Nonlinear model predictive control based on Bernstein global optimization with application to a nonlinear CSTR

Bhagyesh V. Patil[§] and Jan Maciejowski[†] and K. V. Ling[‡]

***Abstract*— We present a model predictive control based tracking problem for nonlinear systems based on global optimization. Specifically, we introduce a 'Bernstein global optimization' procedure and demonstrate its applicability to the aforementioned control problem. This Bernstein global optimization procedure is applied to predictive control of a nonlinear CSTR system. Its strength and benefits are compared with those of a sub-optimal procedure, as implemented in MATLAB using *fmincon* function, and two well established global optimization procedures, BARON and BMIBNB.**

I. Introduction

Model predictive control (MPC), also known as moving horizon control or receding horizon control, is an advanced control scheme for multivariable control systems. Typically, MPC derives a control signal by optimizing a pre-defined performance criterion repeatedly over a finite-time moving horizon within system constraints, and based on a dynamic model of the system to be controlled [1], [2]. MPC has been applied predominantly in the process industries, especially refining and petrochemicals [3]. An excellent survey of industrial applications of MPC can be found in [4], and references therein.

In practice, the majority of MPC applications employ linear models derived from system identification procedures, combined with linear inequality constraints. The MPC scheme in such instances is also known as 'linear MPC'. Linear MPC is widely preferred, due to its simplicity and the applicability of convex optimization algorithms (see [5] for instance). However, some processes may have either semi-batch characteristics, a large operating regime, or other source of significantly nonlinear behaviour. Approaches such as gain scheduling and switching between multiple linear models based on the operating region are possible approaches for such processes. On the other hand, use of a nonlinear process model may come with attractive benefits, such as higher product quality, tighter regulation of process parameters, and the possibility of operating the process (with a good control authority) in different operating regimes. Hence, model predictive control using nonlinear process models, usually called 'nonlinear MPC' (or NMPC), has attracted many researchers over the past decade [6], [7], [8].

An NMPC formulation requires the solution of a (usually *nonconvex*) nonlinear optimization problem at each sampling instant. As such, NMPC is a challenging field, and is dependent on good global optimization procedures. Motivated by this, in the present work we introduce one such global optimization procedure for NMPC applications. This procedure is based on the well-known Bernstein form of polynomials [9], and uses several nice properties associated with this Bernstein form. Optimization procedures based on this Bernstein form, also called *Bernstein global optimization algorithms*, have shown good promise to solve hard nonconvex NLP and MINLP problems (see, for instance, [10], [11], [12]). They are therefore very promising for NMPC applications.

In this work, we present one such Bernstein global optimization algorithm to solve a nonlinear optimization problem at each NMPC iteration. Specifically, we use the nonlinear system model for the predictions, followed by the formulation of a nonlinear programming (NLP) problem based on these predictions. Then, the Bernstein global optimization algorithm is used as a tool to solve this nonlinear optimization problem in terms of the control inputs (u's) as decision variables. The overall approach is demonstrated on a simulation study for predictive control of a nonlinear CSTR system, and the findings are compared with those of a sub-optimal procedure implemented in MATLAB using *fmincon* function, and two well established global optimization procedures, BARON and BMIBNB.

The rest of the paper is organized as follows. In Section II, we introduce a nonlinear MPC formulation. In Section III, we briefly describe the Bernstein form, followed by the presentation of the Bernstein global optimization algorithm. In Section IV, we report the simulation studies on a nonlinear CSTR system with the Bernstein global optimization algorithm and compare with MATLAB *fmincon* function, global optimization procedures, BARON and BMIBNB. Finally, in Section V, we present some concluding remarks.

II. Problem formulation for NMPC

We consider a class of continuous-time systems described by the following nonlinear model

$$\dot{x} = f(x,u), \quad x(0) = x_0 \tag{1}$$

$$y = g(x,u) \tag{2}$$

[§]Bhagyesh V. Patil is with Cambrdige Centre for Advanced Research in Energy Efficiency in Singapore (CARES), 50 Nanyang Ave, Singapore. bhagyesh.patil@gmail.com

[†]Jan Maciejowski is with Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom. jmm@eng.cam.ac.uk

[‡]K. V. Ling is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore. ekvling@ntu.edu.sg

This work was supported by the Singapore National Research Foundation (NRF) under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, and Cambridge Centre for Advanced Research in Energy Efficiency in Singapore (CARES).

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the vectors of states and control inputs, respectively; $y \in \mathbb{R}^p$ is the controlled output. The state of the system and the control input applied at sampling instant k are denoted by $x(k)$ and $u(k)$, respectively. The system is subject to the state and input constraints of the following form:

$$x(k) \in \mathcal{X}, \ \forall k \geq 0 \quad (3)$$

$$u(k) \in \mathcal{U}, \ \forall k \geq 0 \quad (4)$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$. In simplest form, $\mathcal{X}$ and $\mathcal{U}$ are given by bound constraints of the form:

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid x_{\min} \leq x \leq x_{\max}\}.$$

$$\mathcal{U} := \{u \in \mathbb{R}^m \mid u_{\min} \leq u \leq u_{\max}\}.$$

In the present work, we consider the design of an NMPC controller for (1) to track a desired reference x_s, while fulfilling constraints of the form (3)-(4). Further, dropping the index k for simplicity, the general form of NMPC control law can be derived at each sampling instant k by the solution of the following NLP problem.

$$\min_{u_i} \sum_{i=0}^{N-1} \left(\| x_i - x_{i,s} \|_Q^2 + \| \Delta u_i \|_R^2 \right) \quad (5)$$

$$\text{subject to} \quad (1),\ (3),\ \text{and}\ (4)\ \text{for}\ i = 0, 1, \ldots, N-1$$

where $x_{i,s}$ denotes the set-point (reference) at instant i; $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ denote positive definite, symmetric weighting matrices; $\Delta u_i = u_i - u_{i-1}$ denotes the control increment, $N (\geq 1)$ denotes the prediction horizon.

At the outset, the nonlinear model (1) is used for predictions based on the initial state x_0. The predicted control input profile is denoted by $\bar{u}_i$, $i = 0, 2, \ldots, N-1$ Then, assuming that the optimization problem has a feasible solution, an optimizer (in this work, we use Bernstein global optimization algorithm) computes an optimal control sequence based on the NMPC optimization problem formulated in (5), defined as

$$\begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix}. \quad (6)$$

Only the first step of this optimal control sequence, u_0^* is applied to the system (1) to obtain a new updated state. Then the whole process is repeated, with x_0 obtained from the latest measurements, until the state is steered to its desired reference.

Remark 1: The cost function in the optimization problem (5) depends nonlinearly on the state and input variables. Hence, the optimization problem turns out to be a NLP. We note that, in some instances, such as a nonlinear CSTR (see Section IV, Equations (12)-(13)), the nonlinearity appears only in terms of the state variables. Hence, the optimization problem turns out to be a polynomial NLP (that is, (5) comes out to be polynomial in terms of the decision variables u_i). In the present work, we limit ourself to such NMPC problems involving only polynomial NLPs.

III. Bernstein global optimization algorithm

This section briefly presents some notions about the Bernstein form. Due to space limitation, a simple univariate Bernstein form is introduced. A comprehensive background and mathematical treatment for a multivariate case can be found in [12].

We can write a univariate l-degree polynomial p over an interval $\mathbf{x}$ in the form

$$p(x) = \sum_{i=0}^{l} a_i x^i, \qquad a_i \in \mathbb{R}. \quad (7)$$

Now the polynomial p can be expanded into the Bernstein polynomials of the same degree as below [9]

$$p(x) = \sum_{i=0}^{l} b_i(\mathbf{x}) B_i^l(x) \quad (8)$$

where $B_i^l(x)$ are the Bernstein basis polynomials and $b_i(\mathbf{x})$ are the Bernstein coefficients give as below

$$B_i^l(x) = \begin{pmatrix} l \\ i \end{pmatrix} x^i (1-x)^{1-i}. \quad (9)$$

$$b_i(\mathbf{x}) = \sum_{j=0}^{i} \frac{\begin{pmatrix} i \\ j \end{pmatrix}}{\begin{pmatrix} l \\ j \end{pmatrix}} a_j, \ \ i = 0, \ldots, l. \quad (10)$$

Equation (8) is referred as the Bernstein form of a polynomial and obeys the following property:

Theorem 1: (Range enclosure property) Let p be a polynomial of degree l, and let $\overline{p}(\mathbf{x})$ denote the range of p on a given interval $\mathbf{x}$. Then,

$$\overline{p}(\mathbf{x}) \subseteq B(\mathbf{x}) := [\min\ (b_i(\mathbf{x})),\ \max\ (b_i(\mathbf{x}))]. \quad (11)$$

Proof: See [13].

Remark 2: The above theorem says that the minimum and maximum coefficients of $b_i(\mathbf{x})$ provide lower and upper bounds for the range of p. This forms the Bernstein range enclosure, defined by $B(\mathbf{x})$ in equation (11). Figure 1 shows for a univariate polynomial p, its Bernstein coefficients $(b_0, b_1, \ldots, b_5)$. The minimum (b_0, b_4) and maximum (b_1) Bernstein coefficients encloses the range of p. Further, this Bernstein range enclosure can successively be sharpened by the continuous domain subdivision procedure. Figure 2 illustrates this fact.

We now present the global optimization algorithm based on the above Bernstein form. This algorithm uses the Bernstein range enclosing property, followed by a domain subdivision, to correctly locate the global solution (global minimum and global minimizers) for a given NLP problem.

Algorithm Bernstein: $[\widetilde{y}, \widetilde{p}, U]$=BBBC$(N, a_I, \mathbf{x}, \varepsilon_p, \varepsilon_x, \varepsilon_{zero})$

Inputs: Degree N of the variables occurring in the objective and constraint polynomials, the coefficients a_I of the objective and constraint polynomials in the power form, the initial search domain $\mathbf{x}$, the tolerance parameters ε_p and

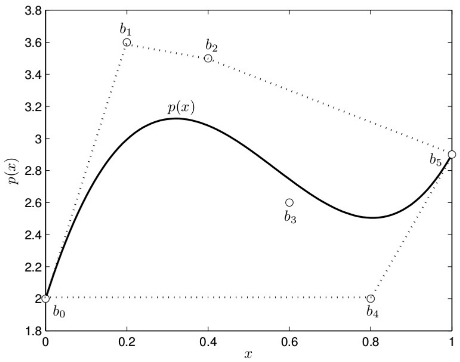

Fig. 1. The polynomial p over $\mathbf{x} = [0,1]$ and its Bernstein coefficients $(b_0, b_1, \ldots, b_5)$ illustrating the range enclosing property for p.

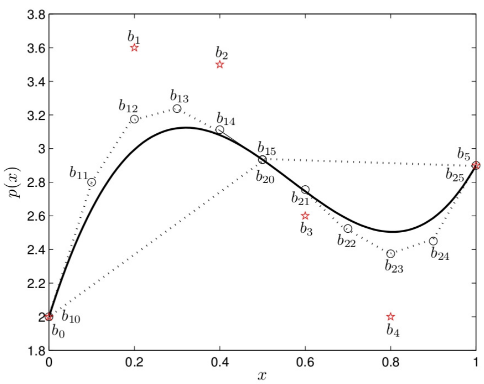

Fig. 2. Improvement in the range enclosure of p with a subdivision of an original interval $\mathbf{x}$. $(b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15})$, and $(b_{20}, b_{21}, b_{22}, b_{23}, b_{24}, b_{25})$ are the Bernstein coefficients over $\mathbf{x}_1 = [0, 0.5]$ and $\mathbf{x}_2 = [0.5, 1]$, respectively.

ε_x on the global minimum and global minimizer(s), and the tolerance parameter ε_{zero} to which the equality constraints are to be satisfied.

Outputs: A lower bound $\widetilde{y}$ and an upper bound $\widetilde{p}$ on the global minimum f^*, along with a set U containing all the global minimizer(s) $\mathbf{x}^{(i)}$.

BEGIN Algorithm

1) Set $\mathbf{y} := \mathbf{x}$.
2) From a_I, compute the Bernstein coefficient arrays of the objective and constraint polynomials on the box $\mathbf{y}$ respectively as $(b_o(\mathbf{y})), (b_{gi}(\mathbf{y})), (b_{hj}(\mathbf{y}))$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$.
3) Set $\widetilde{p} := \infty$ and $y := \min(b_o(\mathbf{y}))$.
4) Initialize list $\mathcal{L} := \{(\mathbf{y}, y)\}$, $\mathcal{L}^{sol} := \{\}$.
5) If $\mathcal{L}$ is empty then go to step 13. Otherwise, pick the first item $(\mathbf{y}, y)$ from $\mathcal{L}$, and delete its entry from $\mathcal{L}$.
6) Choose a coordinate direction λ parallel to which $\mathbf{y}_1 \times \cdots \times \mathbf{y}_l$ has an edge of maximum length, that is $\lambda \in \{i : w(\mathbf{y}) := w(\mathbf{y}_i), i = 1, 2, \ldots, l\}$.
7) Bisect $\mathbf{y}$ normal to direction λ, getting boxes $\mathbf{v}_1$, $\mathbf{v}_2$ such that $\mathbf{y} = \mathbf{v}_1 \cup \mathbf{v}_2$.
8) for $k = 1, 2$
   - (a) Find the Bernstein coefficient array and the corresponding Bernstein range enclosure of the objective function (f) over $\mathbf{v}_k$ as $(b_0(\mathbf{v}_k))$ and $B_0(\mathbf{v}_k)$, respectively.
   - (b) Set $d_k := \min B_o(\mathbf{v}_k)$.
   - (c) If $\widetilde{p} < d_k$ then go to substep (h).
   - (d) for $i = 1, 2, \ldots, m$
     - (i) Find the Bernstein coefficient array and the corresponding Bernstein range enclosure of the inequality constraint polynomial (g_i) over $\mathbf{v}_k$ as $(b_{gi}(\mathbf{v}_k))$ and $B_{gi}(\mathbf{v}_k)$, respectively.
     - (ii) If $B_{gi}(\mathbf{v}_k) > 0$ then go to substep (h).
     - (iii) If $B_{gi}(\mathbf{v}_k) \leq 0$ then go to substep (e)
   - (e) for $j = 1, 2, \ldots, n$
     - (i) Find the Bernstein coefficient array and the corresponding Bernstein range enclosure of the equality constraint polynomial (h_j) over $\mathbf{v}_k$ as $(b_{hj}(\mathbf{v}_k))$ and $B_{hj}(\mathbf{v}_k)$, respectively.
     - (ii) If $0 \notin B_{hj}(\mathbf{v}_k)$ then go to substep (h).
     - (iii) If $B_{hj}(\mathbf{v}_k) \subseteq [-\varepsilon_{zero}, \varepsilon_{zero}]$ then go to substep (f)
   - (f) Set $\widetilde{p} := \min(\widetilde{p}, \max B_o(\mathbf{v}_k))$.
   - (g) Enter $(\mathbf{v}_k, d_k)$ into the list $\mathcal{L}$ such that the second members of all items of the list do not decrease.
   - (h) end (of $k-$loop).
9) {Cut-off test} Discard all items $(\mathbf{z}, z)$ in the list $\mathcal{L}$ that satisfy $\widetilde{p} < z$.
10) Denote the first item of the list $\mathcal{L}$ by $(\mathbf{y}, y)$.
11) If $(w(\mathbf{y}) < \varepsilon_x)$ & $(\max B_o(\mathbf{y}) - \min B_o(\mathbf{y})) < \varepsilon_p$ then remove the item from the list $\mathcal{L}$ and enter it into the solution list $\mathcal{L}^{sol}$.
12) Go to step 6.
13) {Compute the global minimum} Set the global minimum $\widetilde{y}$ to the minimum of the second entries over all the items in $\mathcal{L}^{sol}$.
14) {Compute the global minimizers} Find all those items in $\mathcal{L}^{sol}$ for which the second entries are equal to $\widetilde{y}$. The first entries of these items contain the global minimizer(s) $\mathbf{x}^{(i)}$.
15) Return the lower bound $\widetilde{y}$ and upper bound $\widetilde{p}$ on the global minimum f^*, along with the set U containing all the global minimizer(s) $\mathbf{x}^{(i)}$.

END Algorithm

IV. Simulation Study

In this section, we study the application of a global optimization procedure based on the Bernstein algorithm BBBC, a sub-optimal procedure based on the MATLAB fmincon function, and two well established global optimization solvers, BARON [14] and YALMIP based branch-and-bound BMIBNB [15] procedure to the highly nonlinear model of a CSTR system adapted from [16]. Following assumptions in [16] of constant liquid volume for an exothermic irreversible reaction, $A \rightarrow B$, the CSTR system is described by the following model.

$$\dot{C}_A = \frac{q}{V}\left(C_{Af} - C_A\right) - k_o e^{\left(\frac{-E}{RT}\right)} C_A \tag{12}$$

$$\dot{T} = \frac{q}{V}\left(T_f - T\right) - \frac{\Delta H}{\rho C_p} k_o e^{\left(\frac{-E}{RT}\right)} C_A + \frac{UA}{V\rho C_p}(T_c - T) \tag{13}$$

where C_A is the concentration of A in the reactor, T is the reactor temperature and T_c is the temperature of the coolant stream. The other parameters corresponding to the nominal operating point conditions are reported in Table I.

The objective is to regulate the states $x_1 = C_A$ and $x_2 = T$ by manipulating $u = T_c$ with the state and input constraints of the following form:

$$\mathbf{x} = \left\{ \begin{bmatrix} C_A \\ T \end{bmatrix} \in \mathbb{R}^2 \mid 0 \le C_A \le 1, 280 \le T \le 370 \right\}.$$

$$u = \{T_c \in \mathbb{R} \mid 280 \le T_c \le 370\}.$$

We consider an NMPC scheme for set-point tracking control problem, which involves multiple set-point changes for $y = x_2 = T$. The nonlinear model in (12)-(13) is used as a system model for the simulation, and the NMPC control law is derived by solving an NLP of the form (5). The solution for the updated states is computed based on the set of given initial conditions and first optimal control move derived by a NMPC control law. We adopted the following parameters values for the simulation:

- sampling time of 0.5 seconds
- prediction horizon, N = 3
- $Q = \text{diag}(1 \ \ 0.01)^T$ and $R = 0.01$ as weighting matrices
- initial conditions, $x_0 = [0.2 \ \ 370]^T$ and $u_0 = 300$
- tolerances, $\varepsilon_p = \varepsilon_x = 0.001$ in the algorithm BBBC on the global minimum and minimizers

Similarly, for the sub-optimal procedure using *fmincon*, we choose sequential quadratic programming (SQP) algorithm for the simulation studies.

Figure 3 shows the evolution of the states. Specifically, we compare the results for NMPC of a CSTR system based on the Bernstein algorithm BBBC and on the sub-optimal procedure using *fmincon*. We observed a smooth transition of the both states under multiple set-point changes based on the algorithm BBBC. On the other hand, with the sub-optimal fmincon procedure, significant undershoot and overshoot are observed, although the settling time remains almost the same in both cases. Figure 4 shows the control performance of the two approaches. It is apparent that except at the first few samples ($\approx 0 - 20$), algorithm BBBC computes smooth control moves compared to the sub-optimal fmincon based procedure. In Figure 5, we show the computation time taken to compute the control move at each sampling instant by algorithm BBBC and sub-optimal procedure *fmincon*. In this case too, algorithm BBBC proved superior with an average reduction of 30 % in the computation time. It may be noted that in practice Bernstein algorithm BBBC performs rigorous global search to determine all global solutions. However, to keep the simulation feasible from a practical point of view, we only located one global solution.

Figure 6 plots the cost function values for the NLP problem in (5) obtained with both optimization procedures at each sampling instant. We observed nearly identical cost function values at each sampling instant using both optimization procedures. However, we note that at each set-point change (introduced at samples 0, 50, 100, 150, and 200), algorithm BBBC returned lower cost function values compared to *fmincon*. This is probably associated with the smoother transitions that were obtained using BBBC, and which are visible in Figure 3.

To assess the consistency of the benefits obtained from using algorithm BBBC instead of *fmincon*, we studied one more realistic scenario; set-point tracking problem under a constant input disturbance. Figure 8 shows the result for this constant disturbance handling case. Typically, we consider the CSTR to be operating at steady-state value of $T = 310$ K. Then at 50$^{\text{th}}$ sample a step change is applied requiring the CSTR to operate at 330 K. Further, we assume that a constant input disturbance of 5 K is acting on the system. In this case too, we observed the algorithm BBBC tracks smoothly the given set-point. On the other hand, *fmincon* resulted in a small constant off-set from the desired set-point value.

Further, we choose to compare the Bernstein algorithm BBBC with the two other well-established global optimization solvers, namely, BARON and BMIBNB. The time required to compute the optimal solution (global minimum for NLPs in the NMPC scheme) is considered as a performance metric and is compared in the Figure 7. We observed the average computation time as 0.12, 0.13, and 0.05 for the algorithm BBBC, BMIBNB and BARON, respectively. Algorithm BBBC was found to be faster compared to BMIBNB (which on an average took 8 % more computational time). Similarly, we noted algorithm BBBC to be much slower than BARON (which on an average took 55 % less computational time).

V. Concluding remarks

In this paper, a global optimization procedure based on the Bernstein form was presented to solve the nonlinear optimization problems encountered in model predictive control of nonlinear systems. The main aim of the work was to demonstrate the benefits achieved with the Bernstein global optimization approach compared to the sub-optimal procedures, such as MATLAB's *fmincon* function and global optimization procedures, such as BARON and BMIBNB in the context of NMPC. The presented Bernstein algorithm showed good promise in simulations of a CSTR system for the multiple set-point tracking control problem as well as for the disturbance handling problem. Similarly, the Bernstein algorithm was also found to be well competent with global optimization procedures such as BARON and BMIBNB. Questions such as the sustainability and scalability of Bernstein global optimization algorithms for large-scale nonlinear systems remain to be explored, but we believe that they are very promising contenders for use in NMPC problems.

REFERENCES

[1] J. M. Maciejowski, *Predicitve control with constraints*. UK, Harlow: Prentice Hall, 2000.

[2] E. F. Camacho and C. Bordons, *Model predictive control, 2^{nd} ed.* London: Springer-Verlag, 2004.

[3] M. L. Darby and M. Nikolaou, "MPC: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.

[4] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[5] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control System Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[6] F. Martinsen, L. T. Biegler, and B. A. Fossa, "A new optimization algorithm with application to nonlinear MPC," *Journal of Process Control*, vol. 14, no. 8, pp. 853–865, 2004.

[7] R. Findeisen, F. Allgöwer, and L. Biegler. Assessment and future directions of nonlinear model predictive control, *Lecture Notes in Control and Information Sciences*: Springer-Verlag, 2007.

[8] J. D. Hedengren, R. A. Shishavana, K. M. Powell, and T. F. Edgar, "Nonlinear modeling, estimation and predictive control in APMonitor," *Computers and Chemical Engineering*, vol. 70, no. 5, pp. 133–148, 2014.

[9] H. Ratschek and J. Rokne, *New computer methods for global optimization*. Chichester, England: Ellis Horwood Publishers, 1988.

[10] S. Ray and P. S. V. Nataraj, "An efficient algorithm for range computation of polynomials using the Bernstein form," *Journal of Global Optimization*, vol. 45, no. 3, pp. 403–426, 2009.

[11] P. S. V. Nataraj and M. Arounassalame, "Constrained global optimization of multivariate polynomials using Bernstein branch and prune algorithm," *Journal of Global Optimization*, vol. 49, no. 2, pp. 185–212, 2011.

[12] B. V. Patil, P. S. V. Nataraj, and S. Bhartiya, "Global optimization of mixed-integer nonlinear (polynomial) programming problems: the Bernstein polynomial approach," *Computing*, vol. 94, no. 2-4, pp. 325–343, 2012.

[13] J. Garloff, "The Bernstein algorithm," *Interval Computations*, vol. 2, pp. 154–168, 1993.

[14] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, 2005.

[15] J. Lofberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," *IEEE International Symposium on computer Aided Control Systems Design*, pp. 282–289, 2004.

[16] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini, "A stabilizing model-based predictive control algorithm for nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

TABLE I

LIST OF MODEL PARAMETERS [16].

Parameters	Meaning	Value	Unit
q	Inlet flow	100	l/min
V	Reactor liquid volume	100	l
C_{Af}	Concentration of inlet flow	1	mol/l
k_o	Reaction frequency factor	7.2×10^{10}	l/min
E/R		8750	K
E	Activation energy		
R	Gas constant	8.3196×10^3	J/(mol K)
T_f	Temperature inlet flow	350	K
ΔH	Heat of reaction	-5×10^4	J/mol
ρ	Density	1000	g/l
C_p	Specific heat capacity of the fluid	0.239	J/(g K)
UA		5×10^4	J/(min K)
U	Overall heat transfer coefficient		
A	Heat transfer area		

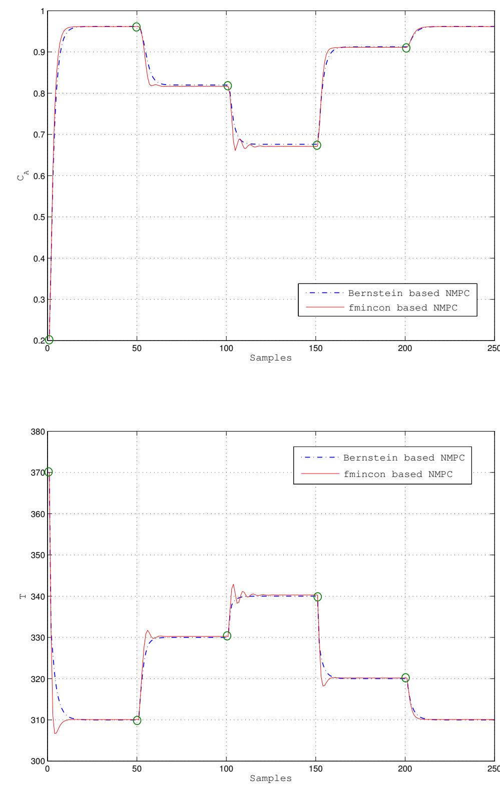

Fig. 3. Evolutions of the state C_A and T with NMPC based on Bernstein algorithm BBBC and MATLAB based *fmincon* function. Green circles at 0, 50, 100, 150, and 200 indicates the samples at which the set-point changes are implemented for T. The set-points are as follows: at 0 sec 310 K, at 50 sec 330 K, at 100 sec 340 K, at 150 s 320 K, and at 200 sec 310 K.

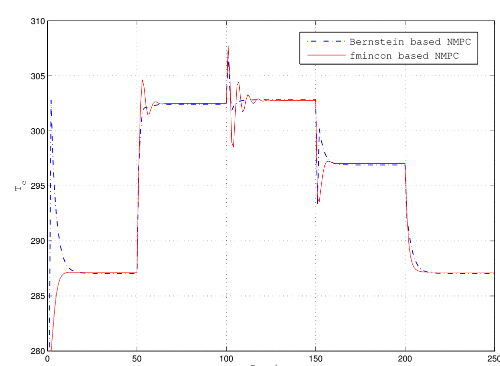

Fig. 4. Control input T_c for the CSTR system with the Bernstein algorithm BBBC and MATLAB based *fmincon* function.

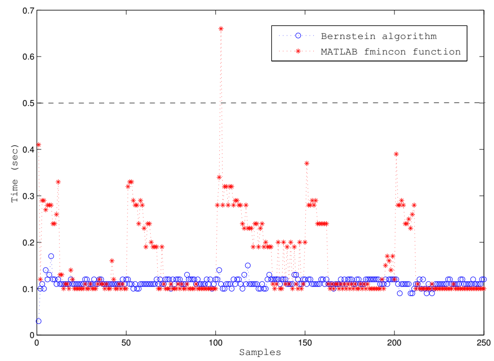

Fig. 5. Comparison of the computation time needed for a solution of an NLP at a each sampling instant with the Bernstein algorithm BBBC and MATLAB based *fmincon* function. Dotted line at 0.5 shows the sampling time.

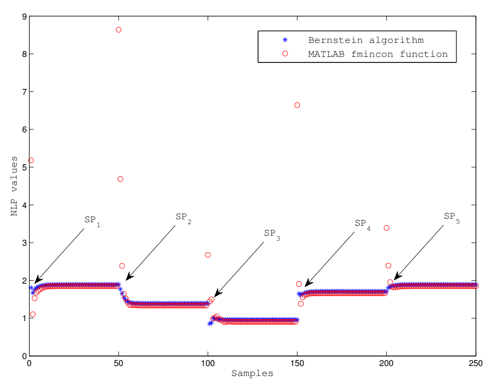

Fig. 6. Cost function values for an NLP problem (of the form (5)) values for a CSTR system based on the Bernstein algorithm BBBC and MATLAB based *fmincon* function. SP_1, SP_2, SP_3, and SP_4 show the time instants at which the set-point changes are implemented.

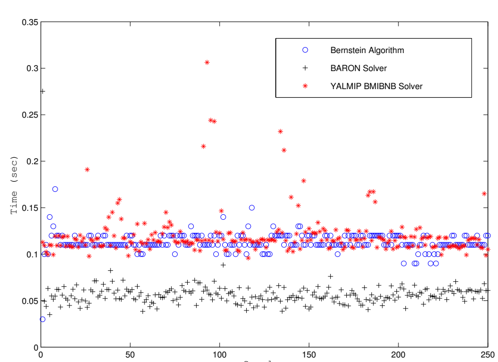

Fig. 7. Comparison of the computation time needed for a solution of an NLP at a each sampling instant with the Bernstein algorithm BBBC, and global optimization solvers, BARON [14] and BMIBNB [15].

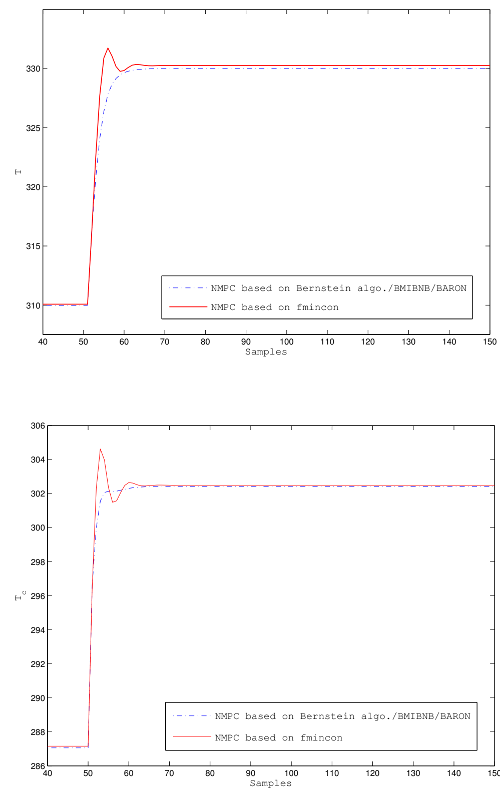

Fig. 8. Comparison of the disturbance handling capability of a NMPC for a CSTR system based on the Bernstein algorithm BBBC, MATLAB based *fmincon* function, global optimization solvers, BARON [14] and BMIBNB [15].