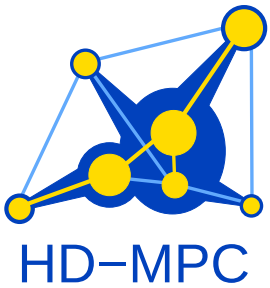
SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC



Deliverable Number:	D4.1.3
Deliverable Type:	Report
Contractual Date of Delivery:	March 1, 2011
Actual Date of Delivery:	**March 1, 2011**
Title of Deliverable:	**Report on new algorithms with guaranteed convergence to an optimum of the global system, at a high rate of convergence, and with intelligent hot-starting**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP4
WP Leader:	Moritz Diehl
Partners:	TUD, KUL, USE, UWM
Author(s):	A. Kozma, H. Scheu

Project co-ordinator

Name:	Bart De Schutter
Address:	Delft Center for Systems and Control Delft University of Technology Mekelweg 2, 2628 Delft, The Netherlands
Phone Number:	+31-15-2785113
Fax Number:	+31-15-2786679
E-mail:	b.deschutter@tudelft.nl
Project web site:	`http://www.ict-hd-mpc.eu`

Executive summary

In this report we discuss newly established optimal control methods that might be possible candidates for future HD-MPC algorithms. First, a sensitivity-driven distributed model predictive control approach is presented for linear continuous systems. Second, a generalization of multiple shooting is introduced for the model predictive control of nonlinear continuous-time systems. The convergence properties of the proposed methods are also discussed.

Chapter 1

Synopsis of the report

In this report we present two different novel algorithms that might be possible candidates for future general HD-MPC algorithms.

First, a sensitivity-driven model predictive control (S-DMPC) approach is introduced. The method implements a cooperative control scheme. The coupled linear subsystems communicate their Jacobians and dual variables to each other, while every local objective is extended by linear correction terms that correspond to other subsystems' objective and constraints. The S-DMPC method can cope with equality constraints as well as coupled inequality constraints and locally linear convergence can be achieved. Extension to the nonlinear case is straightforward and is subject of future investigation. While optimality of the distributed method can be proven for convex problems, convergence of the method depends on the decomposition of the problem, which defines a contraction rate. This rate can be quantified in a conservative way for linear quadratic problems. An extension of the algorithm is proposed to improve the convergence properties or even enforce convergence of the iterative method. A simple hot-start algorithm is proposed to be implemented in the moving-horizon model-predictive control. This hot start-algorithm enables an almost optimal initialization of the optimal control problem on the new horizon, such that only a low number of iterations is required to achieve optimal control performance. The method is validated in a simulated case study for the control of a nonlinear chemical process.

Second, a generalized multiple shooting algorithm is presented. This approach is tailored for the solution of large-scale nonlinear systems composed by multiple subsystems. However, the control scheme is centralized, the integration of subsystems takes place in a parallel fashion. The continuous coupling between the subsystems are discretized by using a polynomial of Legendre orthogonal basis. By this finite dimensional representation multiple shooting can be extended to the state space as well. All the integrations with respect to shooting intervals in time and with respect to subsystems may be carried out simultaneously and independently providing modularity and maintainability. The gap between shooting intervals both in space and time are closed by introducing constraints in the centralized optimization problem. Moreover, this optimal control problem has the ability to cope with both L_1 and L_2 penalties as cost functions. The optimization problem incorporates local state and control input inequalities as well. On the top level we have to solve a large-scale nonlinear program (NLP) with sparse equality constraints. This may be carried out by a standard sequential quadratic programming (SQP) method, where the subproblems are solved by a sparse QP solver. Our experiments show that integration of the subsystems in the framework of generalized multiple shooting takes less time than the average time per system needed to integrate the whole system without decomposition. The convergence rate of an SQP method, depending on how the Lagrange Hessian is approximated, is

at least locally superlinear. In order to decrease the time needed by integrators to solve differential equations and calculate sensitivities, one can use an inexact SQP method. Although, this might be economical in terms of integration time, but the solution of the NLP might take longer, since with inexact derivatives an SQP method has only locally linear convergence.

Chapter 2

Distributed Model Predictive Control of Linear Continuous-Time Systems

This chapter discusses the results published in [4]. A new sensitivity-driven distributed model-predictive control (S-DMPC) method is introduced, which is based on a novel distributed optimization algorithm, relying on a coordination mechanism. Coordination and therefore overall optimality is achieved by means of a linear approximation of the objective functions of neighboring controllers within the objective function of each local controller. As for most of the distributed optimization methods, an iterative solution of the distributed optimal control problems is required. An analysis of the method with respect to its convergence properties is also discussed. The resulting DMPC method is applied to a simulated chemical process to illustrate its capabilities.

2.1 Problem formulation

We consider a system Σ consisting of N linear time-invariant subsystems Σ_i:

$$\dot{x}_i(t) = \sum_{j=1}^{N} A_{ij}x_j(t) + B_{ij}u_j(t), \quad x_i(0) = x_{i,0}, \qquad \forall i \in \{1,\dots,N\}. \tag{2.1}$$

t denotes time. $x(t) = \langle x_1(t),\dots,x_N(t)\rangle$[1], with $x_i(t) \in \mathbb{R}^{n_{x_i}}$, is the state vector with initial condition $x_0 = \langle x_{i,0},\dots,x_{N,0}\rangle$ and $u(t) = \langle u_1(t),\dots,u_N(t)\rangle$ is the aggregated input vector, where $u_i(t) \in \mathbb{R}^{n_{u_i}}$ is the local input vector of subsystem Σ_i. $A = [A_{ij}]_{i,j\in\{1,\dots,N\}}$, with $A_{ij} \in \mathbb{R}^{n_{x_i}\times n_{x_j}}$, refers to the system matrix, and $B = [B_{ij}]_{i,j\in\{1,\dots,N\}}$, with $B_{ij} \in \mathbb{R}^{n_{x_i}\times n_{u_j}}$, denotes the input matrix.

[1] $\langle a,\dots,b\rangle$ is used as a shorthand for $[a^T,\dots,b^T]^T$.

DMPC aims at the solution of the optimal control problem for the overall system Σ, i.e.

$$\min_u \Phi = \sum_{i=1}^{N} \Phi_i(x_i, u_i) \tag{2.2a}$$

$$\text{s.t. } \Phi_i = \frac{1}{2} \int_{t_0}^{t_f} x_i^T Q_i x_i + u_i^T R_i u_i \, dt, \quad i = 1, \ldots, N \tag{2.2b}$$

$$\dot{x} = Ax + Bu, \quad x(t_0) = x_0, \tag{2.2c}$$

$$0 \leq D\langle x, u \rangle + e, \tag{2.2d}$$

on a finite (receding) horizon $[t_0, t_f]$, with a separable (or additive), quadratic objective function, symmetric positive definite weighting matrices Q_i and R_i, $i \in \{1, \ldots, N\}$, with appropriate dimensions, and with[2]

$$D = \langle D_1, \ldots, D_N \rangle, \tag{2.2e}$$

$$e = \langle e_1, \ldots, e_N \rangle. \tag{2.2f}$$

2.2 Distributed Optimization Strategy

The basis for the sensitivity-driven distributed model-predictive control proposed is a new distributed optimization method. The open-loop optimal control problem (2.2) is transcribed into a quadratic programming problem (QP) in two steps :

1. The input vector functions $u_i(t)$ are approximated by an expansion in a complete function space spanned by the base functions $\phi_l(t)$. B-splines provide a flexible class of bases, which includes the piecewise constant representation used exemplarily in this work. In particular, the input functions are approximated by finite sums $u_{i,j}(t) = \sum_l p_{i,j,l} \, \phi_l(t)$, where i refers to the subsystem and j to the j-th component of the corresponding input $u_i(t)$. The parameters $p_{i,j,l}$ are organized in parameter vectors $p_i = \langle p_{i,1}, \ldots, p_{i,n_{u_i}} \rangle$ with $p_{i,j}$ comprising the parameters associated with the j-th scalar input $u_{i,j}(t)$ of system i and in the parameter vector $p = \langle p_1, \ldots, p_N \rangle$ concatenating the control vector parametrization of the overall system.

2. The state equations are solved analytically and the objective function and the constraints are evaluated.

This transcription is straightforward to result in the quadratic program (QP)

$$\min_p \sum_{i=1}^{N} \Phi_i(p) \tag{2.3a}$$

$$\text{s.t. } \Phi_i(p) = \frac{1}{2} p^T A^i p + p^T B^i + C^i, \tag{2.3b}$$

$$c_i(p) = D^{i\,T} p + E^i \geq 0, \tag{2.3c}$$

The distributed solution of QP (2.3) requires its decomposition and the coordination of the resulting subproblems, which are described for a convex nonlinear program (a generalization of QP (2.3), if this QP is convex) in the remainder of this section. The coordination algorithm for a parametric nonlinear optimization problem generalizing QP (2.3) is stated as follows:

Algorithm 1

1. *Choose feasible parameters $p^{[0]}$, and an initial guess of the Lagrange parameters $\lambda^{[0]}$ and set $k := 0$.*

2. *The control parameters $p_i^{[k]}$ and Lagrange parameters $\lambda_i^{[k]}$, $\forall \in \{1,\ldots,N\}$, are communicated to all local controllers.*

3. *Solve the local optimization problems*

$$\min_{p_i} \Phi_i^*(p) \tag{2.4a}$$

$$s.t.\ c_i(p) \geq 0, \tag{2.4b}$$

with the strictly convex objective functions

$$\Phi_i^* = \Phi_i(p) + \left[\sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{\partial \Phi_j}{\partial p_i}\bigg|_{p^{[k]}}^{T} - \lambda_j^{[k]\ T}\ \frac{\partial c_j}{\partial p_i}\bigg|_{p^{[k]}} \right] (p_i - p_i^{[k]}) \tag{2.4c}$$

to obtain the local minimizers $p_i^{[k+1]}$ and the Lagrange multipliers $\lambda_i^{[k+1]}$, $\forall i \in \{1,\ldots,N\}$. Note, that the solution of the local optimization problems for $p_i^{[k+1]}$ assumes that all other control parameters are fixed at the previous iterates $p_j^{[k]}$, $\forall j \neq i$.

4. *Set $k := k+1$ and go back to 2.*

5. *Stop, if $p^{[k]}$ satisfies some convergence criterion.*

Here, k refers to the iteration index, and $\frac{\partial \Phi_j}{\partial p_i}$ and $\frac{\partial c_j}{\partial p_i}$ are the first-order sensitivities of the objective function and the inequality constraints, respectively, corresponding to subsystem Σ_j, $j \neq i$, with respect to the control parameters p_i. c_j are the constraint functions related to system Σ_j, and $\lambda_j^{[k]}$ are the Lagrange multipliers at iteration k related to the j-th NLP.

2.3 Convergence rate of coordination algorithm

It can be easily shown that Algorithm 1 with certain assumptions converges to the optimal solution of the centralized problem. Moreover, in order to improve the convergence properties or even enforce convergence of the iteration, the objective function (2.4c) can be extended to become

$$\Phi_i^+ = \Phi_i^* + \frac{1}{2}(p_i - p_i^{[k]})^T \Omega^i (p_i - p_i^{[k]}), \tag{2.5}$$

with a symmetric positive definite matrix $\Omega_i \in \mathbb{R}^{n_{p_i} \times n_{p_i}}$. If series of iterates $p^{[k]}$ converge, it can be shown that the rate of convergence is linear and the contraction parameter depends on the system matrices A_{ij} and matrix D.

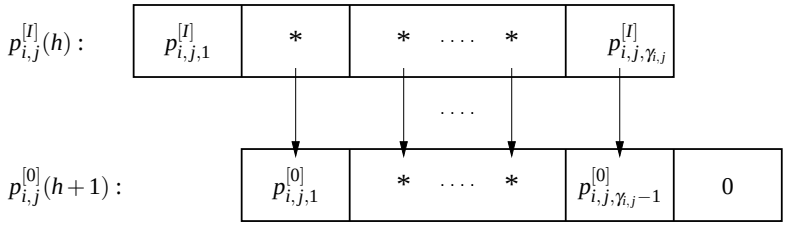


Figure 2.1: Reinitialization of parameters p for piecewise-constant input parametrization.

2.4 Hot-start in S-DMPC

In a moving horizon environment, on a new horizon $h+1$, an almost optimal initialization of the method, i.e. of the parameters $p^{[0]}(h+1)$ is possible by means of the old parameter set $p^{[I]}(h)$, as the QP change only slightly from one moving horizon to the next. The parameters of the piecewise-constant input functions can be reinitialized as depicted in Figure 2.1. Let $p_{i,j,l}$, $l = 1,\ldots,\gamma_{i,j}$, represent the parametrization of the j-th input function $u_{i,j}(t)$ of subsystem Σ_i, where $\gamma_{i,j}$ denotes the number of parameters for input function $u_{i,j}(t)$. Then, all parameters but the first one, which has already been applied to the plant, are reused. The last parameters $p_{i,j,\gamma_{i,j}}$, $\forall i,j$, describing $u_{i,j}(t)$ are initialized to zero. The Lagrange multipliers are initialized with $\lambda_i^{[0]} = 0$, $\forall i \in \{1,\ldots,N\}$, assuming all inequality constraints to be inactive. In order to ensure a feasible initialization, well-known methods such as the phase-I/phase-II approach can be adapted for S-DMPC.

2.5 Numerical results

We consider a simulated chemical process for the alkylation of benzene as depicted in Figure 2.2 to illustrate the performance of the S-DMPC method. The mathematical model consists of material

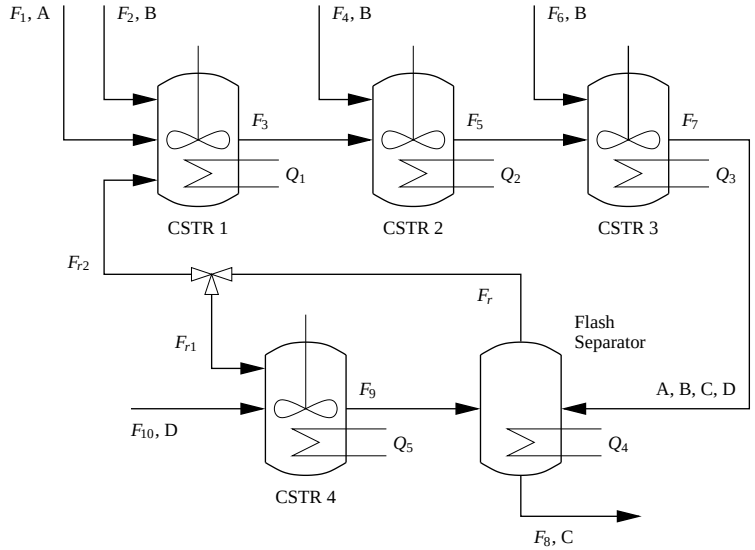


Figure 2.2: Process flow diagram for alkylation of benzene.

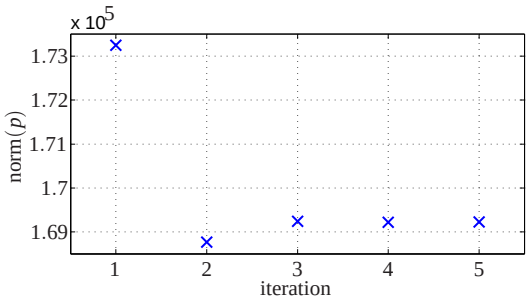


Figure 2.3: Convergence of the distributed method.

balances for each component and an energy balance for each unit of the plant. In addition, the model includes nonlinear reaction kinetics as well as a nonlinear description of the phase equilibrium in the flash separator, leading to a total of approximately 100 equations. The heat flows are assumed to be the manipulated variables $u_i(t)$. The derivation of the model and all data are available in [4]. The goal is to stabilize the plant at its steady-state operating point, while the nonlinear model is linearized around this point. The linearized model is used as the internal model of the controller, while the nonlinear model is used to simulate the plant. The numerical convergence of the method is experimentally analyzed for a set-point change, as depicted in Figure 2.3. The diagram shows the evolution of the 2-norm of p for an increasing number of iterations on a fixed horizon. In this example, convergence is achieved already after 3 iterations.

Chapter 3

Distributed Model Predictive Control of Nonlinear Continuous-Time Systems

In the present chapter we will summarize the idea published in [3] and also introduce developments. Our main goal is to solve a reasonably large nonlinear optimal control problem with multiple shooting for which the centralized solution may not be an option due to the huge computational effort needed to simulate the problem by integrators. In direct methods most of the computation time is spent by integration and the generation of sensitivities. This is the reason why we concentrate on integration and extend standard multiple shooting, while we make use of a centralized controller.

3.1 Problem formulation

We are concerned with systems decomposed into M subsystems,

$$\begin{cases} \dot{x}^i(t) = f^i(x^i(t), u^i(t), z^i(t)) \\ y^i(t) = g^i(x^i(t), u^i(t), z^i(t)) \end{cases} \qquad i = 1, \ldots, M, \tag{3.1}$$

where $x^i(t), u^i(t), z^i(t)$ and $y^i(t)$ denotes the i-th subsystem's states, control inputs, coupling input and system output, respectively. The systems are coupled by

$$z^i(t) = A^i r^i(t) + \sum_{j=1}^{M} A^i_j y^j(t), \tag{3.2}$$

which is a generalization of input-output connections. We would like to solve the following optimal control problem

$$\min_{\substack{x,u,z,\\ y,e}} \quad \int_0^T \ell(e(t))dt + \sum_{i=1}^{M} \int_0^T \ell^i(x^i(t),u^i(t),z^i(t))dt \tag{3.3}$$

$$\text{s.t.} \qquad \dot{x}^i(t) = f^i(x^i(t),u^i(t),z^i(t)) \tag{3.4}$$

$$y^i(t) = g^i(x^i(t),u^i(t),z^i(t)) \tag{3.5}$$

$$x^i(0) = \vec{x}_0^i \tag{3.6}$$

$$z^i(t) = A^i r(t) + \textstyle\sum_{j=1}^{M} A^i_j y^j(t) \tag{3.7}$$

$$Ce(t) = Br(t) + \textstyle\sum_{i=1}^{M} B^i y^i(t) \tag{3.8}$$

$$p^i(x^i(t),u^i(t)) \geq 0, \quad q(e(t)) \geq 0 \quad t \in [0,T]. \tag{3.9}$$

Here (3.3) is the stage cost and a penalty on the $e(t)$ variables, (3.6) is the initial value, (3.7) is the coupling constraint, (3.8) stores the linear combination of a global input signal $r(t)$ and the output signals. Note that $r(t)$ might be a reference signal that must be followed by the system outputs in an appropriate way, thus our formulation incorporates tracking problems as well. Finally, (3.9) denotes local state and input constraints and restrictions to $e(t)$. An application that fits into this framework is discussed later in this chapter.

3.2 Discretization and NLP formulation

An efficient way to solve this problem is multiple shooting, which divides the control horizon $[0,T]$ into smaller intervals and introduces initial values in each shooting interval, while one has to introduce constraints that the propagated states at the end of each shooting interval must be equal to the initial value of the following shooting interval. To discretize $u^i(t)$, we use a constant within every shooting interval, while $x(t)$ is discretized by an underlying integrator, only initial values are determined by the optimization routine. The signals $z(t)$ and $y(t)$ are discretized in a finite dimensional space by using polynomial approximation, i.e.

$$z^i_p(t) = \Gamma_n(t)^T \mathbf{z}^i_{n,p}, \tag{3.10}$$

where $z^i_p(t)$ denotes the p-th coordinate of the coupling input of the i-th subsystem and $\Gamma_n(t)$ is the n-order basis composed by Legendre polynomials. Signals $r(t)$ and $e(t)$ are discretized with the same approach and this already results in an NLP over variables $u^i_n, x^i_n, \mathbf{z}^i_n \mathbf{y}^i_n, \mathbf{e}_n, \quad i = 1,\ldots,N$.

$$\begin{aligned}
\min_{\substack{u^i_n,x^i_n,\mathbf{z}^i_n,\\ \mathbf{y}^i_n,\mathbf{e}_n}} \quad & \sum_{n=0}^{N-1} \left(L_n(\mathbf{e}_n) + \sum_{i=1}^{M} L^i_n(x^i_n,u^i_n,\mathbf{z}^i_n) \right) \\
\text{s.t.} \quad & x^i_{n+1} = F^i_n(x^i_n,u^i_n,\mathbf{z}^i_n) \qquad n = 0,\ldots,N-1 \\
& \mathbf{y}^i_n = G^i_n(x^i_n,u^i_n,\mathbf{z}^i_n) \qquad n = 0,\ldots,N-1 \\
& x^i_0 = \vec{x}^i_0 \\
& \mathbf{z}^i_n = \textstyle\sum_{i=1}^{M} A_{ij} \mathbf{y}^j_n \\
& \mathbf{e}_n = \mathbf{r}_n + \textstyle\sum_{i=1}^{M} B_{ij} \mathbf{y}^j_n \\
& p^i(x^i_n,u^i_n) \geq 0, \qquad Q_n(\mathbf{e}_n) \geq 0
\end{aligned} \tag{3.11}$$

Note that generalized multiple shooting decomposes the problem not only in time, but also in space. By this, we can integrate the subsystems (i.e. evaluate $F_n^i(\cdot)$ and its derivatives) totally independently, with various approaches. This ensures high modularity and parallelizability of our method. As the optimization method proceeds, the gap between states and also polynomials are shrinking.

The solution of (3.11) might happen with a standard NLP solver, but we must keep in mind that the dimension of this problem is large, thus relatively small number of iterations is a demand, while the structure should be exploited. This is the reason why we choose a sequential quadratic programming (SQP) method. The linearization of (3.11) ends up in a quadratic program, which has sparsity pattern shown in Figure 3.1. This feature might be exploited by a sparse QP solver. The state-of-the-art sparse

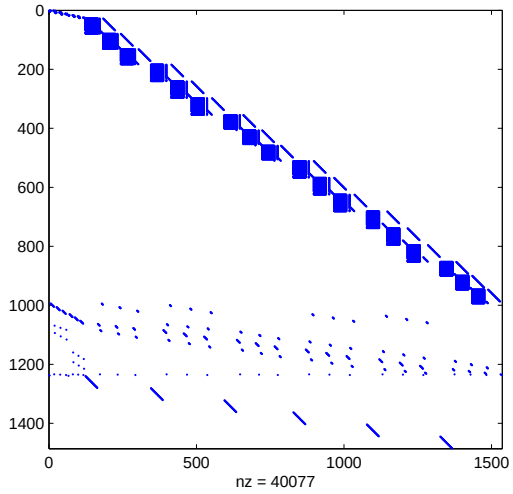


Figure 3.1: Sparsity pattern of equality conditions of a quadratic program arising in generalized multiple shooting

QP solvers are already able to cope with large-scale problems, so we only have to make sure to carry out minimal number of SQP iterations (i.e. number of linearizations and QP solutions). Optimization theory can provide us a result how fast this might be, which we discuss in the following section.

3.3 Convergence rate of generalized multiple shooting

The number of SQP iterations is crucial in terms of the applicability of our approach. If the proposed method is applied in Model Predictive Control (MPC), the time available for solution of an optimal control problem is strictly limited and hence we have expectation of fast convergence.

Our method is capable of fulfilling the demand mentioned above. First we define which type of convergence rates we are interested in. We assume that $a_k \to 0$ and say a_k converges

- sublinearly if $a_k = O(\frac{1}{k^\alpha}), \alpha \in [0, \infty]$,
- linearly if $a_k = O(\alpha^k), \alpha \in [0,1]$,
- superlinearly if $\limsup_{k\to\infty} \frac{a_{+1}}{a_k} = 0$,

- quadratically if $a_k = O(\alpha^{2^k}), \alpha \in [0,1]$.

Depending on the type of objective function the convergence rate of our method may be determined by applying SQP theory [2]. If we have a general nonlinear objective function the Hessian of the Lagrangian (i.e the second order term of the QPs) can be approximated by doing an inexpensive BFGS update, which results in locally superlinear convergence. In case of least-squares objective (i.e. Gauss-Newton objective), by the use of the Jacobian we can approximate the Hessian. This way we have locally linear convergence, but at the end of the iterations quadratic convergence can be reached if we have a perfect fit.

The convergence of the method may be even more developed by hot-starting approaches. One might want to use fast MPC techniques like initial-value embedding and real-time iterations [1] in order to help the controller to recover the optimal control trajectories in case of perturbation.

3.4 Numerical results

The proposed method was implemented to control a hydro power plant (see Figure 3.2), which is composed by a river and three lakes. The river is divided into 6 reaches which terminate with dams equipped with turbines for power production. We regard every reach together with the following dam as a separate subsystem (R1 + D1 , R2 + D2 , R3 +D3 , R4 +D4 , R5 +D5 , R6 +D6). Lakes L1 and L2 are connected by a duct (U1) and lake L1 is connected to the river by duct with a turbine (T1) and a duct with a turbine and a pump (C1). Lakes L1 and L2 form together with U1 , C1 and T1 another subsystem. Lake L3 is connected to the river by a duct with a turbine (T2) and a duct with a turbine and a pump (C2). L3 , C2 and T2 compose the last subsystem. The hydro power plant is composed of 8 subsystems, and it has in total 249 states, 14 inputs, 18 coupling inputs and 18 coupling outputs.

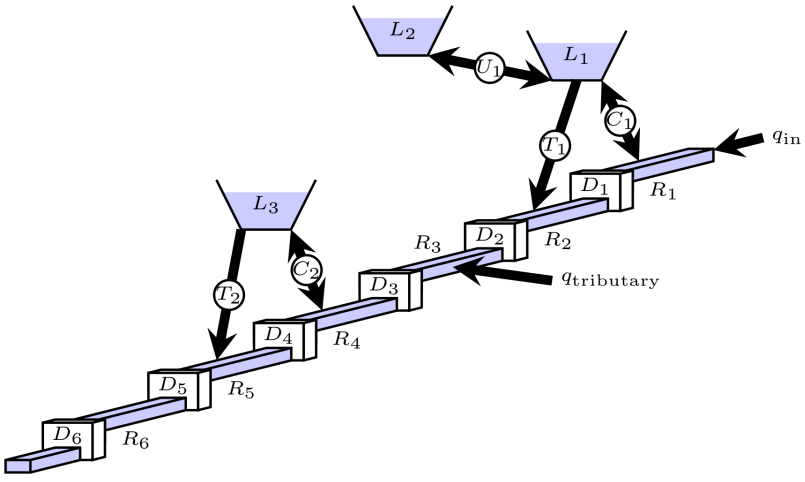


Figure 3.2: Scheme of a hydro power plant controlled by generalized multiple shooting

An optimal control problem of the above problem was solved with 48 shooting intervals, altogether 24 hours of control horizon on a desktop computer with 16 cores. The L_1 cost penalizes the deviation between the produced power and the power reference (see Fig 3.3), the L_2 cost corresponds to the change of water levels from a set point, while respecting all local state and control input constraints. The power reference was tracked with a maximum of 1.71E-4 % relative error as a result of

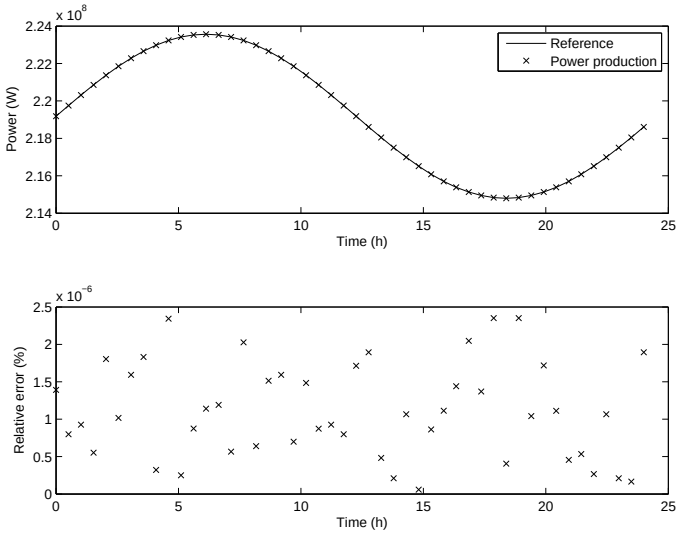


Figure 3.3: The power reference (cont. line) and the produced power (crosses) above and the relative error of the tracking below on the whole control horizon of 24 hours.

21 SQP iterations. This optimization approach took altogether 34 minutes 56 seconds. We also give a lower bound to the runtime of standard multiple shooting. The integration of the whole system on the 48 shooting intervals would take 27 minutes 10 seconds, thus to solve an optimal control problem we would need at least 9 hours 30 minutes and 32 seconds, which is a considerable difference in runtime.

Bibliography

[1] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H.G. Bock, T. Bürner, E.D. Gilles, A. Kienle, J.P. Schlöder, and E. Stein. Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 363–384. Springer, 2001. download at: http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-16.html.

[2] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[3] C. Savorgnan, C. Romani, and A. Kozma. Multiple shooting for distributed systems with applications in hydro electritcity production. *Journal of Process Control*, 2011. Accepted for publication.

[4] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *Journal of Process Control*, 2011. Accepted for publication.