Introduction to Model Based Optimization of Chemical Processes on Moving Horizons

T. Binder[1], L. Blank[2], H.G. Bock[3], R. Bulirsch[4], W. Dahmen[2], M. Diehl[3], T. Kronseder[4], W. Marquardt[1], J.P. Schlöder[3], and O. von Stryk[5]

[1] Lehrstuhl für Prozesstechnik, RWTH Aachen,
[2] Institut für Geometrie und praktische Mathematik, RWTH Aachen,
[3] Interdisziplinäres Zentrum für wissenschaftliches Rechnen, Universität Heidelberg,
[4] Lehrstuhl für Höhere Mathematik und Numerische Mathematik, TU München,
[5] Fachgebiet Simulation und Systemoptimierung, TU Darmstadt.

Abstract. Dynamic optimization problems are typically quite challenging for large-scale applications. Even more challenging are on-line applications with demanding real-time constraints. This contribution provides a concise introduction into problem formulation and standard numerical techniques commonly found in the context of moving horizon optimization using nonlinear differential algebraic process models.

Keywords: dynamic optimization, optimal control, direct methods

1 Introduction

Safe and economical process operation is of crucial importance for the success of chemical companies. Model based optimization is a promising technique to increase the operational profit in process operation. Moving horizon optimization includes model predictive control (MPC) and receding horizon estimations (RHE) and requires on-line dynamic optimization (see, e.g., Helbig et al. 2000). MPC regulates processes whereas RHE is used to estimate unaccessible process states and parameters (see Allgöwer et al. 1999 for an excellent survey on both problems). In these applications a multi-variable optimization problem restricted to a large scale mathematical process model has to be solved on-line. The large scale nature as well as the real-time requirement of the problem is a clear challenge where the cutting edge of currently commercially available technology needs to be pushed further forward. It is the intention of this article to provide a concise introduction into the exciting field of dynamic optimization applied on moving horizons and to summarize the available numerical techniques. However, due to the limitations in space we only focus on techniques which from our point of view are considered as standard technologies currently applied. Recent results developed within the Schwerpunktprogramm are covered elsewhere in this book by additional contributions of each research group (Binder et al., 2001; Diehl et al., 2001; Kronseder et al., 2001).

This introductory article is organized as follows. In Section 2 we introduce the generic problem formulation of control and estimation problems which are closely related. However, we will also illuminate the differences between both. Special attention is given to the mathematical process model in Section 2.1 whereas in Sections 2.2 and 2.3 the optimization problems on fixed and moving horizons are defined respectively. Furthermore, we emphasize the particular real-time character intrinsic to moving horizon optimization.

Closed loop stability of the applied algorithm is of crucial importance. A short introduction into the terminology and basic concepts is given in Section 3. Here, we motivate how stability problems arise when a finite dimensional horizon instead of an infinite dimensional one is chosen.

Section 4 summarizes the basic techniques to solve the dynamic optimization problem on a fixed horizon. We start with a short introduction into optimal feedback controls which are described by the Hamilton-Jacobi-Carathéodory-Bellman partial differential equation, and briefly introduce techniques based on the so called Maximum Principle in Section 4.2.

In Section 5 we focus on the so called direct methods for the numerical solution of optimal control problems. Here, the infinite dynamic optimization problem is transformed into a nonlinear program (NLP) by parameterizing the controls. Special room is given to direct single shooting (Section 5.1), direct multiple shooting (Section 5.2) and direct collocation (Section 5.3), techniques which are commonly used so solve large scale problems. The numerical solution of the NLP by sequential quadratic programming (SQP) is outlined in Section 5.4, and the three presented direct techniques are compared in Section 5.5.

Extensions of fixed horizon optimization to moving horizon optimization are discussed in Section 6. We start with the well-known recursive solution approaches for regulation (Section 6.1) and estimation (Section 6.2) available for unconstrained linear quadratic optimization problems. The simplicity and power of these recursive techniques motivate extensions to nonlinear models. Therefore Section 6.3 discusses optimal feedback control obtained by linearization along a specific reference solution. In moving horizon optimization the numerical cost can be lowered substantially using appropriate initialization techniques. The various options of commonly applied approaches are outlined Section 6.4.

The article is concluded in Section 7 by a short summary.

2 Problem Formulation

2.1 Model

Mathematical process models are an abstraction of real process systems and aim to capture the essential features of concern. In general, the process models are either based on fundamental principles or empirical observations or in the hybrid case on a mixture of both. The basis for virtually all fundamental process models are the general conservation principles of mass, momentum and energy. As long as the underlying assumptions remain valid, fundamental models can be expected to extrapolate to new operating regions where no data sets are available. However, it is a rather difficult and time consuming task to construct and validate good fundamental process models (see, e.g., Aris, 1994; Bauer et al., 2000; Marquardt 2001). An empirical model built from available process data might be more convenient in some instances since a detailed process understanding is not required for the model development, although a suitable model structure has to be selected as well. Artificial neural networks are the most popular framework for empirical model development (Su and McAvoy, 1997), but other techniques based on Hammerstein and Wiener models (Norquay et al., 1999; Pearson and Pottmann, 2000; Wellers and Rake, 2000), Volterra models (Maner et al., 1996), and polynomial ARMAX models (Sriniwas and Arkun, 1997) might be considered alternatively. In this contribution a detailed discussion of the particular advantages and disadvantages of fundamental or empirical modeling are off focus since from an optimization point of view we only need a sufficiently good process model. The underlying principles of the building process are of minor importance, although they very well might affect the applicability of the model and the particular choice of the numerical solution method. We assume that a fundamental process model is available, but we keep in mind that other model types might be used as well. There is a wide variety of phenomena in chemical process systems such that we have various types of process models which vary over a large range starting from simple algebraic equation systems, to ordinary (ODE) or differential-algebraic (DAE) equations systems, and to more complicated (partial-) integro-differential equations. Despite this richness our discussion is limited to mathematical process models which can be represented as DAE systems given by

$$\mathbf{0} = \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}, t), \quad \forall t \in I, \tag{1}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}, t), \quad \forall t \in I, \tag{2}$$

Here, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and $\mathbf{z}(t) \in \mathbb{R}^{n_z}$ denote the differential and the algebraic system state vectors, respectively. $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are operational variables which can be directly manipulated by process operators. Modeling uncertainties and disturbances are concatenated without further specification into a vector function $\mathbf{w}(t) \in \mathbb{R}^{n_w}$. $\mathbf{p} \in \mathbb{R}^{n_p}$ denotes a vector of time-invariant system parameters. The function $\mathbf{f}$ (with $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}}$ invertible) describes the differential portion while the function $\mathbf{g}$ represents the algebraic portion of the process model. In general the Jacobian $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$ might be singular such that the DAE could be of a higher index, where roughly spoken, the index denotes the minimum number the system has to be differentiated with respect to time to be able to transform the DAE system into an ODE system. Details on the theory of DAE's can be found for example in Brenan et al. 1996 or Unger et al. 1995. Because of the richness of phenomena occuring in higher index problems we limit ourselves to problems of index one. The time interval of interest is denoted in the sequel by $I := [t_0, t_f]$ where t_0, t_f are starting and final times respectively. The process model (1) and (2) might be used for simulation. Given particular values of $\mathbf{u}^\star(t)$[1], $\mathbf{w}^\star(t)$, $t \in I$, $\mathbf{p}^\star$ and appropriate initial conditions the process model (1) and (2) can be solved using a suitable integration routine. For notational convenience we assume in the remainder that initial conditions are provided for the differential states, i.e., $\mathbf{x}(t_0) = \mathbf{x}_0$. A more general discussion on the specification of initial conditions can be found in e.g., Kröner et al. 1992, Brenan et al. 1996 or Unger et al. 1995 for index one and higher index problems.

Some functions of the system states are measurable, therefore we augment the model (1) and (2) by the sensor model

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}, t) \in \mathbb{R}^{n_y}, \quad \forall t \in I, \tag{3}$$

that determines the output variables $\mathbf{y}$ as a function of the other system variables.

2.2 Off-line Optimization on a Fixed Horizon

Before we start to outline moving horizon dynamic optimization we consider first an off-line problem on a fixed horizon such as the optimization of batch processes. These problems require the minimization of an objective function by adjusting the free operational variables $\mathbf{u}$, also referred to as controls, in an appropriate manner within the finite interval $I^r = [t_0^r, t_f^r]$ which denotes an operational phase of the process, such as the time required for a grade change of a continuous process or the reaction phase in a batch process. The final time may be fixed or subject to optimization.

The controls $\mathbf{u}$ cannot be adjusted arbitrarily since they might be restricted by constraints which are typically associated with physical limits such as, e.g., restrictions on valve position or rate of change.

Further (mixed) constraints on controls *and* states comprise, e.g., limits on capacity of production units and quality specifications on the product, as well as safety constraints. For notational simplicity, both types of restrictions are concatenated in a general constraint vector function $\mathbf{c}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}, t)$. The constraints $\mathbf{c}$ have to be enforced during process operation at any time $t \in I$.

Optimal operation of the process with respect to the specified cost functional could be achieved if a perfect process model (1), (2) and (3) of the process would be available and if the initial state $\mathbf{x}_0$ at t_0, the parameters $\mathbf{p}$, and the disturbances $\mathbf{w}$ were known exactly. Then the controls and therefore the operational trajectory could be determined entirely off-line through the solution of the following dynamic optimization problem (provided that it is

[1] The superscript $\star$ denotes specific but arbitrary values.

solvable)

$$\min_{\substack{\mathbf{x}^r(\cdot),\mathbf{z}^r(\cdot),\\ \mathbf{u}^r(\cdot),t_f^r}} \quad E^r(\mathbf{x}^r(t_f^r),\mathbf{z}^r(t_f^r),\mathbf{p}^r) + \int_{t_0^r}^{t_f^r} L^r(\mathbf{x}^r,\mathbf{z}^r,\mathbf{u}^r,\mathbf{p}^r,\tau)\,d\tau \tag{4}$$

$$\begin{aligned}
\text{s.t.} \qquad\qquad \mathbf{0} &= \mathbf{f}(\dot{\mathbf{x}}^r(t),\mathbf{x}^r(t),\mathbf{z}^r(t),\mathbf{u}^r(t),\mathbf{w}^r(t),\mathbf{p}^r,t), \quad t \in I^r,\\
\mathbf{x}^r(t_0^r) &= \mathbf{x}_0^r,\\
\mathbf{0} &= \mathbf{g}(\mathbf{x}^r(t),\mathbf{z}^r(t),\mathbf{u}^r(t),\mathbf{w}^r(t),\mathbf{p}^r,t), \quad t \in I^r,\\
\mathbf{0} &\le \mathbf{c}^r(\mathbf{x}^r(t),\mathbf{z}^r(t),\mathbf{u}^r(t),\mathbf{p}^r,t), \quad t \in I^r,\\
\mathbf{0} &= \mathbf{r}^r(\mathbf{x}^r(t_f^r),\mathbf{z}^r(t_f^r),\mathbf{p}^r).
\end{aligned}$$

In the case of tracking problems the Lagrange term L^r may be given by an appropriate norm of the difference between the output trajectory $\mathbf{y}$ and a given reference trajectory $\boldsymbol{\eta}^r(t)$, such as a weighted Euclidean norm with the particular weighting $\mathbf{S}$:

$$L^r(\mathbf{x}^r,\mathbf{z}^r,\mathbf{u}^r,\mathbf{p}^r,t) := ||\mathbf{h}(\mathbf{x}^r(t),\mathbf{z}^r(t),\mathbf{u}^r(t),\mathbf{p}^r,t) - \boldsymbol{\eta}^r(t)||_{\mathbf{S}}^2.$$

E^r is then the penalty for the final states. In a more general case L^r and E^r may denote an economical cost function. The vector function $\mathbf{r}^r$ is used to account for endpoint constraints. The superscript r in all quantities indicates that (4) is typically an optimal control problem which aims to determine the regulating optimal trajectory for the control $\mathbf{u}$. Furthermore, problem (4) is commonly referred to as an open loop optimal control problem, since no feedback from the process enters the problem formulation.

A similar dynamic optimization problem can be formulated if one aims to determine unknown or hardly accessible process quantities such as initial conditions, process parameters, process disturbances, or model uncertainty from process measurements on $I^e = [t_0^e, t_f^e]$. We can formulate the following dynamic optimization problem for the off-line estimation of the unknown quantities using data that have been collected during an operational phase I^e.

$$\min_{\substack{\mathbf{x}_0^e,\mathbf{x}^e(\cdot),\mathbf{z}^e(\cdot),\\ \mathbf{w}^e(\cdot),\mathbf{p}^e}} \quad E^e(\mathbf{x}^e(t_0^e),\mathbf{z}^e(t_0^e),\mathbf{p}^e) + \int_{t_0^e}^{t_f^e} L^e(\mathbf{x}^e,\mathbf{z}^e,\mathbf{w}^e,\mathbf{p}^e,\tau)\,d\tau \tag{5}$$

$$\begin{aligned}
\text{s.t.} \qquad\qquad 0 &= \mathbf{f}(\dot{\mathbf{x}}^e(t),\mathbf{x}^e(t),\mathbf{z}^e(t),\mathbf{u}^e(t),\mathbf{w}^e(t),\mathbf{p}^e,t), \quad t \in I^e,\\
\mathbf{x}^e(t_0^e) &= \mathbf{x}_0^e,\\
0 &= \mathbf{g}(\mathbf{x}^e(t),\mathbf{z}^e(t),\mathbf{u}^e(t),\mathbf{w}^e(t),\mathbf{p}^e,t), \quad t \in I^e,\\
0 &\le \mathbf{c}^e(\mathbf{x}(t),\mathbf{z}^e(t),\mathbf{u}^e(t),\mathbf{p}^e,t), \quad t \in I^e.
\end{aligned}$$

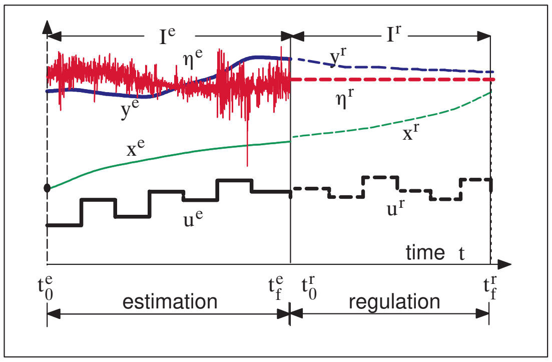

Fig. 1. Moving horizon approach.

Now, the superscript e is used to indicate the estimation. Here, the Lagrange term is typically given as a weighted Euclidean norm of the difference between the measurements $\boldsymbol{\eta}^e(t)$ and the model response $\mathbf{y}$

$$L^e(\mathbf{x}^e, \mathbf{z}^e, \mathbf{w}^e, \mathbf{p}^e, t) := ||\mathbf{h}(\mathbf{x}^e(t), \mathbf{z}^e(t), \mathbf{u}^e(t), \mathbf{p}^e, t) - \boldsymbol{\eta}^e(t)||^2_{\mathbf{S}}.$$

A typical weighting matrix $\mathbf{S}$ is the inverse of the covariance matrix of the measurement error. Nevertheless, more general weights like, e.g., time dependent operators are possible, too (Binder et al. 1998). The measurement function $\boldsymbol{\eta}^e(t)$ has to be generated appropriately from the measurements taken from the process at discrete sampling times. The measurements might be as well included pointwise by substituting the integral by a finite sum. The initial conditions $\mathbf{x}^e_0$, the parameters $\mathbf{p}^0$, and the disturbances $\mathbf{w}^e$ are free variables to be determined by the optimizer.

A reference value $\bar{\mathbf{x}}_0$ which could be close to the true initial conditions can be incorporated into the initial penalty E^e, e.g., by

$$E^e(\mathbf{x}^e(t^e_0), \mathbf{z}^e(t^e_0), \mathbf{p}^e) := ||\mathbf{x}^e(t^e_0) - \bar{\mathbf{x}}^e_0||^2_{\mathbf{S}},$$

where the particular weight $\mathbf{S}$ reflects the confidence in such a reference value. The controls $\mathbf{u}^e$ are typically accessible and therefore assumed to be known.

So far we considered the regulating and estimation problems independently of each other, each formulated on a fixed horizon. It is obvious that both problems can be also coupled. For example, consider an operational phase of a production process where first process data is collected in some interval to estimate unknown quantities. Then, based on these estimates an optimal operational trajectory is determined on the remaining time interval. Let's assume the time interval $[t_0, t_f]$ is split in two parts, i.e., $I^e = [t^e_0, t^e_f]$, $I^r = [t^r_0, t^r_f]$ where $t^e_0 = t_0$, $t^e_f = t^r_0$, $t^r_0 = t_f$; see Figure 1. The optimal solution of (5) provides estimates on I^e of parameters $\mathbf{p}$, disturbances $\mathbf{w}(t)$, and measured or unmeasured states $\mathbf{x}(t), \mathbf{z}(t)$ which are consistent with the process model. The estimates can then be used in the regulator problem (4) on I^r. In (4) the uncertainty and disturbance vector $\mathbf{w^r}$ is assumed to be fixed and known. Typically its values are suitable predictions based upon the estimates $\mathbf{w^e}$ computed on I^e, e.g. $\mathbf{w^r} = \mathcal{P}_\mathbf{w}(\mathbf{w^e})$ where $\mathcal{P}_w$ denotes a prediction operator. The predictions are computed by extrapolation or by use of simple disturbance models as discussed, e.g., in Ricker (1991). Furthermore, the problems (4) and (5) are coupled by the initial condition $\mathbf{x^r}(\mathbf{t_0^r}) = \mathbf{x^e}(\mathbf{t_f^e})$ and the parameters $\mathbf{p}^r = \mathcal{P}_p(\mathbf{p}^e)$ which have been determined from (5) for further use in (4) using the prediction operator $\mathcal{P}_p$. Note that in this example the solution to (4) and (5) cannot be computed off-line anymore.

The concept of estimation and regulation has been outlined for a simple setting. It is clear that the methodology also applies in more general situations, to better deal with the uncertainty in the model and the disturbances acting on the real process. Next, the horizons are repetitively shifted with time by a sampling time interval ΔT such that a moving horizon optimization problem is obtained.

2.3 On-line Optimization on Moving Horizons

In moving horizon optimization problems (4) and (5) are solved repeatedly. Unknown process quantities are estimated from the collected process measurements using (5). Based on this estimates an optimal trajectory $\mathbf{u}^{r*}(t)$ $t \in I^r$ is determined by solving (4)[2], but $\mathbf{u}^{r*}(t)$ $t \in I^r$, is applied to the process only during an interval ΔT, i.e., $\mathbf{u}^{r*}(t)$, $t_0^r \leq t < t_0^r + \Delta T$. Then, new measurement information is collected, the estimation and regulation horizons I^e, I^r are shifted by ΔT, and (4), (5) are resolved. We now have several horizons which typically overlap such that we introduce a horizon index k which is also used as subscript in notation, i.e., the horizons are denoted by $I_k^e := [t_{0,k}^e, t_{f,k}^e]$ and $I_k^r := [t_{0,k}^r, t_{f,k}^r]$. Furthermore, we include I_k^e, I_k^r as a second argument in all quantities appearing in (4) and (5) to distinguish the solutions computed on different horizons. Therefore $\mathbf{u}^{r*}(t|I_k^r)$, $t \in I_k^r$, denotes the optimal solution $\mathbf{u}^{r*}$ in problem (4) obtained on horizon I_k^r. Similar notation applies for all other quantities in problems (4) and (5). Note, that in this problem setting we estimate all quantities simultaneously. In practice the unknowns might live on different time-scales such that there is no need to estimate slowly varying quantities in each time step (Helbig et al., 2000).

So far, the framework introduced is quite idealistic since it assumes that problems (4) and (5) can be solved instantly. For practical applications the estimation and prediction horizons have to be separated by some time to perform the necessary numerical computations and data input/output operations. Therefore, the estimation and prediction horizons are separated typically by one sampling time, i.e., $t_{0,k}^r = t_{f,k}^e + \Delta T$. An illustration of the

[2] The superscript $*$ denotes the optimal values.

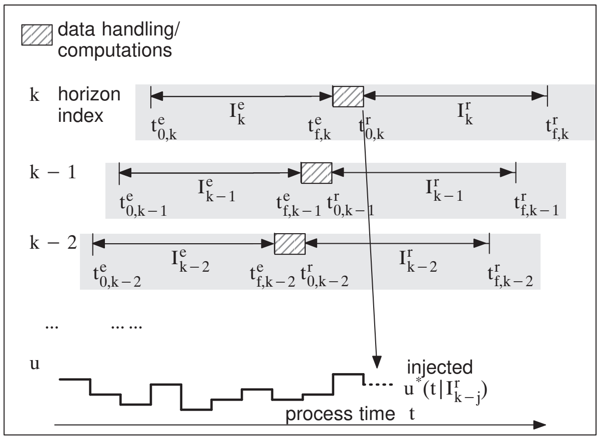

Fig. 2. On-line optimization.

moving horizon approach is given in Figure 2, where however $\mathbf{w}, \mathbf{z}, \mathbf{p}$ are not displayed to avoid an overload of the graph. With these definitions the basic moving horizon algorithm is given by:

1. While $t^e_{f,k} \leq t < t^r_{0,k}$:
   - Apply $\mathbf{u}^r(t|I^r_{k-1})$ to the process.
   - Access the measurement values $\boldsymbol{\eta}^e(t)$, $t \in I^e_k$.
   - Solve the estimation problem (5), use the so far injected controls as $\mathbf{u}^e(t)$ for $t \in I^e_k$, i.e., compute $\mathbf{x}^e(t|I^e_k)$, $\mathbf{z}^e(t|I^e_k)$, $\mathbf{w}^e(t|I^e_k)$, $\mathbf{p}^e_{I^e_k}$.
   - Compute $\mathbf{x}^r(t^r_{0,k})$ using the model $\mathbf{f}, \mathbf{g}$ in (5) as prediction model with initial condition $\mathbf{x}^e(t^e_{f,k})$, control $\mathbf{u}^r(t|I^r_{k-1})$ and suitable prediction models $\mathcal{P}_w, \mathcal{P}_p$ to extrapolate $\mathbf{w}(t^e|I^e_k)$ and $\mathbf{p}^e_{I_k}$ on $[t^e_{f,k}, t^r_{0,k}]$. Extend the extrapolation to I^r_k such that $\mathbf{w}^r$, $\mathbf{p}^r_{I_k}$ is obtained.
   - Solve the control problem (4) with the extrapolated quantities and determine $\mathbf{u}^r(t|I^r_k)$ to be injected into the process in the upcoming step $t^e_{f,k+1} < t < t^r_{0,k+1}$.
2. $k := k + 1$.
3. Goto 1.

The underlying assumption of the algorithm is that the prediction models are of sufficient quality such that the initial guess $\mathbf{x}^r(t^r_{0,k})$ and the extrapolations $\mathbf{w}^e(t)$, $t^e_{f,k} \leq t \leq t^r_{0,k}$, $\mathbf{w}^r(t)$, $t \in I^r_k$, are close to the true values.

Obviously, the time ΔT should be as small as possible where at least ΔT has to be sufficiently smaller than the dominating process time constants. These time constants depend on a number of factors such as for example on the particular chemical species involved or on the particular unit operations used. While distillation column time constants with regard to product concentration are in the range of hours the product concentration of chemical reactors can change in seconds. The computational complexity of an algorithm to solve (4) and (5) depends in addition on a number of other factors such as the used model (type, structure, dimension), numerical solution approaches (optimization method, discretization), choice of cost functional and the horizon length. While the time constant is given by the process, the computational complexity is affected by engineer and mathematician through modeling and algorithmic design decisions. The designed algorithm has to prompt in any event the optimal values (or at least suitable approximations) of (4) and (5) within the available time span ΔT since otherwise proper function of the on-line optimization scheme cannot be guaranteed. This is an important real-time requirement which should be addressed by the design of the algorithms.

The functionality of the process is further affected by the closed loop stability properties (e.g., Bitmead, Gevers and Wertz, 1990) of the moving horizon approach which will be addressed in the next section.

3 Remarks on Closed Loop Stability

First, we discuss closed loop stability for the regulator problem (4) assuming fully accessible differential states, i.e., $\mathbf{y}^r = \mathbf{x}^r$, and given $\mathbf{p}^r$, $\mathbf{w}^r(t)$. Let's assume that we want to find an optimal control $\mathbf{u}^r(t), t_0^r \leq t \leq \infty$ which moves the process state $\mathbf{x}^r$ from some given initial conditions $\mathbf{x}^r(t_{0,k}^r)$ to a target state which for simplicity is chosen to be the origin, i.e., $\boldsymbol{\eta}^r(t) = 0, \forall t$. Suppose, that the considered system is controllable (see, e.g, Ogunnaike and Ray, 1994, for an introduction to the concept of controllability) and that no unknown disturbances, unknown parameters and model uncertainties are present. Furthermore, we assume to know the true initial conditions $\mathbf{x}(t_{0,k}^r)$.

It follows from Bellman's Principle of Optimality (e.g. Anderson and Moore, 1989), that in each horizon k the predicted state and control trajectories $\mathbf{x}^r(t, I_k^{r,\infty}), \mathbf{z}^r(t, I_k^{r,\infty}), \mathbf{u}^r(t, I_k^{r,\infty})$ of problem (4), where $I^{r,\infty} := [t_{0,k}^r, \infty]$, are equal to the optimal process trajectories $\mathbf{x}^r(t, I_0^{r,\infty}), \mathbf{z}^r(t, I_0^{r,\infty}), \mathbf{u}^r(t, I_0^{r,\infty})$ of the process system determined on $[t_0^r, \infty]$. This holds only if the first problem $k = 0$ is feasible and if the initial conditions $\mathbf{x}^r(t_{0,k}^r)$ are known for all k (Keerthi and Gilbert, 1988). Therefore, for infinite horizons there is no difference between the subsequent control sequences determined at certain time steps and the control trajectory obtained by solving a single problem. This implies closed loop stability, as any feasible optimized trajectory goes to the origin (Keerthi and Gilbert, 1988).

When instead a (small) finite horizon I_k^r, $t_{f,k} < \infty$, is chosen the actual closed loop control and state trajectories will differ in general from the predicted open loop trajectories even if no model uncertainty and unknown disturbances are present which is nicely illustrated by Bitmead, Gevers, and Wertz (1990). The solutions computed on $I_k^{r,\infty}$ and I_k^r may differ significantly the shorter I_k^r is chosen. Since from a theoretical perspective the minimum requirement of a model based controller is that it yields a stable closed–loop system if a perfect model of the plant is available and if the state is completly accessible by measurements (Henson, 1998), intense research has been undertaken in the last decade to develop schemes with guaranteed nominal stability properties. The major developments are summarized in excellent surveys given by Mayne (1995), Morari and Lee (1999), Allgöwer et al. (1999), and Mayne et al. (2000). However, the developed approaches are yet computationally expensive, difficult to design and therefore limited to processes with low state dimensions. So far, moving horizon schemes with guaranteed stability have been only applied in academia. Besides the inherent drawbacks of the approaches with guaranteed stability this might as well be due to the fact that it is typically not difficult for practical problems to find long enough horizons by trial and error such that closed loop stability is obtained. However, it should be admitted that it is difficult to come up with a generally applicable horizon design procedure, which, given a specific problem, determines stabilizing prediction and control horizons based on the process model and the cost functional chosen (Allgöwer et. al, 1999).

Similar stability considerations apply to the estimation problem. Here, stability of the estimator is defined as the convergence of the estimated states to the true states for $t \to \infty$ for arbitrarily specified initial conditions, if the measurements contain no errors, the model is correct and the disturbances and parameters are known. Stability is trivially obtained by dropping the Mayer term in the objective ($E^e = 0$), because the minimization of the cost immediately moves the initial state to the correct value. However, such a strategy would lead to poor estimation quality if measurement noise and model uncertainty would be present as in any real situation. In these cases one should include an appropriate guess of the initial condition, say $\bar{\mathbf{x}}^e_{0,k}$, to improve estimation quality. This can be accomplished by introducing the Mayer term $E^e(\mathbf{x}^e(t^e_{0,k}) - \bar{\mathbf{x}}^e_{0,k})$ where E^e typically reflects some kind of least-squares formulation of the error. Alternatively, one could account for all available measurement information for the current estimation by keeping $t^e_{0,k} = t_{0,0},\ \forall k$. The resulting problems are not computationally tractable since the problem dimension grows as the estimation horizon grows. Instead in the k-th horizon past data in $[t^e_{0,0}, t^e_{f,k}]$ are indirectly accounted for by $\bar{\mathbf{x}}^e_{0,k}$ which is used to reflect the past estimate and thus indirectly the information content of the past measurement data. Thus the weights in E^e reflect the confidence in the past estimates. E^e has to be chosen rather carefully to ensure proper weighting of the old data. Estimator divergence may result if the initial penalty E^e biases the old data by too strongly weighting the past estimates, while performance may suffer if the initial penalty neglects the old data by not sufficiently weighting them. Stability and performance implications for several choices of E^e for a number of problems are discussed in a rigorous manner by a number of contributions, e.g., see Michalska and Mayne (1995), Muske and Rawlings (1995), Robertson et al. (1996), and Rao and Rawlings (2000).

So far stability has been illuminated separately for the regulator and estimation problem. For linear time invariant models with quadratic cost functionals and no inequality restrictions present the separation principle holds and closed loop stability of the combined problem follows if the estimation and regulation problem are stable independently. Furthermore, it can be proven for general systems (Meadows and Rawlings, 1995) that if an exponentially converging estimator is combined with a stable control algorithm where all states are measurable, then this observer-controller system is stable. This holds even for nonlinear regulators where the separation principle obviously does not hold.

4 Overview of Solution Methods for Optimal Control Problems on Fixed Horizon

Next, we discuss the numerical techniques which are commonly applied to solve dynamic optimization problems. First we review available methods to solve optimal control problems on a fixed horizon before we examine particular extensions towards a moving horizon.

Many methods for the on-line solution of optimal control problems on moving horizons are based on algorithms designed for the off-line computation of solutions to optimal control problems on a fixed interval $I := [t_0, t_f]$ in time (including problems where t_f is as well a degree of freedom in the optimization problem). Therefore, in this section we give an overview of the most common off-line optimal control methods which will form the core of any receding horizon strategy.

In the previous sections the necessity to distinguish the estimation and regulator problems required an extended notation which we drop here for convenience, since both types of optimal control problems can be solved using similar numerical techniques. Additionally, we restrict our attention to ODE models as the DAE case introduced above poses additional theoretical and practical difficulties which are beyond the scope of this general discussion. However, for further information on dynamic optimization with DAE systems we refer to Pytlak (1999) as well as to the articles by Büskens et al. (2001), Diehl et al. (2001), and Kröner et al. (2001) within this book as a starting point.

We consider a deterministic optimal control problem in Bolza form on a fixed horizon $I := [t_0, t_f]$ with

$$\min_{\mathbf{u}(\cdot),\mathbf{x}(\cdot)} J[\mathbf{u}(\cdot),\mathbf{x}(\cdot)] := E\left(\mathbf{x}(t_f)\right) + \int_{t_0}^{t_f} L\left(\mathbf{x}(t),\mathbf{u}(t),t\right)\mathrm{d}t \tag{6}$$

subject to

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right), \quad t \in I \tag{7}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{8}$$

$$\mathbf{0} \leq \mathbf{c}\left(\mathbf{x}(t), \mathbf{u}(t), t\right), \quad t \in I \tag{9}$$

$$\mathbf{0} = \mathbf{r}\left(\mathbf{x}(t_f)\right), \tag{10}$$

where $\mathbf{x} : I \to \mathbb{R}^{n_x}$, $n_x \geq 1$, and $\mathbf{u} : I \to \mathbb{R}^{n_u}$, $n_u \geq 1$, denote the state and control variables. The model ODE is denoted by $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times I \to \mathbb{R}^{n_x}$, $\mathbf{c} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times I \to \mathbb{R}^{n_c}$, $n_c \geq 1$, is a general nonlinear inequality constraint function, and $\mathbf{r} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_r}$, $n_r \geq 0$ describes the end point constraints. The objective incorporates a Mayer term $E : \mathbb{R}^{n_x} \to \mathbb{R}$ and a Lagrange term with $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times I \to \mathbb{R}$.

For simplicity, the final time $t_f > t_0$ as well as the initial conditions $\mathbf{x}_0$ and the model parameters are assumed to be known and fixed, but an extension of the solution methods presented towards a free end time and unknown initial initial conditions and model parameters can be obtained straightforwardly. For convenience, the model parameters have been suppressed in Eqns. (6)-(10).

The functions E, L, $\mathbf{f}$, $\mathbf{c}$, and $\mathbf{r}$ are assumed to be twice continuously differentiable with respect to their arguments.

There are three basic approaches to solving optimal control problems of the form (6)-(10):

(I) Hamilton-Jacobi-Carathéodory-Bellman (HJCB) partial differential equations (PDEs) and Dynamic Programming,
(II) Calculus of Variations, Euler-Lagrange differential equations (EL-DEQ), and the Maximum Principle (indirect methods), and
(III) direct methods based on a finite dimensional parameterization of the controls.

We will briefly comment on the first two approaches in Subsections 4.1 and 4.2. The direct methods will be presented in detail in Section 5, as they have proven to be most successful for the treatment of real life large scale optimal control problems.

4.1 Hamilton-Jacobi-Carathéodory-Bellman Partial Differential Equation, Dynamic Programming

In HJCB the optimal *feedback* control $\mathbf{u}^*(\mathbf{x}, t)$ is obtained by solving a PDE for a so-called value function (e.g., Pesch and Bulirsch, 1994). In practice, however, the PDE can be solved numerically for very small state dimensions only. A further severe drawback is that inequality constraints on the state variables as well as dynamical systems with switching points usually lead to discontinuous partial derivatives and cannot easily be included. Discretization methods to compute numerical approximations of the value function by solving a first order PDE with dynamic programming are described by Bardi and Dolcetta (1996), Falcone and Ferretti (1994), and Lions (1982) (*viscosity solutions* of the HJCB equation). It is worth mentioning here that for the subclass of *linear-quadratic regulator* problems, the HJCB-PDE can be solved analytically or numerically by solving either an algebraic or dynamic matrix Riccati equation. This approach is described in more detail in Section 6.2.

A similar solution methodology is obtained by *dynamic programming* (Bellman 1957), which provides the *global* optimal control. Unfortunately, its application is severely restricted in the case of continuous states systems – at most three state dimensions seem feasible so far because of the *curse of dimensionality*. Recently, the application of neural network approximations has been investigated to handle the curse of dimensionality and the *curse of modeling* if dynamic programming is applied to higher dimensional, nonlinear and also stochastic problems (*neuro-dynamic programming*, Bertsekas and Tsitsiklis 1996). Another new development is the adaptive critic method which relies on neural network approximations, reinforcement learning strategies and dynamic programming (e.g., Naumer 1999; Werbos 1992). However, these approaches are still restricted to problems with small state dimension.

4.2 Calculus of Variations, Euler-Lagrange Differential Equations, Maximum Principle (Indirect Methods)

A common approach to compute the optimal control is based on the Maximum Principle, that we will sketch for the case of optimal control problems with the control constrained to the (nonempty) set $\mathcal{U}(t) := \{\mathbf{u} \in \mathbb{R}^{n_u} | \mathbf{0} \leq \mathbf{c}(\mathbf{u}, t)\}$.

First, a Hamiltonian is defined as

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) := -L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}(t)^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{11}$$

where the vector $\boldsymbol{\lambda}(t) : I \to \mathbb{R}^{n_x}$ denotes the so-called adjoint variables. Necessary conditions for optimality of solution trajectories $\mathbf{x}^*(t)$ and $\mathbf{u}^*(t)$, $t \in I$, can then be given by the following boundary value problem in the states $\mathbf{x}^*(t)$ and in the adjoints $\boldsymbol{\lambda}^*(t)$ [3], which form the EL-DEQ for the situation considered

$$\begin{aligned}
\mathbf{x}^*(t_0) &= \mathbf{x}_0 \\
\mathbf{0} &= \mathbf{r}(\mathbf{x}^*(t_f)) \\
\boldsymbol{\lambda}^*(t_f) &= \nabla_{\mathbf{x}} E(\mathbf{x}^*(t_f)) - \nabla_{\mathbf{x}} \mathbf{r}(\mathbf{x}^*(t_f))\boldsymbol{\alpha} \\
&\text{and for almost all } t \in [t_0, t_f] \\
\dot{\mathbf{x}}^*(t) &= \nabla_{\boldsymbol{\lambda}} \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \\
\dot{\boldsymbol{\lambda}}^*(t) &= -\nabla_{\mathbf{x}} \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t).
\end{aligned} \tag{12}$$

[3] $\nabla_{\boldsymbol{\lambda}} \mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) := \left[\frac{\partial \mathcal{H}(\cdot)}{\partial \boldsymbol{\lambda}_1}, \ldots, \frac{\partial \mathcal{H}(\cdot)}{\partial \boldsymbol{\lambda}_{n_x}}\right]^T$ for $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{n_x}]^T$

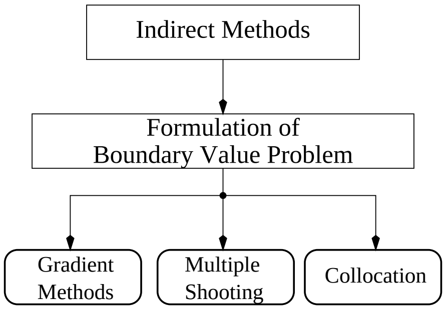

Fig. 3. Overview of numerical methods based on the indirect approach

The vector $\boldsymbol{\alpha} \in \mathbb{R}^{n_r}$ denotes Lagrange multipliers for the end point constraints. The optimal controls are obtained by a *pointwise* maximization of the Hamiltonian, which may lead to discontinuities:

$$\mathbf{u}^*(t) = \arg \max_{\mathbf{u} \in \mathcal{U}(t)} \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t), t). \tag{13}$$

Early developments of the Maximum Principle have been carried out by Pontryagin et al. (1964), Isaacs (1965), and Hestenes (1966). The approach has been extended to handle general constraints (9) on the control and state variables (for an overview see, e.g., Hartl, Sethi, and Vickson 1995). Then the EL-DEQ form an intricate multi-point boundary value problem (MPBVP) with a priori unknown interior switching points denoting the times when one of the constraints becomes active or inactive. Activation or deactivation of a state constraint generally leads to jumps in the adjoint variables.

Several families of numerical methods are based on the EL-DEQ and the Maximum Principle, some of which are listed in Figure 3.

Gradient methods are intended to iteratively improve an approximation of the optimal control by minimizing the Hamiltonian subject to a boundary value problem (Cauchy 1847; Kelley 1960; Tolle 1975; Bryson and Ho 1975; Miele 1980; Chernousko and Luybushin 1982). In each iteration step, the model (7) is numerically integrated forward in time while the adjoint differential equations are integrated backwards in time.

Multiple shooting is one of the most powerful numerical methods for solving the resulting MPBVP derived from the necessary conditions of optimality of a constrained nonlinear optimal control problem, generating highly accurate and verified (with respect to necessary conditions of optimality) solutions. Numerical multiple shooting methods have been developed by Fox (1960), Keller (1968), Bulirsch (1971), Deuflhard (1974), Bock (1978a, 1978b), Oberle (1982), Bock (1987), Kiehl (1989), Hiltmann (1990), and Callies (1999). For an introduction into multiple shooting we refer to Ascher et al. (1988) or Stoer and Bulirsch (1993).

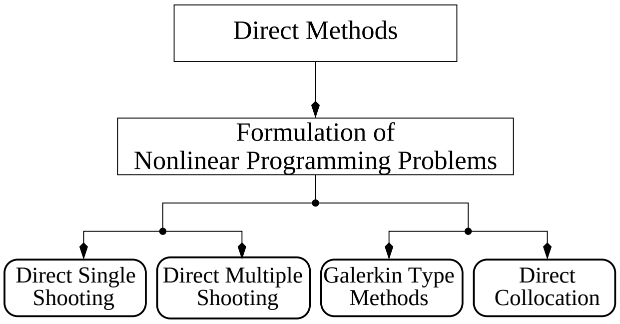

Fig. 4. Overview of numerical methods based on the direct approach

Collocation methods have also been investigated to solve the boundary value problem of the EL-DEQ (e.g., Dickmanns and Well, 1975; Bär, 1983; Ascher et al. 1979) but they have been applied more successfully in the context of direct methods (Section 5).

The practical drawbacks of indirect methods are:

– Proper formulations of the necessary conditions (EL-DEQ etc.) in a numerically suitable way must be derived. The application of automatic differentiation (e.g., Griewank, 2000) may help to partly reduce the efforts to formulate the MBPVP (e.g., Mehlhorn and Sachs, 1994). In spite of this, significant knowledge and experience in optimal control is still required by the user of an indirect method.
– In order to handle active constraints properly, their switching structure must be guessed.
– Suitable initial guesses of the state and adjoint trajectories must be provided to start the iterative methods.
– Changes in the problem formulation (e.g., by a modification of the model equations), or low differentiability properties of the model functions (e.g., by low order interpolation of tabular data), are difficult to include in the solution procedure.

5 Introduction into Direct Solution Algorithms

The basic idea of direct methods for the solution of optimal control problems introduced above is to transcribe the original infinite dimensional problem (6)-(9) into a finite dimensional Nonlinear Programming problem (NLP) (Kraft, 1985; Bock and Plitt, 1984; Biegler 1984; Betts 1998; von Stryk and Bulirsch 1992), which has been pushed by the progress in nonlinear optimization (Han, 1976; Powell, 1978; Barclay, Gill, and Rosen 1998; Betts 1998). Two basically different solution strategies for the reformulated problem exist (see Pytlak, 1999, for a survey):

(i) Sequential simulation and optimization:
In every iteration step of the optimization method, the model equations (7) are solved "exactly" by a numerical integration method for the current guess of control parameters. This method is also referred to as control vector parameterization.

(ii) Simultaneous simulation and optimization:
The discretized differential equations (7) enter the transcribed optimization problem as nonlinear constraints that can be violated during the optimization procedure. At the solution, however, they have to be satisfied.

Figure 4 outlines four particular methods which differ in the way the transcription is achieved. Collocation methods arise from general Galerkin type approaches by an appropriate choice of the approximation spaces and quadrature rules (see, e.g., Fletcher, 1984). Therefore, we will not comment any further on Galerkin type methods since the statements made for direct collocation apply as well for the more general Galerkin type methods. In this section we will only elaborate on direct single shooting, direct multiple shooting, and direct collocation.

Direct single shooting represents a pure sequential approach, whereas collocation is a pure simultaneous approach; direct multiple shooting may be considered a hybrid method, as the model equations are solved "exactly" only on intervals during the solution iterations.

5.1 Direct Single Shooting

In the direct single shooting method (e.g., Kraft 1985, 1994), the infinitely many degrees of freedom $\mathbf{u}(t)$ for $t \in I$ are reduced by a control parameterization $\tilde{\mathbf{u}}(t, \mathbf{q})$ that depends on a finite dimensional vector $\mathbf{q} \in \mathbb{R}^{n_q}$. The parameterization of the control can be based on general functions with local or global support or a mixture of both. An example based on a parameterization using functions with global support, e.g., a polynomial with N coefficients $\mathbf{q}_0, \dots, \mathbf{q}_{N-1}$, is given by

$$\tilde{\mathbf{u}}(t, \mathbf{q}_0, \dots, \mathbf{q}_{N-1}) := \sum_{i=0}^{N-1} \mathbf{q}_i t^i, \quad t \in I.$$

A second example (see Fig. 5) employing a localized parameterization is obtained using a piecewise constant control representation on a partition of the interval I into N subintervals I_i, $i = 0, 1, \dots, N-1$, such that

$$\tilde{\mathbf{u}}(t, \mathbf{q}_0, \dots, \mathbf{q}_{N-1}) := \mathbf{q}_i, \quad t \in I_i.$$

Besides these two explicit parameterizations of the controls one can also define controls implicitly via additional parameterized ODEs (or – if DAE

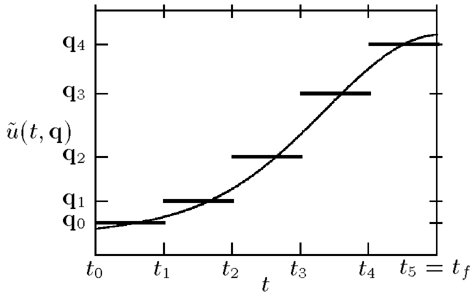

Fig. 5. Piecewise constant representation of a control ($N = 5$). The control intervals are given as $I_i = [t_i, t_{i+1}]$ for $i = 0, \ldots, 4$ with intermediate time points $t_1, \ldots, t_4$.

models are admissible – by additional algebraic equations containing the so called shape parameters), e.g.,

$$\begin{aligned} \dot{\tilde{\mathbf{u}}}(t; \mathbf{q}) &= \tilde{\mathbf{f}}\left(\mathbf{x}(t), \tilde{\mathbf{u}}(t; \mathbf{q}), t, \mathbf{q}\right), \quad t \in I \\ \tilde{\mathbf{u}}(t_0; \mathbf{q}) &= \tilde{\mathbf{u}}_0(\mathbf{q}). \end{aligned}$$

The additional equations can be added to the model equations Eq. (7). In this case, the parameterized controls $\tilde{\mathbf{u}}$ are reinterpreted as (parameter dependent) states.

Given an initial value $\mathbf{x}_0$ and a parameter vector $\mathbf{q}$, the following Initial Value Problem (IVP) can be solved:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \tilde{\mathbf{u}}(t, \mathbf{q}), t\right), \quad t \in I, \\ \mathbf{x}(t_0) &= \mathbf{x}_0. \end{aligned}$$

The solution of this problem is a trajectory $\mathbf{x}(t)$ which is a function of $\mathbf{q}$ only. To keep this dependency in mind we will denote this solution by $\tilde{\mathbf{x}}(t; \mathbf{q})$ in the following. By substituting this trajectory into the objective functional defined in (6) we can define the cost function $\tilde{J} : \mathbb{R}^{n_q} \to \mathbb{R}$ as

$$\tilde{J}(\mathbf{q}) := E\left(\tilde{\mathbf{x}}(t_f; \mathbf{q})\right) + \int_{t_0}^{t_f} L\left(\tilde{\mathbf{x}}(t; \mathbf{q}), \tilde{\mathbf{u}}(t, \mathbf{q}), t\right) \mathrm{d}t$$

In order to incorporate the path inequality constraints $\mathbf{c}$ into the NLP, different methods have been developed (cf. Vassiliadis, Sargent, and Pantelides 1994b). Two popular methods are

1. Introduction of a penalty term in the objective function:

$$\widehat{J}[\mathbf{u}(\cdot), \mathbf{x}(\cdot)] := J[\mathbf{u}(\cdot), \mathbf{x}(\cdot)] + \sum_{j=1}^{n_h} \kappa_j \cdot \int_{t_0}^{t_f} \Big(\max\big(0, -c_j(\cdot)\big)\Big)^2 \mathrm{d}t$$

   where $\kappa_j \in \mathbb{R}^+$, $j = 1, \ldots, n_h$ are large positive constants. A difficulty with the max operator is that it hides all information about a constraint as long as it is inactive, and that its smoothness is limited.

2. Using a time grid $t_0 < t_1 < \ldots < t_N = t_f$ the infinite dimensional path inequality constraints (9) are reformulated into $N+1$ vector inequality constraints

$$\mathbf{0} \leq \tilde{\mathbf{c}}_i(\mathbf{q}) := \mathbf{c}\left(\tilde{\mathbf{x}}(t_i;\mathbf{q}), \tilde{\mathbf{u}}(t_i,\mathbf{q}), t_i\right), \quad i = 0, \ldots, N\,.$$

By construction, this method enforces the path inequality constraints at the points on the time grid only. A sufficiently good approximation of the original constraint can be obtained by a sufficiently fine grid. Also a combination with the first method is possible.

In the sequel we adopt the second approach.

The endpoint constraint is similarly reformulated as

$$\mathbf{0} = \tilde{\mathbf{r}}(\mathbf{q}) := \mathbf{r}\left(\tilde{\mathbf{x}}(t_N;\mathbf{q})\right).$$

In summary, the finite dimensional NLP in the direct single shooting parameterization is given as

$$\min_{\mathbf{q}\in\mathbb{R}^{n_q}} \tilde{J}(\mathbf{q})$$

subject to

$$\begin{aligned} \mathbf{0} \leq \tilde{\mathbf{c}}_i(\mathbf{q}), \quad i = 0, \ldots, N, \\ \mathbf{0} = \tilde{\mathbf{r}}(\mathbf{q})\,. \end{aligned} \tag{14}$$

The numerical effort to solve the NLP (14) is determined to a large extent by the complexity of the parameterization of the control vector. Clearly, a piecewise constant parameterization with a uniform mesh length might not be the best for general problems such that adaptive parameterization schemes should be employed to resolve the trajectory at the right place. However, it is by no means trivial to generate such problem adapted meshes a-priori, i.e., before the actual optimal solution is known (see Waldraff et al. 1997, Betts and Huffmann 1998, Binder et al. 1998).

The solution of the NLP (14) requires sensitivity information of the states with respect to the control parameters $\mathbf{q}$. The computation of these sensitivities should be done according to the principle of *Internal Numerical Differentiation (IND)* (Bock, 1981) and *not* by trying to generate derivates by finite differences of independently computed approximations of the solution of disturbed initial value problems [4] . Many ODE and DAE solvers exist that can efficiently compute sensitivities according to the principle of IND; see, e.g., Caracotsios and Stewart (1985), Leis and Kramer (1985), Heim (1992), Buchauer, Hiltmann, and Kiehl (1994), Bock, Schlöder, and Schulz (1995), Maly and Petzold (1996), Kiehl (1998), Engl et al. (1998), Bauer (2000).

[4] This is also valid in the context of multiple shooting which we will introduce in the next section.

In many practical applications the problem functions have only low, local differentiability properties, i.e., discontinuities in the first or second order derivatives occur. In these cases, obtaining a useful gradient approximation is rather involved, since a numerical sensitivity analysis for initial value problems with switching points must be carried out, e.g., Rozenvasser (1967), Bock (1987), von Schwerin, Winckler, and Schulz (1996), Galán, Feehery, and Barton (1998).

5.2 Direct Multiple Shooting

In the direct multiple shooting method (Plitt, 1981; Bock and Plitt, 1984), the transcription of the optimal control problem (6)-(9) into an NLP starts similar to the direct (single) shooting method with a local control representation. First, the time horizon $I = [t_0, t_f]$ is divided into N subintervals $I_i := [t_i, t_{i+1}]$, $i = 0, 1, \ldots, N-1$, with $t_0 < t_1 < \ldots < t_N = t_f$. Then, the control trajectory is parameterized by a piecewise representation

$$\tilde{\mathbf{u}}_i(t, \mathbf{q}_i) \text{ for } t \in [t_i, t_{i+1}]$$

with N *local* control parameter vectors $\mathbf{q}_0, \mathbf{q}_1, \ldots \mathbf{q}_{N-1}$, $\mathbf{q}_i \in \mathbb{R}^{n_q}$. The trivial example for such a parameterization is again the piecewise constant representation shown in Fig. 5.

In a crucial second step, $N+1$ additional vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_N$ of the same dimension n_x as the system state are introduced, to which we will refer to as the multiple shooting *node values*. All but the last serve as initial values for N independent IVPs on the intervals I_i:

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{f}\left(\mathbf{x}_i(t), \tilde{\mathbf{u}}_i(t, \mathbf{q}_i), t\right), \quad t \in [t_i, t_{i+1}] \\ \mathbf{x}_i(t_i) &= \mathbf{s}_i. \end{aligned}$$

The solutions of these problems are N independent trajectories $\mathbf{x}_i(t)$ on $[t_i, t_{i+1}]$, which are a function of $\mathbf{s}_i$ and $\mathbf{q}_i$ only. To keep this dependency in mind, we will denote these solutions by $\tilde{\mathbf{x}}_i(t; \mathbf{s}_i, \mathbf{q}_i)$ in the following. For an illustration, see Fig. 6.

By substituting the independent trajectories $\tilde{\mathbf{x}}_i(t; \mathbf{s}_i, \mathbf{q}_i)$ into the Lagrange term L in Eq. (6) we can calculate the objective contributions $\tilde{J}_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_q} \to \mathbb{R}$ for $i = 0, \ldots, N-1$ as

$$\tilde{J}_i(\mathbf{s}_i, \mathbf{q}_i) := \int_{t_i}^{t_{i+1}} L\left(\tilde{\mathbf{x}}_i(t; \mathbf{s}_i, \mathbf{q}_i), \tilde{\mathbf{u}}_i(t, \mathbf{q}_i), t\right) \mathrm{d}\, t.$$

The decoupled IVPs are connected by matching conditions which require that each node value should equal the final value of the preceding trajectory:

$$\mathbf{s}_{i+1} = \tilde{\mathbf{x}}_i(t_{i+1}; \mathbf{s}_i, \mathbf{q}_i), \quad i = 0, \ldots, N-1. \tag{15}$$

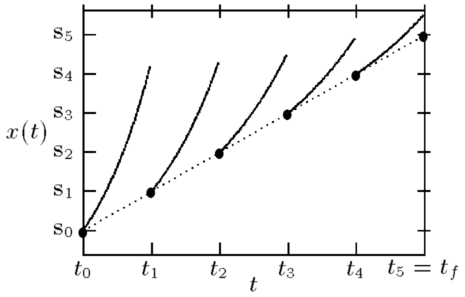

Fig. 6. Five trajectories in the multiple shooting parameterization ($N = 5$).

The first multiple shooting node variable $\mathbf{s}_0$ is required to be equal to the initial value $\mathbf{x}_0$ of the optimization problem:

$$\mathbf{s}_0 = \mathbf{x}_0. \tag{16}$$

Together, the constraints (15) and (16) remove the additional degrees of freedom which were introduced with the parameters $\mathbf{s}_i$, $i = 0, \ldots, N$. It is by no means necessary that the constraints (15) and (16) are satisfied *during* the optimization iterations – on the contrary, it is a crucial feature of the direct multiple shooting method that it can deal with *infeasible* initial guesses of the variables $\mathbf{s}_i$ and $\mathbf{q}_i$.

Using for notational convenience the same time grid as for the multiple shooting parameterization (finer or coarser grids are equally possible), the infinite dimensional path inequality constraints (9) are transcribed into $N+1$ vector inequality constraints

$$0 \leq \tilde{\mathbf{c}}_i(\mathbf{s}_i, \mathbf{q}_i) := \mathbf{c}\left(\mathbf{s}_i, \tilde{\mathbf{u}}_i(t_i, \mathbf{q}_i), t_i\right), \quad i = 0, \ldots, N.$$

Summarizing, the finite dimensional NLP in the direct multiple shooting parameterization is given as

$$\min_{\mathbf{s}_0, \ldots, \mathbf{s}_N, \mathbf{q}_0, \ldots, \mathbf{q}_{N-1}} E(\mathbf{s}_N) + \sum_{i=0}^{N-1} \tilde{J}_i(\mathbf{s}_i, \mathbf{q}_i)$$

subject to

$$\begin{aligned} \mathbf{s}_{i+1} &= \tilde{\mathbf{x}}_i(t_{i+1}; \mathbf{s}_i, \mathbf{q}_i), \quad i = 0, \ldots, N-1, \\ \mathbf{s}_0 &= \mathbf{x}_0, \\ 0 &\leq \tilde{\mathbf{c}}_i(\mathbf{s}_i, \mathbf{q}_i), \\ 0 &= \mathbf{r}(\mathbf{s}_N). \end{aligned}$$

An important feature of the direct multiple shooting method is the sparse structure of this large scale NLP. Its Lagrangian function $\mathcal{L}$ is *partially separable*, i.e., its Hessian matrix $\nabla^2_{\mathbf{s},\mathbf{q}}\mathcal{L}$ is block diagonal with non-zero blocks $\nabla^2_{\mathbf{s}_i,\mathbf{q}_i}\mathcal{L}$ that correspond to *local* variables $\mathbf{s}_i$, $\mathbf{q}_i$ only (Bock and Plitt, 1984).

Extensions of the Direct Multiple Shooting Method to treat DAE systems efficiently are described by Bock, Eich, Schlöder (1988), Schulz, Bock, Steinbach (1998), and Heim and von Stryk (1996) for the case of parameter estimation and for very general multistage optimal control problems by Leineweber (1999).

5.3 Direct Collocation

We now consider a general direct collocation discretization of the optimal control problem Eqs. (6)-(10). For ease of notation, we assume that the functional Eq. (6) is in Mayer form $J[\mathbf{u}, \mathbf{x}] = E(\mathbf{x}(t_f))$. This is no restriction of generality, as the transformation of the Bolza functional (6) to Mayer form is easily done: As a first step, an additional state $\mathbf{x}_{n_x+1}$ and an additional differential equation

$$\dot{\mathbf{x}}_{n_x+1}(t) = L(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}_{n_x+1}(t_0) := 0$$

are introduced. In the second step, the objective $E(\mathbf{x}(t_f))$ is redefined as $E(\mathbf{x}(t_f)) + \mathbf{x}_{n_x+1}(t_f)$ (in order to keep notation at a minimum, no new symbol for the *redefined* objective is introduced).

Both state and control variables are approximated by piecewise defined functions $\tilde{\mathbf{x}}(t;\cdot)$ and $\tilde{\mathbf{u}}(t;\cdot)$ on the time grid

$$t_0 < t_1 < \ldots < t_{N+1} = t_f .$$

Within each collocation interval $[t_i, t_{i+1}[$, $0 \leq i \leq N$, these functions are chosen as parameter dependent polynomials of order $k, l \in \mathbb{N}$ respectively:

$$\begin{aligned}
\tilde{\mathbf{x}}(t;\mathbf{s})\big|_{[t_i,t_{i+1}[} &:= \tilde{\mathbf{x}}_i(t;\mathbf{s}_i) := \pi_i^{\mathrm{x}}(t;\mathbf{s}_i) \in \boldsymbol{\Pi}_k^{n_x}, \\
\tilde{\mathbf{u}}(t;\mathbf{q})\big|_{[t_i,t_{i+1}[} &:= \tilde{\mathbf{u}}_i(t;\mathbf{q}_i) := \pi_i^{\mathrm{u}}(t;\mathbf{q}_i) \in \boldsymbol{\Pi}_l^{n_c} .
\end{aligned}$$

Here, $\boldsymbol{\Pi}_\mu^\nu$ denotes the space of ν-dimensional vectors of polynomials up to degree μ. The coefficients of the polynomials (*shape parameters*) are collected in the vectors

$$\begin{aligned}
\mathbf{s} &:= (\mathbf{s}_0^T, \ldots, \mathbf{s}_N^T)^T \in \mathbb{R}^{N\cdot(k+1)\cdot n_x}, \quad \mathbf{s}_i \in \mathbb{R}^{(k+1)\cdot n_x},\ i = 0, \ldots, N, \\
\mathbf{q} &:= (\mathbf{q}_0^T, \ldots, \mathbf{q}_N^T)^T \in \mathbb{R}^{N\cdot(l+1)\cdot n_c}, \quad \mathbf{q}_i \in \mathbb{R}^{(l+1)\cdot n_c},\ i = 0, \ldots, N .
\end{aligned}$$

Matching conditions of the form

$$\pi_i(t_{i+1}^-, \cdot) = \pi_{i+1}(t_{i+1}^+, \cdot), \quad i = 0, \ldots, N-1$$

have to be imposed at the boundaries of the subintervals to enforce continuity of the approximating functions in $[t_0, t_f]$. Additionally, higher order differentiability may be imposed by

$$\frac{d^\kappa}{dt^\kappa}\pi_i(t_{i+1}^-, \cdot) = \frac{d^\kappa}{dt^\kappa}\pi_{i+1}(t_{i+1}^+, \cdot), \quad \begin{cases} \kappa = 1, \ldots, J \\ i = 0, \ldots, N-1 \end{cases}$$

where J denotes the desired order of differentiability.

In order to formulate a nonlinear optimization problem, the model equations and the continuous constraints are explicitly discretized:

1. The model equations (7) are only to be satisfied at the *collocation points* $t_{i\,\mu}$, $\mu = 1, \ldots, M$, within each subinterval $[t_i, t_{i+1}[$, $i = 0, \ldots, N-1$, and $[t_N, t_{N+1}]$:
$$\begin{aligned} t_i \le t_{i0} < \ldots < t_{iM} < t_{i+1}, &\qquad i = 0, \ldots, N-1 \\ t_N \le t_{N0} < \ldots < t_{NM} \le t_{N+1}, & \end{aligned}$$
2. The inequality constraints $\mathbf{c}(\cdot)$ are sampled on a second grid within $[t_0, t_f]$:
$$t_0 \le t_1^c < \ldots < t_L^c \le t_f$$

Altogether, this leads to the formulation of the discretized optimal control problem derived from (6)-(10) (in Mayer form) by collocation:

$$\min_{\mathbf{s},\mathbf{q}} \tilde{E}(\mathbf{s}) = E\left(\tilde{\mathbf{x}}(t_f; \mathbf{s})\right) \tag{17}$$

subject to the nonlinear (point) constraints

$$\mathbf{f}\left(\tilde{\mathbf{x}}(t_{il}; \mathbf{s}), \tilde{\mathbf{u}}(t_{il}; \mathbf{q}), t\right) - \dot{\tilde{\mathbf{x}}}(t_{il}; \mathbf{s}) = \mathbf{0}, \quad \begin{cases} i = 0, \ldots, N \\ l = 0, \ldots, M \end{cases} \tag{18}$$

$$\mathbf{c}\left(\tilde{\mathbf{x}}(t_\gamma^c; \mathbf{s}), \tilde{\mathbf{u}}(t_\gamma^c; \mathbf{q}), t_\gamma^c\right) \ge \mathbf{0}, \quad \gamma = 1, \ldots, L \tag{19}$$

$$\tilde{\mathbf{x}}(t_0; \mathbf{s}) - \mathbf{x}_0 = \mathbf{0}, \tag{20}$$

$$\mathbf{r}\left(\tilde{\mathbf{x}}(t_f; \mathbf{s})\right) = \mathbf{0}. \tag{21}$$

If the solution is restricted to (higher order) continuously differentiable state and control variables, the matching conditions have to be fulfilled additionally:

$$\frac{d^\kappa}{dt^\kappa}\pi_i^{\mathrm{x}}(t_{i+1}^-; \mathbf{s}_i) - \frac{d^\kappa}{dt^\kappa}\pi_{i+1}^{\mathrm{x}}(t_{i+1}^+; \mathbf{s}_{i+1}) = 0\ , \quad \begin{cases} \kappa = 0, \ldots, J_{\mathrm{s}} \\ i = 0, \ldots, N-1 \end{cases} \tag{22a}$$

$$\frac{d^\kappa}{dt^\kappa}\pi_i^{\mathrm{u}}(t_{i+1}^-; \mathbf{q}_i) - \frac{d^\kappa}{dt^\kappa}\pi_{i+1}^{\mathrm{u}}(t_{i+1}^+; \mathbf{q}_{i+1}) = 0\ , \quad \begin{cases} \kappa = 0, \ldots, J_{\mathrm{c}} \\ i = 0, \ldots, N-1 \end{cases} \tag{22b}$$

where J_{s} is the order of differentiability in the state variables and J_{c} is the order of differentiability in the control variables.

The constrained nonlinear optimization problem Eqs. (17), (18)-(20), (22a)-(22b) can be efficiently solved using SQP algorithms that will be discussed in Section 5.4. SQP methods are based on the availability of gradient information. This gradient information can be obtained very easily, e.g.,

$$\frac{d}{d\mathbf{s}_i}(\mathbf{f} - \dot{\tilde{\mathbf{x}}})\Big|_{t=t_{il};\mathbf{s},\mathbf{q}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \cdot \frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{s}_i}(t_{il}; \mathbf{s}_i) - \frac{\partial \dot{\tilde{\mathbf{x}}}_i}{\partial \mathbf{s}_i}(t_{il}; \mathbf{s}_i).$$

Due to the full discretization of both control and state space, the NLPs generated by direct collocation tend to become very large for practically interesting problems. Thus, special care has to be taken in the implementation of a collocation algorithm to account for the special structure and the high sparsity of the Jacobian of the constraints Eqs. (18)-(20), (22a)-(22b); see, e.g., Betts (1998), von Stryk (2000).

5.4 Numerical Solution of the NLP by Sequential Quadratic Programming

Sequential quadratic programming (SQP)(Han, 1976; Powell, 1978) is a very efficient iterative method for the solution of NLP arising from the discretization of optimal control problems by direct transcription methods as described above. Now, let $\boldsymbol{\xi}$ be the set of *parameters* introduced by the discretization of an infinite dimensional optimal control problem. In each SQP iteration a current guess of the optimal set of *optimization variables* $\boldsymbol{\xi}^*$ is improved by the solution of a quadratic subproblem derived from a quadratic approximation of the Lagrangian of the NLP subject to the linearized constraints (for a description see, e.g., Barclay, Gill, and Rosen 1998; Gill, Murray, and Saunders 1997).

In the sequel, we consider an NLP of the form

$$\min_{\boldsymbol{\xi}} \varphi(\boldsymbol{\xi}) \qquad (23)$$
$$\text{subject to} \quad \mathbf{a}(\boldsymbol{\xi}) = 0, \quad \mathbf{b}(\boldsymbol{\xi}) \geq 0,$$

and their solution by SQP methods equipped with a relaxation strategy based on line search.

For the class of SQP methods considered, the vector of optimization variables $\boldsymbol{\xi}_k \in \mathbb{R}^{n_y}$ itself and the vector of multipliers $\mathbf{v}_k := (\boldsymbol{\mu}, \boldsymbol{\sigma})_k \in \mathbb{R}^{n_a+n_b}$ are changed from (the major SQP) iteration number k to iteration number $k+1$ by

$$\begin{pmatrix} \boldsymbol{\xi}_{k+1} \\ \mathbf{v}_{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\xi}_k \\ \mathbf{v}_k \end{pmatrix} + \alpha_k \begin{pmatrix} \mathbf{d}_k \\ \mathbf{u}_k - \mathbf{v}_k \end{pmatrix}, \quad k = 0, 1, 2, \ldots$$

where the *search direction* $(\mathbf{d}_k, \mathbf{u}_k)$ is obtained as the solution of a linearly constrained quadratic problem (QP) resulting from a quadratic approximation of the Lagrangian $\mathcal{L}$

$$\mathcal{L}(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) := \varphi(\boldsymbol{\xi}) - \sum_{i=1}^{n_a} \mu_i a_i(\boldsymbol{\xi}) - \sum_{j=1}^{n_b} \sigma_j b_j(\boldsymbol{\xi}), \quad \boldsymbol{\mu} \in \mathbb{R}^{n_a}, \quad \boldsymbol{\sigma} \in \mathbb{R}^{n_b} :$$

$$\begin{aligned} &\min_{\mathbf{d}\in\mathbb{R}^{n_y}} \frac{1}{2}\mathbf{d}^T\mathbf{C}_k\mathbf{d} + \nabla\varphi(\boldsymbol{\xi}_k)^T\mathbf{d} \qquad (24) \\ \text{subject to} \quad &\nabla a_i(\boldsymbol{\xi}_k)^T\mathbf{d} + a_i(\boldsymbol{\xi}_k) = 0,\ i = 1,\dots,n_a, \\ &\nabla b_j(\boldsymbol{\xi}_k)^T\mathbf{d} + b_j(\boldsymbol{\xi}_k) \geq 0,\ j = 1,\dots,n_b. \end{aligned}$$

Usually, $\mathbf{C}_k$ is a positive definite approximation of the Hessian $\mathbf{H}_k$ of the Lagrangian $\mathcal{L}(\boldsymbol{\xi}_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$. The search direction $\mathbf{d}_k$ is the solution of the QP (24) and $\mathbf{u}_k$ is the corresponding multiplier. The quadratic (sub-)problem Eqs. (24) itself is solved by an iterative method (usually, an active set strategy or an interior point method is employed).

The *step size* $\alpha_k \in \mathbb{R}$ is obtained by a (approximate) one-dimensional minimization of a merit function (*line search*)

$$\psi_r\left(\begin{pmatrix}\boldsymbol{\xi}\\ \mathbf{v}\end{pmatrix} + \alpha\begin{pmatrix}\mathbf{d}\\ \mathbf{u}-\mathbf{v}\end{pmatrix}\right)$$

with respect to α. A suitable merit function is, e.g., the Lagrangian augmented by penalty terms (*augmented Lagrangian*, e.g., Gill, Murray, Saunders, and Wright (1992))

$$\begin{aligned} \psi_r(\boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \varphi(\boldsymbol{\xi}) &- \sum_{i=1}^{n_a}\left(\mu_i a_i(\boldsymbol{\xi}) - \frac{1}{2}r_i a_i^2(\boldsymbol{\xi})\right) \\ &- \sum_{j\in J}\left(\sigma_j b_j(\boldsymbol{\xi}) - \frac{1}{2}r_{n_a+j} b_j^2(\boldsymbol{\xi})\right) - \frac{1}{2}\sum_{j\in K}\frac{\sigma_j^2}{r_{n_a+j}}. \end{aligned}$$

The index sets J and K are chosen according to $J = \{j | 1 \leq j \leq n_b, b_j(y) \geq \sigma_j / r_{n_a+j}\}$, $K = \{1,\dots,n_b\} \backslash J$, where $r_i > 0$, $i = 1,\dots,n_a+n_b$.

A widely used and robust general-purpose line search based SQP method is NPSOL (Gill, Murray, Saunders, and Wright 1998), which is suitable for small to medium sized problems. The new, sparse SQP method SNOPT is a successor of NPSOL and one of the most advanced, efficient and robust, general-purpose SQP methods currently available for large-scale problems (Gill, Murray, and Saunders 1997; Gould and Toint 1999).

A discussion of other SQP methods, e.g., of the trust-region method, can be found in Gould and Toint (1999) or in Nocedal and Wright (1999).

5.5 Comparison of Direct Methods

We will try to develop the advantages and disadvantages of the previously described three methods – direct single shooting, direct multiple shooting, and direct collocation – when the resulting NLPs are solved by appropriately designed SQP methods. A brief summary of this discussion is given in Table 7. Additional background information and a broad list of references regarding direct methods can be found, e.g., in von Stryk (2000).

Direct (single) shooting

- In each major SQP iteration an initial value problem is numerically solved with high solution accuracy (even though the controls may be far from from their optimal solution values).
- Possible use of existing dynamic simulation facilities (Engl, Kröner, Kronseder, and von Stryk 1998) can increase the confidence of users not deeply familiar with optimization techniques.
- Use of efficient state-of-the-art ODE and DAE solvers allows to profit from recent developments in the field.
- Small size of NLP facilitates the use of off-the-shelf NLP/QP solvers.
- Only initial guesses for the control parameters (and if free, for the initial values) are needed.
- For highly unstable systems (i.e., initial value problems with a strong dependence on the initial values) the optimization algorithm inherits the ill-conditioning of the initial value problem, even if the optimization problem itself is well-conditioned (this well-conditioning may, e.g., be due to end point constraints or an objective function penalizing trajectory deviations as, e.g., in tracking/estimation problems).
- The dynamic model is fulfilled during all SQP iterations (up to integrator accuracy), so that in time critical cases a premature stop with a physical system trajectory is possible. However, state and end point constraints (9), (10) may still be violated – roughly spoken, they have only second priority in the single shooting formulation.
- If the initial value is fixed (as in the optimal control problem (4), but not in the estimation problem (5)), the number of derivatives corresponds to the number of control parameters – this may limit the numerical effort very efficiently for large scale systems with few control parameters.
- The single shooting algorithm can, e.g., be found in the software packages gOPT (Process Systems Enterprise, 2000), DYNOPT (Abel et al., 1999), OPTISIM (Engl et al., 1998, Kröner et al., 2001). These packages have been successfully applied to solve large scale industrial problems.

Direct multiple shooting

- Similar to single shooting, the underlying initial value problems are numerically solved with prespecified accuracy in each SQP iteration.
- Use of existing dynamic simulation facilities, and of efficient state-of-the-art DAE solvers is possible, as for single shooting.
- The relatively large number of variables requires specially tailored NLP/QP algorithms. On the other hand, the structure can be exploited to yield even faster convergence than for direct single shooting ("high rank updates", Bock and Plitt, 1984), which is especially useful in the case of long horizons with many control parameters. For the QP solution, recursive schemes allow to reduce the linear algebra effort to essentially

the same as for single shooting ("condensing", Bock and Plitt, 1984). Alternatively, an efficient QP solution based on dynamic programming (Steinbach, 1995) is possible which is *linear* in the number of control intervals.
– Initial guesses for the whole state trajectory are needed. This is an advantage, if a-priori knowledge about the state trajectory is available, as, e.g., in tracking problems, where it can damp the influence of poor initial guesses for the controls (which are usually much less known).
– The optimization of highly unstable or even chaotic systems can be possible (cf. Baake et al. 1992; Kallrath et al. 1993). A detailed numerical stability analysis for the case of parameter estimation is given by Bock (1987).
– The method is well suited for parallel computation, since the IVP solutions and derivative computations are decoupled on different multiple shooting intervals (Gallitzendörfer and Bock, 1994).
– The continuity (Eq. (15)) of the system trajectory is only fulfilled after successful termination of the SQP solution procedure (up to the solution tolerance). At premature stops, both, continuity conditions (15) and state and end point constraints (9), (10) may be equally violated.
– An implementation of the multiple shooting method is found, e.g, within the highly advanced optimal control package MUSCOD-II (Leineweber, 1999), or in Petzold et al. (1997).

Direct collocation

– The ODE simulation (7) and the control optimization problems (6)-(10) are solved simultaneously, which leads to potentially faster computations compared to shooting techniques.
– Existing dynamic simulation facilities and DAE solvers cannot be reused directly.
– The very large number of variables requires tailored NLP/QP algorithms. On the other hand, similar as for the direct multiple shooting method, a careful exploitation of the structure can lead to excellent convergence behaviour and very efficient QP solutions. Furthermore, sparsity can be exploited at all levels.
– As for multiple shooting, initial guesses for the whole state trajectory are needed, which may be an advantage, if a-priori knowledge about the state trajectory is available.
– The optimization of highly unstable systems is also possible.
– The discretized DAE model equations (7) are only fulfilled after succesful termination of the SQP solution procedure (up to the solution tolerance). At premature stops, all constraints (18)-(22b) are equally violated.
– A reliable estimation of the adjoint variables is available on the entire state variable discretisation grid. Moreover, the estimates are also valid along arcs with active state constraints. The estimation of the adjoint

	Direct Single Shooting	Direct Multiple Shooting	Direct Collocation
general solution approach	sequential	hybrid	simultaneous
use of (state of the art) DAE solvers	yes	yes	no
number of variables / size of NLP	small	intermediate	large
initial guess for system states	initial state	all node values	all node values
applicable to highly unstable sytems	no	yes	yes
DAE model fullfilled in each iteration step	yes	partially	no

Fig. 7. Comparison of direct methods.

variables from the Lagrange multipliers at the solution of the NLP corresponding to the infinite dimensional optimal control problem has been described, e.g., in von Stryk (1993) for the case without state constraints and in von Stryk (1995) for problems including state constraints.

In this way, collocation can be used within a *hybrid* approach (von Stryk and Bulirsch, 1992) to provide information required for a highly accurate indirect multiple shooting method (see Section 4.2), i.e., good start estimates for all optimal trajectories *including* the adjoint states (e.g., Bulirsch et al., 1993), as well as for the switching structure (e.g., von Stryk and Schlemmer 1994).

– Highly advanced collocation algorithms have been implemented by Betts and Huffmann (1998) (SOCS), Cervantes and Biegler (1998), Schulz, (1996, 1998) (OCPRSQP), and von Stryk 2000 (DIRCOL).

The development of these pieces of software has been facilitated by the advent of new optimization methods which allow the solution of very large scale NLP. [5]

[5] OCPRSQP uses a partially reduced SQP method.
DIRCOL employs SNOPT (Gill et al., 1997). SNOPT approximates the Hessian of the NLP Lagrangian by limited-memory quasi-Newton updates and uses a reduced Hessian algorithm for solving the QP subproblems. The null-space matrix of the working set in each iteration is obtained from a sparse LU factorization.
In the code SOCS of Betts and Frank (1994) a Schur-complement QP method is implemented instead of a reduced-Hessian QP method.

6 Optimization Techniques on Moving Horizons

When a sequence of moving horizon optimization problems is solved on-line, several questions regarding the employed numerical algorithm arise:

- Can the solution of each optimization problem be computed in a time ΔT that is known a-priori?
- If not so, what are suitable approximations of the feedback control that can be used instead?
- What can in advance be computed off-line, what has necessarily to be done on-line?
- How can the similarity of subsequent optimization problems be exploited to reduce computation times?

As the approaches to address these questions vary broadly and are not easily classified, we will here only mention some classical approaches which we consider a useful basis for understanding current developments. We explicitly encourage the reader to consult the research articles of Binder et al., Kronseder et al. and Diehl et al. in this book for some recent approaches.

Before briefly introducing some classical approaches, let us first go a step backwards and formulate what is the aim of numerical moving horizon optimization algorithms. Further, we will distinguish between those problem specific data that are known a priori, and those that are only available online. For simplicity, we will here only treat the optimal control problem (4). In Subsection 6.1, however, we will briefly address the estimation problem (5) for linear systems and introduce the Kalman filter algorithm.

Optimal moving horizon feedback control

Let us recall that the task of on-line optimization on moving horizons is to compute an open-loop control $\mathbf{u}(t|I_k^r)$ for all $t \in I_k^r$. Only the first part on the time interval $t \in [t_{0,k}^r, t_{0,k}^r + \Delta T]$ is applied to the process. In the limit of negligible computation times the sampling time ΔT could be set to zero, so that the only essentially needed output of the algorithm is the first value of the open-loop control, i.e., the vector $\mathbf{u}^r(t_{0,k}|I_k^r) \in \mathbb{R}^{n_u}$.

On the other hand, what data are necessary to specify the k-th optimal control problem (4)? First, a DAE model, constraint functions and an objective functional have to be given a priori – however, some of the model parameters $\mathbf{p}^r$ and similarly the disturbance prediction $\mathbf{w}^r(t), t \in I_k^r$ may not be known before the process runs. In practice, we have to provide in advance a disturbance model that provides explicitly the predicted disturbance trajectory $\mathbf{w}^r(t)$, depending on some additional parameters. We will assume that this parameterized disturbance model is contained in the model equations, and that the vector $\mathbf{p}^r$ of a priori unknown parameters is suitably enlarged. Secondly, the objective function, or more precisely, the reference trajectory $\boldsymbol{\eta}^r(t), t \in I_k^r$, may be changed during process operation – e.g., due to a change in the desired operating point. Again, we have to assume that a parameterization of all possible reference trajectories $\boldsymbol{\eta}^r(t)$ exists, and that the additional parameters are again added to the general parameter vector $\mathbf{p}^r$.

Thus, the only quantities that are on-line inputs to our optimization algorithm are

- the parameter vector $\mathbf{p}_k^r$,
- the initial value $\mathbf{x}_{0,k}^r$, and
- the starting time $t_{0,k}^r$.

In summary, the purpose of idealized on-line optimization on moving horizons is to compute the *optimal moving horizon feedback control* function, that we define as follows:

$$\begin{aligned} \mathbf{u} : D \subset \mathbb{R}^{n_p} \times \mathbb{R}^{n_x} \times \mathbb{R} &\rightarrow \mathbb{R}^{n_u} \\ (\mathbf{p}_k^r, \mathbf{x}_{0,k}^r, t_{0,k}^r) &\mapsto \mathbf{u}(\mathbf{p}_k^r, \mathbf{x}_{0,k}^r, t_{0,k}^r) := \mathbf{u}^r(t_{0,k}^r | I_k^r), \end{aligned} \tag{25}$$

where we have introduced the bounded domain $D \subset \mathbb{R}^{n_p} \times \mathbb{R}^{n_x} \times \mathbb{R}$ to account for the fact that all inputs are expected to vary in a finite range only. Note that the control vector $\mathbf{u}(\mathbf{p}_k^r, \mathbf{x}_{0,k}^r, t_{0,k}^r)$ is computed as the first value of an open-loop optimal control, but that the idea of optimal moving horizon feedback control is to apply exactly this value to the real system. If $\mathbf{p}_k^r, \mathbf{x}_{0,k}^r$ were directly accessible (and not the result of on-line estimation), the optimal moving horizon feedback control function alone would define the closed-loop system behaviour.

In principle, this function could be precalculated off-line on a sufficiently fine grid on its domain D, thus eliminating the need for any on-line calculations. In practice, even for moderate state and parameter dimensions n_p and n_x, the necessary off-line calculation time and the storage requirements would be excessive, thus creating the need for on-line optimization.

For notational convenience, we go back to the problem formulation (6)-(10) introduced at the beginning of Section 4, and therefore omit the parameters $\mathbf{p}_k^r$ in the rest of this section. In the presented framework they can be treated in the same way as the initial values $\mathbf{x}_{0,k}^r$.

Time Dependence of Moving Horizon Problems

We can divide the possible moving horizon problem formulations into three major classes:

Finite Moving Horizon Problems: In this class of problem, the initial and the final time of the horizon move simultaneously, i.e., the horizon length $T = t_{f,k}^r - t_{0,k}^r$ is constant for all k. If the model equations and objective function are time independent, the output of the optimization algorithm looses its direct dependence on $t_{0,k}^r$. This can be exploited in the numerical solution of subsequent problems.

Shrinking Horizon Problems: This class comprises problems with a finite horizon length $t^r_{f,k} - t^r_{0,k}$ which is typically decreasing with growing k. Two cases are distinguished:

a) Fixed end time problems, where $t^r_{f,k} = t^r_{f,k-1} = t^r_f$. This may, e.g., occur in batch processes with a prespecified delivery time. Even when the system model and objective are time independent, the optimal control problems differ in the horizon length $t_{f,k} - t_{0,0}$, so that the resulting feedback control $\mathbf{u}(t^r_{0,k}|I^r_k)$ usually has a time dependence.
b) Open end time problems, which leave the final time $t^r_{f,k}$ as a degree of freedom of the optimization (or restrict them by a state dependent constraint). This may occur, e.g., in batch processes that should stop when the product or conversion specifications are attained. This formulation leads again to a *time independent* feedback control, if the system model and objective is time invariant.

Infinite Horizon Problems: Though so far not numerically tractable for general systems, it is worth mentioning here that they again lead to time independent control laws if the problem formulation is time invariant. This can so far only be exploited in the linear quadratic regulator problem investigated in Subsection 6.2.

Unconstrained optimal control problems for linear systems with quadratic cost can be solved very elegantly by dynamic programming techniques that will be reviewed in the first two subsections for both, the estimation and the regulator problem. It will be seen that subsequent problems can essentially be solved with negligible computational cost. Many textbooks consider this topic in far greater detail (see, e.g., Anderson and Moore, 1989). Let us first treat the estimation problem.

6.1 Linear Quadratic Estimation problem

The typical problem formulation of such systems is given by

$$\begin{aligned}
\min_{\mathbf{x}^e(\cdot),\mathbf{y}^e(\cdot),\mathbf{w}^e(\cdot)} \quad & \tfrac{1}{2}\left(\mathbf{x}^e(t^e_{0,k}) - \bar{\mathbf{x}}^e_{0,k}\right)^T \mathbf{E}^e \left(\mathbf{x}^e(t^e_{0,k}) - \bar{\mathbf{x}}^e_{0,k}\right) \\
& + \tfrac{1}{2}\int_{t^e_{0,k}}^{t^e_{f,k}} (\boldsymbol{\eta}^e(\tau) - \mathbf{y}^e(\tau))^T \mathbf{Q}^e (\boldsymbol{\eta}^e(\tau) - \mathbf{y}^e(\tau)) + \mathbf{w}^e(\tau)^T \mathbf{R}^e \mathbf{w}^e(\tau) d\tau \\
\text{s.t.} \quad & \dot{\mathbf{x}}^e(t) = \mathbf{A}\mathbf{x}^e(t) + \mathbf{B}\mathbf{u}^e(t) + \mathbf{w}^e(t), \quad \forall\, t \in I^e_k, \\
& \mathbf{y}^e(t) = \mathbf{C}\mathbf{x}^e(t), \quad \forall\, t \in I^e_k.
\end{aligned} \tag{26}$$

The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ reflect the time-invariant model matrices and $\mathbf{E}^e, \mathbf{Q}^e, \mathbf{R}^e$ are time-invariant positive semi-definite weighting matrices. $\bar{\mathbf{x}}^e_{0,k}$ refers to a reference value of the initial state. The control $\mathbf{u}^e(t)$ is assumed to be known.

Since we have a linear quadratic problem the optimal solution $\mathbf{x}^e(t|I_k^e)$ of (26) can also be written as (Kailath, 1980)

$$\mathbf{x}^e(t) = \boldsymbol{\xi}^e(t) + \mathbf{P}^e(t)\boldsymbol{\lambda}^e(t),\ t \in I_k^e, \tag{27}$$

where $\mathbf{P}^e(t)$ and $\boldsymbol{\lambda}^e(t), \boldsymbol{\xi}^e(t)$ denote differentiable time dependent matrix and vector functions, respectively. Explicit equations for $\mathbf{P}^e(t)$, $\boldsymbol{\lambda}^e(t)$, and $\boldsymbol{\xi}^e(t)$ can be derived exploring the necessary optimality conditions commonly referred to as Euler-Lagrange Equations, which have been discussed in Section 4.2 (cf. Eqs. (12) and (13), and more specifically Kailath, 1980):

$$\dot{\mathbf{P}}^e(t) = \mathbf{A}\mathbf{P}^e(t) + \mathbf{P}^e(t)\mathbf{A}^T + \mathbf{R}^{e-1} - \mathbf{P}^e(t)\mathbf{C}^T\mathbf{Q}^e\mathbf{C}\mathbf{P}^e(t), \quad t \in I_k^e, \tag{28}$$

$$\dot{\boldsymbol{\lambda}}^e(t) = (\mathbf{C}^T\mathbf{Q}^e\mathbf{C}\mathbf{P}^e(t) - \mathbf{A}^T)\boldsymbol{\lambda}^e(t) - \mathbf{C}^T\mathbf{Q}^e(\boldsymbol{\eta}^e(t) - \mathbf{C}\boldsymbol{\xi}^e(t)),\ t \in I_k^e, \tag{29}$$

$$\dot{\boldsymbol{\xi}}^e(t) = \mathbf{A}\boldsymbol{\xi}^e(t) + \mathbf{B}\mathbf{u}^e(t) + \mathbf{P}^e(t)\mathbf{C}^T\mathbf{Q}(\boldsymbol{\eta}^e(t) - \mathbf{C}\boldsymbol{\xi}^e(t)), \quad t \in I_k^e. \tag{30}$$

The initial conditions arise from transversality conditions and are given by

$$\mathbf{P}^e(t_{0,k}) = \mathbf{E}^{e-1}, \tag{31}$$

$$\boldsymbol{\lambda}^e(t_{0,k}) = \mathbf{0}, \tag{32}$$

$$\boldsymbol{\xi}^e(t_{0,k}) = \bar{\mathbf{x}}_{0,k}^e. \tag{33}$$

Equation (28) is commonly referred to as the matrix Riccati equation, (29) is the governing equation for the dual variable $\boldsymbol{\lambda}^e(t)$ and (30) denotes a filter equation which will be further discussed at the end of this section. Note that initial conditions (31) and (33) are specified at $t_{0,k}^e$ while (32) is a final condition at $t_{f,k}^e$. $\boldsymbol{\xi}^e(t)$ and $\mathbf{P}^e(t)$ can be solved by forward integration using the derived initial conditions. If the trajectories for $\mathbf{P}^e, \boldsymbol{\xi}^e$ are available, $\boldsymbol{\lambda}^e(t)$ can be computed by an integration backwards in time starting at $t_{f,k}^e$. Thus forward and backward integration are necessary to determine $\mathbf{x}^e(t|I_k),\ t \in I_k^e$. However, the integration of (29) becomes unnecessary if only the end value $\mathbf{x}(t_{f,k}|I_k^e)$ is of interest. It refers to a filtered state estimate using all preceeding data collected in the interval I_k^e. On the other hand, the estimates $\mathbf{x}^e(t|I_k^e)$, $t_{0,k}^e \leq t < t_{f,k}^e$ in the interior of I_k^e, which require a backward integration of (29), are referred to as smoothed states.

The Kalman Filter

The solution of (26) has been outlined using a deterministic problem formulation. The equations can also be derived using a stochastic approach. Then, $\mathbf{w}^e(t)$ and $\mathbf{v}^e(t) := \boldsymbol{\eta}^e(t) - \mathbf{y}^e(t)$ are assumed to follow an uncorrelated zero-mean Gaussian statistic with covariances $E\{\mathbf{w}^e(t)\mathbf{w}^{eT}(\tau)\} = \mathbf{R}^{e-1}\delta(t-\tau)$ and $E\{\mathbf{v}^e(t)\mathbf{v}^{eT}(\tau)\} = \mathbf{Q}^{e-1}\delta(t-\tau)$ where δ denotes the Dirac distribution and E is the expected value. Furthermore, let $E\{(\mathbf{x}^e(t_{0,k}^e)\} = \bar{\mathbf{x}}_{0,k}^e$ and $E\{(\mathbf{x}^e(t_{0,k}^e) - \bar{\mathbf{x}}_{0,k}^e)(\mathbf{x}^e(t_{0,k}^e) - \bar{\mathbf{x}}_{0,k})^T\} = \mathbf{E}^{e-1}$. Then, (26) defines a maximum likelihood problem. Bias free estimates of minimal variance are obtained and

$\mathbf{P}^e$ can be interpreted to be the covariance matrix of the state estimation error. However, one should be aware that the statistical assumptions might not be justified in practical applications. The problem (26) based on a statistical formulation was originally formulated and solved by Kalman (1960).

If subsequent estimation problems differ only by an increasing end time, i.e., if $t^e_{f,k} > t^e_{f,k-1}$, but $t^e_{0,k} = t^e_{0,k-1} (= t_{0,0})$, and if only the filtered state estimates $\mathbf{x}^e(t_{f,k}|I^e_k)$ are of interest, a solution can be obtained efficiently as follows: Starting with the end values $\mathbf{P}^e(t^e_{f,k-1})$ and $\boldsymbol{\xi}^e(t^e_{f,k-1})$ of the previous problem, Eqs. (28) and (30) have to be integrated on the appended part $[t^e_{f,k-1}, t^e_{f,k}]$ of the interval only. The end value $\boldsymbol{\xi}^e(t^e_{f,k})$ provides already the new filtered state estimate because of Eq. (32) and (27) evaluated at $t^e_{f,k}$. In fact, the integration of (28) and (30) can be performed simultaneously with the data acquisition, providing a continuous stream of filtered state estimates. Equation (30) is commonly referred to as the Kalman filter equation for continuous problems where $\mathbf{K}^e(t) := \mathbf{P}^e(t)\mathbf{C}^{eT}\mathbf{Q}^e$ denotes the filter gain.

For $t \to \infty$, the matrix $\mathbf{P}^e(t)$ approaches a constant steady state $\bar{\mathbf{P}}^e$ that can be calculated a priori as the solution of the algebraic Riccati equation that is obtained by setting $\dot{\mathbf{P}}^e(t) = \mathbf{0}$ in Eq. (28). In this case, the (relatively expensive) integration of the matrix Riccati equation (28) can be omitted, and only a constant gain matrix $\bar{\mathbf{K}}^e = \bar{\mathbf{P}}^e\mathbf{C}^{eT}\mathbf{Q}^e$ has to be kept for use in the Kalman filter equation (30).

The Extended Kalman Filter (EKF)

The Kalman filter algorithm for linear systems can be extended to non-linear systems to obtain a heuristic algorithm that is known as the *Extended Kalman Filter*. Though successful in practical applications, this algorithm does neither provide a solution to a general non-linear optimization problem of a similar form as (26), nor does it have a statistical interpretation. However, for a discrete-time system and a horizon length of one time step, the extended Kalman-filter can be related to moving horizon estimation (Robertson et al., 1996).

The extension of the Kalman filter equations is as follows: Assuming a nonlinear ODE system with outputs

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right),\\ \mathbf{y}(t) &= \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right),\end{aligned}$$

the matrix Riccati and Kalman filter equations (28) and (30) can be generalized to obtain nonlinear analogues: Eq. (28) can directly be used with the substitutions

$$\begin{aligned}\mathbf{A}(t) &:= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\left(\boldsymbol{\xi}(t), \mathbf{u}(t)\right),\\ \mathbf{C}(t) &:= \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\left(\boldsymbol{\xi}(t), \mathbf{u}(t)\right),\end{aligned}$$

and Eq. (30) is modified to

$$\dot{\boldsymbol{\xi}}^e(t) = \mathbf{f}\left(\boldsymbol{\xi}^e(t), \mathbf{u}(t)\right) + \mathbf{P}^e(t)\mathbf{C}(t)^T \mathbf{Q}\left(\boldsymbol{\eta}^e(t) - \mathbf{h}\left(\boldsymbol{\xi}^e(t), \mathbf{u}(t)\right)\right).$$

The initial conditions (31) and (33) are the same.

Note that the EKF, when applied to linear systems, coincides with the Kalman filter.

6.2 Linear Quadratic Regulator problem

Similar analysis as for the linear quadratic estimation problem holds for the Linear Quadratic Regulator (LQR) problem which is given by

$$\begin{aligned}
\min_{\mathbf{x}^r(\cdot),\mathbf{u}^r(\cdot)} \quad & \frac{1}{2}\mathbf{x}^r(t^r_{f,k})^T \mathbf{E}^r \mathbf{x}^r(t^r_{f,k}) \qquad (34)\\
& + \frac{1}{2}\int_{t^r_{0,k}}^{t^r_{f,k}} \left(\mathbf{x}^r(\tau)^T \mathbf{Q}^r \mathbf{x}^r(\tau) + 2\mathbf{u}^r(\tau)^T \mathbf{S}^r \mathbf{x}^r(\tau) + \mathbf{u}^r(\tau)^T \mathbf{R}^r \mathbf{u}^r(\tau)\right) d\tau \\
\text{s.t.} \quad & \dot{\mathbf{x}}^r(t) = \mathbf{A}\mathbf{x}^r(t) + \mathbf{B}\mathbf{u}^r(t), \quad \forall\, t \in I^r_k \quad , \\
& \mathbf{x}^r(t_{0,k}) = \mathbf{x}^r_{0,k}.
\end{aligned}$$

The optimal control $\mathbf{u}^r(t|I^r_k)$, $t \in I^r_k$, has to be determined by the optimizer. For the sake of simplicity, $\mathbf{w}$ has been left out in (34) but extensions to include *known* forcing functions are only a matter of notation. The optimal solution to (34) for any initial state is a linear function of the state $\mathbf{u}(t) = -\mathbf{K}^r(t)\mathbf{x}(t)$ where the time-variant gain $\mathbf{K}^r(t)$ is given by

$$\mathbf{K}^r(t) = \mathbf{R}^{r\,-1}\left(\mathbf{B}^T \mathbf{P}^r(t) + \mathbf{S}\right). \qquad (35)$$

Similar to the solution of the estimation problem, $\mathbf{P}^r(t)$ denotes a differentiable time dependent matrix which has to satisfy a matrix Riccati equation given by

$$\begin{aligned}
\dot{\mathbf{P}}^r(t) &= -\mathbf{P}^r(t)\mathbf{A} - \mathbf{A}^T\mathbf{P}^r(t) - \mathbf{Q}^r \qquad (36)\\
&\quad + \left(\mathbf{P}^r(t)\mathbf{B} + \mathbf{S}^T\right)\mathbf{R}^{r\,-1}\left(\mathbf{B}^T\mathbf{P}^r(t) + \mathbf{S}\right), \quad t \in I^r_k, \\
\mathbf{P}^r(t^r_{f,k}) &= \mathbf{E}^r. \qquad (37)
\end{aligned}$$

Equation (36) can be solved by integration backwards in time starting at $t^r_{f,k}$.

Three interesting cases of moving horizons allow very efficient on-line schemes to calculate the optimal moving horizon feedback control $\mathbf{u}(\mathbf{x}^r_{0,k}, t_{0,k})$ for problem k. All of them make use of the fact that the solution of the matrix Riccati equation (36) is independent of the initial value $\mathbf{x}^r_{0,k}$ and can thus be solved before $\mathbf{x}^r_{0,k}$ is specified.

Shrinking Horizon

For a sequence of problems with fixed end time $t^r_{f,k} = t^r_{f,k-1}$, but $t^r_{0,k} > t^r_{0,k-1}$, e.g. for batch problems with a fixed end time, we can use the fact that the backwards integration of the matrix Riccati equation (37) starts at the identical "initial" condition (37) and thus gives identical trajectories $\mathbf{P}^r(t)$, but on shrinking time intervals. It is possible to perform the computation of $\mathbf{P}^r(t)$ on the interval $[t^r_{0,0}, t^r_{f,0}]$ off-line, and to store just the gain matrix trajectory $\mathbf{K}^r(t)$, $t \in [t^r_{0,0}, t^r_{f,0}]$. This allows to obtain the optimal feedback control law $\mathbf{u}(\mathbf{x}^r_{0,k}, t^r_{0,k}) := -\mathbf{K}^r(t^r_{0,k})\mathbf{x}^r_{0,k}$, that can be evaluated in negligible time.

Note that this method is equally applicable to linear time-variant systems. It also provides the basis for the linearized neighboring feedback control method for non-linear systems presented in Subsection 6.3.

Moving Horizon:

A second interesting simplification arises in the case that the initial and the final time of the horizon move simultaneously, i.e., that $T = t^r_{f,k} - t^r_{0,k}$ is constant for all k. An inspection of Eqs. (36) and (37) shows that the solution $\mathbf{P}^r(t)$ of problem k does *not* depend on the index k. In particular, $\mathbf{P}^r(t^r_{0,k})$ is identical for different problems k, and therefore also the gain matrix $\mathbf{K}^r(t^r_{0,k}) = \bar{\mathbf{K}}^r$. The optimal moving horizon feedback control is therefore simply given by a matrix multiplication $\mathbf{u}(\mathbf{x}^r_{0,k}) = -\bar{\mathbf{K}}^r\mathbf{x}^r_{0,k}$. In contrast to the shrinking horizon case, this feedback law is time independent and requires storage of one matrix $\bar{\mathbf{K}}^r$ only. Unfortunately, this method cannot be generalized to time variant linear systems, because Eq. (36) would loose its time invariance.

Infinite Horizon:

A third and very prominent case arises when $t^r_{f,k} = \infty$ for all k. Here, the matrix $\mathbf{P}(t)$ is simply constant for all times in all problems; it is the solution of an algebraic Riccati equation that can be obtained by requiring $\dot{\mathbf{P}}^r(t) = 0$ in Eq. (36). As in the moving horizon case, the gain matrix is constant, $\mathbf{K}^r(t_{0,k}) = \bar{\mathbf{K}}^r$ for all k, and can be computed off-line. The resulting linear controller is commonly referred to as the *Linear Quadratic Regulator (LQR)*.

For both problems, the linear state estimation (26) and the linear quadratic regulator (34), efficient and robust numerical techniques have been developed which are also applicable to large scale processes (see, e.g. Mehrmann 1991; Jacobson et al. 1980). However, the problem formulations are restricted to linear process models and general inequality restrictions cannot be considered.

6.3 Linearized Neighboring Feedback Control along Reference Solutions

The simplicity and the power of the recursive techniques that are applicable to linear systems with quadratic cost motivates the question how they can help to provide an approximation to the optimal moving horizon control for non-linear systems. One such technique will be briefly described in this subsection. The method is applicable to a much wider class of problems than considered. Numerical techniques to solve them have been developed, e.g., by Pesch (1978), Krämer-Eis et al., (1985, 1987), and Kugelmann and Pesch (1990a, 1990b). Linearized neighboring techniques have also been used in similar approaches, e.g., by Terwiesch and Agarwal (1994) and de Oliveira and Biegler (1995).

Let us assume that we have found an optimal solution to the problem (6)-(10) for some $\mathbf{x}_0$ and t_0 by the indirect approach. The result are trajectories $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ and $\boldsymbol{\lambda}^*(t)$, which have to satisfy the necessary conditions for optimality stated in Eqs. (12) and (13). We rephrase these equations here for a slightly simplified problem:

$$\begin{aligned}
&\mathbf{0} = \mathbf{x}^*(t_0) - \mathbf{x}_0 \\
&\mathbf{0} = \boldsymbol{\lambda}^*(t_f) - \nabla_{\mathbf{x}} E(\mathbf{x}^*(t_f)) \\
&\text{and for almost all } t \in [t_0, t_f] \\
&\mathbf{0} = \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t), t) - \dot{\mathbf{x}}^*(t) \\
&\mathbf{0} = \nabla_{\mathbf{x}} \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) + \dot{\boldsymbol{\lambda}}^*(t) \\
&\mathbf{0} = \nabla_{\mathbf{u}} \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t).
\end{aligned} \tag{38}$$

The final state constraint (10) and all path constraints (9) are omitted and we assume that the Hamiltonian $\mathcal{H}$ from Eq. (11) depends twice continuously differentiable on $\mathbf{x}$ and $\mathbf{u}$ and is concave in $\mathbf{u}$, so that the last equation is equivalent to the maximization of $\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t), t)$ with respect to $\mathbf{u}$.

Let us now investigate how the solution trajectories change if the initial value changes to a slightly disturbed value $\mathbf{x}_0' = \mathbf{x}_0 + \epsilon$. Under mild regularity assumptions, the solution trajectories depend continuously differentiable on $\mathbf{x}_0$; let us introduce the shorthands

$$\begin{aligned}
\mathbf{x}^\epsilon(t) &= \tfrac{d\mathbf{x}^*(t;\mathbf{x}_0)}{d\mathbf{x}_0}(\mathbf{x}_0' - \mathbf{x}_0), \\
\mathbf{u}^\epsilon(t) &= \tfrac{d\mathbf{u}^*(t;\mathbf{x}_0)}{d\mathbf{x}_0}(\mathbf{x}_0' - \mathbf{x}_0), \\
\boldsymbol{\lambda}^\epsilon(t) &= \tfrac{d\boldsymbol{\lambda}^*(t;\mathbf{x}_0)}{d\mathbf{x}_0}(\mathbf{x}_0' - \mathbf{x}_0).
\end{aligned}$$

We can apply the implicit function theorem to compute these derivatives. A linearization of system (38) along the reference trajectories $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ and

$\boldsymbol{\lambda}^*(t)$ yields[6] :

$$\begin{aligned}
\mathbf{0} &= \mathbf{x}^\epsilon(t_0) - (\mathbf{x}_0' - \mathbf{x}_0), \\
\mathbf{0} &= \boldsymbol{\lambda}^\epsilon(t_f) - \frac{\partial^2 E}{\partial \mathbf{x}^2}(t_f)\mathbf{x}^\epsilon(t_f), \\
&\text{and for all } t \in [t_0, t_f], \\
\mathbf{0} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t)\mathbf{x}^\epsilon(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(t)\mathbf{u}^\epsilon(t) - \dot{\mathbf{x}}^\epsilon(t), \\
\mathbf{0} &= \frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}^2}(t)\mathbf{x}^\epsilon(t) + \frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}\partial \mathbf{u}}(t)\mathbf{u}^\epsilon(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t)^T \boldsymbol{\lambda}^\epsilon(t) + \dot{\lambda}^\epsilon(t), \\
\mathbf{0} &= \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}\partial \mathbf{x}}(t)\mathbf{x}^\epsilon(t) + \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}^2}(t)\mathbf{u}^\epsilon(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(t)^T \boldsymbol{\lambda}^\epsilon(t).
\end{aligned}$$

It turns out that this system of linear equations is nothing else than the indirect approach applied to a time variant linear quadratic regulator problem of the same form as (34). This problem can be formulated as follows:

$$\begin{aligned}
\min_{\mathbf{x}^\epsilon(\cdot),\mathbf{u}^\epsilon(\cdot)} \quad & \frac{1}{2}\mathbf{x}^\epsilon(t_f)^T \frac{\partial^2 E}{\partial \mathbf{x}^2}\mathbf{x}^\epsilon(t_f) \\
& + \frac{1}{2}\int_{t_0}^{t_f} \left(\mathbf{x}^{\epsilon T} \frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}^2}\mathbf{x}^\epsilon + 2\mathbf{u}^{\epsilon T} \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}\partial \mathbf{x}}\mathbf{x}^\epsilon + \mathbf{u}^{\epsilon T} \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}^2}\mathbf{u}^\epsilon \right) dt \qquad (39) \\
\text{s.t.} \quad & \dot{\mathbf{x}}^\epsilon(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{x}^\epsilon(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{u}^\epsilon(t), \quad \forall\, t \in [t_0, t_f], \\
& \mathbf{x}^\epsilon(t_0) = (\mathbf{x}_0' - \mathbf{x}_0).
\end{aligned}$$

The matrix Riccati equation (36) can be solved on the horizon $[t_0, t_f]$ for the linearized problem (39) along the reference trajectory with initial value $\mathbf{x}_0$ (cf. Eq. (38)). Then the feedback matrix $\mathbf{K}(t_0)$ can be precalculated to provide a first order approximation $\tilde{\mathbf{u}}$ to the optimal feedback for a system state $\mathbf{x}_0'$ at time t_0:

$$\tilde{\mathbf{u}}(\mathbf{x}_0', t_0) := \mathbf{u}^*(t_0) - \mathbf{K}(t_0)(\mathbf{x}_0' - \mathbf{x}_0). \qquad (40)$$

Shrinking Horizon: For shrinking horizon problems, the matrix function $\mathbf{K}(t)$ can be precomputed along the reference solution for $t \in [t_0, t_f]$ and can

[6] For the Jacobian of a vector valued function $\mathbf{f}(\mathbf{x})$ we write $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ which denotes the matrix with entries $\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{ij} := \frac{\partial f_i}{\partial x_j}$. The second derivative matrix of a scalar function $\mathcal{H}(\mathbf{x}, \mathbf{u})$ is denoted, e.g., by $\frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}\partial \mathbf{u}}$ with $\left(\frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}\partial \mathbf{u}}\right)_{ij} := \frac{\partial^2 \mathcal{H}}{\partial x_i \partial u_j}$. For brevity, we do not repeat all function arguments, but only the time t, and implicitly assume that the derivatives are evaluated at the corresponding point of the trajectories $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ and $\boldsymbol{\lambda}^*(t)$.

serve to provide an immediate feedback analogous to the shrinking horizon method described in Section 6.2. For given $\mathbf{x}_{0,k}$ and $t_{0,k} \in [t_0, t_f]$ we compute a first order approximation $\tilde{\mathbf{u}}$ of the optimal moving horizon feedback control that is given by

$$\tilde{\mathbf{u}}(\mathbf{x}_{0,k}, t_{0,k}) := \mathbf{u}^*(t_{0,k}) - \mathbf{K}(t_{0,k})(\mathbf{x}_{0,k} - \mathbf{x}^*(t_{0,k})).$$

The motivation for this approximation is the expectation that the real system trajectory stays sufficiently close to the reference trajectory. In particular, we assumed that no model uncertainties and disturbances have been present. However, if severe model uncertainty and disturbances are present the approach will encounter difficulties.

6.4 Initialization techniques for direct methods

A straightforward approach to moving horizon optimization is to apply one of the direct methods described in Section 5 to solve the moving horizon optimization problems. Though they are originally designed for off-line use, their on-line application can lead to good results, depending on the real-time requirements of the problem, as the advantages of direct methods (flexibility, robustness, handling of constraints) can be fully exploited (see, e.g., Leineweber, 1997). It should be kept in mind, however, that no general run-time guarantees can be given for these methods as the number of SQP iterations is not limited (an interesting approach that requires only one iteration per sampling time can be found in the research article by Diehl et al. (2001) in this book).

The computing times for the subsequent NLP solutions depend considerably on the initial guess $\boldsymbol{\xi}_0$ for the optimization variables and the initial setup of the SQP algorithm (in particular the Hessian). We will present some apparent approaches to find a good initial guess $\boldsymbol{\xi}_0^k$ for the optimization variables in the NLP (23) that arises after the discretization of the k-th optimal control problem (6)-(10). We will briefly discuss them for moving and shrinking horizon problems.

Moving horizon problems: For time independent moving horizon problems, three possibilities seem to suggest themselves for the initialization $\boldsymbol{\xi}_0^k$ of the NLP (23):

- *Set-point initialization:* If the optimization problem is formulated with the objective to steer the system into a desired steady state (the setpoint state $\mathbf{x}_{ss}$ and controls $\mathbf{u}_{ss}$), the (constant) setpoint trajectory is the solution of an optimization problem (6)-(10) with initial value $\mathbf{x}_0 = \mathbf{x}_{ss}$. The NLP solution $\boldsymbol{\xi}_*^{ss}$ of this optimization problem in the chosen transcription may be used to serve as an initial guess for the NLP solution iterations (5.4): $\boldsymbol{\xi}_0^k := \boldsymbol{\xi}_*^{ss}$. As long as the real system state $\mathbf{x}_{0,k}$ stays close to $\mathbf{x}_{ss}$ this may be a good initial guess. The setpoint initialization provides every optimization problem with the same initial guess.

- *Simple warm start:* This strategy is based on the conjecture that the solution $\boldsymbol{\xi}_*^{k-1}$ of the previous optimization problem $k-1$ would provide a good initial guess for the current problem k: $\boldsymbol{\xi}_0^k := \boldsymbol{\xi}_*^{k-1}$. This may be justified if the new initial state $\mathbf{x}_{0,k}$ has not changed much compared to $\mathbf{x}_{0,k-1}$, as can be expected if the sampling time ΔT is short relative to the time constant of the system.
- *Shift strategy:* The third strategy is motivated by the following observation: for a fictitious *undisturbed* system controlled by a moving horizon algorithm with *infinite* horizon, the (open-loop) solution of the first optimization problem on $[t_{0,0}, \infty]$ would already provide the whole *closed-loop* control trajectory – thanks to the dynamic programming property, the part of the precalculated control strategy that remains at problem k on the horizon $[t_{0,k}, \infty]$ is still optimal (this is similar for shrinking horizon problems). In the finite moving horizon framework the dynamic programming property does no longer hold strictly, but the idea to shift the problem in time may still be advantageous if the horizon is chosen to be sufficiently long. We will illustrate this strategy in the context of the direct single shooting method described in Section 5.1; we choose a piecewise constant control representation with N intervals I_i each of length ΔT. Using the $(k-1)$st solution $\boldsymbol{\xi}_*^{k-1} = (\mathbf{q}_{*,0}^{k-1}, \ldots, \mathbf{q}_{*,N-1}^{k-1})$, the initial guess $\boldsymbol{\xi}_0^k$ of the kth problem would be determined by a "shift" in the controls

$$\mathbf{q}_{0,i}^k := \mathbf{q}_{*,i+1}^{k-1} \quad \text{for} \quad i = 0, 1, \ldots N-2.$$

  The new initial value for the last control variable cannot be obtained by the shift and must be extrapolated; a convenient initialization is, e.g.: $\mathbf{q}_{0,N-1}^k := \mathbf{q}_{*,N-1}^{k-1}$. This method is applicable to general time-variant nonlinear systems.

The setpoint initialization provides every optimization problem with the same initialization and thus leads to optimization outcomes that are independent of the optimization history. In practice, however, both the warm start and shift strategy perform clearly faster (cf. Diehl et al., 1999, for a test in the context of the direct multiple shooting method). From the programmer's point of view, the warm start technique can often easier be incorporated into existing off-line optimization software and may therefore be preferable.

Shrinking horizon problems: An initialization method very similar to the shift strategy can be applied for shrinking horizon problems with fixed end time $t_{f,k} = t_{f,0}$. Here, only the part of the old solution $\boldsymbol{\xi}_*^{k-1}$ that corresponds to the new horizon $[t_{0,k}, t_{f,0}] \subset [t_{0,k-1}, t_{f,0}]$ is used to initialize the (reduced) optimization variable vector $\boldsymbol{\xi}_0^k$ of the new problem. For the direct single shooting method this would, e.g., mean that the reduced new piecewise control vector $\boldsymbol{\xi}_0^k = \boldsymbol{\xi}_0^k = (\mathbf{q}_{0,0}^k, \ldots, \mathbf{q}_{0,N^k}^k)$, where $N^k = N^{k-1} - 1$, is initialized by

$$\mathbf{q}_{0,i}^{k} := \mathbf{q}_{*,i+1}^{k-1} \quad \text{for} \quad i = 0, 1, \ldots N^k - 1.$$

As the shift strategy this method is applicable to general time-variant non-linear systems.

7 Summary

An introduction has been given to dynamic optimization on moving horizons. We first focused on the generic problem formulation for both, control and estimation problems where we illuminated the special on-line character of the problem. Secondly, we reviewed standard numerical techniques to solve the problem on a fixed horizon. Special emphasis has been given to direct optimization methods which are typically used in practise. Furthermore we discussed basic extensions of the fixed horizon approaches to the moving horizon case. An extended discussion of more advanced concepts to solve these demanding dynamic optimization problems, proposed by the authoring research groups, can be found in this book elsewhere.

In particular, a methodology using a multiscale approach is suggested by Binder et al. (2001) where a hierarchy of successively refined finite dimensional problems are constructed and solved as long as time permits. Therefore an approximate solution is provided at any time where the approximation quality of the solution scales with the used computation time.

Diehl et al. (2001) develop a real-time iteration scheme for the direct multiple shooting method that is aimed for large-scale real-time optimization problems arising in nonlinear MPC. They perform closed-loop experiments with a high purity distillation column that is described by a DAE model involving 164 state equations; sampling times of a few seconds are feasible with this approach.

In Kronseder et.al. (2001) a concept for model predictive control of very large-scale dynamical systems that arise in the control of air separation plants and consist in thousands of DAEs is developed. The concept considers the different time scales prescribed by the nature of the process. Emphasis is put on mid term and short term computations, which are here represented by online computation of parameterized optimal set point trajectories on a moving horizon and by update of set point trajectories via linearization of neighboring parameterized extremals respectively. Additionally, fundamental issues of the notion of real-time optimality are discussed.

References

Abel, O., Helbig, A., Marquardt, W.: DYNOPT User Manual, Release 2.4, Lehrstuhl für Prozesstechnik, RWTH Aachen (1999)

Anderson, B.D.O., Moore, J.B.: Optimal Control: Linear Quadratic Methods. Prentice-Hall (1989)

Allgöwer, F., Badgwell, T.A., Qin, J.S., Rawlings, J.B., Wright, S.J.: Nonlinear Predictive Control and Moving Horizon Estimation - An introductory overview. In: P.M. Frank (ed.): Advances in Control. Highlights of ECC'99, Springer (1999) 391-449

Aris R.: Mathematical Modeling Techniques. Dover Publications (1994)

Arwell, J.A., King, B.B.: Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. Report 99-01-01, Interdisciplinary Center for Applied Mathematics, Virginia Tech (1999)

Ascher, U., Christiansen, J., Russell, R.D.: A collocation solver for mixed order systems of boundary value problems. Math. Comp. **33**, (1979) 659–679

Ascher, U., Mattheij, R., Russell, R.D.: Numerical Solution of Boundary Value Problems for Differential Equations. SIAM (1988)

Baake, E., Baake, M., Bock, H.G., Briggs, K.: Fitting Ordinary Differential Equations to Chaotic Data. Phys. Rev. A, **45** (1992)

Bär, V.: Ein Kollokationsverfahren zur numerischen Lösung allgemeiner Mehrpunktrandwertaufgaben mit Schalt- und Sprungbedingungen mit Anwendungen in der Optimalen Steuerung und der Parameteridentifizierung. Diplomarbeit, Bonn (1983)

Bardi, M., Dolcetta, I.C.: Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Birkhäuser (1996)

Barclay, A., Gill, P.E., Rosen, J.B.: SQP methods and their application to numerical optimal control. In: W.H. Schmidt, K. Heier, L. Bittner, R. Bulirsch (eds.): Variational Calculus. Optimal Control and Application (1998)

Bauer, I.: Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik. PhD thesis, University of Heidelberg (2000).

Bauer, I., Bock, H.G., Körkel, S., and Schlöder, J.P.: Numerical Methods for Optimum Experimental Design in DAE Systems. J. Comp. Appl. Math. Vol. **120** (2000) 1-25

Bellman, R.E.: Dynamic Programming. Princeton University Press (1957)

Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific (1996)

Betts, J.T.: Survey of numerical methods for trajectory optimization. AIAA J. Guidance, Control, and Dynamics **21**, 2 (1998) 193-207.

Betts, J.T., Frank, P.D.: A sparse nonlinear optimization algorithm. J. Optimization Theory and Applications **82**, 3 (1994) 519-541.

Betts, J.T., Huffmann, W.P.: Mesh refinement in direct transcription methods for optimal control. Optim. Control Appl. Meth. **19**, (1998) 1-21

Biegler, L.T.: Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comput. chem. Engng. **8**, 3/4 (1984) 243-248.

Binder T., Blank L., Dahmen W., Marquardt W.: Towards Multiscale Dynamic and Data Reconciliation. In: R. Berber, C. Kravaris (eds.): Nonlinear Model Based Process Control Kluwer Academic Publishers, (1998), 623-665

Binder, T., Blank, L., Dahmen W., Marquardt W.: Multiscale Concepts for Moving Horizon Optimization. In: M. Groetschel, S.O. Krumke, J. Rambau (eds.): Online Optimization of Large Scale Systems: State of the Art, Springer (2001)

Bitmead, R.R, Gevers, M., Wertz, V.: Adaptive optimal Control - The Thinking Man's GPC. Prentice Hall (1990)

Bock, H.G.: Numerische Berechnung zustandsbeschränkter optimaler Steuerungen. Carl-Cranz-Gesellschaft, Tech. Rep 1.06/78 Heidelberg (1978)

Bock, H.G.: Numerical Solution of Nonlinear Multipoint Boundary Value Problems with Applications to Optimal Control. ZAMM **58**, T407 (1978)

Bock, H.G.: Numerical treatment of inverse problems in chemical reaction kinetics. In: K. H. Ebert, P. Deuflhard, and W. Jäger, (eds.): Modelling of Chemical Reaction Systems. volume 18 of Springer Series in Chemical Physics, Springer (1981)

Bock, H.G.: Recent advances in parameter identification techniques for O.D.E. In: P. Deuflhard und E. Hairer (eds.): Numerical Treatment of Inverse Problems in Differential and Integral Equations. Birkhäuser (1983)

Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In Proc. 9th IFAC World Congress Budapest, July 2-6, Pergamon Press (1984), 242-247

Bock, H.G.: Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. Bonner Mathematische Schriften 183, University of Bonn (1987)

Bock, H.G., Eich, E., Schlöder, J.P.: Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In: K. Strehmel (ed.): Numerical Treatment of Differential Equations. Teubner (1988)

Bock, H.G., Schlöder, J.P., Schulz, V.: Numerik großer Differentiell-Algebraischer Gleichungen — Simulation und Optimierung. In: H. Schuler (ed.): Prozeßsimulation. VCH Verlagsgesellschaft mbH (1995) 35-80

Bulirsch, R., Nerz, E., Pesch, H.J., von Stryk, O.: Combining direct and indirect methods in optimal control: range maximization of a hang glider. In: R. Bulirsch, A. Miele, J. Stoer, K.-H. Well (eds.): Optimal Control - Calculus of Variations, Optimal Control Theory and Numerical Methods, International Series of Numerical Mathematics **111**, Birkhäuser (1993) 273-288.

Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM (1996)

Bryson, A.E., Ho, Y.C.: Applied Optimal Control. Ginn and Company, (1969), Rev. printing, Hemisphere (1975)

Buchauer, O., Hiltmann, P., Kiehl, M.: Sensitivity analysis of initial-value problems with application to shooting techniques. Numerische Mathematik **67** (1994) 151-159

Bulirsch, R.: Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. Report of the Carl-Cranz-Gesellschaft e.V., DLR, Oberpfaffenhofen (1971)

Bulirsch, R., Kraft, D. (eds.): Computational Optimal Control. International Series in Numerical Mathematics **115** Birkhäuser-Verlag (1994)

Burns, J.A., King, B.B.: A reduced basis approach to the design of low order feedback controllers for nonlinear continuous systems. To appear in: Journal of Vibration and Control

Büskens, C., Gerdts, M.: Real-time optimization of DAE-systems. In: M. Groetschel, S.O. Krumke, J. Rambau (eds.): Online Optimization of Large Scale Systems: State of the Art. Springer (2001)

Callies, R.: Habilitationsschrift. Technische Universität München, submitted

Caracotsios, M., Stewart, W.E.: Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations. Computers & Chemical Engineering **9**, 4 (1985) 359-365

Cauchy, A.L.: Méthode générale pour la résolution systémes d'équations simultanées. Compt. rend. acad. sci. **25** (1847) 536-538

Cervantes, A., Biegler, L.T.: Large-scale DAE optimization using a simultaneous NLP formulation. AIChE Journal **44**, 5 (1998) 1038-1050.

Chernousko, F.L., Luybushin, A.A.: Method of successive approximations for optimal control problems (survey paper). Opt. Contr. Appl. and Meth. **3**, (1982) 101-114

Deuflhard, P.: A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. Numer. Math. **22** (1974) 289-315

Dickmanns, E.D., Well, K.H.: Approximate solution of optimal control problems using third order Hermite polynomial functions. Lec. Notes in Comp. Sc., Vol. 27, Springer (1975) 158-166

Diehl, M., Bock, H.G., Leineweber, D.B., Schlöder J.P.: Efficient direct multiple shooting in nonlinear model predictive control. In: F. Keil, W. Mackens, H. Voß, and J. Werther (eds.): Scientific Computing in Chemical Engineering II, volume 2, Springer (1999), 218-227

Diehl, M., Disli-Uslu, I., Schwarzkopf, S., Allgöwer, F., Bock, H.G., Bürner, T., Findeisen, R., Gilles, E.D., Kienle, A., Schlöder, J.P., Stein, E.: Real-Time Optimization of Large Scale Process Models: Nonlinear Model Predictive Control of a High Purity Distillation Column. In: M. Groetschel, S.O. Krumke, J. Rambau (eds.): Online Optimization of Large Scale Systems: State of the Art. Springer (2001)

Engl, G., Kröner, A., Kronseder, T., von Stryk, O.: Numerical simulation and optimal control of air separation plants. In: H.-J. Bungartz, F. Durst, Chr. Zenger (eds.): High Performance Scientific and Engineering Computing. Lecture Notes in Computational Science and Engineering **8**, Springer (1999) 221-231

Falcone, M., Ferretti, R.: Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations. Numerische-Mathematik **67**, 3 (1994) 315-344

Fletcher, C.A.: Computational Galerkin Methods. Springer (1984)

Fox, L.: Some numerical experiments with eigenvalue problems in ordinary differential equations. In: Langer R.E. (ed): Boundary Value Problems in Differential Equations. (1960)

Galán, S., Feehery, W.F., Barton, P.I.: Parametric sensitivity functions for hybrid discrete/continuous systems. Applied Numerical Mathematics **31**, 1 (1999) 17-47

Gallitzendörfer, J.V., Bock, H.G.: Parallel algorithms for optimization boundary value problems in DAE. In: H. Langendörfer (ed.): Praxisorientierte Parallelverarbeitung. Hanser (1994)

Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. Report NA 97-2, Department of Mathematics, University of California, San Diego (1997)

Gill, Ph.E., Murray, W., Saunders, M.A., Wright, M.H.: Some theoretical properties of an augmented Lagrangian merit function. In: P.M. Pardalos (ed.): Advances in Optimization and Parallel Computing. Elsevier Science Publishers (1992) 101-128

Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H.: User's Guide for NPSOL (Version 5.0): a Fortran package for nonlinear programming. Numerical Analysis Report 98-2, Department of Mathematics, University of California, San Diego (1998)

Gould, N.I.M, Toint, P.L.: SQP Methods for Large-Scale Nonlinear Programming. Proceedings of the 19th IFIP TC7 Conference on System Modelling and Optimization, Cambidge, England, July 12th to 16th 1999, also available as: Rutherford Appleton Laboratory Technical Report RAL-TR-1999-055

Griewank, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. Frontiers in Applied Mathematics **19** SIAM (2000)

Han, S.P.: Superlinearly convergent variable-metric algorithms for general nonlinear programming problems. Math. Progr. **11** (1976) 263-282

Hartl, R.F., Sethi, S.P., Vickson, R.G.: A survey of the Maximum Principles for optimal control problems with state constraints. SIAM Review **37**, 2 (1995) 181-218

Helbig A., Abel O., Marquardt W.: Structural concepts for optimization based control of transient processes, In: F. Allgöwer, A. Zheng (eds.): Nonlinear Predictive Control. Birkhäuser (2000) 295-312

Heim, A.: Parameteridentifizierung in differential-algebraischen Gleichungssystemen. Diploma thesis, Department of Mathematics, Technische Universität München (1992)

Heim, A., von Stryk, O.: Documentation of PAREST — A multiple shooting code for optimization problems in differential-algebraic equations. Report TUM-M9616, Mathematisches Institut, Technische Universität München (1996)

Henson, M. A.: Nonlinear model predictive control: Current status and future directions. Comp. Chem. Eng. **23** (1998) 187-202

Hestenes, M.R.: Calculus of Variations and Optimal Control Theory. Wiley (1966)

Hiltmann, P.: Numerische Lösung von Mehrpunkt-Randwertproblemen und Aufgaben der optimalen Steuerung mit Steuerfunktionen über endlichdimensionalen Räumen. Dissertation, Fakultät für Mathematik und Informatik, Technische Universität München (1990)

Hinsberger, H.: Ein direktes Mehrzielverfahren zur Lösung von Optimalsteuerungsproblemen mit großen, differential-algebraischen Gleichungssystemen und Anwendungen aus der Verfahrenstechnik. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Technische Universität Clausthal (1997)

Isaacs, R.: Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization. J. Wiley & Sons (1965)

Jacobson, D.H., Martin, D.H., Pachter, M., Geveci, T.: Extensions of Linear-Quadratic Control Theory. Lecture Notes in Control and Inform. Sciences **27**, Springer (1980)

Kallrath, J., Bock, H.G., Schlöder, J.P.: Least Squares Parameter Estimation in Chaotic Differential Equations. Celestial Mechanics and Dynamical Astronomy, **56** (1993)

Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. ASME, J. Basic Engineering (1960) 35-45

Kailath, T.: Linear systems. Prentice Hall (1980)

Keerthi, S.S, Gilbert, E.G: Optimal infinite-horizon feedback laws for a general class of constraint discrete-time systems: Stability and moving horizon approximations. J. Opt. Theory and Appl. **57**(2) (1988) 265-293

Kelley, H.J.: Gradient theory of optimal flight paths. Journal of the American Rocket Society **30** (1960) 947-953

Keller, H.J.B.: Numerical Methods for Two-Point Boundary Value problems. Waltham: Blaisdell (1968)

Kiehl, M.: Vektorisierung der Mehrzielmethode zur Lösung von Mehrpunkt-Randwertproblemen und Aufgaben der optimalen Steuerung. PhD Thesis, Mathematisches Institut, Technische Universität München (1989)

Kiehl, M.: Sensitivity analysis of ODEs and DAEs — theory and implementation guide. Optimization Methods and Software **10**, 6 (1999) 803-821

King, B.B., Sachs, E.W.: Semidefinite programming techniques for reduced order systems with guaranteed stability margins. Submitted to: Computational Optimization and Applications

Kraft, D.: On converting optimal control problems into nonlinear programming problems. In: K. Schittkowski (ed.): Computational Mathematical Programming. NATO ASI Series, Vol. **F15**, Springer (1985) 261-280.

Kraft, D.: Algorithm 733: TOMP–Fortran modules for optimal control calculations. ACM Transactions on Mathematical Software **20**, 3 (1994) 262-281.

Krämer-Eis, P.: Ein Mehrzielverfahren zur numerischen Berechnung optimaler Feedback-Steuerungen bei beschränkten nichtlinearen Steuerungsproblemen. Bonner Mathematische Schriften 166, University of Bonn (1985)

Krämer-Eis, P., Bock, H.G.: Numerical Treatment of State and Control Constraints in the Computation of Feedback Laws for Nonlinear Control Problems. In: P. Deuflhard et al. (eds.): Large Scale Scientific Computing. Birkhäuser (1987), 287-306

Kröner, A., Marquardt, W., Gilles, E.D.: Computing consistent initial conditions for differential algebraic process models. Comp. Chem. Eng., **16**S (1992) 131-138

Kröner, A., Kronseder, T., Engl, G., von Stryk, O.: Dynamic optimization for air separation plants. Proceesdings of the European Symposium on Computer Aided Process Engineering (ESCAPE-11), Kolding, Denmark, May 27-30, 2001 (2001)

Kronseder, T., von Stryk, O., Bulirsch, R.: Towards Nonlinear Model Based Predictive Optimal Control of Large-Scale Process Models with Application to Air Separation Plants. In: M. Groetschel, S.O. Krumke, J. Rambau, (eds.): Online Optimization of Large Scale Systems: State of the Art. Springer (2001)

Kugelmann, B., Pesch, H.J.: New general guidance method in constrained optimal control, Part 1: Numerical method. J. Optimization Theory and Applications **67**, 3 (1990) 421-435.

Kugelmann, B., Pesch, H.J.: New general guidance method in constrained optimal control, Part 2: Application to space shuttle guidance. J. Optimization Theory and Applications **67**, 3 (1990) 437-446.

Kunisch, K, Volkwein, S.: Control of Burger's equation by a reduced order approach using proper orthogonal decomposition. Karl-Franzens-Universität Graz, Spezialforschungsbereich F003, Bericht Nr. 138 (Sep. 1998)

Leineweber, D.B.: Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models. volume 613 of Fortschr.-Ber. VDI Reihe 3, Verfahrenstechnik. VDI Verlag (1999)

Leineweber D.B., Bock H.G., Schlöder, J.P.: Fast direct methods for real-time optimization of chemical processes. Proc. 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics Berlin, Wissenschaft- und Technik-Verlag (1997)

Leis, J.R., Kramer, M.A.: Sensitivity analysis of systems of differential and algebraic equations. Comput. Chem. Eng. **9** (1985) 93-96

Lions, P.L.: Generalized Solutions of Hamilton-Jacobi Equations. Pittman (1982)

Maly, T., Petzold, L.R.: Numerical methods and software for sensitivity analysis of differential-algebraic equations. Applied Numerical Mathematics **20** (1996) 57-79

Maner, B.R, Doyle, F.J., Ogunnaike, B.A., Pearson, R.K.: Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. Automatica **32**, (1996) 1285-1301

Marquardt, W.: Nonlinear model reduction for optimization based control of transient chemical processes. Proceedings of Chemical Process Control-6, Tuscon USA (2001)

Mayne, D.Q.: Optimization in model based control. In: J.B. Rawlings (ed.): The 4-th IFAC symposium on dynamics and control of chemical reactors, distillation columns, and batch processes. DYCORD 95 (1995) 229-242

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica **36** (2000) 789-814

Meadows, E.S., Rawlings, J.B.: Topics in model predictive control. In: R. Berber (ed.): Methods of Model-Based Control. NATO-ASI Series, Kluwer Press (1995) 331-347

Mehlhorn, R., Sachs, G.: A new tool for efficient optimization by automatic differentiation and program transparency. Optimization Methods and Software **4** (1994) 225-242

Mehrmann, V.L.: The Autonomous Linear Quadratic Control Problem. Lecture Notes in Control and Information Sciences **163** Springer (1991)

Michalska, H., Mayne, D.: Moving horizon observers-based control. IEEE Transactions on Automatic Control **40**(6) (1995) 995-1006

Miele, A.: Gradient algorithms for the optimization of dynamic systems. In: C.T. Leondes (ed.): Control and Dynamic Systems **16** (1980) 1-52

Morari, M., Lee, J.H.: Model predictive control: past, present and future. Computers and Chemical Engineering **23** (1999) 667-682

Muske, K.R., Rawlings, J.B.: Nonlinear receding horizon state estimation. In: R. Berber (ed.): Methods of Model-Based Control. NATO-ASI Series. Kluwer Press (1995) 489-504

Naumer, B.: Über approximative Methoden der Dynamischen Programmierung in der Optimalen Steuerung. Utz (1999)

Nocedal, J., Wright, S.J.: Numerical optimization. Springer (1999)

Norquay, S.J., Palazoglu, A., Romagnoli, J.A.: Application of Wiener model predictive control (WMPC) to an industrial C2-splitter. J. Process Control **9** (1999) 461-473

Oberle, H.J.: Numerische Berechnung optimaler Steuerungen von Heizung und Kühlung für ein realistisches Sonnenhausmodell. Habilitationsschrift, Technische Universität München, Germany (1982)

Ogunnaike, B.A., Ray, W.H.: Process Dynamics, Modelling and Control. Oxford University Press (1994)

de Oliveira, N.M.C., Biegler, L.T. An extension of Newton-type algorithms for nonlinear process control. Automatica **31** (2) (1995) 281-286

Pearson, R.K., Pottmann, M.: Gray-box identification of block-oriented nonlinear models. J. Process Control **10** (2000) 301-314

Pesch, H.J.: Numerische Berechnung optimaler Flugbahnkorrekturen in Echtzeitrechnung. Ph.D. thesis, TU München (1978)

Pesch, H.J., Bulirsch, R.: The Maximum Principle, Bellman's equation, and Caratheodory's work. J. Optimization Theory and Applications **80**, 2 (1994) 199-225

Petzold, L., Rosen, J.B., Gill, P.E., Jay, L.O., Park, K.: Numerical optimal control of parabolic PDEs using DASOPT. In: Biegler, Coleman, Conn, Santosa (eds.): Large Scale Optimization with Applications, Part II. Springer (1997), 271-299

Plitt, K.J.: Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen, Diplomarbeit, Bonn (1981)

Pontryagin, L.S., Boltyanski, V.G., Gamkrelidze, R.V., Miscenko, E.F.: The Mathematical Theory of Optimal Processes. Wiley (1962)

Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. in G.A. Watson (Hrsg.), Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics **630**, Springer (1978).

Process Systems Enterprise Ltd: gPROMS Advanced User Guide (Release 1.8), London (2000)

Pytlak, R.: Numerical Methods for Optimal Control Problems with State Constraints. Springer (1999)

Rao, C.V., Rawlings, J.B.: Nonlinear Horizon State Estimation, In: F. Allgöwer, A. Zheng (eds.): Nonlinear Predictive Control. Birkhäuser (2000) 45-70

Ricker, N.L.: Model predictive control: State of the art. In: Y. Arkun and W.H. Ray (eds.): Chemical Process Control - CPC IV, CACHE. Elsevier (1991) 271-296

Robertson, D., Lee, K.H., Rawlings, J.B.: A moving horizon–based approach for least–squares estimation. AIChE Journal **42**(8) (1996) 2209-2223

Rozenvasser, E.N.: General sensitivity equations of discontinuous systems. Automat. Remote Control (1967) 400-404

Schulz, V.: Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots. Dissertation, Naturwiss.-Math. Gesamtfakultät, Universität Heidelberg (1996).

Schulz, V.H.: Solving discretized optimization problems by partially reduced SQP methods. Comput. Visual. Sci. **1** (1998) 83-96

Schulz, V. H., Bock, H. G., and Steinbach, M. C.: Exploiting invariants in the numerical solution of multipoint boundary value problems for DAEs. SIAM J. Sci. Comput. **19** (1998) 440-467

Sriniwas, G.R., Arkun Y.: A global solution to the nonlinear model predictive control algorithms using polynomial ARX models. Comp. Chem. Engng **21**, (1997) 431-439

Steinbach, M.C.: Fast recursive SQP methods for large-scale optimal control problems. PhD thesis. University of Heidelberg (1995)

Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. 2nd ed., Springer (1993)

Su, H.T., McAvoy, T.J.: Artificial neural networks for nonlinear process identification and control. In: Henson M.A., Seborg, D.E (eds.): Nonlinear Process Control. Prentice-Hall (1997) 371-428

Terwiesch, P., Agarwal, M.: On-line corrections of pre-optimized input profiles for batch reactors. Comp. Chem. Eng. **18** (1994) S433-S437

Tolle, H.: Optimization Methods. Springer (1975)

Unger, J., Kröner, A., Marquardt, W.: Structural analysis of DAE-systems - Theory and applications. Comput. Chem. Eng. **19**, 8 (1995) 867-882

Vassiliadis, V.S., Sargent, R.W.H., Pantelides, C.C.: Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints. Ind. Eng. Chem. Res. **33** (1994) 2111-2122

Vassiliadis, V.S., Sargent, R.W.H., Pantelides, C.C.: Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. Ind. Eng. Chem. Res. **33** (1994) 2123-2133

von Schwerin, R., Winckler, M.J., Schulz, V.H.: Parameter estimation in discontinuous descriptor models. In: Bestle and Schiehlen (eds.): IUTAM Symposium on Optimization of Mechanical Systems. Kluwer Academic Publishers (1996) 269-276

von Stryk, O.: Numerical solution of optimal control problems by direct collocation. In: Bulirsch, R., Miele, A., Stoer, J., Well, K.H. (eds.): Optimal Control — Calculus of Variations, Optimal Control Theory and Numerical Methods. International Series of Numerical Mathematics **111** Birkhäuser (1993) 129-143

von Stryk, O.: Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen. Fortschritt-Berichte VDI, Reihe 8, Nr. 441, VDI-Verlag (1995).

von Stryk, O.: Numerical Hybrid Optimal Control and Related Topics. Habilitation, Department of Mathematics, Technische Universität München (2000), submitted

von Stryk, O., Bulirsch, R.: Direct and indirect methods for trajectory optimization. Annals of Operations Research **37** (1992) 357-373

von Stryk, O., Schlemmer, M.: Optimal control of the industrial robot Manutec r3. In: R. Bulirsch, D. Kraft (eds.): Computational Optimal Control. International Series of Numerical Mathematics **115**, Basel (1994) 367-382

Waldraff, W., King, R., Gilles E.D.: Optimal feeding strategies by adaptive mesh selection for fed–batch bioprocesses. Bioprocess Engineering **17** (1997) 221-227

Werbos, P.J.: Approximate dynamic programming for real-time control and neural modeling. In: White, D.A. and Sofge, D.A. (eds.): Handbook of Intelligent Control. Van Nostrand Reinhold (1992) 493-559

Wellers, M., Rake, H: Nonlinear model predictive control based on stable Wiener and Hammerstein models. In: F. Allgöwer, A. Zheng (eds.): Nonlinear Predictive Control. Birkhäuser (2000) 357-368