Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Polynomial B-Spline Methods For Nonconvex Optimization Problems

Deepak Gawali

Systems & Control Engineering Department, Indian Institute of Technology, Bombay.

Abstract:
Constrained optimization problems with multivariate polynomial objective functions provide a useful framework for addressing several problem types in engineering. For the purpose of constrained global optimization of multivariate polynomial functions, we offer techniques based on the polynomial B-spline form. Using a bound-and-prune structure, the proposed algorithms are developed. The suggested fundamental constrained global optimization algorithms were put to the test using test issues drawn from the field of systems analysis. The findings are consistent with what has been published.

Keywords: B-spline expansion, System analysis, Constrained optimization.

I. INTRODUCTION

The primary issue in system analysis is determining the minimal distance to the surface that is determined by the polynomial constraint $f(x) = 0$. It is possible to frame it as a problem of constrained optimization.

$$\rho^* = \min_{z \in R^n} \|z\|_2^2$$
$$\text{s.t. } f(z) = 0.$$

The majority of approaches found in works for addressing the minimal distance problem rely on the utilization of linear matrix inequality techniques. [1][2]. These methods needs the correct homogeneous form of the polynomials. In general, the challenge of determining the minimal distance may be formulated as a constrained global optimization task within the realm of nonconvex programming problems. In this field of research concerned with identifying the optimal value, or the best possible outcome, for a given problem. The problem of constrained global optimization for non-linear programming (NLP) can be formulated as follows:

$$\min_{z \in \mathbf{z}} f(z)$$
$$\text{s.t. } c_p(z) \le 0, p = 1,2,\ldots,n \quad (1)$$
$$ceq_q(z) = 0, q = 1,2,\ldots,m$$

The branch-and-bound framework is a widely employed approach for addressing global optimization problems with constraints [3]. For example, some interval approaches [4][5] employ this framework in order to ascertain the global minimum of a particular non-linear programming problem (NLP). This study shows new ways to use B-splines to solve constrained multivariate optimization problems in the field of systems that are not convex and are not linear. The proposed approach focuses on optimizing objective functions and satisfying constraints that are only represented by polynomial functions. By transforming the power-form polynomial objective function and constraints into a polynomial B-spline form [6], [7], we can improve our search results. The coefficients of B-spline expansion then establish a minimum and maximum bound for the allowed variation in the goal function and restrictions.

Within the article, we study one example of the fundamental global optimization under constraints. This example include the issue of minimum distance problem. The aforementioned issues are simplified down to the form of a quadratic optimization problem that include multivariate polynomials, and then the suggested technique for constrained global optimization is applied to find a solution.

The benefits of the suggested strategy are: (i) it doesn't need to evaluate f and constraints (c_i & c_{eq_j}); (ii) it doesn't need an initial guess to kick off optimization; (iii) it ensures that the local minimum will be located within an accuracy threshold set by the user; and (iv) it doesn't need prior knowledge of stationary points.

II. BACKGROUND: B-SPLINE EXPANSION

In the first place, we will provide a quick introduction to B-spline expansion. The range of in power from polynomial is obtained by using the B-spline expansion. After that, the B-spline shape is used as the foundation for the primary zero finding procedure in section 3.

So as to acquire the B-spline expansion, we follow the approach described in [7] and [6]. Consider $F(x_1, \cdots x_v)$ represent a multivariate polynomial in v real variables, where the polynomial has the largest degree $(d_1 + \cdots + d_v)$ (2).

$$F(x_1, \cdots x_v) = \sum_{p_1=0}^{d_1} \cdots \sum_{p_v=0}^{d_v} c_{p_1 \cdots p_v} x_1^{p_1} \cdots x_v^{p_v}. \quad (2)$$

2.1 Univariate polynomial

Lets consider univariate polynomial case first, (3)

$$\mathrm{F(x)} = \sum_{\mathrm{p=0}}^{\mathrm{d}} \mathrm{c_v t x^v}, \mathrm{x} \in [\mathrm{a,b}], \quad (3)$$

For a given degree m, this is equivalent to an order of m+1. The B-spline expansion is defined on a compact interval I=[a,b], where the condition $\mathrm{m} \geq \mathrm{d}$ holds. The splines with a degree of m on a partition of the uniform grid is referred to as the Periodic or Closed knot vector, and it is denoted by the letter, $\mathbf{w}$, and denoted as $\Omega_{\mathrm{m}}(\mathrm{J}, \mathbf{w})$, and $\mathbf{w}$ is given as,

$$\mathbf{w} := \{\mathrm{x_0} < \mathrm{x_1} < \cdots < \mathrm{x_{s-1}} < \mathrm{x_s}\}. \quad (4)$$

The value of $\mathrm{x_j} := \mathrm{a} + \mathrm{jz}$, $0 \leq \mathrm{j} \leq \mathrm{s}$, where s denotes number segments of B-spline and $\mathrm{z} := (\mathrm{b} - \mathrm{a})/\mathrm{s}$.

Let's say that $\mathbf{N}_{\mathrm{q}}$ represents the space occupied by splines of degree q. The degree q splines with $\mathrm{C}^{\mathrm{q}-1}$ continue on $[\mathrm{a,b}]$ and $\mathbf{w}$ as knot vector is thus designated by the following notation:

$$\Omega_{\mathrm{q}}(\mathrm{I}, \mathbf{w}) := \{\Omega \in \mathrm{C}^{\mathrm{q}-1}(\mathrm{I}) : \Omega|[\mathrm{z_j}, \mathrm{z_{j+1}}] \in \mathbf{N}_{\mathrm{q}},\ \mathrm{j} = 0, \cdots, s-1\}. \quad (5)$$

Since $\Omega_q(I, \boldsymbol{w})$ is $(s+q)$ dimension linear space [8]. To provide a foundation for locally supported splines, $\Omega_q(I, \boldsymbol{w})$, we required some extra knots $z_{-q} \leq \cdots \leq z_{-1} \leq a$ and $b \leq z_{s+1} \leq \cdots \leq z_{s+q}$ clamed at the ends of knot vector which are called as Clamped knot vectors, (6). Elements of Open or Clamped knot vector $\boldsymbol{w}$ is obtained as $z_j := a + ju$,

$$\boldsymbol{w} := \{z_{-q} \leq \cdots \leq z_{-1} \leq a = z_0 < z_1 < \cdots < z_{s-1} < b = z_s \leq z_{s+1} \leq \cdots \leq z_{s+q}\}. \quad (6)$$

The B-spline basis $\left\{B_j^q(z)\right\}_{j=1}^{s-1}$ of $\Omega_q(I, \boldsymbol{w})$ is defined in terms of divided differences:

$$B_j^q(z) := \left(z_{j+q} - z_j\right)[z_j, z_{j+1}, \cdots, z_{j+q+1}](.-z)_+^q, \quad (7)$$

where $(.)_+^q$ represent degree truncation. This can be simply shown as

$$B_j^q(z) := \Omega_d\left(\frac{z-a}{h} - i\right), -q \leq j \leq s-1, \quad (8)$$

where

$$\Delta_q(z) := \frac{1}{q!}\sum_{i=0}^{q+1}(-1)^i \binom{q+1}{v} (z-v)_+^q,$$

(9)

$B_j^q(z) := (z_{j+q} - z_j)[z_j, z_{j+1}, \cdots, z_{j+q+1}](.-z)_+^q$, is degree q basis function. The expression for basis in B-spline form is facilitated by following Cox-deBoor recursion formula,

$$B_j^q(z) := \beta_{j,q}(z) B_j^{q-1}(z) + \left(1 - \beta_{j+1,q}(z)\right) B_{j+1}^{q-1}(z),\ q \geq 1, \qquad (10)$$

where

$$\beta_{j,q}(z) = \begin{cases} \frac{z-x_j}{z_{j+q}-z_j}, & \text{if } z_j \leq z_{j+q}, \\ 0, & \text{otherwise}, \end{cases} \qquad (11)$$

and

$$B_j^0(z) := \begin{cases} 1, & \text{if } z \in [z_j, z_{j+1}), \\ 0, & \text{otherwise}. \end{cases} \qquad (12)$$

The spline basis set $\{B_j^q(z)\}_{j=1}^{s-1}$ has the following desirable characteristics:

1. Every $B_j^q(z)$ is greater than zero on $[z_j, z_{j+q+1}]$.
2. The spline basis set $\{B_j^q(z)\}_{j=1}^{s-1}$ shows a partition of unity, i.e. $$\sum_{j=1}^{v-1} B_j^q(z) = 1.$$

The following relation may be used to express the $\{z^l\}_{l=0}^m$ in (3) in terms of B-spline.

$$z^l := \sum_{r=-q}^{v-1} \pi_r^{(l)} B_r^q(z), l = 0, \cdots, q,$$

(13)

and the symmetric polynomial $\pi_r^{(l)}$ defined as

$$\pi_r^{(l)} := \frac{\mathrm{Sym}_s(r+1, \cdots, r+q)}{s^l \binom{q}{l}},\ l = 0, \cdots, q. \qquad (14)$$

Then by substituting (13) in (3) we get the power form polynomial (3)'s B-spline extension as follows:

$$F(z) := \sum_{p=0}^{m} c_p \sum_{r=-q}^{v-1} \pi_r^{(l)} B_r^q(z) = \sum_{r=-q}^{v-1} \left[\sum_{p=0}^{m} c_p \pi_r^{(l)} \right] B_r^q(z) = \sum_{r=-q}^{s-1} D_n B_r^q(z), \tag{15}$$

where

$$D_n := \sum_{p=0}^{m} c_p \pi_r^{(l)} . \tag{16}$$

2.2 Multivariate polynomial case

Let us now investigate B-spline form of following power form polynomial in a number of variables (17),

$$P(z_1, \cdots, z_v) := \sum_{g_1=0}^{k_1} \cdots \sum_{g_v=0}^{k_v} c_{g_1 \cdots g_v} z_1^{k_1} \cdots z_v^{k_v} = \sum_{\boldsymbol{g} \le \boldsymbol{k}} a_{\boldsymbol{g}}\, z^{\boldsymbol{k}}, \tag{17}$$

where $\boldsymbol{g} := (g_1, \cdots, g_v)$ and $\boldsymbol{k} := (k_1, \cdots, k_v)$. Substituting (13) for each z^k, (17) may also be expressed as

$$F(z_1, z_2, \ldots, z_v) = \sum_{l_1=0}^{m_1} \ldots \sum_{l_v=0}^{m_v} c_{l_1 \ldots l_v} \sum_{u_1=-q_1}^{k_1-1} \pi_{u_1}^{(l_1)} B_{u_1}^{q_1}(z_1) \ \ldots \sum_{u_v=-q_v}^{k_v-1} \pi_{u_v}^{(l_v)} B_{u_v}^{q_v}(z_v),$$

$$= \sum_{u_1=-q_1}^{k_1-1} \ldots \sum_{u_v=-q_v}^{k_v-1} \left(\sum_{l_1=0}^{m_1} \ldots \sum_{l_v=0}^{m_v} c_{l_1 \ldots l_v} \pi_{u_1}^{(l_1)} \ldots . \pi_{u_v}^{(l_v)} \right) B_{u_1}^{q_1}(z_1) \ldots B_{u_v}^{q_v}(z_v), \tag{18}$$

$$= \sum_{u_1=-q_1}^{k_1-1} \ldots \sum_{u_v=-q_v}^{k_v-1} D_{u_1 \ldots u_v} B_{u_1}^{q_1}(z_1) \ldots B_{u_v}^{q_v}(z_v),$$

we can write (18) as

$$F(z) := \sum_{u \le k} D_u \, B_u^k(z). \tag{19}$$

where $u := (u_1, \cdots, u_v)$ and D_u is B-spline coefficient given as

$$D_{u_1 \ldots u_v} = \sum_{l_1=0}^{m_1} \ldots \sum_{l_v=0}^{m_v} c_{l_1 \ldots l_v} \pi_{u_1}^{(l_1)} \ldots . \pi_{u_v}^{(l_v)}. \tag{20}$$

Equation (18) gives B-spline expansion of equation (17). A polynomial derivative in a specific direction may be determined by using the values of D_u, these are the coefficients of the equation (18) for $\boldsymbol{y} \subseteq I$. The derivative of $F(x)$ in direction x_r is represented by equation (21).

$$F_r'(\boldsymbol{y}) = \frac{m_r}{\boldsymbol{w}_{s+m_r+1} - \boldsymbol{w}_{s+1}} \times \sum_{I \le \boldsymbol{m}_{r,-1}} \left[D_{\boldsymbol{s}_{r,1}}(\boldsymbol{y}) - D_s(\boldsymbol{y}) \right] B_{\boldsymbol{m}_{r,-1},\boldsymbol{s}}(x), \; 1 \le r \le v, x \in \boldsymbol{y}, \tag{21}$$

If $\boldsymbol{w}$ is a knot vector then partial derivative $F_r'(\boldsymbol{y})$ gives the bound of the range enclosure for the derivative of F with respect to $\boldsymbol{y}$. In their work, Lin and Rokne proposed (14) for symmetric polynomials, using a closed or periodic knot vector. As a result of the modification in the knot vector from (4) to (6), we suggest a revised formulation of (14) in the subsequent manner,

$$\pi_u^{(l)} := \frac{\mathrm{Sym}_v(u+1, \cdots, u+q)}{\binom{q}{l}}. \tag{22}$$

2.3 B-spline range enclosure property

$$F(z) := \sum_{i=1}^{m} D_i B_i^q(z), z \in \boldsymbol{y}. \tag{23}$$

Consider the B-spline expansion (23) representing the polynomial $g(t)$ in power form. Let $\bar{g}(\boldsymbol{y})$ indicate the range of $g(t)$ on subbox $\boldsymbol{y}$. The array $D(\boldsymbol{y})$ consists B-spline coefficients. Then for $D(\boldsymbol{y})$ it holds

$$\bar{g}(\boldsymbol{y}) \subseteq D(\boldsymbol{y}) = [min\, D\,(\boldsymbol{y}), max\, D\,(\boldsymbol{y})]. \tag{24}$$

The interval formed by the lowest and maximum values of B-spline coefficients gives bound for the range of equation (17) g on $\boldsymbol{y}$.

2.4 Domain division procedure

The enclosure of range achieved by B-spline expansion may be enhanced by using the technique of domain division of subbox $\boldsymbol{y}$. Let

$$\boldsymbol{y} := [\underline{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_1] \times \cdots \times [\underline{\boldsymbol{y}}_r, \bar{\boldsymbol{y}}_r] \times \cdots \times [\underline{\boldsymbol{y}}_v, \bar{\boldsymbol{y}}_v],$$

the box that has to be consider for domain subdivison in the rth direction $(1 \leq r \leq v)$. It results in two subboxes $\boldsymbol{y_A}$ and $\boldsymbol{y_B}$ as follows

$$\boldsymbol{y_A} := [\underline{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_1] \times \cdots \times [\underline{\boldsymbol{y}}_r, m(\boldsymbol{y}_r)] \times \cdots \times [\underline{\boldsymbol{y}}_v, \bar{\boldsymbol{y}}_v],$$
$$\boldsymbol{y_B} := [\underline{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_1] \times \cdots \times [m(\boldsymbol{y}_r), \bar{\boldsymbol{y}}_r] \times \cdots \times [\underline{\boldsymbol{y}}_v, \bar{\boldsymbol{y}}_v],$$

where $m(\boldsymbol{y}_r)$ is a midpoint of $[\underline{\boldsymbol{y}}_r, \overline{\boldsymbol{y}}_r]$.

III. SUMMARY OF THE PROPOSED ALGORITHM

The underlying B-spline algorithm approach is similar to the one described in [9] for global optimization of nonlinear polynomials. This is a summary of the algorithm.

Step 1: The algorithm makes use of the array of polynomial coefficients of the objective function, denoted by A_o, as well as the arrays denoting the inequality constraints, denoted by A_{g_i} and the equality constraints, denoted by A_{h_j}. A cell structure known as A_c is used to hold these arrays of coefficients.

Step 2: Consider N_c comprises degree vectors N, $N_{c_i}$and $N_{c_{eq_j}}$, $i = 0, \dots, n$. How often a certain variable occurs in f and constraints $(c_i\ \&\ c_{eq_j})$ is represented by the length of the corresponding degree vector.

Step 3: Since the B-spline is having order of the B-spline plus one segments equal, the degree vector is used to calculate the number of segments. The vectors K_o, K_{c_i}, and $K_{c_{eq_j}}$ are computed as $K = N + 2$ using degree vectors N, N_{c_i} and $N_{c_{eq_j}}$ and entered in K_c cell like structure.

Step 4: Using the proposed method coefficients of B-spline for f and constraints $(c_i \, \& \, c_{eq_j})$ on the starting search box $\boldsymbol{x}$ are then calculated and kept in the arrays $D_o(\boldsymbol{y})$, $D_{g_i}(\boldsymbol{y})$ and $D_{h_j}(\boldsymbol{y})$, respectively.

Step 5: We begin by setting current lowest estimate, denoted by $\tilde{e}$ as largest coefficient of polynomial B-spline form of f on $\boldsymbol{x}$, i.e. $\tilde{e} = max\, D_o\,(\boldsymbol{y})$.

Step 6: The next step is to zero out all of the components of a flag vector designated as $F := (F_1, \ldots, F_p, F_{p+1}, \ldots, F_{p+q}) = (0, \ldots, 0)$. The efficiency of the method is improved by the use of the flag vector F. Consider, $c_i\,(y) \leq 0$ meets the requirement on y belong to the box $\boldsymbol{y}$, i.e. $c_i\,(y) \leq 0$ for $y \in \boldsymbol{y}$. If such is the case, there is no requirement to verify it once again $c_i(y) \leq 0$ for all other subbox $\boldsymbol{y}_0 \subseteq \boldsymbol{y}$. The same can be said about c_{eq_j}. We make use of flag vector in order to manage this information $F = (F_1, \ldots, F_p, , F_{p+1}, \ldots, F_{p+q})$ where the elements of F_f, takes either the value 0 or 1, as will be seen below:

a) $F_f = 1$ In either case if the f^{th}, c_i or c_{eq_j} is met.

b) $F_f = 0$ In either case if the f^{th}, constraint of c_i or c_{eq_j} is not met.

Step 7: Consider $\mathcal{L}$ as a running list assigned with the item $\mathcal{L} \leftarrow \{\boldsymbol{y}, D_o(\boldsymbol{y}), D_{g_i}(\boldsymbol{y}), D_{h_j}(\boldsymbol{y}), F\}$, and a list of possible solutions $\mathcal{L}^{sol}$to the empty list.

Step 8: Place items in descending order of $(min\, D_o\,(\boldsymbol{y}))$ order in $\mathcal{L}$.

Step 9: Start the algorithm. If $\mathcal{L}$ has no item to process then implement Step 14. Else select the last item from $\mathcal{L}$, represent it as $\{\boldsymbol{y}, D_o(\boldsymbol{y}), D_{g_i}(\boldsymbol{y}), D_{h_j}(\boldsymbol{y}), F\}$, and discard it's entry in $\mathcal{L}$.

Step 10: Implement speed accelerating algorithm as: the bounds of the function's range enclosure is determined by the lowest and maximum B-spline coefficients. Let $\tilde{e}$ is a current lowest estimate, and $\{\boldsymbol{y}, D(\boldsymbol{y})\}$ be the item that is being processed at the moment, in which case $\tilde{e} \leq min\, D\,(\boldsymbol{y})$. Then, the global minimizer cannot be contained by $\{\boldsymbol{y}, D_o(\boldsymbol{y}), D_{g_i}(\boldsymbol{y}), D_{h_j}(\boldsymbol{y}), F\}$ and must be discard this item if $min\, D_o\,(\boldsymbol{y}) > \tilde{p}$ and return to Step 9.

Step 11: Decision on subdivision. If

$$(\text{wid}\ \boldsymbol{y})\ \text{and}\ (max\, D_o\,(\boldsymbol{y}) - min\, D_o\,(\boldsymbol{y})) < \epsilon$$

then augment the item $\{\boldsymbol{x}, \text{min}\ D_0(\boldsymbol{x})\}$ to $\mathcal{L}^{sol}$and go to step 9. Else go to Step 12. Here ϵ represents a margin of error.

Step 12: Domain subdivision results into two sub boxes. Domain subdivision is done in the most distant direction of $\boldsymbol{y}$ at midpoint. It results into two subboxes $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ such that $\boldsymbol{y} =$

$\boldsymbol{y}_1 \cup \boldsymbol{y}_2$.

Step 13: For $r = 1,2$

1. Set $F^r := (F_1^r, \dots, F_p^r, F_{p+1}^r, \dots, F_{p+q}^r) = F$
2. Calculate the objective and constraints polynomial B-spline coefficient arrays on $\boldsymbol{y}_r$ and get range enclosure $\mathbb{D}_o(\boldsymbol{y}_r)$, $\mathbb{D}_{g_i}(\boldsymbol{y}_r)$, and $\mathbb{D}_{h_j}(\boldsymbol{y}_r)$ for for f and constraints (c_i & c_{eq_j}).
3. Consider $\tilde{e}_{local} = min(\mathbb{D}_o(\boldsymbol{y}_r))$.
4. If $\tilde{e}_{local} > \tilde{e}$ then go to Step 9.
5. for $i = 1, \dots, p$ if $F_i = 0$ then
    a. If $\mathbb{D}_{g_i}(\boldsymbol{b}_r) > 0$ then implement Step 6.
    b. If $\mathbb{D}_{g_i}(\boldsymbol{b}_r) \le 0$ then set $F_i^r = 1$.
6. for $j = 1, \dots, q$ if $F_{p+j} = 0$ then
    a. If $0 \notin \mathbb{D}_{h_j}(\boldsymbol{b}_r)$ then implement Step 9.
    b. If $\mathbb{D}_{h_j}(\boldsymbol{b}_r) \subseteq [-\epsilon_{zero}, \epsilon_{zero}]$ then set $F_{p+j}^r = 1$.
7. If $F^r = (1, \dots, 1)$ then set $\tilde{e} := min(\tilde{e}, max(\mathbb{D}_o(\boldsymbol{b}_r)))$.
8. Add item $\{\boldsymbol{b}_r, D_o(\boldsymbol{b}_r), D_{g_i}(\boldsymbol{b}_r), D_{h_j}(\boldsymbol{b}_r), F^r\}$ to the list $\mathcal{L}$.
9. For loop End

Step 14: Equalize current minimal approximation to the global minimum as, $\hat{e} = \tilde{e}$.

Step 15: Set all global minimizer(s) $\boldsymbol{z}^{(i)}$ as the initial entries of items in $\mathcal{L}^{sol}$ for which $min\, D_o(\boldsymbol{x}) = \hat{e}$.

Step 16: Terminate the algorithm and retrieve the global minimum $\hat{e}$ and all minimizers $\boldsymbol{z}^{(i)}$found.

IV. NUMERICAL RESULTS

The calculations are carried out on a personal computer with an PC having i3-370M, 2.40 GHz processor and 6 GB of RAM, while the techniques themselves are performed in MATLAB [10]. For the purpose of determining the $\hat{e}$ and $\boldsymbol{z}^{(i)}$, an accuracy of at least $\epsilon = 10^{-6}$ is required. The computation time in seconds is reported. This problem is derived from sources [1] and [12]. The following state-space system should be taken into consideration.

$$\dot{x} = A(y)x(t),$$

consider $x \in \mathbb{R}^n$ is the vector representing the state and $y = (y_1, y_2, \dots, y_n)' \in \mathbb{R}^n$ is vector denoted as is representative of the uncertain parameters. Considering that A(0) is a Hurwitz matrix, The parametric stability margin, denoted as l_2 is defined as

$$\rho_2 = \sqrt{\rho^*} = \sqrt{min\{\rho_R, \rho_I\}}.$$

The variable ρ_R represents the answer of the optimization problem subject to equality constraints.

$$\rho_R = min_{z \in R^n}\, z_1^2 + z_2^2$$

$$\text{s.t. } det[\,A(z)] = 0,$$

and ρ_I represents the answer to a separate optimization problem that is subject to equality constraints.

$$\rho_I = min_{z \in R^n}\, z_1^2 + z_2^2,$$

$$\text{s.t. } H_{n-1}[A(z)] = 0.$$

This minimal distance problem transforms into a quadratic optimization problem. Consider i $A(y)$ is a polynomial on $\boldsymbol{y}$. In the case of the specific illustration given in [12], we have

$$det[\,A(y)] = -3y_1^3 - 7y_1^2y_2 - 2y_1y_2^2 - 2y_2^3 - 4y_1^2 + y_2^2 + 2y_1 + 2x_2 - 1,$$
$$H_{n-1}[\mathrm{A(y)}] = -8\mathrm{y}_1^3 - 4\mathrm{y}_1\mathrm{y}_2 - 2\mathrm{y}_1\mathrm{y}_2^2 - 28\mathrm{y}_1^2 + \mathrm{y}_1\mathrm{y}_2 - 3\mathrm{y}_2 - 22\mathrm{y}_1 - 7\mathrm{y}_2 + 8,$$
$$\mathbf{y}_1 = [0{,}0.5], \mathbf{y}_2 = [0{,}0.5].$$

The suggested approach solves the initial constrained optimization issue by locating its global minimum as

$$\rho_R = 0.2083,$$

whereas solving a second constrained optimization problem, it locates its global minimum,

$$\rho_I = 0.0664.$$

Therefore, the smallest possible stability margin on a global scale is

$$\rho^* = \min\{\rho_R, \rho_I\} = 0.0664,$$

expressing the parametric stability margin for l_2 as

$$\rho_2 = \sqrt{\rho^*} = 0.2576.$$

The findings presented in this study are consistent with the results published in previous studies [1][12].

V. CONCLUSION

We proposed a constrained global optimization algorithm to solve the minimum distance problem using polynomial B-spline form as an inclusion function to bound the range of nonlinear multivariate polynomial function. The approach solves the issue to the required precision without resorting to linearization or relaxation techniques.

REFERENCES

[1] Floudas CA, Pardalos PM. Handbook of Test Problems in Local and Global Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands; 1999.
[2] Horst R, Pardalos PM. Handbook of Global Optimization. Kluwer Academic, Netherlands; 1995.
[3] Hansen E, Walster G. Global Optimization Using Interval Analysis: Revised and Expanded. Vol 264. (2, ed.). Marcel Dekker, New York; 2004.
[4] Jaulin L. Applied Interval Analysis: with Examples in Parameter and State Estimation. Robust Control Robot Springer. 2001
[5] Stahl V. Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations. PhD Thesis, Johannes Kepler University, Linz.; 1995.
[6] Lin Q, Rokne JG. Interval approximation of higher order to the ranges of functions. Comput Math with Appl. 1996;31(7):101-109.
[7] Lin Q, Rokne JG. Methods for bounding the range of a polynomial. J Comput Appl Math. 1995;58:193-199.
[8] DeVore RA, Lorentz GG. Constructive Approximation. Vol 303. Springer Science & Business Media, Berlin; 1993.
[9] Patil B. V., Nataraj P. S. V., Bhartiya S. Global optimization of mixed-integer nonlinear (polynomial) programming problems: the {B}ernstein polynomial approach. Computing. 2012;94(2-4):325-343.
[10] Mathworks Inc. MATLAB Version 8.0.0.783 ({R} 2012 B). Inc., Natick, Massachusetts, United States; 2012.
[1] Henrion, D., Lasserre JB. Solving Nonconvex Optimization Problems. Control Syst IEEE. 2004;(24):72-83.
[2] Chesi G, Garulli A, Tesi A, Vicino A. Solving quadratic distance problems: an LMI-based approach. IEEE Trans Automat Contr. 2003;48(2):200-212.
[3] Horst R, Pardalos PM. Handbook of Global Optimization. Kluwer Academic, Netherlands; 1995.
[4] Hansen E, Walster G. Global Optimization Using Interval Analysis: Revised and Expanded. Vol 264. (2, ed.). Marcel Dekker, New York; 2004.

[5] Jaulin L. Applied Interval Analysis: with Examples in Parameter and State Estimation. Robust Control Robot Springer. 2001;1.
[6] Stahl V. Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations. PhD Thesis, Johannes Kepler University, Linz.; 1995.
[7] Lin Q, Rokne JG. Interval approximation of higher order to the ranges of functions. Comput Math with Appl. 1996;31(7):101-109.
[8] Lin Q, Rokne JG. Methods for bounding the range of a polynomial. J Comput Appl Math. 1995;58:193-199.
[9] DeVore RA, Lorentz GG. Constructive Approximation. Vol 303. Springer Science & Business Media, Berlin; 1993.
[10] Patil B. V., Nataraj P. S. V., Bhartiya S. Global optimization of mixed-integer nonlinear (polynomial) programming problems: the Bernstein polynomial approach. Computing. 2012;94(2-4):325-343.
[11] Mathworks Inc. MATLAB Version 8.0.0.783 (R 2012 B). Inc., Natick, Massachusetts, United States; 2012.
[12] Chesi G, Tesi A, Vicino A, Genesio R. An LMI approach to constrained optimization with homogeneous forms. Syst Control Lett. 2001;42(1):11-19.