratic subproblems, where the two major alternatives are "condensing" and band structure exploiting approaches. The second part of the paper discusses how the algorithms can be adapted to the real-time challenge of NMPC and MHE. We recall an important sensitivity result from parametric optimization, and show that a tangential solution predictor for online data can easily be generated in Newton type algorithms. We point out one important difference between SQP and IP methods: while both methods are able to generate the tangential predictor for fixed active sets, the SQP predictor even works across active set changes. We then classify many proposed real-time optimization approaches from the literature into the developed categories.

1 Introduction

Nonlinear optimal control algorithms are at the core of all nonlinear MPC or moving horizon estimation (MHE) schemes. In contrast to linear MPC, where convex quadratic programs are mostly solved exactly at each sampling time, nonlinear MPC faces a dilemma: either the nonlinear iteration procedure is performed until a pre-specified convergence criterion is met, which might introduce considerable feedback delays, or the procedure is stopped prematurely with only an approximate solution, so that a pre-specified computation time limit can be met. Fortunately, considerable progress has been achieved in the last decade that allows to reduce both, computational delays and approximation errors. This progress would not have been possible by using just off-the-shelf

Int. Workshop on Assessment and Future Directions of NMPC
Pavia, Italy, September 5-9, 2008

optimal control codes; it is the development of dedicated real-time optimization algorithms for NMPC and MHE that allows to nowadays apply NMPC to plants with tens of thousands of states or to mechatronic applications.

While several excellent numerical optimization textbooks exist [25, 28, 44], in the field of numerical *optimal control* there are only a few [2, 11], and when it comes to real-time optimal control algorithms there is even less overview material [5]. The aim of the present article is to help closing this gap and to summarize the state-of-the-art in this field by presenting those algorithmic ideas that appear to be crucial to the authors. We choose a rather simplified setting, leaving many important special cases aside, in order to present the major ideas as clearly as possible.

The article is organized as follows: In Section 2 the NMPC and MHE problems are stated, in Section 3 we review Newton type optimization methods of different flavor, and in Section 4 we discuss how to exploit the optimal control structure of the linear equation systems to be solved in each Newton type iteration. In Section 5 we present online initialization strategies for subsequent NMPC problems, and in Section 6 the online algorithms of different flavours are discussed, and we finally conclude the paper in Section 7.

2 Problem Formulation

Throughout this paper we regard discrete time dynamical systems augmented with algebraic equations, as follows:

$$x_{k+1} = f_k(x_k, z_k, u_k) \tag{1a}$$

$$g_k(x_k, z_k, u_k) = 0 \tag{1b}$$

Here, $x_k \in \mathbb{R}^{n_x}$ is the differential state, $z_k \in \mathbb{R}^{n_z}$ the algebraic state, and $u_k \in \mathbb{R}^{n_u}$ is the control. Functions f_k and g_k are assumed twice differentiable and map into $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_z}$, respectively. The algebraic state z_k is uniquely determined by (1b) when x_k and u_k are fixed, as we assume that $\frac{\partial g_k}{\partial z}$ is invertible everywhere.

We choose to regard this difference-algebraic system form because it covers several parametrization schemes for continuous time dynamic systems in differential algebraic equation (DAE) form, in particular direct multiple shooting with DAE relaxation [39] and direct collocation [3, 59]. Note that in the case of collocation, all collocation equations on a collocation interval would be collected within the function g_k and the collocation node values in the variables z_k.

2.1 NMPC Optimal Control Problem

Based on this dynamic system form, we regard the following simplified optimal control problem in discrete time:

$$
\begin{aligned}
\underset{x,\,z,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) \quad + \quad E\,(x_N) && \text{(2a)}\\
\text{subject to} \quad x_0 - \bar{x}_0 &= 0, && \text{(2b)}\\
x_{i+1} - f_i(x_i, z_i, u_i) &= 0, \quad i = 0, \ldots, N-1, && \text{(2c)}\\
g_i(x_i, z_i, u_i) &= 0, \quad i = 0, \ldots, N-1, && \text{(2d)}\\
h_i(x_i, z_i, u_i) &\leq 0, \quad i = 0, \ldots, N-1, && \text{(2e)}\\
r\,(x_N) &\leq 0. && \text{(2f)}
\end{aligned}
$$

Here, the free variables are the differential state vector $x = (x_0^T, x_1^T \ldots, x_{N-1}^T, x_N^T)^T$ at all considered time points and the algebraic and control vector on all but the last time points: $z = (z_0^T, z_1^T \ldots, z_{N-1}^T)^T$ and $u = (u_0^T, u_1^T \ldots, u_{N-1}^T)^T$.

Remark on fixed and free parameters: In most NMPC applications there are some *constant* parameters $\bar{p}$ that are assumed constant for the NMPC optimization, but that change for different problems, like $\bar{x}_0$. We do not regard them here for notational convenience, but note that they can be regarded as constant system states with fixed initial value $\bar{p}$. In some NMPC applications *free* parameters p exist that are part of the optimization variables, but that are – in contrast to the controls u_k – constant in time. Again, we disregard this case for notational simplicity.

2.2 Moving Horizon Estimation: Nearly a Dual Problem

For moving horizon estimation (MHE), see e.g. [21, 48, 65], we typically choose convex functions to penalize the mismatch between the real measurements y_k and the corresponding model predictions $m_k(x_k, z_k, u_k, w_k)$. For notational simplicity, we regard only weighted Euclidean norms here, $\|y_k - m_k(x_k, z_k, u_k, w_k)\|_Q^2$, but point out that it is often useful to regard other penalty functions like e.g. the ℓ_1 penalty, which necessitate slight adaptations in the numerical solution algorithms presented later. The controls u_k are here regarded as fixed and known and enter the system dynamics only as constant but time varying parameters. However, time varying disturbances w_k are often introduced in the MHE problem to account for plant-model mismatch. They take the same role as the controls in the NMPC problem and are often ℓ_2 penalized.

$$
\begin{aligned}
\underset{x,\,z,\,w}{\text{minimize}} \quad & \|x_0 - \bar{x}_0\|_P^2 \; + \; \sum_{i=0}^{N-1} \; \|y_i - m_i(x_i, z_i, u_i, w_i)\|_Q^2 + \|w_i\|_R^2 && \text{(3a)}\\
\text{subject to} \quad & \\
x_{i+1} - f_i(x_i, z_i, u_i, w_i) &= 0, \quad i = 0, \ldots, N-1, && \text{(3b)}\\
g_i(x_i, z_i, u_i, w_i) &= 0, \quad i = 0, \ldots, N-1, && \text{(3c)}\\
h_i(x_i, z_i, u_i, w_i) &\leq 0, \quad i = 0, \ldots, N-1, && \text{(3d)}
\end{aligned}
$$

Due to the fact that the MHE problem has the same optimal control structure as the NMPC problem, they are often called "dual" to each other, in a slight abuse of terminology. However, the starkest contrast to the NMPC problem is the fact that the MHE problem has a free initial value x_0 and often has a much higher dimensional "control vector" w_k. This necessitates possibly different linear algebra solvers in the solution procedures described below.

2.3 Sequential vs. Simultaneous Optimal Control

For simplicity of presentation, we will in this subsection only focus on the NMPC problem (2a)-(2f). Here, the equality constraints (2b)-(2d) uniquely determine the variables x and z if the vector u is fixed. Thus, they can be inverted to yield the implicit functions $\tilde{x}(u)$ and $\tilde{z}(u)$ that satisfy (2b)-(2d) for all u, by a system simulation. It allows to reduce the optimization problem to

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) \quad + \quad E\left(\tilde{x}_N(u)\right) && \text{(4a)} \\ \text{subject to} \quad & h_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) \quad \leq \quad 0, \quad i = 0, \ldots, N-1, && \text{(4b)} \\ & r\left(\tilde{x}_N(u)\right) \quad \leq \quad 0. && \text{(4c)} \end{aligned}$$

This problem has a strongly reduced variable space compared to the original problem, and it is thus an appealing idea to use the reduced problem within an optimization procedure. This gives rise to the so called "sequential" approach to optimal control problems, where in each optimization iteration the two steps, system simulation and optimization, are performed sequentially, one after the other. This approach emerged early in the nonlinear optimal control literature [50].

In contrast to the sequential approach, the so called "simultaneous" approach addresses the full nonlinear program as stated above in (2a)-(2f) directly by a Newton type optimization algorithm, i.e., optimization and simulation are performed simultaneously. It comes in the form of direct collocation methods [3, 59, 64] as well as in form of direct multiple shooting [9, 39].

The optimization problem of the sequential approach has much less variables, but also less structure in the linear subproblems than the simultaneous approach (an interesting structure preserving sequential algorithm was however presented in [58]). Even more important, the Newton type optimization procedure behaves quite differently for both approaches: typically, faster local convergence rates are observed for the simultaneous approach, in particular for unstable or highly nonlinear systems, because – intuitively speaking – the nonlinearity is equally distributed over the nodes.

3 Newton Type Optimization

Newton's method for solution of a nonlinear equation $R(W) = 0$ starts with an initial guess W^0 and generates a series of iterates W^k that each solves a linearization of the system at the previous iterate, i.e., for given W^k the next iterate W^{k+1} shall satisfy $R(W^k) + \nabla R(W^k)^T(W^{k+1} - W^k) = 0$. The hope is that the linearizations – that can be solved w.r.t. W^{k+1} by standard linear algebra tools – are sufficiently good approximations of the original nonlinear system and that the iterates converge towards a solution W^*. Newton's method has locally a quadratic convergence rate, which is as fast as making any numerical analyst happy. If the Jacobian $\nabla R(W^k)^T$ is not computed or inverted exactly, this leads to slower convergence rates, but cheaper iterations, and gives rise to the larger class of "Newton type methods". An excellent overview of the field is given in [13]. But how are these ideas generalized to nonlinear optimization?

The NMPC and MHE problems as stated above are specially structured cases of a generic nonlinear program (NLP) that has the form

$$\underset{X}{\text{minimize}} \ F(X) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} G(X) & = & 0 \\ H(X) & \leq & 0 \end{array} \right. \tag{5}$$

Under mild assumptions, any locally optimal solution X^* of this problem has to satisfy the famous Karush-Kuhn-Tucker (KKT) conditions: there exist multiplier vectors λ^* and μ^* so that the following equations hold:

$$\nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) \ = \ 0 \tag{6a}$$

$$G(X^*) \ = \ 0 \tag{6b}$$

$$0 \ \geq \ H(X^*) \quad \perp \quad \mu^* \ \geq \ 0. \tag{6c}$$

Here we have used the definition of the Lagrange function

$$\mathcal{L}(X, \lambda, \mu) = F(X) + G(X)^T \lambda + H(X)^T \mu \tag{7}$$

and the symbol $\perp$ between the two vector valued inequalities in Eq. (6c) states that also the complementarity condition

$$H_i(X^*) \, \mu_i^* = 0, \quad i = 1, \ldots, n_H, \tag{8}$$

shall hold. All Newton type optimization methods try to find a point satisfying these conditions by using successive linearizations of the problem functions. Major differences exist, however, on how to treat the last condition (6c) that is due to the inequality constraints, and the two big families are Sequential Quadratic Programming (SQP) type methods and Interior Point (IP) methods.

3.1 Sequential Quadratic Programming

A first variant to iteratively solve the KKT system is to linearize all nonlinear functions appearing in Eqs. (6a)–(6c). It turns out that the resulting linear complementarity system can be interpreted as the KKT conditions of a quadratic program (QP)

$$\underset{X}{\text{minimize}} \ F_{\mathrm{QP}}^k(X) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} G(X^k) + \nabla G(X^k)^T (X - X^k) & = & 0 \\ H(X^k) + \nabla H(X^k)^T (X - X^k) & \leq & 0 \end{array} \right. \tag{9}$$

with objective function

$$F_{\mathrm{QP}}^k(X) = \nabla F(X^k)^T X + \frac{1}{2}(X - X^k)^T \nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)(X - X^k). \tag{10}$$

In the case that the so called Hessian matrix $\nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)$ is positive semidefinite, this QP is convex so that global solutions can be found reliably. This general approach to address the nonlinear optimization problem is called Sequential Quadratic Programming (SQP). Apart from the presented "exact Hessian" SQP variant presented above, several other – and much more widely used – SQP variants exist, that make use of inexact Hessian or Jacobian matrices.

3.1.1 Powell's Classical SQP Method

One of the most successfully used SQP variants is due to Powell [47]. It uses exact constraint Jacobians, but replaces the Hessian matrix $\nabla_X^2 \mathcal{L}(X^k, \lambda^k, \mu^k)$ by an approximation A_k. Each new Hessian approximation A_{k+1} is obtained from the previous approximation A_k by an update formula that uses the difference of the Lagrange gradients, $\gamma = \nabla_X \mathcal{L}(X^{k+1}, \lambda^{k+1}, \mu^{k+1}) - \nabla_X \mathcal{L}(X^k, \lambda^{k+1}, \mu^{k+1})$ and the step $\sigma = X^{k+1} - X^k$. Aim of these "Quasi-Newton" or "Variable-Metric" methods is to collect second order information in A_{k+1} by satisfying the secant equation $A_{k+1}\sigma = \gamma$. The most widely used update formula is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update $A_{k+1} = A_k + \gamma\gamma^T/(\gamma^T\sigma) - A_k\sigma\sigma^T A_k/(\sigma^T A_k \sigma)$, see e.g. [44]. Quasi-Newton methods can be shown to converge superlinearly under mild conditions, and had a tremendous impact in the field of nonlinear optimization. Successful implementations are the packages NPSOL and SNOPT for general NLPs [27], and MUSCOD-II [39] for optimal control. Note that in this paper we omit all discussion on the usually crucial issue of globalisation strategies, because these are less important in online optimization.

3.1.2 Constrained Gauss-Newton Method

Another particularly successful SQP variant – the Constrained (or Generalized) Gauss-Newton method – is also based on approximations of the Hessian. It is applicable when the objective function is a sum of squares:

$$F(X) = \frac{1}{2}\|R(X)\|_2^2. \tag{11}$$

In this case, the Hessian can be approximated by

$$A_k = \nabla R(X^k) \nabla R(X^k)^T \tag{12}$$

and the corresponding QP objective is easily seen to be

$$F_{\mathrm{QP}}^k(X) = \frac{1}{2}\|R(X^k) + \nabla R(X^k)^T(X - X^k)\|_2^2 \tag{13}$$

The constrained Gauss-Newton method has only linear convergence but often with a surprisingly fast contraction rate. The contraction rate is fast when the residual norm $\|R(X^*)\|$ is small or the problem functions R, G, H have small second derivatives. It has been developed and extensively investigated by Bock and coworkers, see e.g. [6, 53]. The constrained Gauss-Newton method is implemented in the packages PARFIT [6], FIXFIT [53], and also as one variant within MUSCOD-II [14, 39].

Remark on adjoint based SQP variants: Newton type SQP methods may not only use an approximation of the Hessian, but also of the constraint Jacobians. The most general formulation including inexact inequalities, which is originally due to [7] and was analysed in [61], uses approximations A_k, B_k, C_k of the matrices $\nabla_X^2 \mathcal{L}(\cdot), \nabla G(X^k), \nabla H(X^k)$, and a so called “modified gradient”

$$a_k = \nabla_X \mathcal{L}(X^k, \lambda^k, \mu^k) - B_k \lambda^k - C_k \mu^k \tag{14}$$

in the QP objective

$$F_{\mathrm{adjQP}}^k(X) = a_k^T X + \frac{1}{2}(X - X^k)^T A_k (X - X^k). \tag{15}$$

The following QP is solved in each iteration:

$$\underset{X}{\text{minimize}} \; F_{\mathrm{adjQP}}^k(X) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} G(X^k) + B_k^T(X - X^k) & = & 0 \\ H(X^k) + C_k^T(X - X^k) & \leq & 0. \end{array} \right. \tag{16}$$

It can be shown that using a modified gradient a_k allows to locally converge to solutions of the original nonlinear NLP even in the presence of inexact inequality constraint Jacobians [7, 20, 61]. A crucial ingredient of the adjoint based SQP scheme is the fact that the Lagrange gradient needed for a_k in (14) can be evaluated efficiently by adjoint based techniques or, equivalently, by the reverse mode of automatic differentiation [30]. Adjoint based SQP schemes are at the core of the multi-level real-time iterations described in Section 6.1. Even quasi Newton update schemes can be used in order to approximate the Jacobians [32].

3.2 Interior Point Methods

In contrast to SQP methods, an alternative way to address the solution of the KKT system is to replace the last nonsmooth KKT condition in Eq. (6c) by a smooth nonlinear approximation, with $\tau > 0$:

$$\begin{aligned} \nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) &= 0 && \text{(17a)} \\ G(X^*) &= 0 && \text{(17b)} \\ H_i(X^*)\, \mu_i^* &= \tau, \quad i = 1, \ldots, n_H. && \text{(17c)} \end{aligned}$$

This system is then solved with Newton's method. The obtained solution is not a solution to the original problem, but to the problem

$$\underset{X}{\text{minimize}} \; F(X) - \tau \sum_{i=1}^{n_H} \log(-H_i(X)) \quad \text{s.t.} \quad G(X) \; = \; 0. \tag{18}$$

Thus, the solution is in the interior of the set described by the inequality constraints, and closer to the true solution the smaller τ gets. The crucial feature of the family of “interior point methods” is the fact that, once a solution for a given τ is found, the parameter τ can be reduced by a constant factor without jeopardising convergence of Newton's method. After only a limited number of Newton iterations a quite accurate solution of the original NLP is obtained. We refer to the excellent textbooks [10, 63] for details. A widely used implementation of nonlinear Interior Point methods is the open source code IPOPT [60].

Remark on the structure of the linear subproblems: It is interesting to note that the linearization of the smoothed KKT system (17a)-(17c) is a linear system that is equivalent – after elimination of the variable μ^{k+1} – to the KKT conditions of an equality constrained quadratic program. It is important to remark that most structure exploiting features of SQP type methods also have an equivalent in IP methods, like globalisation strategies, use of adjoints, structure preserving linear algebra, etc., and we will mention them when applicable.

Remark on Ohtsuka's inequality treatment: An interesting treatment of inequality constraints that is similar to interior point methods was proposed and successfully used in the context of NMPC by Ohtsuka [45]. He proposes to approximate the inequality constrained NLP (5) by a formulation

$$\underset{X,Y}{\text{minimize}} \;\; F(X) - \tau \sum_{i=1}^{n_H} Y_i \quad \text{s.t.} \; \left\{ \begin{array}{rcl} G(X) & = & 0 \\ H_i(X) + Y_i^2 & = & 0, \; i = 1, \ldots, n_H. \end{array} \right. \tag{19}$$

which is equivalent to

$$\underset{X}{\text{minimize}} \;\; F(X) - \tau \sum_{i=1}^{n_H} \sqrt{-H_i(X)} \quad \text{s.t.} \quad G(X) \;\; = \;\; 0. \tag{20}$$

This barrier is not self-concordant and does not connect easily to the dualtiy theory of interior-point methods, but we will nevertheless call this approach a variant of IP methods in this paper.

4 Numerical Optimal Control

When Newton type optimization strategies are applied to the optimal control problem (2a)-(2f), the first question is, if a simultaneous or a sequential approach is used. In the case of a sequential approach, where all state variables x, z are eliminated and the optimization routine only sees the control variables u, the specific optimal control problem structure plays a minor role. Thus, often an off-the-shelf code for nonlinear optimization can be used. This makes practical implementation very easy and is a major reason why the sequential approach is used by many practitioners. It is in strong contrast to the simultaneous approach, that addresses the optimal control problem (2a)-(2f) in the full variable space x, z, u, and thus allows – and necessitates – to exploit the specific problem structure. In all Newton type optimization routines there are two crucial and often costly computational steps, namely (a) Derivative Computation and (b) Solution of the Quadratic Subproblems. In both areas, specific structures can be exploited. In this paper we will focus on (b), the solution of the QPs, but add suitable remarks on how to treat derivative computations when necessary.

4.1 The Linearized Optimal Control Problem

Let us regard the linearization of the optimal control problem (2a)-(2f) within an SQP method, which is a structured QP. It turns out that due to the dynamic system structure the Hessian of the Lagrangian function has the same separable structure as the Hessian of the original objective function (2a), so that the quadratic QP objective is still representable as a sum of linear-quadratic stage costs, which was first observed by Bock and Plitt [9]. Thus, the QP subproblem has the following form, where we left out the SQP iteration index k for notational simplicity, and where the summands of the objective each are linear-quadratic.

$$\begin{aligned}
\underset{x,\,z,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_{\mathrm{QP},i}(x_i, z_i, u_i) \quad + \quad E_{\mathrm{QP}}\,(x_N) && && \text{(21a)}\\
\text{subject to} \quad x_0 - \bar{x}_0 &= 0, && && \text{(21b)}\\
x_{i+1} - f'_i - F^x_i x_i - F^z_i z_i - F^u_i u_i &= 0, && i = 0, \ldots, N-1, && \text{(21c)}\\
g'_i + G^x_i x_i + G^z_i z_i + G^u_i u_i &= 0, && i = 0, \ldots, N-1, && \text{(21d)}\\
h'_i + H^x_i x_i + H^z_i z_i + H^u_i u_i &\leq 0, && i = 0, \ldots, N-1, && \text{(21e)}\\
r' + R x_N &\leq 0. && && \text{(21f)}
\end{aligned}$$

When the linear algebra within the QP solution is concerned, the dynamic system structure can be exploited in different ways.

Remark on high rank Hessian updates: The fact that the Hessian matrix of the optimal control problem is block diagonal does not only allow to write down the objective (21a) in a separable form and exploit this sparsity in the linear algebra; when quasi Newton Hessian update methods are used, it also allows to perform "partitioned variable metric" or "high rank updates" of the Hessian, by updating all Hessian blocks separately [9, 31].

4.2 Elimination of Algebraic Variables

We consider now several algorithmic building blocks helping to solve the QP problem (21a)-(21f). Let us first regard Eq. (21d). Due to our assumptions in the problem statement of (2a)-(2f), we know that the Jacobian matrix G^z_i is invertible. Thus, Eq. (21d) can directly be inverted by a factorization of the matrix G^z_i, yielding an explicit expression for z_i:

$$z_i = -\left(G^z_i\right)^{-1} \left[g'_i + G^x_i x_i + G^u_i u_i\right] \tag{22}$$

Note that the matrix G^z_i is often sparse and might best be factorized by a direct sparse solver. Once this factorization is performed, it is possible to reduce problem (21a)-(21f) to a smaller scale QP in the variables x and u only, which has the following form:

$$\begin{aligned}
\underset{x,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_{\mathrm{redQP},i}(x_i, u_i) \quad + \quad E_{\mathrm{QP}}\,(x_N) && && \text{(23a)}\\
\text{subject to} \quad x_0 - \bar{x}_0 &= 0, && && \text{(23b)}\\
x_{i+1} - c_i - A_i x_i - B_i u_i &= 0, && i = 0, \ldots, N-1, && \text{(23c)}\\
\bar{h}_i + \bar{H}^x_i x_i + \bar{H}^u_i u_i &\leq 0, && i = 0, \ldots, N-1, && \text{(23d)}\\
r' + R x_N &\leq 0. && && \text{(23e)}
\end{aligned}$$

This partially reduced QP can be post-processed either by a condensing or a band structure exploiting strategy.

Remark on Leineweber's Partially Reduced SQP Method: In the context of a direct multiple shooting method, the evaluation of the Jacobian matrices F_i^x, F_i^z, F_i^u in (21c) is a very CPU intensive step. Given the fact that finally only the reduced matrices A_i and B_i are needed in the reduced QP, Leineweber [39] proposed a partially reduced SQP method that never computes the matrices needed in the QP (21a)-(21f). Instead, it first performs the sparse matrix factorization of G_i^z needed for elimination of the variables z_i via Eq. (22), and then it computes the matrices A_i and B_i directly as directional derivatives of $f_i(x_i, z_i, u_i)$:

$$A_i = \frac{\partial f_i(\cdot)}{\partial(x,z,u)} \begin{bmatrix} \mathbb{I} \\ -\left(G_i^z\right)^{-1} G_i^x \\ 0 \end{bmatrix} \quad \text{and} \quad B_i = \frac{\partial f_i(\cdot)}{\partial(x,z,u)} \begin{bmatrix} 0 \\ -\left(G_i^z\right)^{-1} G_i^u \\ \mathbb{I} \end{bmatrix}. \tag{24}$$

This allows to reduce the computational burden significantly in case of many algebraic variables z_i and expensive evaluation of f_i.

4.3 Condensing

In order to see how the variable space of a QP can be reduced further in a very simple way, let us recall that it was possible to reduce the large scale NLP via a nonlinear system simulation in the sequential approach. The basic idea of the "condensing" approach that was first proposed by Bock and Plitt [9] is to use the same fact, but apply it only to the linearized dynamic system. For this aim let us note that Eqs. (23b) and (23c) describe nothing else than a linear time varying discrete time system, and that for fixed u the values for x can easily be obtained by a forward simulation of the linear dynamics. Hence, the vector x is completely determined by the vector u and the given initial state $\bar{x}_0$. Therefore, the states can be eliminated from the linearized problem resulting in a smaller, but dense quadratic program of the form

$$\underset{u}{\text{minimize}} \quad f_{\text{condQP},i}(\bar{x}_0, u) \tag{25a}$$

$$\text{subject to} \quad \bar{r} + \bar{R}^{x_0}\bar{x}_0 + \bar{R}^u u \quad \leq \quad 0. \tag{25b}$$

Here, the inequalities (25b) contain both types of the original inequalities, (23d) and (23e), in "condensed" form. If the dimension of the vector $u = (u_0^T, u_1^T, \ldots, u_{N-1}^T)^T$ is not too large, this QP can be solved fast using dense general purpose QP solvers. By doing so, the cost of solving one QP subproblem grows with $O(N^3 n_u^3)$, i.e. cubically with the horizon length N.

Remark on Schlöder's Reduction Trick: In the context of direct multiple shooting methods, the generation of the matrices A_i and B_i in (23c) is expensive if the differential state dimension n_x is large. It needs $O(N(n_x + n_u))$ stage wise directional derivatives. We might instead, as in Leineweber's partially reduced SQP method, directly compute the quantities needed in the objective and the constraints of the condensed QP (25a)-(25b). This idea was first proposed by Schlöder [53], in the context of the Generalized Gauss-Newton method. The method is implemented in the codes FIXFIT [53] and MSOPT [52]. It is only advantageous for large state but small control dimensions ($n_u \ll n_x$), and it exploits the fact that the initial value x_0 is fixed in the NMPC problem. Thus, it offers no advantages in the MHE problem where the initial value is free for optimization.

4.4 Band Structure Exploiting Riccati Based Solutions

Instead of condensing the linearized problem, one can opt to keep the constraints (23b) and (23c) and the variables x as unknowns in the QP. To sketch the idea, let us regard a QP without the inequalities (23d) and (23e). The KKT conditions of this equality constrained QP (23a)-(23c) in the primal and dual variables $w = (\lambda_0^T, x_0^T, u_0^T, \lambda_1^T, x_1^T, u_1^T, \ldots, \lambda_N^T, x_N^T)^T$ are a symmetric linear system $Mw = b$ with KKT matrix

$$M = \begin{bmatrix} & \mathbb{I} & & & \\ \mathbb{I} & Q_0 & S_0 & -A_0^T & \\ & S_0^T & R_0 & -B_0^T & \\ & -A_0 & -B_0 & \ddots & \mathbb{I} \\ & & & \mathbb{I} & Q_N \end{bmatrix} \tag{26}$$

The almost block diagonal structure of this linear system allows it to be efficiently factorized by a (discrete time) Riccati recursion. This was shown for optimal control problems within an active set framework in [29] and within an interior point framework in [56]. For linear model predictive control, Riccati based solutions are described in [35, 49]. The cost of this factorization, which is usually dominating the cost for solving the QP, is $O(N(n_x + n_u)^3)$. The cost grows only linearly with the horizon length N, in contrast to condensing with its cubic growth $O(N^3 n_u^3)$. This makes the Riccati method favorable for larger horizon lengths N and when $n_x \approx n_u$. A Riccati based factorization is particularly advantageous for the MHE problem where the dimension of the "controls" w is typically as big as the state dimension.

Remark on direct or iterative sparse solvers: Note that it is not necessary to use a Riccati based solution in order to obtain the complexity $O(N(n_x + n_u)^3)$, but that this can also be achieved by using a direct sparse solver, as e.g. done in the general purpose and open-source NLP package IPOPT [60]. Also, iterative linear solvers might be used.

Remark on Tenny's Feasibility Perturbed SQP Method: An interesting method for optimal control and NMPC was proposed by Tenny, Wright and Rawlings [58], who regard a simultaneous formulation within an SQP type framework, but "perturb" the result of each SQP iteration in order to make the state trajectory consistent, i.e., they close all nonlinear continuity conditions (2c). This can be done by a simple "open loop" forward simulation of the system given the new controls, or by more complex "closed loop" simulations. In the open loop variant, this is nearly a sequential approach and performs, if exact Hessians are used, even the same SQP iterations. But it differs in one important aspect: it allows to exploit the same sparsity structure as a simultaneous approach, e.g. full space derivative computation, Riccati based linear algebra, or high rank updates for the block structured Hessian [9]. This makes it an interesting cross-over between typical features of sequential and simultaneous methods.

4.5 A Classification of Optimal Control Methods

It is an interesting exercise to try to classify Newton type optimal control algorithms. Let us have a look at how nonlinear optimal control algorithms perform their major algorithmic components, each of which comes in several variants:

(a) Treatment of Inequalities: Nonlinear IP vs. SQP
(b) Nonlinear Iterations: Simultaneous vs. Sequential
(c) Derivative Computations: Full vs. Reduced
(d) Linear Algebra: Banded vs. Condensing

In the last two of these categories, we observe that the first variants each exploit the specific structures of the simultaneous approach, while the second variant reduces the variable space to the one of the sequential approach. Note that reduced derivatives imply condensed linear algebra, so the combination [Reduced,Banded] is excluded. In the first category, we might sometimes distinguish two variants of SQP methods, depending on how they solve their underlying QP problems, via active set QP solvers (SQP-AS) or via interior point methods (SQP-IP).

Based on these four categories, each with two alternatives, and one combination excluded, we obtain seven possible combinations. In these categories, the classical single shooting method [50] could be classified as [SQP,Sequential,Reduced] or as [SQP,Sequential,Full,Condensing] because some variants compute directly the reduced derivatives $\bar{R}^u$ in (25b), while others compute first the matrices A_i and B_i in (23c) and condense then. Tenny's feasibility perturbed SQP method [58] could be classified as [SQP,Sequential,Full,Banded], and Bock's multiple shooting [9] as well as the classical reduced SQP collocation methods [2, 3, 59] as [SQP,Simultaneous,Full,Condensing]. The band structure exploiting SQP variants from Steinbach [56] and Franke [26] are classified as [SQP-IP,Simultaneous,Full,Banded], while the widely used interior point direct collocation method in conjunction with IPOPT by Biegler and Wächter [60] as [IP,Simultaneous,Full,Banded]. The reduced Gauss-Newton method of Schlöder [53] would here be classified as [SQP,Simultaneous,Reduced].

5 Online Initialization and NLP Sensitivities

For exploiting the fact that NMPC requires the solution of a whole sequence of "neighboring" NLPs and not just a number of stand-alone problems, we have first the possibility to *initialize* subsequent problems efficiently based on previous information. In this section we introduce several concepts for such initializations, in particular the important concept of NLP sensitivities. On the other hand, in Section 6 we will give an overview of specially tailored *online algorithms* for approximately solving each NLP, that deliver on purpose inaccurate solutions and postpone calculations from one problem to the next.

5.1 Shift Initialization

A first and obvious way to transfer solution information from one solved NMPC problem to the initialization of the next one is based on the principle of optimality of subarcs, also called the dynamic programming principle. It states the following: Let us assume we have computed an optimal solution $(x_0^*, z_0^*, u_0^*, x_1^*, z_1^*, u_1^*, \dots, x_N^*)$ of the NMPC problem (2a)-(2f) starting with initial value $\bar{x}_0$. If we regard a shortened NMPC problem without the first interval, which starts with the initial value $\bar{x}_1$ chosen to be x_1^*, then for this shortened problem the vector $(x_1^*, z_1^*, u_1^*, \dots, x_N^*)$ is the optimal solution.

Based on the expectation that the measured or observed true initial value for the shortened NMPC problem differs not much from x_1^* – i.e. we believe our prediction model and expect no disturbances – this "shrinking" horizon initialization is canonical, and it is used in MPC of batch or finite time processes, see e.g. [15, 34].

However, in the case of moving (finite) horizon problems, the horizon is not only shortened by removing the first interval, but also prolonged at the end by appending a new terminal interval – i.e. the horizon is moved forward in time. In the moving horizon case, the principle of optimality is thus not strictly applicable, and we have to think about how to initialize the appended new variables z_N, u_N, x_{N+1}. Often, they are obtained by setting $u_N := u_{N-1}$ or setting u_N as the steady state control. The states z_N and x_{N+1} are then obtained by forward simulation. This transformation of the variables from one problem to the next is called "shift initialization". It is not as canonical as the "shrinking horizon" case, because the shifted solution is not optimal for the new undisturbed problem. However, in the case of long horizon lengths N we can expect the shifted solution to be a good initial guess for the new solution. Moreover, for most NMPC schemes with stability guarantee (for an overview see e.g. [42]) there exists a canonical choice of u_N that implies feasibility (but not optimality) of the shifted solution for the new, undisturbed problem. The shift initialization is very often used e.g. in [4, 19, 41, 43].

A comparison of shifted vs. non-shifted initializations was performed in [8] with the result that for autonomous NMPC problems that shall regulate a system to steady state, there is usually no advantage of a shift initialization compared to the "primitive" warm start initialization that leaves the variables at the previous solution. In the extreme case of short horizon lengths, it turns out to be even advantageous NOT to shift the previous solution, as subsequent solutions are less dominated by the initial values than by the terminal conditions. On the other hand, shift initialization are a crucial prerequisite in periodic tracking applications [19] and whenever the system or cost function are not autonomous.

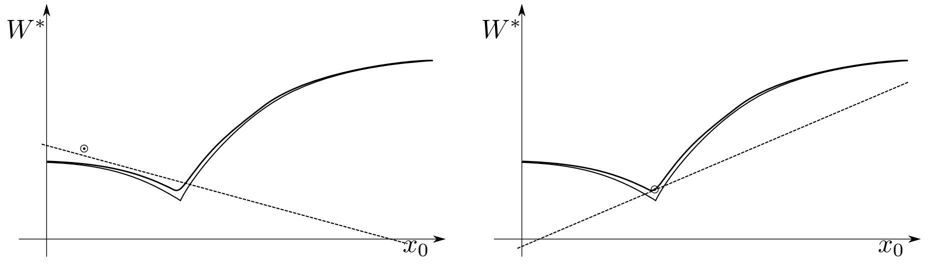
5.2 Parametric Sensitivities

In the shift initialization discussed above we did assume that the new initial value corresponds to the model prediction. This is of course never the case, because exactly the fact that the initial state is subject to disturbances motivates the use of MPC. By far the most important change from one optimization problem to the next one are thus the unpredictable changes in the initial value. Is there anything we can do about this in the initialization of a new problem?

It turns out that we can, if we use the concept of parametric NLP sensitivities to construct a new initial guess. To illustrate the idea, let us first regard the parametric root finding problem $R(\bar{x}_0, W) = 0$ that results from the necessary optimality conditions of an IP method, i.e. the system (17a)–(17c) in variables $W = (X, \lambda, \mu)$. In the NMPC context, this system depends on the uncertain initial value $\bar{x}_0$. We denote the solution manifold by $W^*(\bar{x}_0)$. When we know the solution $W = W^*(\bar{x}_0)$ for a previous initial value $\bar{x}_0$ and want to compute the solution for a new initial value $\bar{x}_0'$, then a good solution predictor for $W^*(\bar{x}_0')$ is provided by $W' = W + \frac{dW^*}{d\bar{x}_0}(\bar{x}_0)(\bar{x}_0' - \bar{x}_0)$ where $\frac{dW^*}{d\bar{x}_0}(\bar{x}_0)$ is given by the implicit function theorem. An important practical observation is that an approximate tangential predictor can also be obtained when it is computed at a point W that does not exactly lie on the solution manifold. This more general predictor is given by the formula

$$W' = W - \left(\frac{\partial R}{\partial W}(\bar{x}_0, W)\right)^{-1} \left[\frac{\partial R}{\partial \bar{x}_0}(\bar{x}_0, W) \left(\bar{x}_0' - \bar{x}_0\right) + R(\bar{x}_0, W)\right]. \qquad (27)$$

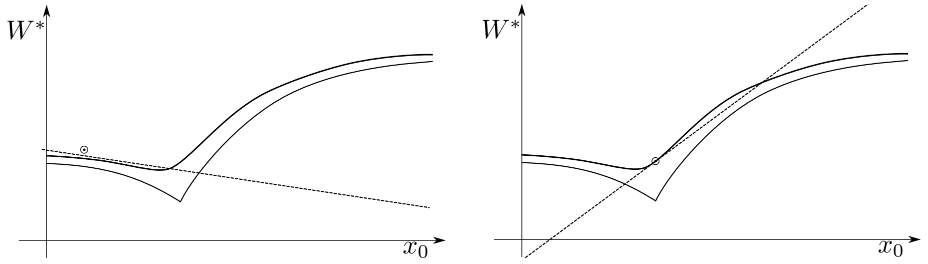
This fact, that is illustrated in Fig. 1(a), and that leads to a combination of a predictor and corrector step in one linear system, is exploited in the continuation method by Ohtsuka [45] and in a generalized form in the real-time iteration scheme [16], both described below. When $R(\bar{x}_0, W) = 0$ the formula simplifies to the tangential predictor of the implicit function theorem, which is e.g. employed in the advanced step controller[64].



(a) Linearizing at approximate solution

(b) Linearizing at active set change

Figure 1: Tangential predictors for interior point method using a small τ.



(a) Linearizing at approximate solution

(b) Linearizing at active set change

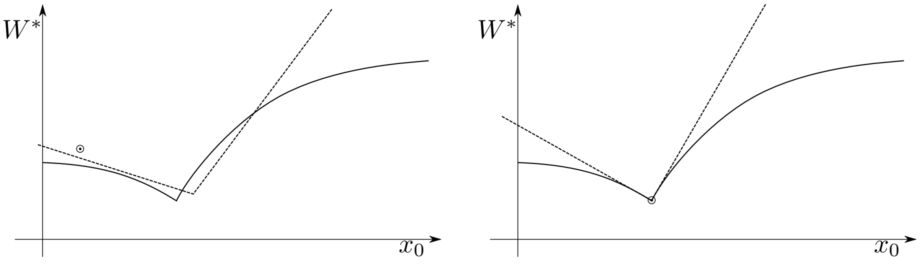
Figure 2: Tangential predictors for interior point method using a larger τ.

Remark on IP Sensitivities at Active Set Changes: Unfortunately, the interior point solution manifold is strongly nonlinear at points where the active set changes, and the tangential predictor is not a good approximation when we linearize at such points, as visualized in Fig. 1(b). One remedy would be to increase the path parameter τ, which decreases the nonlinearity, but comes at the expense of generally less accurate IP solutions. This is illustrated in Figs. 2(a) and 2(b) for the same two linearization points as before. In Fig. 2(b) we see that the tangent is approximating the IP solution manifold well in a larger area around the linearization point, but that the IP solution itself is more distant to the true NLP solution.

5.3 Generalized Tangential Predictors via SQP Methods

In fact, the true NLP solution is not determined by a smooth root finding problem (17a)–(17c), but by the KKT conditions (6a)–(6c). It is a well-known fact from parametric optimization, cf. [33], that the solution manifold has smooth parts when the active set does not change (and bifurcations are excluded), but that non-differentiable points occur whenever the active set changes. Is there anything we can do in order to “jump” over these non-smooth points in a way that delivers better predictors than the IP predictors discussed before?

In fact, at points with weakly active constraints, we have to regard *directional* derivatives of the solution manifold, or “generalized tangential predictors”. These can be computed by suitable quadratic programs [33, Thm 3.3.4] and are visualized in Fig. 3(b). The theoretical results can be made a practical algorithm by the following procedure proposed in [14]: first, we have to make sure that the parameter $\bar{x}_0$ enters the NLP linearly, which is automatically the case for simultaneous optimal control formulations, cf. Eq. (2b). Second, we address the problem with an exact Hessian SQP method. Third, we just take our current solution guess W for a problem $\bar{x}_0$, and then solve the QP subproblem (21a)–(21f) for the new parameter value $\bar{x}'_0$, but initialized at W. It can be shown [14, Thm. 3.6] that this “initial value embedding” procedure delivers exactly the generalized tangential predictor when started at a solution $W = W^*(\bar{x}_0)$, as in Fig. 3(b). It is important to remark that the predictor becomes approximately tangential when (a) we do not start on the solution manifold, see Fig. 3(a), or (b) we do not use an exact Hessian or Jacobian matrix or (c) we do not evaluate the Lagrange gradient or constraint residuals



(a) Linearizing at approximate solution

(b) Linearizing at active set change

Figure 3: Generalized tangential predictors for SQP method.

exactly.

6 Online Algorithms

In NMPC and MHE we would dream to have the solution to a new optimal control problem instantly, which is impossible due to computational delays. Several ideas help to deal with this issue, which we discuss before explaining in detail several of the existing online algorithms. We focus on the NMPC problem but remark that all ideas are also transferable to the MHE problem, which we sometimes mention explicitly.

Offline precomputations: As consecutive NMPC problems are similar, some computations can be done once and for all before the controller starts. In the extreme case, this leads to an explict precomputation of the NMPC control law that has raised much interest in the linear MPC community [1], or a solution of the Hamilton-Jacobi-Bellman Equation, both of which are prohibitive for state and parameter dimensions above ten. But also when online optimization is used, code optimization for the model routines is often essential, and it is in some cases even possible to precompute and factorize Hessians or even Jacobians in Newton type Optimization routines, in particular in the case of neighboring feedback control along reference trajectories [12, 37].

Delay compensation by prediction: When we know how long our computations for solving an NMPC problem will take, it is a good idea *not* to address a problem starting at the current state but to simulate at which state the system will be when we will have solved the problem. This can be done using the NMPC system model and the open-loop control inputs that we will apply in the meantime [24]. This feature is used in many practical NMPC schemes with non-negligible computation time.

Division into preparation and feedback phase: A third ingredient of several NMPC algorithms is to divide the computations in each sampling time into a preparation phase and a feedback phase [16]. The more CPU intensive preparation phase (a) is performed with an old predicted state $\bar{x}_0$ before the new state estimate, say $\bar{x}_0'$, is available, while the feedback phase (b) then delivers quickly an *approximate* solution to the optimization problem for $\bar{x}_0'$. Often, this approximation is based on one of the tangential predictors discussed in the previous section.

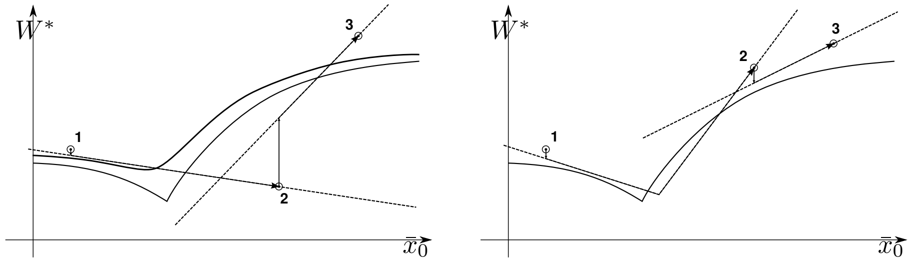
Iterating while the problem changes: A fourth important ingredient of some NMPC algorithms is the idea to work on the optimization problem while it changes, i.e., to never iterate the Newton type procedure to convergence for an NMPC problem getting older and older during the iterations, but to rather work with the most current information in each new iteration. This idea is used in [16, 41, 45].

6.1 A Survey of Online Optimization for NMPC

We will in the following review several of the approaches suggested in the literature, in a personal and surely incomplete selection, and try to classify them along the algorithmic lines discussed in this paper.

The Newton-Type Controller of Li and Biegler [40]: This was probably one of the first true online algorithms for NMPC. It is based on a sequential optimal control formulation, thus it iterated in the space of controls $u = (u_0, u_1, \ldots, u_{N-1})$ only. It uses an SQP type procedure with Gauss-Newton Hessian and line search, and in each sampling time, only one SQP iteration is performed. The transition from one problem to the next uses a shift of the controls $u^{\text{new}} = (u_1, \ldots, u_{N-1}, u_N^{\text{new}})$. The result of each SQP iterate is used to give an approximate feedback to the plant. As a sequential scheme without tangential predictor, it seems to have had sometimes problems with nonlinear convergence, though closed-loop stability was proven for open-loop stable processes [41], and in principle, the theoretical NMPC stability analysis from [18] is applicable.

The Continuation/GMRES Method of Ohtsuka [45]: Similar to the Newton-Type controller, the Continuation/GMRES method performs only one Newton type iteration in each sampling time, and is based on a sequential formulation. It is different in that it is based on an IP treatment of the inequalities with fixed path parameter $\tau > 0$, see Section 3.2, that it uses an exact Hessian, and that it uses the iterative GMRES method for linear system solution in each Newton step. Most important, it makes no use of a shift, but instead use of the tangential predictor described in Eq. (27). This features seems to allow it to follow the nonlinear IP solution manifold well – which is strongly curved at active set changes. For a visualization, see Fig. 4(a). In each sampling time, only one linear system is built and solved by the GMRES method, leading to a predictor-corrector pathfollowing method. The closed-loop stability of the method is in principle covered by the stability analysis for the real-time iterations without shift given in [17]. A variant of the method is given in [54], which uses a simultanous approach and condensing and leads to improved accuracy and lower computational cost in each Newton type iteration.



(a) Ohtsuka's C/GMRES method

(b) Real-Time Iteration scheme

Figure 4: Subsequent solution approximations.

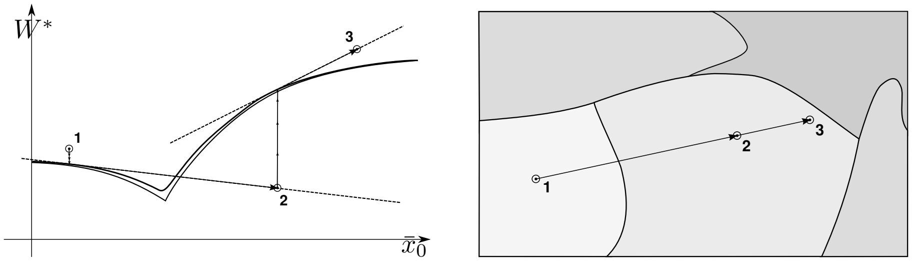
The Real-Time Iteration Scheme [16]: Similar to the Newton-Type controller, the real-time iteration scheme presented in [14, 16] performs one SQP type iteration with Gauss-Newton Hessian per sampling time. However, it employs a simultaneous NLP parameterization, Bock's direct multiple shooting method, with full derivatives and condensing. Moreover, it uses the generalized tangential predictor of the "initial value embedding" discussed in Section 5.3 to correct for the mismatch between the expected state $\bar{x}_0$ and the actual state $\bar{x}_0'$. In contrast to the C/GMRES method, where the predictor is based on one linear system solve from Eq. (27), here an inequality constrained QP is solved. The computations in each iteration are divided into a long "preparation phase" (a), in which the system linearization, elimination of algebraic variables and condensing are performed, as described in Sections 4.1–4.3, and a much shorter "feedback phase" (b). The feedback phase solves just one condensed QP (25a)–(25b), more precisely, an "embedded" variant of it, where the expected state $\bar{x}_0$ is replaced by the actual one, $\bar{x}_0'$. Depending on the application, the feedback phase can be several orders of magnitude shorter than the feedback phase. The iterates of the scheme are visualized in Fig. 4(b). The same iterates are obtained with a variant of the scheme that uses Schlöder's trick for reducing the costs of the preparation phase in the case of large state dimensions [51]. Note that only one system linearization and one QP solution are performed in each sampling time, and that the QP corresponds to a linear MPC feedback along a time varying trajectory. In contrast to IP formulations, the real-time iteration scheme gives priority to active set changes and works well when the active set changes faster than the linearized system matrices. In the limiting case of a linear system model it gives the same feedback as linear MPC. Error bounds and closed loop stability of the scheme have been established for shrinking horizon problems in [15] and for NMPC with shifted and non-shifted initializations in [18] and [17].

Advanced Step Controller by Zavala and Biegler [64]: In order to avoid the convergence issues of predictor-corrector pathfollowing methods, in the "advanced step controller" of Zavala and Biegler a more conservative choice is made: in each sampling time, a complete Newton type IP procedure is iterated to convergence (with $\tau \to 0$). In this respect, it is just like offline optimal control – IP, simultaneous, full derivatives with exact Hessian, structure exploiting linear algebra. However, two features qualify it as an online algorithm: first, it takes computational delay into account by solving an "advanced" problem with the expected state $\bar{x}_0$ as initial value – similar as in the real-time iterations with shift – and (b), it applies the obtained solution not directly, but computes first the tangential predictor which is correcting for the differences between expected state $\bar{x}_0$ and the actual state $\bar{x}_0'$, as described in Eq. (27) with $R(W, \bar{x}_0) = 0$. Note that in contrast to the other online algorithms, several Newton iterations are performed in part (a) of each sampling time, the "preparation phase". The tangential predictor (b) is computed in the "feedback phase" by only one linear system solve based on the last Newton iteration's matrix factorization. As in the C/GMRES method, the IP predictor cannot "jump over" active set changes as easily as the SQP based predictor of the real-time iteration. Roughly speaking, the advanced step controller gives lower priority to sudden active set changes than to system nonlinearity. As the advanced step controller solves each expected problem exactly, classical NMPC stability theory [42] can relatively easily be extended to this scheme [64].

Multi-Level Real-Time Iterations [7]: While the advanced step controller deviates from the other online NMPC schemes in that it performs many Newton iterations per sampling time, the opposite choice is made in the multi-level real-time iterations presented in [7], where even cheaper calculations are performed in each sampling time than one Newton step usually requires. At the lowest level (A), only one condensed QP (25a)–(25b) is solved, for the most current initial value $\bar{x}_0$. This provides a form of linear MPC at the base level, taking at least active set changes into account with a very high sampling frequency. On the next two intermediate levels, that are performed less often than every sampling time, only the nonlinear constraint residuals are evaluated (B), allowing for feasibility improvement, cf. also [12], or the Lagrange gradient is evaluated (C), allowing for optimality improvement, based on the adjoint based SQP presented in Section 3.1.2. Note that in all three levels A, B, and C mentioned so far, no new QP matrices are computed and that even system factorizations can be reused again and again. Level C iterations are still considerably cheaper than one full SQP iteration [61], but also for them optimality and NMPC closed-loop stability can be guaranteed by the results in [17] – as long as the system matrices are accurate enough to guarantee Newton type contraction. Only when this is not the case anymore, an iteration on the highest level, D, has to be performed, which includes a full system linearization and is as costly as a usual Newton type iteration.

Remark on Critical Regions and Online Active Set Strategies: It is interesting to have a look at the parameter space $\bar{x}_0$ visualized in Fig.5(b). The picture shows the "critical regions" on each of which the active set in the solution is stable. It also shows three consecutive problems on a line that correspond to the scenario used in Figures 4(a), 4(b), and 5(a). Between problem 1 and 2 there is one active set change, while problems 2 and 3 have the same active set, i.e., are in the same critical region. The C/GMRES method and Advanced Step Controller exploit the smoothness on each critical region in order to obtain the conventional Newton predictor that, however, looses validity when a region boundary is crossed. The real-time iteration basically "linearizes" the critical regions which then become polytopic, by using the more accurate, but also more expensive QP predictor.

As the QP cost can become non-negligible for fast MPC applications, a so-called online active set strategy was proposed in [23]. This strategy goes on a straight line in the space of linearized regions from the old to the new



(a) Solutions of Advanced Step Controller (b) Critical regions of a parametric NLP

Figure 5: Subsequent solution approximations (left), and critical regions (right).

QP problem. As long as one stays within one critical region, the QP solution depends affinely on $\bar{x}_0$ – exactly as the conventional Newton predictor. Only if the homotopy crosses boundaries of critical regions, the active set is updated accordingly. The online active set strategy is available in the open-source QP package qpOASES [22], and is particularly suitable in combination with real-time iterations of level A, B, and C, where the QP matrices do not change, see [62].

Remark on Online MHE Algorithms: Many algorithmic NMPC ideas have been generalized to MHE problems. For example, a Newton-type control framework was used for MHE in [43], the C/GMRES method in [55], cf. also [46], the real-time iteration in [21] and [38], and the advanced step framework in [65]. A somewhat interesting online MHE approach related to the Newton-type control framework was presented in [36], which uses *backwards* single shooting making it not suitable for stiff systems. Other numerical MHE schemes were presented in [35] and [57].

7 Conclusions

In this paper we have tried to give a self-contained overview of Newton type methods for online solution of nonlinear optimal control problems. We first reviewed several categories in which offline algorithms differ, such as simultanous vs. sequential approaches, Interior Point (IP) vs. Sequential Quadratic Programming (SQP) methods, band structure exploiting linear algebra vs. condensing, and different ways to compute the derivatives needed in Newton type iterations. We then categorized several offline approaches along these lines. The second part started by a discussion of online initializations. We stressed the importance of sensitivity results from parametric optimization, which in SQP type frameworks even allow to obtain cheaply a solution predictor across active set changes. We then classified many proposed real-time optimization approaches from the literature into the developed categories, starting with the ”Newton-type controller” [40] and the related ”continuation method” [45], both based on sequential approaches, and then went over to the ”real-time iteration scheme” [16], a simultaneous approach characterized by an SQP type solution predictor and iterations that perform only one system linearization at each sampling time. We also discussed the recently proposed simultaneous ”advanced step controller” [64] and ”multi-level real-time iterations” [7], as well as fast online QP solutions [23].

Acknowledgements M. Diehl is a professor at the Katholieke Universiteit Leuven, H. J. Ferreau and N. Haverbeke are research assistants there. Research supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering(OPTEC), GOA AMBioRICS, IOF-SCORES4CHEM, PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0302.07, G.0320.08, G.0558.08, G.0557.08, research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite; Helmholtz: viCERP; EU: ERNSI,HD-MPC; Contracts: AMINAL; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011).

References

[1] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[2] J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming.* SIAM, Philadelphia, 2001.

[3] L.T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering*, 8:243–248, 1984.

[4] L.T. Biegler and J.B Rawlings. Optimization approaches to nonlinear model predictive control. In W.H. Ray and Y. Arkun, editors, *Proc. 4th International Conference on Chemical Process Control - CPC IV*, pages 543–571. AIChE, CACHE, 1991.

[5] T. Binder, L. Blank, H.G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J.P. Schlöder, and O.v. Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S.O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 295–340. Springer, 2001.

[6] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften.* Universität Bonn, Bonn, 1987.

[7] H.G. Bock, M. Diehl, E.A. Kostina, and J.P. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time and Online PDE-Constrained Optimization*, pages 3 – 22. SIAM, 2007.

[8] H.G. Bock, M. Diehl, D.B. Leineweber, and J.P. Schlöder. Efficient direct multiple shooting in nonlinear model predictive control. In F. Keil, W. Mackens, H. Voß, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 218–227, Berlin, 1999. Springer.

[9] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization.* University Press, Cambridge, 2004.

[11] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control.* Wiley, New York, 1975.

[12] C. Büskens and H. Maurer. Sqp-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, 120:85–108, 2000.

[13] P. Deuflhard. *Newton Methods for Nonlinear Problems.* Springer, New York, 2004.

[14] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*, volume 920 of *Fortschr.-Ber. VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik.* VDI Verlag, Düsseldorf, 2002. Download also at: http://www.ub.uni-heidelberg.de/archiv/1659/.

[15] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.

[16] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Proc. Contr.*, 12(4):577–585, 2002.

[17] M. Diehl, R. Findeisen, and F. Allgöwer. A stabilizing real-time implementation of nonlinear model predictive control. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time and Online PDE-Constrained Optimization*, pages 23–52. SIAM, 2007.

[18] M. Diehl, R. Findeisen, F. Allgöwer, H.G. Bock, and J.P. Schlöder. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proc.-Control Theory Appl.*, 152(3):296–308, 2005.

[19] M. Diehl, L. Magni, and G. De Nicolao. Online NMPC of unstable periodic systems using approximate infinite horizon closed loop costing. *Annual Reviews in Control*, 28:37–45, 2004.

[20] M. Diehl, A. Walther, H. G. Bock, and E. Kostina. An adjoint-based SQP algorithm with quasi-newton jacobian updates for inequality constrained optimization. Technical Report Preprint MATH-WR-02-2005, TU Dresden, 2005.

[21] Moritz Diehl, P. Kühl, H.G Bock, and J.P. Schlöder. Schnelle algorithmen für die zustands- und parameterschätzung auf bewegten horizonten. *Automatisierungstechnik*, 54(12):602–613, 2006.

[22] H.J. Ferreau. *qpOASES User's Manual*, 2007–2008. http://homes.esat.kuleuven.be/~optec/software/qpOASES/.

[23] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[24] R. Findeisen and F. Allgöwer. Computational delay in nonlinear model predictive control. Proc. Int. Symp. Adv. Control of Chemical Processes, ADCHEM, 2003.

[25] R. Fletcher. *Practical Methods of Optimization*. Wiley, Chichester, 2nd edition, 1987.

[26] R. Franke. *Integrierte dynamische Modellierung und Optimierung von Systemen mit saisonaler Wärmespeicherung*. PhD thesis, Technische Universität Ilmenau, Germany, 1998.

[27] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report, Numerical Analysis Report 97-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.

[28] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, London, 1999.

[29] T. Glad and H. Johnson. A method for state and control constrained linear-quadratic control problems. In *Proceedings of the 9th IFAC World Congress, Budapest, Hungary*, pages 1583–1587, 1984.

[30] A. Griewank. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, 2000.

[31] A. Griewank and Ph.L. Toint. Partitioned variable metric updates for large structured optimization problems. *Numerische Mathematik*, 39:119–137, 1982.

[32] A. Griewank and A. Walther. On constrained optimization by adjoint based quasi-Newton methods. *Optimization Methods and Software*, 17:869 – 889, 2002.

[33] J. Guddat, F. Guerra Vasquez, and H.T. Jongen. *Parametric Optimization: Singularities, Pathfollowing and Jumps.* Teubner, Stuttgart, 1990.

[34] A. Helbig, O. Abel, and W. Marquardt. Model predictive control for on-line optimization of semi-batch reactors. In *Proc. Amer. Contr. Conf.*, pages 1695–1699, Philadelphia, 1998.

[35] J.B. Jorgensen, J.B. Rawlings, and S.B. Jorgensen. Numerical methods for large-scale moving horizon estimation and control. In *Proceedings of Int. Symposium on Dynamics and Control Process Systems (DYCOPS)*, 2004.

[36] Wei Kang. Moving horizon numerical observers of nonlinear control systems. *IEEE Transactions on Automatic Control*, 51(2):344–350, 2006.

[37] P. Krämer-Eis and H.G. Bock. Numerical treatment of state and control constraints in the computation of feedback laws for nonlinear control problems. In P. Deuflhard et al., editor, *Large Scale Scientific Computing*, pages 287–306. Birkhäuser, Basel Boston Berlin, 1987.

[38] T. Kraus, P. Kühl, L. Wirsching, H. G. Bock, and M. Diehl. A moving horizon state estimation algorithm applied to the tennessee eastman benchmark process. In *Proc. of IEEE Robotics and Automation Society conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006.

[39] D.B. Leineweber, I. Bauer, A.A.S. Schäfer, H.G. Bock, and J.P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (Parts I and II). *Computers and Chemical Engineering*, 27:157–174, 2003.

[40] W.C. Li and L.T. Biegler. Multistep, Newton-type control strategies for constrained nonlinear processes. *Chem. Eng. Res. Des.*, 67:562–577, 1989.

[41] W.C. Li and L.T. Biegler. Newton-type controllers for constrained nonlinear processes with uncertainty. *Industrial and Engineering Chemistry Research*, 29:1647–1657, 1990.

[42] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: stability and optimality. *automatica*, 26(6):789–814, 2000.

[43] A. M'hamdi, A. Helbig, O. Abel, and W. Marquardt. Newton-type receding horizon control and state estimation. In *Proc. 13rd IFAC World Congress*, pages 121–126, San Francisco, 1996.

[44] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, Heidelberg, 1999.

[45] T. Ohtsuka. A continuation/gmres method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.

[46] T. Ohtsuka and H.A. Fujii. Nonlinear receding-horizon state estimation by real-time optimization technique. *Journal of Guidance, Control, and Dynamics*, 19(4), 1996.

[47] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G.A. Watson, editor, *Numerical Analysis, Dundee 1977*, volume 630 of *Lecture Notes in Mathematics*, Berlin, 1978. Springer.

[48] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

[49] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99:723–757, 1998.

[50] R.W.H. Sargent and G.R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Proceedings of the 8th IFIP Conference on Optimization Techniques (1977), Part 2*, Heidelberg, 1978. Springer.

[51] A. Schäfer, P. Kühl, M. Diehl, J.P. Schlöder, and H.G. Bock. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chemical Engineering and Processing*, 46(11):1200–1214, 2007.

[52] A.A.S. Schäfer. *Efficient reduced Newton-type methods for solution of large-scale structured optimization problems with application to biological and chemical processes.* PhD thesis, Universität Heidelberg, 2005.

[53] J.P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung*, volume 187 of *Bonner Mathematische Schriften.* Universität Bonn, Bonn, 1988.

[54] Y. Shimizu, T. Ohtsuka, and M. Diehl. A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/krylov method. *International Journal of Robust and Nonlinear Control*, 2008. (accepted).

[55] Y. Soneda and T. Ohtsuka. Nonlinear moving horizon state estimation for a hovercraft with continuation/gmres method. In *Proc. of IEEE Conf. on Control Applications*, 2002.

[56] M.C. Steinbach. A structured interior point sqp method for nonlinear optimal control problems. In R. Bulirsch and D. Kraft, editors, *Computation Optimal Control*, pages 213–222, Basel Boston Berlin, 1994. Birkhäuser.

[57] M.J. Tenny and J.B. Rawlings. Efficient moving horizon estimation and nonlinear model predictive control. In *Proceedings of the American Control Conference*, Anchorage, AK, 2002.

[58] M.J. Tenny, S.J. Wright, and J.B. Rawlings. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Computational Optimization and Applications*, 28(1):87–121, April 2004.

[59] T.H. Tsang, D.M. Himmelblau, and T.F. Edgar. Optimal control via collocation and non-linear programming. *International Journal on Control*, 21:763–768, 1975.

[60] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering.* PhD thesis, Carnegie Mellon University, 2002.

[61] L. Wirsching. An SQP algorithm with inexact derivatives for a direct multiple shooting method for optimal control problems. Master's thesis, University of Heidelberg, 2006.

[62] L. Wirsching, H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy for fast adjoint based nonlinear model predictive control. In *Preprints of the 7th Symposium on Nonlinear Control Systems (NOLCOS), Pretoria*, 2007.

[63] S.J. Wright. *Primal-Dual Interior-Point Methods.* SIAM Publications, Philadelphia, 1997.

[64] V. M. Zavala and L.T. Biegler. The advanced step nmpc controller: Optimality, stability and robustness. *Automatica*, 2008. (accepted for publication).

[65] V.M. Zavala and L.T. Biegler. Nonlinear programming sensitivity for nonlinear state estimation and model predictive control. In *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, 2008.