

Optimal nonlinear model predictive control based on Bernstein polynomial approach

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

Abstract—In this paper, we compare the performance of Bernstein global optimization algorithm based nonlinear model predictive control (NMPC) with a power system stabilizer and linear model predictive control (MPC) for the excitation control of a single machine infinite bus power system. The control simulation studies with Bernstein algorithm based NMPC show improvement in the system damping and settling time when compared with respect to a power system stabilizer and linear MPC scheme. Further, the efficacy of the Bernstein algorithm is also compared with global optimization solver BMIBNB from YALMIP toolbox in terms of NMPC scheme and results are found to be satisfactory.

I. INTRODUCTION

Over the past decades, model predictive control (MPC) has emerged as one of the prominent advanced control methodology for multivariable control. At the heart of MPC lies a system model, which predicts the future evolution of the system states. It generates control actions by iteratively optimizing a performance criterion over a finite-time moving window with reference to system constraints, and based on predictions of the system model [1], [2]. In practice, MPC implementations utilizing linear models (a.k.a ‘linear MPC’) are preferred. This facilitates use of linear/convex programming techniques to exactly solve optimization problems at each sampling instant. However, some applications (like power systems) have significant nonlinear behaviour, and for such applications, a linear MPC scheme may not yield desirable closed-loop performance [3], [4]. Hence, to mitigate problems arising from the system nonlinearities, many researchers have pursued MPC approach based on nonlinear system models. This approach under large variation of dynamic behaviour of the system, may provide more satisfactory control than linear MPC. Model predictive control using nonlinear system models, usually called ‘nonlinear MPC’ (or NMPC), hence has attracted many researchers over the past decade [5], [6], [7], [8], [9], [10].

We note that an NMPC formulation usually requires the solution of a nonlinear, usually *nonconvex*, optimization problem at each sampling instant. Therefore, NMPC requires

[§]Bhagyesh V. Patil is with Cambridge Centre for Advanced Research and Education in Singapore (CARES), 50 Nanyang Ave, Singapore. bvpatil@ntu.edu.sg

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

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

This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

global optimization procedures (cf. [11], [12]). The challenge involved in NMPC is two fold:

(i) Can we complete the nonlinear iteration procedure until a pre-specified convergence criterion is met so as to guarantee the optimal solution of the optimization problem?

(ii) can we achieve (i) in a pre-specified sampling time limit? Concerning these facts, our previous works have introduced Bernstein global optimization procedures for NMPC applications (cf. [13], [14]). Optimization procedures based on this Bernstein form, also called *Bernstein global optimization algorithms*, have shown good promise to solve hard nonconvex optimization problems. This procedure is based on the Bernstein form of polynomials [15], and uses several nice ‘geometrical’ properties associated with this Bernstein form.

The current scope of the work is based on the sequential improvement of our previous works in [13], [14]. Specifically, in this work we implement a hull pruning feature for the Bernstein global optimization algorithm to solve a nonconvex optimization problem at each NMPC iteration. The hull pruning aids in discarding regions from solution search spaces that surely do not contain the global solution. As this feature provides the means to narrow the search region for the optimization problem, we call it a narrowing (‘hull pruning’) operator. The applicability of the Bernstein algorithm with this hull pruning feature is demonstrated by simulating a nonlinear model predictive control scheme for a classical single machine infinite bus (SMIB) power system [16]. SMIB has a strong nonlinear characteristics and exhibits accurate representation of the synchronous generator behaviour. The findings of our nonlinear MPC scheme based on the Bernstein algorithm with hull pruning operator are compared with respect to well-established power system stabilizer (PSS) [16] and linear MPC. We also investigate our findings with nonlinear MPC based on the YALMIP global optimization solver BMIBNB [17].

In the rest of the paper, we first introduce a nonlinear MPC formulation (Section 2). Next, we briefly describe the Bernstein form and the hull pruning operator, followed by the presentation of the Bernstein global optimization algorithm (Section 3). We then report the simulation studies on a nonlinear SMIB power system (Section 4). Finally, some concluding remarks are given in Section 5.

II. NMPC CONTROLLER FORMULATION

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

$$\dot{x} = f(x, u), \quad x(t_0) = x_0 \quad (1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the vectors of states and control inputs, respectively. In practice, the continuous-time model (1) used for predictions is discretized with a sampling time Δt , such as Euler's method

$$x_{k+1} = x_k + \Delta t \cdot f(x_k, u_k) \quad (2)$$

where k denotes sampling instant.

The nonlinear optimization problem in the NMPC formulation at each sampling instant k is stated by (3)-(6). The control objective is to maintain equilibrium point (origin), by minimizing the cost (3) subjected to discretized nonlinear predictive model (2), and fulfilling constraints of the form (5).

$$\min_{u_k} \sum_{k=0}^{N-1} \mathcal{L}(x_k, u_k) \quad (3)$$

$$\text{subject to } x_{k+1} = x_k + \Delta t \cdot f(x_k, u_k) \quad (4)$$

$$c(x_k, u_k) \leq 0 \quad (5)$$

$$\text{for } k = 0, 1, \dots, N-1 \quad (6)$$

where $N(> 1)$ denotes the prediction horizon and $c(x_k, u_k)$ are the nonlinear constraints arising due to safety and operational requirements of the control system. It may be noted that, $c(x_k, u_k)$ subsume the constraints on the state and input of the following form:

$$x_k^{\min} \leq x_k \leq x_k^{\max}. \quad (7)$$

$$u_k^{\min} \leq u_k \leq u_k^{\max}. \quad (8)$$

We assume at the equilibrium point (x_k^*, u_k^*) , the cost (3) should be zero, *i.e.* $\mathcal{L}(x_k^*, u_k^*) = 0$.

The NMPC algorithm at each sampling instant k involves the following steps:

- Measure the state x_k of the system (we assume in the above NMPC formulation all states are available for measurement).
- Solve the nonlinear optimization problem (3)-(6) with initial state x_k . Denote the obtained optimal control sequence as $u_0^*, u_1^*, \dots, u_{N-1}^*$.
- Implement the first step of the optimal control sequence, u_0^* to the system (1) to obtain a new updated state until the next sampling instant.
- Repeat from (a).

The overall scheme of the Bernstein algorithm based NMPC is depicted in Fig. 1. The Bernstein algorithm can be applied when the system model, cost function and the constraints are polynomials. In many other applications the problem can be approximated (as closely as desired) by polynomials (such as our example in Section IV).

III. BERNSTEIN GLOBAL OPTIMIZATION APPROACH

In the NMPC formulation (Section II), we solve a nonlinear optimization problem at each sampling instant to derive a control law for the nonlinear system (1). [5] and [7] report some optimization approaches to achieve this goal. This section briefly introduces one such (global) optimization approach based on the Bernstein form of polynomials. For

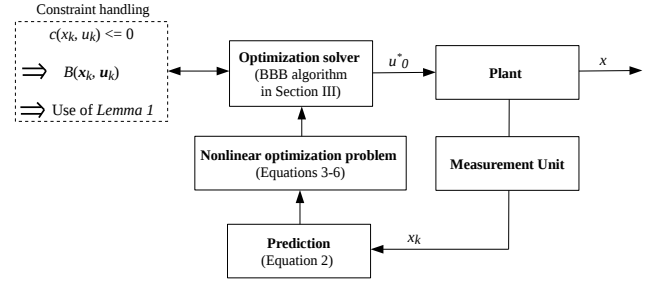


Fig. 1. Schematic of Bernstein algorithm based NMPC.

brevity, we only present notions about the univariate Bernstein form (see [18] for the multivariate case).

We can write a generic polynomial of degree l as

$$p(x) = \sum_{i=0}^l a_i x^i, \quad a_i \in \mathbb{R} \quad (9)$$

where x is the variable, and $\{a_0, a_1, \dots, a_l\}$ are the coefficients of the power basis \mathcal{B}_p given by the following set of monomials

$$\mathcal{B}_p = \{1, x, x^2, \dots, x^l\}. \quad (10)$$

We assume that $p(x)$ is defined over a real bounded and closed interval $\mathbf{x} = [0, 1]$. The unit interval is not a restriction, since any nonempty compact interval can be mapped affinely onto it.

Now the polynomial p can be expressed into the Bernstein polynomial form of the same degree [15]:

$$p(x) = \sum_{i=0}^l b_i B_i^l(x) \quad (11)$$

where $B_i^l(x)$ are the Bernstein basis polynomials and b_i are the Bernstein coefficients:

$$B_i^l(x) = \binom{l}{i} x^i (1-x)^{l-i}. \quad (12)$$

$$b_i = \sum_{j=0}^i \frac{\binom{i}{j}}{\binom{l}{j}} a_j, \quad i = 0, \dots, l. \quad (13)$$

Equation (11) is referred as the Bernstein form of (9) and satisfies the following range enclosure property [15]:

$$\bar{p}(\mathbf{x}) \subseteq B(\mathbf{x}) := [\min b_i, \max b_i]. \quad (14)$$

where $\bar{p}(\mathbf{x})$ denote the range of p on a given interval \mathbf{x} .

Remark 1: Equation (14) says that the minimum and maximum coefficients of b_i provide lower and upper bounds for the range of p . This forms the Bernstein range enclosure, defined by $B(\mathbf{x})$. Further, this Bernstein range enclosure

can successively be sharpened by the continuous domain subdivision procedure (see, for instance [19]).

The following properties follow immediately from the Bernstein range enclosure (14).

Lemma 1: Let $B(\mathbf{x})$ be the Bernstein range enclosure for a polynomial $p(x)$ on a given box \mathbf{x} . Then, the following identities hold

- 1) $B(\mathbf{x}) \leq 0 \Rightarrow p(x) \leq 0$ for all $x \in \mathbf{x}$.
- 2) $B(\mathbf{x}) > 0 \Rightarrow p(x) > 0$ for all $x \in \mathbf{x}$.
- 3) $0 \notin B(\mathbf{x}) \Rightarrow p(x) \neq 0$ for all $x \in \mathbf{x}$.
- 4) $B(\mathbf{x}) \subseteq [-\varepsilon_{zero}, \varepsilon_{zero}] \Rightarrow p(x) \in [-\varepsilon_{zero}, \varepsilon_{zero}]$ for all $x \in \mathbf{x}$, where $\varepsilon_{zero} > 0$.

Vertex condition [15] Consider the Bernstein form in equation (14) for a polynomial p of degree l , and let the range $\bar{p}(\mathbf{x}) = [a, b]$. Then

$$a = \min_{0 \leq i \leq l} (b_i) \text{ if and only if } \min_{0 \leq i \leq l} (b_i) = \min_{i \in \{0, l\}} (b_i)$$

$$b = \max_{0 \leq i \leq l} (b_i) \text{ if and only if } \max_{0 \leq i \leq l} (b_i) = \max_{i \in \{0, l\}} (b_i)$$

Remark 2: The above vertex condition says that the lower bound (respectively, upper bound) is sharp if and only if $\min b_i$ (respectively, $\max b_i$) is attained at a vertex Bernstein coefficient (that is, b_0, b_l). Further, the vertex condition is said to be met within a given tolerance ε_f , if

$$b_k - \min_{0 \leq i \leq l} b_i \leq \varepsilon_f, \quad k = 0, l. \quad (15)$$

A. Hull pruning operator

As mentioned earlier, NMPC solves a nonlinear programming (NLP) problem of the form (3)-(6). This NLP has a nonlinear set of constraints and a search space (box) representing the variable domains. The hull pruning operation removes inconsistent values from a box that surely do not contribute in locating global solution. It achieves this goal using the constraint inversion procedure and interval arithmetic operations[†].

Consider a multivariate equality constraint $h(x) = 0$. To apply hull pruning, we keep one term on the left hand side (say variable x_r from which has to be pruned) and remaining all other terms are taken on the right hand side, that is, we write the constraint in the form $a_l x^l = h_1(x)$ where, $x = (x_1, \dots, x_r, \dots, x_l)$ and $l = (i_1, i_2, \dots, i_r, \dots, i_l)$. Now, replacing all x^l by their respective intervals and $h_1(x)$ by its interval evaluation, we can write interval \mathbf{x}'_r (for the variable x_r , which has to be pruned) as

$$\mathbf{x}'_r = \left(\frac{\mathbf{h}'_1}{a_l \prod \mathbf{x}_k^{i_k}} \right)^{1/i_r}, \quad k = 1, 2, \dots, l. \quad (16)$$

Then the new pruned interval for the variable x_r can be obtained as below

$$\mathbf{x}_{r, \text{consistent}} = \mathbf{x}'_r \cap \mathbf{x}_r \quad (17)$$

[†]Interval arithmetic operations are set theoretic extensions of the corresponding real (floating-point) operations [20].

Example: Consider a constraint $x_2 = x_1^2$, $x_1 \in [1, 2]$ and $x_2 \in [0, 2.5]$. First perform constraint inversion and interval arithmetic operations for x_1

$$\begin{aligned} \mathbf{x}'_1 &= (\mathbf{x}_2)^{\frac{1}{2}} \\ &= [0, 2.5]^{\frac{1}{2}} \\ &= [0, 1.5812] \\ \mathbf{x}_{1, \text{consistent}} &= \mathbf{x}'_1 \cap x_1 \\ &= [0, 1.5812] \cap [1, 2] \\ &= [1, 1.5812] \end{aligned}$$

Similarly, we can achieve pruning for x_2 as $\mathbf{x}_{2, \text{consistent}} = [1, 2.5]$. Fig. 2 illustrates the complete hull pruning for constraint $x_2 = x_1^2$ for $x_1 \in [1, 2]$ and $x_2 \in [0, 2.5]$.

In our hull pruning operator, we perform constraint inversion and interval arithmetic operations using the Bernstein range enclosure $B(\mathbf{x})$ in (14).

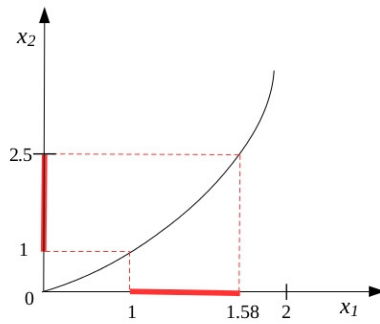


Fig. 2. Computation of consistent set of values (pruning) from $\mathbf{x}_1 \in [1, 2]$ and $\mathbf{x}_2 \in [0, 2.5]$ with respect to the constraint $x_2 = x_1^2$.

B. Main Bernstein branch-and-prune algorithm

$$[f^*, \mathbf{x}^*] = \text{BBB}(f, g_i, h_j, \mathbf{x}, \varepsilon_f, \varepsilon_{zero})$$

Inputs: The cost function (3) as f , equality constraints (4) as h_j , and inequality constraints (5) as g_i , the initial search box for u_k as \mathbf{x} , the tolerance parameter ε_f on the global minimum, and the tolerance parameter ε_{zero} to which the equality constraints are to be satisfied.

Outputs: The global minimum f^* and global minimizer \mathbf{x}^* .

BEGIN Algorithm

Initialization step (R)

- Compute the range enclosures for f , g_i , and h_j , respectively as b_f, b_{g_i} , and b_{h_j} over \mathbf{x} . Set \hat{f} to the minimum Bernstein coefficient of b_f and $\hat{f} = \hat{f}$.
- Construct $\mathcal{L} \leftarrow \{(\hat{f}, b_f, b_{g_i}, b_{h_j}, \mathbf{x})\}$, $\mathcal{L}^{\text{sol}} \leftarrow \{\}$.

Sorting and Pruning step (SP)

- If \mathcal{L} is empty, go to step T. Else sort item(s)[†] in \mathcal{L} in ascending order of \hat{f} .

[†]Each item in the list \mathcal{L} is of the form: $(\hat{f}, b_f, b_{g_i}, b_{h_j}, \mathbf{x})$.

- Pick the first item from \mathcal{L} removing its entry.
- Apply hull pruning operator based on Bernstein range enclosure property to the set of constraints (4)-(5) and obtain the contracted box as \mathbf{x}' . Compute the range enclosures for f , g_i , and h_j , respectively as b_f, b_{g_i} , and b_{h_j} over \mathbf{x}' .

Feasibility and Bounding step (FB)

- Check the constraint feasibility (see **Lemma 1**). If the constraint is not strictly feasible, then go to the branching step.
- Check the vertex condition for item (see **Remark 2**). If 'true', then update $\tilde{f} = b_k$ and add that item to \mathcal{L}^{sol} . Go to step SP.

Branching step (B)

- Partition the (feasible) search space \mathbf{x} into two subregions (such that, $\mathbf{x} = \mathbf{x}_1 \cup \mathbf{x}_2$).
- Compute the Bernstein range enclosures for f , g_i , and h_j over \mathbf{x}_1 and \mathbf{x}_2 .
- Discard \mathbf{x}_k , $k = 1, 2$ for which $\min(b_{f,k}) > \tilde{f}$. Enter $(\hat{f}_k, b_{f,k}, b_{g_i,k}, b_{h_j,k}, \mathbf{x})$ into the list \mathcal{L} ($\hat{f}_k := \min(b_{f,k})$).

Termination step (T)

- If maximum number of the subdivisions are not reached or \mathcal{L} is not empty, then repeat SP-FB-B. Else go to the next step.
- Find that item in \mathcal{L}^{sol} for which the first entry is equal to \tilde{f} . Denote that item by \mathcal{J}_f .
- \mathcal{J}_f : the first entry is a global minimum f^* , last entry is the global minimizer \mathbf{x}^* .
- Return the global solution (f^*, \mathbf{x}^*) .

END Algorithm

IV. NMPC APPLICATION: POWER SYSTEM

In this section, we present our findings with Bernstein algorithm (BBB) based NMPC (henceforth, referred as BNMPC) for nonlinear excitation control of a single machine infinite bus (SMIB) power system (shown in Fig. 3). We use the following SMIB system dynamical model in a $d-q$ reference frame [16].

$$\dot{\delta} = \Omega_B(\omega - \omega_r) \quad (18)$$

$$\dot{\omega} = \frac{1}{2H} (P_m - P_e - K_d(\omega - \omega_r)) \quad (19)$$

$$\dot{e}'_q = \frac{1}{T'_{d0}} \left(-\frac{X_{dr}e'_q}{X'_{dr}} + \frac{(X_{dr} - X'_{dr})}{X'_{dr}} v_r \cos(\delta) + e_{fd} \right) \quad (20)$$

$$\dot{e}_{fd} = -\frac{1}{T_A} e_{fd} + \frac{K_A}{T_A} (1 - V_t) + \frac{K_A}{T_A} u \quad (21)$$

where

$$P_e = \frac{e'_q v_r \sin(\delta)}{X'_{dr}} + \frac{(X_{dr} - X'_{dr})}{X_{qr} X'_{dr}} v_r^2 \cos(\delta) \sin(\delta).$$

$$V_t = \sqrt{(e'_q - X'_d I_d)^2 + (X'_q I_q)^2}.$$

$$X_{dr} = X_d + X_r, X_{qr} = X_q + X_r, X'_{dr} = X'_d + X_r.$$

where δ is the rotor angle of the generator, ω is the rotor speed, H is the inertia constant of the generator, P_m is the

mechanical input power to the generator which is assumed to be constant, K_d is the damping constant of the generator, P_e is the electrical power delivered by the generator, e_{fd} is the field voltage of the generator, V_t is the terminal voltage of the generator, and u is the control input from the controller which modulates e_{fd} . All parameters are expressed in per unit (pu) and their equilibrium points values used in the simulation are listed in Table I. For all simulations, we have shifted the actual equilibrium point to the origin using the appropriate transformation.

TABLE I
LIST OF SMIB PARAMETERS AND DATA [16].

Parameter	Parameter	Value
Base angular frequency	Ω_B	$2\pi \times 50$ rad/sec
Network bus voltage	v_r	0.90081 pu
d-axis transient time constant	T'_{d0}	8 sec
d-axis reactance	X_d	1.81 pu
d-axis transient reactance	X'_d	0.3 pu
q-axis reactance	X_q	1.76 pu
Transmission line reactance	X_r	0.475 pu
Mechanical power	P_m	0.9 pu
Generator exciter gain	K_A	200
Generator exciter time constant	T_A	0.001 sec
Generator inertia constant	H	3.5 sec
Stable equilibrium point	$[\delta \ \omega \ e'_q \ e_{fd}]$	[1.225 1 1.023 2.42]

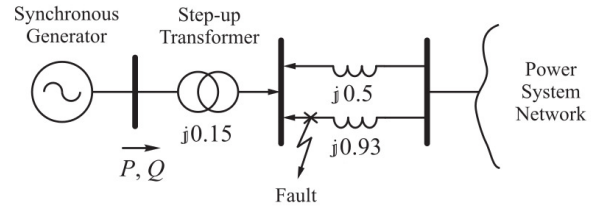


Fig. 3. Classical single machine infinite bus power system network [16].

For the nonlinear control simulation studies we consider the following two scenarios:

Scenario I: A short-circuit fault occurring at one of the two parallel transmission lines shown in Fig. 3 (mechanical power P_m is assumed to be constant at 0.9 pu).

Scenario II: 5 % and 10 % step changes of the input mechanical power to the generator.

For both scenarios, we simulate an BNMPC scheme which regulates the SMIB system about its equilibrium point. We consider the nonlinear model in (18)-(21) as the system, and the NMPC control law is derived by solving an NLP of the form (3)-(6) using the Bernstein algorithm BBB. The solution for the updated states is computed based on the set of given initial conditions and first optimal control move derived by a BNMPC control law. We adopted the cost \mathcal{L} in (3) as $\sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k$ with the following parameter values for the simulation:

- sampling time of 0.03 seconds
- prediction horizon, $N = 3$
- $Q = \text{diag}(1 \ 1 \ 1 \ 1)^T$ and $R = 1$ as weighting matrices

- initial condition, $x_0 = [0 \ 0 \ 0 \ 0]^T$ and $u_0 = 0$
- constraints on the control input, $-0.1 \leq u \leq 0.1$
- tolerances, $\varepsilon_f = \varepsilon_{zero} = 0.001$ in the algorithm BBB
- maximum number of subdivisions to be 100 in the algorithm BBB

To compare the performance of an NMPC scheme with the Bernstein algorithm (BNMPC), we choose a well-established control scheme from the power systems literature, namely a power system stabilizer (PSS). The tuned parameters of PSS were adopted from [16, p. 865]. Further, we also choose to compare with respect to linear MPC scheme and NMPC scheme based on YALMIP global optimization solver BMIBNB [17]. The linear MPC is designed based on the linearized model of (18)-(21) around the equilibrium point, $\delta = 1.225$, $\omega = 1$, $e'_q = 1.023$, and $e_{fd} = 2.42$. **The resulting optimization problems were solved using MATLAB 'quadprog' solver [21].** Note that for all control schemes, the control law (u) derived was injected directly to the nonlinear system (18)-(21). All control simulations were performed in the MATLAB environment [21] on a desktop PC running an Intel®Core i7-5500U CPU processor at 2.40 GHz with a 8 GB RAM. We now briefly discuss the simulation results of the two scenarios.

Scenario I: In order to validate the effectiveness of an BNMPC scheme under a disturbance, a short-circuit fault lasting 100 milliseconds is considered between the two parallel transmission lines. The fault occurs at $t = 3$ seconds and is cleared at $t = 3.1$ seconds. This event perturbs the states (δ , ω , e'_q , e_{fd}) from their equilibrium. The system may become unstable during the post-fault period due to insufficient damping. Hence, the controller has two roles: i) to stabilize the system in the post-fault period; and ii) to bring the states back to their equilibrium point.

Fig. 4 shows the rotor angle (δ) response of the synchronous generator. The dotted line indicates the system response with the classical PSS; solid red line indicates case with our BNMPC; whereas solid blue line shows the system behaviour with the linear MPC scheme. It can be observed that the BNMPC scheme results in better damping compared to the PSS with quickly bringing the rotor angle back to the equilibrium point ($\approx 28\%$ reduction in the settling time) and slightly better settling time when compared with the linear MPC.

Similarly, we note that the synchronous generator speed deviation ($\Delta\omega$) is also disturbed by the short-circuit event. Fig. 5 shows the speed deviation response of the generator, where the dotted line indicates the system response with the classical PSS; solid red line indicates case with our BNMPC; whereas solid blue line shows the system behaviour with the linear MPC scheme. In this case too BNMPC ensures good transient stability compared with the PSS and slightly better settling time when compared with the linear MPC.

Similar findings can be seen in transient responses in the generator terminal voltage from Fig. 6. Fig. 7 shows the control signal delivered to the generator. We note that PSS had a large oscillating damping before settling. On the other hand, the BNMPC and linear MPC control moves had small

oscillations at the cost of offering better damping during the post-fault state. This is evident from the system responses shown in Figs. 4-6.

Scenario II: We now simulate the power system with initial nominal mechanical power ($P_m = 0.9$). Then we assume a sudden disturbance on the input side of the generator (such as, drop in the steam pressure used to rotate the turbines) which result in a change in the mechanical power (P_m) of the generator. We consider a 5% step change in P_m from its nominal value (*i.e.* 0.9 to 0.8548) at 2 seconds and again 20% step change in P_m (*i.e.* 0.8548 to 1.0258) at 5 seconds. For the first step change in P_m and constant load, the difference between the electrical power generated and the desired load is met by reducing the rotor speed (ω) and as a result the rotor angle (δ) settles down to a lower equilibrium value. The opposite is observed when P_m is increased by 20% from its nominal value.

The resulting rotor angle (δ) is shown in Fig. 8. The dotted line indicates the system response with the classical PSS; solid red line indicates case with our BNMPC; whereas solid blue line shows the system behavior with the linear MPC scheme. It can be observed that the BNMPC scheme is superior than the PSS in terms of the damping and settling time. The performance of linear MPC and BNMPC are almost equal for the first step change in P_m at 2 seconds. On the other hand, for the second step change in P_m at 5 seconds our BNMPC performs better than linear MPC ($\approx 7\%$ reduction in the overshoot).

Fig. 9 shows the generator terminal voltage response to the changes in the mechanical power P_m . It can be observed that in this large transient period the PSS scheme responds poorly. On the other hand, the BNMPC scheme performs satisfactorily, keeping the terminal voltage close to its nominal value of 1. In this case too the performance of our BNMPC is better than linear MPC, particularly for the second step change in P_m at 5 seconds.

To assess computational time demand of our BNMPC scheme, we compare it with YALMIP global optimization solver BMIBNB embedded in an NMPC and the linear MPC schemes. Specifically, we compare the computational times under the aforementioned Scenario I. Fig. 10 shows the time taken to compute the control move at each sampling instant (*i.e.* to solve an NLP of the form (3)-(6)) by a BMIBNB solver, algorithm BBB, and MATLAB linprog solver. **We observe that BMIBNB and quadprog solvers resulted in an average time of 0.12 and 0.003 seconds, respectively for the control move computation.** Further, we observed that for BNMPC without the use of hull pruning operator resulted in a average time of 0.035 seconds for the control move computation. On the other hand, in presence of the hull pruning operator, the control move computation time was well within the sampling period of 0.03 seconds—on average 0.015 seconds. Similar, findings are observed for Scenario II.

Finally, to study the scalability of our BNMPC scheme against the linear MPC scheme, we vary the prediction horizon N from 3 to 7. An increase in the prediction horizon results in an increase in number of decision variables

for the optimization problem (3)-(6). Fig. 11 shows the resulting average computational times for the solution of an optimization problem. In all cases, BNMPC was found to be slower than its counterpart, linear MPC, as expected. However, in both cases the average computational time rise was observed to be almost linear with the prediction horizon varying from 3 to 7.

V. CONCLUSIONS

In this work a nonlinear model predictive control (NMPC) scheme was presented. The specific highlight of our NMPC scheme was an improved branch-and-prune type Bernstein global optimization procedure to solve the optimization problems at each sampling instant. This NMPC scheme holds good potential for power systems and other applications, because it uses hull pruning procedure to handle constraints efficiently in the optimization problems at each sampling instant. Further, such NMPC scheme also benefits from the Bernstein algorithm due to its ability to locate correct globally optimal solutions with any desired accuracy, and quickly enough for solving online optimization problems. The scalability of our NMPC scheme for a varying prediction horizon and the computational comparison with respect to the linear MPC scheme were also found to be satisfactory. In summary, this paper provides further evidence that the Bernstein algorithm is an effective tool for NMPC.

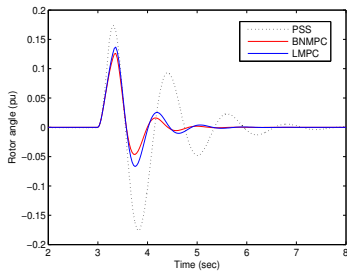


Fig. 4. Generator rotor angle response under a 100 ms short circuit fault (Scenario I).

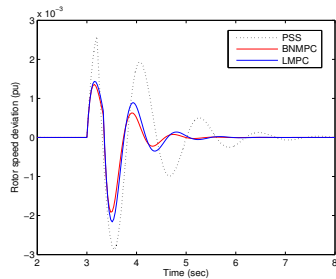


Fig. 5. Generator rotor speed response under a 100 ms short circuit fault (Scenario I).

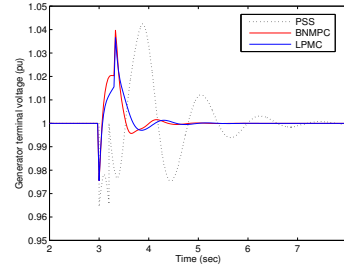


Fig. 6. Generator terminal voltage under a 100 ms short-circuit fault (Scenario I).

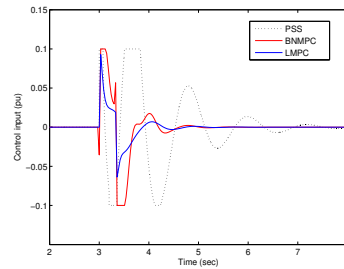


Fig. 7. Control signals under a 100 ms short-circuit fault (Scenario I).

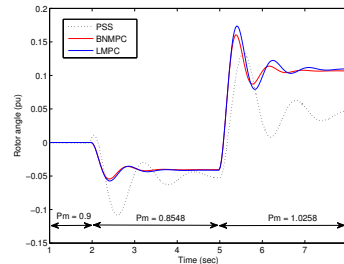


Fig. 8. Generator rotor angle response with changes in mechanical power input (Scenario II).

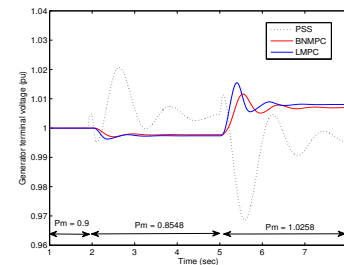


Fig. 9. Generator terminal voltage response with changes in mechanical power input (Scenario II).

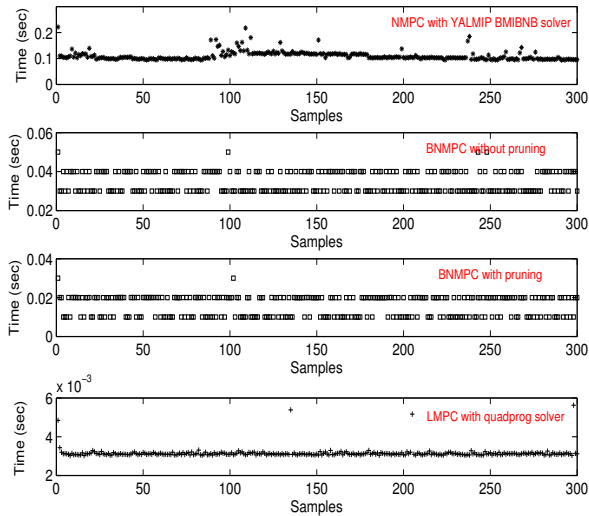


Fig. 10. Comparison of the computation time needed for a solution of an optimization problem (in MPC scheme) at each sampling instant: YALMIP global optimization solver BMIBNB, Bernstein algorithm (with and without pruning operator), and linear MPC (Scenario I). Sampling time is 30 ms.

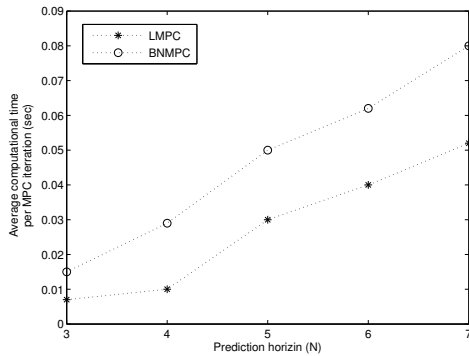


Fig. 11. Comparison of the average computation time for a solution of an optimization problem (in MPC scheme) at each sampling instant for varying prediction horizon (N): linear MPC (LMPC) and Bernstein algorithm based NMPC (BNMPC) (Scenario I).

REFERENCES

- [1] J. M. Maciejowski, *Predictive control with constraints*. UK, Harlow: Prentice Hall, 2002.
- [2] E. F. Camacho and C. Bordons, *Model predictive control, 2nd ed.* London: Springer-Verlag, 2004.
- [3] Y. Wan and J. Zhao, "Extended backstepping method for single-machine infinite-bus power systems with SMES," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 915–923, 2013.
- [4] S. S. Kaddah, K. M. Abo-Al-Ez, and T. F. Megahed, "Application of nonlinear model predictive control based on swarm optimization in power systems optimal operation with wind resources," *Electric Power Systems Research*, vol. 143, pp. 415–430, 2017.
- [5] 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.
- [6] 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.

- [7] M. Diehl, H. J. Ferreau, and N. Haverbeke, *Efficient numerical methods for nonlinear MPC and moving horizon estimation*. In: L. Magni, D. M. Raimondo, and F. Allgöwer (Eds.), *Nonlinear Model Predictive Control*. Springer: Lecture Notes in Control and Information Sciences, pp. 391–417, 2009.
- [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] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, pp. 1–19, 2016.
- [10] Inga J. Wolf and W. Marquardt, "Fast NMPC schemes for regulatory and economic NMPC— A review," *Journal of Process Control*, vol. 44, pp. 162–183, 2016.
- [11] L. Biegler, *Efficient solution of dynamic optimization and NMPC problems*. In: F. Allgöwer and A. Zheng (Eds.), *Nonlinear Model Predictive Control*. Springer Basel AG, pp. 219–243, 2000.
- [12] X. Wang, V. Mahalec, and F. Qian, "Globally optimal nonlinear model predictive control based on multi-parametric disaggregation," *Journal of Process Control*, vol. 57, pp. 1–13, 2017.
- [13] B. V. Patil, J. Maciejowski, and K. V. Ling, "Nonlinear model predictive control based on Bernstein global optimization with application to a nonlinear CSTR," *IEEE Proceedings of European Control Conference (ECC)*, pp. 471–476, 2016.
- [14] B. V. Patil, J. Maciejowski, and K. V. Ling, "Nonlinear model predictive control based on improved Bernstein global optimization algorithm with application to power systems," *20th IFAC World Congress*, July 9–14, Toulouse, 2017.
- [15] H. Ratschek and J. Rokne, *New computer methods for global optimization*. Chichester, England: Ellis Horwood Publishers, 1988.
- [16] P. Kundur, *Power system stability and control, 1st edition*. New York: McGraw-Hill Education, 1994.
- [17] J. Lofberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," *IEEE International Symposium on computer Aided Control Systems Design*, pp. 282–289, 2004.
- [18] 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.
- [19] J. Garloff, "The Bernstein algorithm," *Interval Computations*, vol. 2, pp. 154–168, 1993.
- [20] E. R. Hansen and G. W. Walster, *Global optimization using interval analysis, 2nd edition*. New York: Marcel Dekker, 2005.
- [21] The Mathworks Inc., MATLAB version 8.3 (R2014a), Natick, MA, 2014.