Nonlinear Model Predictive Control Using Multiple Shooting Combined with Collocation on Finite Elements

Jasem Tamimi and Pu Li

Simulation and Optimal Processes Group, Institute of Automation and Systems Engineering, Ilmenau University of Technology, P. O. Box 10 05 65, 98684 Ilmenau, Germany. (Tel.:+49-3677-691427, e-mail:{jasem.tamimi/pu.li}@tu-ilmenau.de)

Abstract: A new approach to nonlinear model predictive control (NMPC) is proposed in this paper. The multiple shooting method is used for discretizing the dynamic system, through which the optimal control problem is transformed to a nonlinear program (NLP). To solve this NLP problem state variables and their gradients at the end of each shooting need to be computed. Here we employ the method of collocation on finite elements to carry out this task. Due to its high numerical accuracy, the computation efficiency for the integration of model equations can be enhanced, in comparison to the existing multiple shooting method where an ODE solver is applied for the integration and the chain-rule for the gradient computation. The numerical solution framework is implemented in C++. Two examples are taken to demonstrate the effectiveness of the proposed NMPC algorithm.

Keywords: Optimal control, NMPC, multiple shooting, collocation on finite elements.

1. INTRODUCTION

Solving optimal control problem is highly motivated nowadays, since these solutions are very important in almost all industrial fields such as chemical, electrical, mechanical, and economical systems. One of the optimal control algorithms is MPC which refers to a class of computer control strategies that utilize an explicit process model to predict the future response of the plant (Qin and Badgwell, 2003). MPC, also known as receding horizon control, has the ability to handle input as well as output constraints and transparent tuning capabilities (Gatlu and Zafiriou, 1992).

The main goal of MPC is to find an optimal vector of control functions that minimize or maximize a performance index subject to a given process model (usually a nonlinear differential equation system) as equality constraints, and boundary conditions as inequality constraints on the states and controls. Simple problems can be solved by the so-called *indirect method* which is based on the first order optimality condition of variation (Diehl *et al.,* 2006, Schäfer *et al.*, 2007). This leads to a two-point-boundary value problem in ordinary differential equations (ODE). For more details see e.g. Bryson and Ho (1975), Kirk (1970), and Lewis and Syrmos (1995).

On the other hand, the *direct method* which follows the philosophy of "first discretize then optimize", transforms the optimal control problem into a NLP problem which can then be solved by the method of sequential quadratic programming (SQP). In this way inequality constraints and equality path constraints can be easily treated, and we can also successfully deal with highly nonlinear complex optimal control problems (Diehl *et al.*, 2006, Schäfer *et al.*, 2007).

In all direct methods, the control trajectory will be parameterized and the state trajectories computed using either sequential or simultaneous approaches. In the *sequential approach*, the state variables are considered as an implicit function of control trajectories, where the ODEs are addressed as an initial value problem using one of the dedicated integration methods like Runge-Kutta or Euler algorithms (Sargent and Sullivan, 1977; Kraft, 1985; Biegler *et al.,* 2002). In the *simultaneous approach*, state trajectories are parameterized, too, and we deal with all of parameterized variables (states and controls) as optimization variables in the NLP. The ODEs will be represented as equality constraints, either with collocation on finite elements (Biegler *et al.,* 2002; Hong *et al.,* 2006; Li, 2007) or with multiple shooting (Bock and Plitt, 1984; Leineweber, 1995; Diehl, 2001; Diehl *et al.,* 2002a; Diehl *et al*., 2002b).

In this work, we propose a new approach to the solution of nonlinear model predictive control (NMPC) problems. This control strategy is a combination of the multiple shooting and the collocation method. We use multiple shooting for discretizing the dynamic system, so that the optimal control problem is transformed to a NLP problem in which continuity conditions in each shooting are considered as equalities and state constraints at the end of each shooting as inequalities. To solve this NLP problem the values of state variables and their gradients at the end of each shooting have to be computed. Here we employ collocation on finite elements to carry out this task. Due to its high numerical accuracy, the computation efficiency for the integration of the ODEs can be enhanced, in comparison to the existing multiple shooting method where an ODE solver is applied for the integration and chain-rules for the gradient computation. We implement the proposed approach with a numerical solution framework in C++. Two examples are taken to demonstrate the effectiveness of the proposed NMPC algorithm. The results from our approach are compared with those achieved from the multiple shooting method (using the software MUSCOT II (Diehl *et al.*, 2001)).

2. NONLINEAR MODEL PREDICTIVE CONTROL

2.1 Optimal control problem

We will consider the following optimal control problem

$$\min \int_{t_0}^{t_f} L(x(t), u(t), t)dt + E\left(x(t_f)\right)$$

s.t.

(i) $x(t_0) = x(0)$,

(ii) $\dot{x}(t) = f(x(t), u(t), t),\ t \in [t_0, t_f]$ (1)

(iii) $g(x(t), u(t), t) \geq 0\ ,\ t \in [t_0, t_f]$

(iv) $r\left(x(t_f)\right) = 0$,

where $x(t), u(t)$ are the state and control variables, respectively, t_0 and t_f are initial and final time of the receding horizon, and constraint (i) is the initial value condition, (ii) the nonlinear ODE model, (iii) the path constraint, and (iv) the terminal constraint.

2.2 Direct multiple shooting scheme

The direct multiple shooting algorithm proposed by Bock and Plitt (1984) for solving problem (1) can be summarized in the following steps:

1) Discretize the time horizon $[t_0, t_f]$ into equal subintervals $[t_i, t_{i+1}]$, such that

$$t_0 < t_1 < \cdots < t_n = t_f \quad (2)$$

where *n* is the total number of subintervals.

2) Parameterize the control function $u(t)$ for each subinterval:

$$u(t) = v_i \quad for \quad t \in [t_i, t_{i+1}] \quad (3)$$

$$i = 0,1, \dots, n-1$$

3) Parameterize the initial condition of the state vector for each subinterval:

$$x(t_i) = h_i \quad (4)$$

$$i = 0,1, \dots, n-1$$

4) Evaluate the state trajectories in each subintervals and the value of h_i from the final state subinterval considering the parameterized state initial value in the previous step:

$$\dot{x}_i(t) = f(x_i(t), v_i, t),\ t \in [t_i, t_{i+1}] \quad (5a)$$

$$x_i(t_i) = h_i \quad (5b)$$

5) Define the continuity constraints:

$$h_{i+1} - x_i(t_{i+1}; h_i, v_i) = 0 \quad (6)$$

6) Compute the objective function for each subinterval, so we need to solve the following NLP

$$\min_{h_i, v_i} \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} L(x_i(t), v_i)dt + E(x_n(t_n))$$

s.t. (7)

$$h_0 - x(0) = 0$$

$$h_{i+1} - x_i(t_{i+1}; h_i, v_i) = 0,\ i = 0,1, \dots, n-1,$$

$$g(h_i, v_i) \geq 0$$

Eq. (7) can be described as

$$\min_w A(w) \qquad s.t. \quad \begin{cases} B(w) = 0 \\ C(w) \geq 0 \end{cases} \quad (8)$$

where $w = [h_0, v_0, h_1, v_1, \dots, h_n, v_{n-1}]$,

$$B(w) = \begin{bmatrix} h_0 - x(0) \\ h_1 - x_0(t_1; h_0, v_0) \\ \vdots \\ h_{n-1} - x_{n-2}(t_{n-1}; h_{n-2}, v_{n-2}) \end{bmatrix},$$

$$C(w) = \begin{bmatrix} g(h_0, v_0) \\ g(h_1, v_1) \\ \vdots \\ g(h_{n-1}, v_{n-1}) \end{bmatrix}.$$

We can use the spars nonlinear optimizer (SNOPT) to solve the above NLP problem. In SNOPT equality constrains will be transformed into inequality constraints by introducing a set of slack variables, i.e.

$$\min_w A(w)$$

s.t. (9)

$$\left(\begin{pmatrix} B(w) \\ C(w) \end{pmatrix} - s = 0\right), \text{ and } \ l \leq \begin{pmatrix} w \\ s \end{pmatrix} \leq u$$

where $s = (s_0, \cdots, s_{n-1}, s_n, \cdots s_{2n-2})^T$. For more information on SNOPT see Gill *et al.* (2005) and Gill *et al.* (2008). Consequently, problem (8) can now be rewritten as:

$$\min_w A(w)$$

s.t. (10)

$$l \leq D(w) \leq u$$

$$\text{where, } D(w) = \begin{bmatrix} h_0 - x(0) \\ h_1 - x_0(t_1; h_0, v_0) \\ \vdots \\ h_{n-1} - x_{n-2}(t_{n-1}; h_{n-2}, v_{n-2}) \\ g(h_0, v_0) \\ g(h_1, v_1) \\ \vdots \\ g(h_{n-1}, v_{n-1}) \end{bmatrix},$$

$$l = \begin{bmatrix} l_0 \\ \vdots \\ l_{n-1} \\ l_n \\ \vdots \\ l_{2n-2} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \ and \ u = \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \\ u_n \\ \vdots \\ u_{2n-2} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \infty \\ \vdots \\ \infty \end{bmatrix}.$$

2.3 SQP iteration

Fig. 1 shows all of the information needed for each SQP iteration.

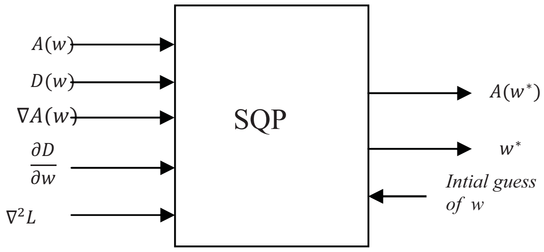

Fig. 1: Inputs and outputs of each SQP iteration.

In Fig. 1, $\nabla A(w)$ and $\frac{\partial D}{\partial w}$ are the gradient of the objective function and Jacobian of the equality constraints in (7), respectively. $A(w^*)$ *and* w^* are the objective function value and the optimization variables at the solution. $\nabla^2\mathrm{L}$ denotes the Hessian of the Lagrangian. The sensitivity information, i.e. $\nabla A(w)$ and $\frac{\partial D}{\partial w}$, plays the most important role in the SQP iteration and its computation requires much CPU-time. In the existing multiple shooting algorithm it is done by integrating the ODEs with an ODE solver and then using the chain-rule for the sensitivity computation. In this work we employ the method of collocation on finite elements to carry out the ODE integration and compute these sensitivities for each shooting. This proposed method is described in the next section.

3. SOLVING ODE AND SENSITIVITIES

To solve NLP (9) we have to solve the set of ODEs (5a). If we use piece-wise constant parameters for v_i, we can rewrite the ODE in each subinterval as

$$\begin{pmatrix} \dot{x}_i(t) \\ \dot{v}_i \end{pmatrix} = \begin{pmatrix} f(x_i(t), v_i, t) \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{z}_i = f(z_i(t), t) \\ z_i(t_i) = [x_i \; v_i]^T \end{pmatrix} \qquad (11)$$

Using collocation method the state variables $z_i(t)$ will be approximated by the following Lagrangian polynomials (Finlayson, 1980)

$$\mathrm{z}(t) = \sum_{j=0}^{M} \left[\prod_{\substack{k=0 \\ j \neq k}}^{M} \frac{(t - t_k)}{(t_j - t_k)} \right] z_j \qquad (12)$$

where *M* is the number of the collocation points. Using the three-point-collocation to compute the vector *z*, we yield

$$z(t) = T_0 z_0 + T_1 z_1 + T_2 z_2 + T_3 z_3 \qquad (13)$$

where $T_j = \prod_{\substack{k=0 \\ k \neq j}}^{3} \frac{t - t_k}{t_j - t_k}$.

To define the end time point of a subinterval to be the beginning point of the next one, we yield inside each shooting

$$t_0 = t_i, \qquad t_1 = \alpha_1(t_{i+1} - t_i),$$
$$t_2 = \alpha_2(t_{i+1} - t_i), \text{ and } t_3 = t_{i+1} \qquad (14)$$

where $\alpha_1 = 0.127$ and $\alpha_2 = 0.5635$. Then

$$\dot{T}_{i,k} Z_{i,k} + \dot{T}_{i,0} Z_{i,0} - f_{i,k}(z_i(t), t) = 0 \qquad (15)$$

where $T_{i,k} = \begin{bmatrix} T_{i,1}(t_1) & T_{i,2}(t_1) & T_{i,3}(t_1) \\ T_{i,1}(t_2) & T_{i,2}(t_2) & T_{i,3}(t_2) \\ T_{i,1}(t_3) & T_{i,2}(t_3) & T_{i,3}(t_3) \end{bmatrix}, Z_{i,k} = \begin{bmatrix} z_{i,1} \\ z_{i,2} \\ z_{i,3} \end{bmatrix}$

$$T_{i,0} = \begin{bmatrix} T_{i,1}(t_0) & 0 & 0 \\ 0 & T_{i,2}(t_0) & 0 \\ 0 & 0 & T_{i,3}(t_0) \end{bmatrix}, Z_{i,0} = \begin{bmatrix} z_{i,0} \\ z_{i,0} \\ z_{i,0} \end{bmatrix}$$

We solve the nonlinear equations (15) on the collocation points by using the Newton-Raphson method to find $Z_{i,k}$ and z_i. The first Taylor-expansion of (15) leads to

$$\dot{T}_{i,k} \frac{\partial Z_{i,k}}{\partial z_{i,0}} + \dot{T}_{i,0} - \frac{\partial f_{i,k}(z_i(t), t)}{\partial z_{i,0}} = 0 \qquad (16)$$

We define $\frac{\partial Z_{i,k}}{\partial z_{i,0}} = \Psi_{i,k}$, then

$$\dot{T}_{i,k} \Psi_{i,k} + \dot{T}_{i,0} - \frac{\partial f_{i,k}(z_i(t), t)}{\partial z_{i,k}} \Psi_{i,k} = 0 \qquad (17)$$

or

$$\Psi_{i,k} = -\left[\dot{T}_{i,k} - \frac{\partial f_{i,k}(z_i(t), t)}{\partial z_{i,k}} \right]^{-1} \dot{T}_{i,0} \qquad (18)$$

In fact, equation (18) is a linear equation system and thus can be solved by a LU factorization using forward and backward substitution, for more details see Golub and van Loan (1996). From the computed value of $\Psi_{i,3}$ we receive the Jacobian $\frac{\partial D}{\partial w}$, since

$$\frac{\partial D}{\partial w} = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ & \Psi_{0,3} & I & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & & \Psi_{1,3} & I & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & & \Psi_{2,3} & I & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \Psi_{n-1,3} & I \end{bmatrix} \qquad (19)$$

where *I* is a unit matrix. In the same way, we can calculate the gradient vector of $\nabla A(w)$.

4. THE PROPOSED ALGORITHM

As we have seen above, the multiple shooting method depends mainly on the SQP iteration. Inside each SQP iteration the gradient values of the objective function and Jacobian of the constraints as well as the approximated Hessian need to be computed. Based on the theoretical development in Sections 2 and 3, we propose the following algorithm to solve the nonlinear optimal control problem.

Algorithm 1:

1. *Initialize SQP*
   1.1. *Time horizon.*
   1.2. *Subintervals.*
   1.3. *Upper and lower bounds for states, controls and constraints.*
   1.4. *Fixed initial value constraint.*
   1.5. *Initial guess.*

2. *Define the continuity constraints $B(w)$ (8).*
3. *Define the continuity constraints $C(w)$ (8).*
4. *Initialize the three collocation points for each subinterval (14).*
5. *Compute the constraint equations and their sensitivities*
    5.1. *Define collocation equations (15).*
    5.2. *Solve (15) using Newton-Raphson.*
    5.3. *Define sensitivity equations (1).*
    5.4. *Solve (17) using LU factorization.*
6. *Compute objective function and its sensitivity.*
7. *Solve SQP iteration*
    7.1. *If KKT is not satisfied go to 4.*
8. *End*

This algorithm is realized in the framework of the numerical algorithm group (NAG) library Mark 8 (Numerical Algorithms Group Ltd, 2005) and IPOPT (Wächter, 2008) for SQP and in C/C++ for the rest of computations.

5. A CASE STUDEIS

We consider the following optimal control problems to demonstrate the performance of the proposed algorithm.

Example 1: Batch reactor - temperature profile. Maximize yield of x_2 after one hour's operation by manipulating a transformed temperature $u(t)$. This example is taken from Diehl *et al.* (2001).

$$\max_{u,x1,x2} \quad x_2(t_f)$$

s.t. (20)

$$\dot{x}_1(t) = -(u(t) + \frac{u^2(t)}{2})x_1(t)$$
$$\dot{x}_2(t) = u(t)x_1(t), \qquad t \in [0,1]$$
$$x_1(0) = 1, x_2(0) = 0.$$
$$0 \le x_1(t), x_2(t) \le 1$$
$$0 \le u(t) \le 5$$

We discretize the dynamic system with 20 subintervals. The computation was done using a PC with an intel processor "Pentium 4", 3 GHz and 1G Byte RAM. The solution took 350 ms and provides the final value of objective function with $x_2(t_f) = 0.57329$. Fig. 2 shows the optimal control trajectory and Fig. 3 the corresponding state trajectory x_1 while x_2 is shown in Fig. 4. These profiles of states (x_1 and x_2) and optimal control trajectory are identical, by using both MSCOD II and the proposed algorithm.

If we solve this problem with different number of subintervals, e.g. 5, 10, 20, 40, 80 and 160 subintervals, we can note from the results, as shown in Table 1, that the number of optimization variables (z) and the number of constrains will be increased when the number of subintervals increases. The CPU-time will increase exponentially. However, if we compare the CPU-time taken by MUSCOD II with that of the proposed algorithm, it can be seen at a large number of subintervals (i.e. a high dimension of the NMPC), the proposed algorithm will be more effective.

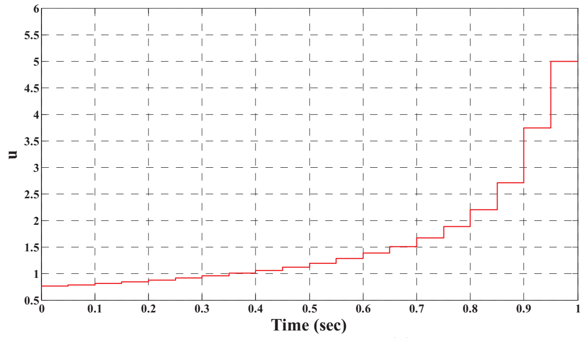

Fig. 2: The optimal control trajectory $u(t)$

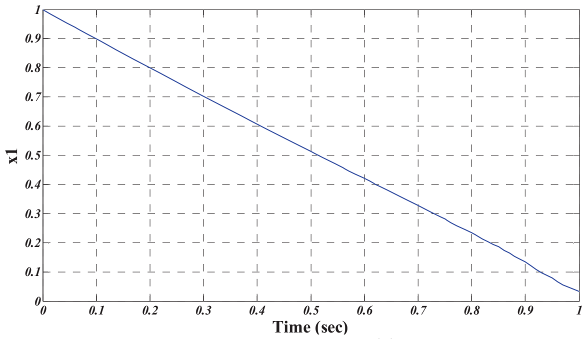

Fig. 3: The optimal state trajectory $x_1(t)$

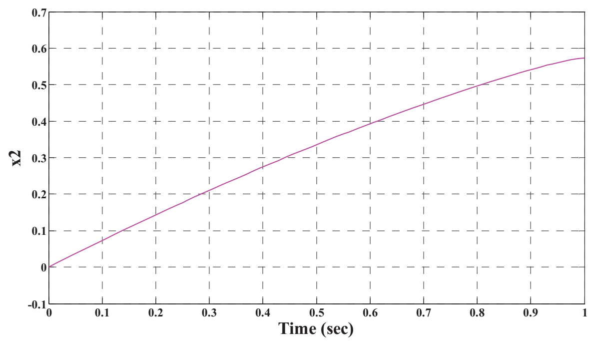

Fig. 4: The optimal state trajectory $x_2(t)$

Table 1: Results of using different number of subintervals

n	*z's*	*Co. eq.*	MUSCOD II		Algorithm 1	
			CPU-Time (ms)	*J*	CPU-Time (ms)	*J*
5	18	12	43	0.573117	188	0.568171
10	33	22	53	0.573080	290	0.572162
20	63	42	146	0.573527	350	0.573290
40	123	82	940	0.573544	480	0.573478
80	243	162	3620	0.573545	547	0.573528
160	483	322	21612	0.573545	735	0.573541

n: number of subintervals; *z's*: total number of variables; *Co. eq.*: total number of constraints; *J*: value of objective function.

Example 2: Optimal control of a continuous stirred tank reactor (CSTR): We consider a CSTR as shown in Fig. 5.

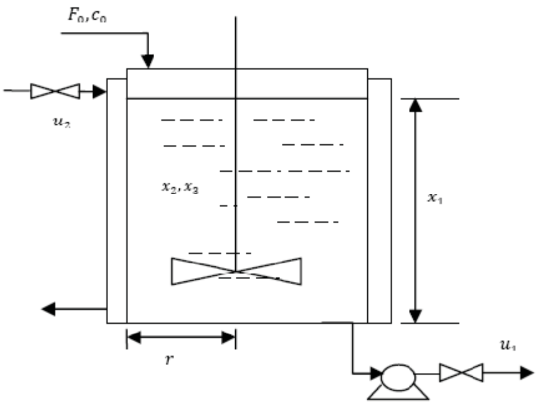

Fig. 5: CSTR example

An exothermic, irreversible, first order reaction A B occurs in the liquid phase and the temperature is regulated with external cooling. This highly nonlinear example is taken form Henson and Seborg (1997) or Pannocchia and Rawlings (2003) with the assumption that the level liquid is not constant. The constrained optimal control problem is formulated as follows:

$$\min_{x,u} \int_0^{t_f} [(x_1 - x_1^s)^2 + 100(x_2 - x_2^s)^2$$
$$+0.1(u_1 - u_1^s)^2 + 0.1(u_2 - u_2^s)^2]dt$$

s.t. (21)

$$\dot{x}_1 = \frac{F_0 - u_1}{\pi r^2}$$
$$\dot{x}_2 = \frac{F_0(c_0 - x_2)}{\pi r^2 x_1} - k_0 x_2 e^{-E/RT}$$
$$\dot{x}_3 = \frac{F_0(T_0 - x_3)}{\pi r^2 x_1} + \frac{-\Delta H}{\rho C_p} k_0 x_2 e^{-E/RT} + \frac{2U}{r\rho C_p}(u_2 - x_3)$$
$$x_1(0) = 0.659, \quad x_2(0) = 0.877 \text{ and } x_1(0) = 324.5$$
$$0.5 \le x_1 \le 2.5, \quad 0.8 \le x_2 \le 1.0$$
$$85 \le u_1 \le 115, \quad 299 \le u_2 \le 301$$

where x_1 is the level of the tank in meter, x_2 the product concentration in mol and x_3 the reaction temperature in (K), and the controls are u_1 and u_2 the outlet flow rate in (L/min) and coolant liquid temperature, respectively. In addition the inlet flow rate F_0 or the inlet concentration c_0 is acting as a disturbance to CSTR. The desired steady-state operating points: x_1^s, x_2^s, u_1^s and u_2^s are 0.659 meter, 0.877 mol/L, 100L/min and 300K, respectively. The model parameters in nominal conditions are shown in Table 2. We consider the operation case that at the tenth minute a disturbance enters the plant at a level of 0.05 mol/L on the inlet molar concentration c_0. A time horizon of $t_f = 50$ min is considered.

Table 2: Parameters of the CSTR

F_0	100 L/min	E/R	8750 K
T_0	305 K	U	915.6 $Wm^{-2}K^{-1}$
c_0	1.0 mol/L	ρ	1 kg/L
R	0.219	C_p	0.239 $Jg^{-1}K^{-1}$
k_0	0.219	ΔH	-5×10^4 J/mol

To solve problem (21) using the proposed algorithm we divide the time horizon into 50 subintervals and so that the number of resulted NLP will be 306 variables with 204 constraints, and the same PC is used to make the optimization. We used the IPOPT 3.4.0 to solve the NLP and NAG mark 8 to solve the Newton-Raphson equations and linear equation systems.

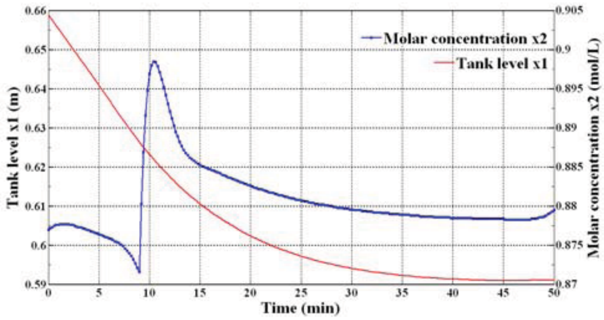

Fig. 6: The optimal output flow $x_1(t)$ and coolant temperature $x_2(t)$.

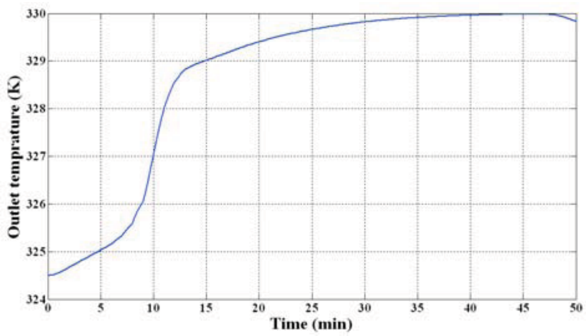

Fig. 7: The optimal outlet temperature $x_3(t)$.

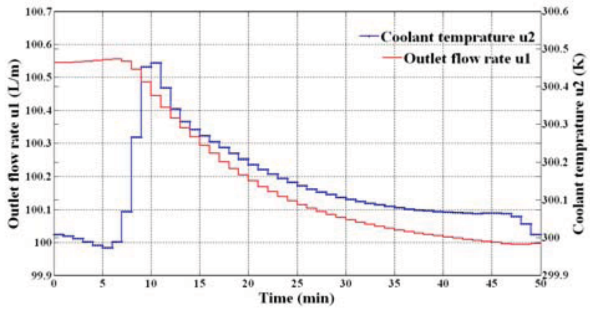

Fig. 8: The optimal control profiles $u_1(t)$ and $u_2(t)$.

Figures 6 and 7 show the optimal control profiles of the states $x_1(t)$, $x_1(t)$ and $x_3(t)$, respectively and Fig. 8 shows the optimal control profiles $u_1(t)$ and $u_2(t)$. The objective function value at the optimum is 0.9015886. Moreover, the algorithm was converged in 35 iterations and with the CPU-time in 0.954s. In comparison, this problem was also solved by Hong *et al.* (2006) using a quasi-sequential approach and it was converged in 16 iterations and 5.56 s of CPU time of SUN Ultra 10 Station with identical solutions.

6. CONCLUSIONS

In this paper we proposed a novel algorithm for NMPC. It is a combination of the multiple shooting, where the NLP problem will be handled, with the collocation method, where function values and gradients required in the NLP will be computed. We use piecewise constant for controls and the three-point collocation for states to parameterize the vector of optimization variables. The proposed algorithm has been realized in the framework of the numerical algorithm group (NAG) and IPOPT in the C/C++ environment. In addition, two demonstrative examples have been taken as case studies to show and compare the results from our algorithm and the well known MUSCOD II code. From these results it can be seen that the proposed algorithm is more efficient when a large-scale NMPC problem is to be solved. Stability and error control issues as well as practical applications of this algorithm will be considered in our future work.

7. ACKNOWLEDGEMENT

The financial support from the German Academic Exchange Service (DAAD) for this work is gratefully acknowledged.

8. REFERENCES

Biegler, L. T., Cervantes, A. M. and Wächter, A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*. 4(57), 575–593.

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

Bryson, A. E. and Ho, Y.C. (1975). *Applied Optimal control*, Hemisphere publication corporation, Levittown.

Diehl, M. (2001). Real-time optimization for large scale nonlinear processes. PhD Thesis, *University of Heidelberg*.

Diehl, M., Findeisen, R., Schwarzkopf, S., Uslu, I., Allgöwer, F., Bock, H.G., Gilles, E.D., and Schlöder, J.P. (2002a). An efficient algorithm for nonlinear model predictive control of large-scale systems. Part I: Description of the method. *At-Automatisierungstechnik.* 50(12), 557-567.

Diehl, M., Bock, H.G. Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002b) Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control.* 12(4)**,** 577-585.

Diehl, M., Schäfer, A. and Leineweber, D. B. (2001). *MOSCOD II user manual*, Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg.

Diehl, M., Bock, H.G., Diedam, H. and Wieber P.B. (2006). Fast direct multiple shooting algorithms for optimal robot control. LNCIS, *Fast Motions in Biomechanics and Robotics*, 340, 65-93.

Finlayson, B. A. (1980). *Nonlinear analysis in chemical engineering*, McGraw-Hill, New York.

Gill, P. E., Murray, W. and Saunders, M. A. (2002). SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM J. Optim*. 12(1). 979–1006.

Gill, P. E., Murray, W. and Saunders, M. A. (2008). *User's Guide for SNOPT version 7: Software for large-scale nonlinear programming.* Department of Mathematics, University of California, San Diego.

Gatlu, G. and Zafiriou, E. (1992). Nonlinear quadratic dynamic matrix control with state estimation. *Ind. Eng. Chem. Res*. 31(4), 1096-1104.

Golub, G. H. and van Loan, C. F. (1996). *Matrix computations.* (3rd ed.). Johns Hopkins University Press. Baltimore.

Henson, MA., Seborg, DE. (1997). *Nonlinear Process Control. Upper Saddle River.* NJ, Prentice Hall.

Hong, W. R., Wang, S. Q., Li, P., Wozny, G., Biegler, L.T. (2006), A quasi-sequential approach to large-scale dynamic optimization problems, *AIChE Journal*, 52(1), 255-268.

Kirk, D. E. (1970). *Optimal Control*, Prentice Hall Inc, Englewood Cliffs.

Kraft, D. (1985). On converting optimal control problems into nonlinear programming problems. *In K. Schittkowski. Computational Mathematical Programming,* vol. F15 of NATO ASI, 261–280. Springer.

Leineweber, D. B. (1995). Analyse und Restrukturierung eines Verfahrens zur direkten Lösung von Optimal-Steuerungsproblem, The Theory of MUSCOD II in a Nutshell. Master Thesis, University of Heidelberg.

Lewis, F. L., and Syrmos, V. L. (1995). *Optimal Control Theory*, Wiley Interscience.

Li, P. (2007). Prozessoptimierung: Methoden, Anwendungen und Herausforderungen, *Chemie Ingenieur Technik.* 79(10), 1567-1580.

Numerical Algorithms Group Ltd. (2005), *NAG C Library Manual, Mark 8.* Oxford, UK.

Pannocchia, G., Rawlings, J., Disturbance models for offset-free model predictive control. *AIChE J.* 49, 426-437

Qin, S.J., Badgwell, T.A. (2003). A survey of industrial model predictive control technology, *Control Engineering Practice.* 11(7), 1096-1104.

Sargent, R.W.H. and Sullivan, G.R. (1977). The development of an efficient optimal control package. In J. Stoer, *Proceedings of the 8th IFIP Conference on Optimization Techniques,* Part 2. **7**(1978), 158-168. Springer. Heidelberg.

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

Wächter, A. (2008), Introduction to IPOP: A tutorial for downloading, installing and using IPOPT. Department of Mathematical Sciences, IBM T.J. Watson Research Center, Yorktown Heights, NY.

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