ISSN: 2633-4828 Vol. 3 No.2, December, 2021

International Journal of Applied Engineering & Technology

ANALYSIS OF ROBUST STABILITY UTILIZING THE POLYNOMIAL B-SPLINE EXPANSION

Deepak Gawali
Systems & Control Engineering Department, Indian Institute of Technology, Bombay
ddgawali2002@gmail.com

ABSTRACT

Constrained global optimization of stringent inequalities (or equations) utilizing multivariate polynomials can be used to solve the challenges posed by robust stability analysis. For the purpose of constrained global optimization of multivariate functions, we offer techniques that are constructed using B-spline expansion. The branch-and-bound structure serves as the foundation for the proposed algorithms. By taking into account the robust stability analysis problem, we put the suggested fundamental restricted global optimization algorithms through their paces. The results obtained are consistent with those that have been published in the relevant literature.

Keywords: B-spline expansion, Stability analysis, Global optimization.

I. INTRODUCTION

It is well knowledge that the solution to the characteristic equation gives us insight into the stability of the feedback system. The expression that describes the characteristics of a system, taking into account uncertainty in the parameters may be expressed in polynomial form, and the uncertainty boundaries may be seen as system constraints. Both of these can be accomplished through the use of a polynomial equation. Finding the zone of parameter uncertainty in a linear system that allows the controller to stabilizing any systemic turbulence is what meant by *robust stability analysis* [1]. Finding the optimum solution for a problem is referred to as nonlinear programming problems (NLP), which is an acronym that stands for global optimization with constraints of nonlinear programming problems. In general, the robust stability analysis issues reduce to this type of optimization. This is a statement of the global constrained optimization of NLPs:

$$\min_{y \in y} f(z)$$

$$\text{s.t. } c_p(z) \le 0, p = 1,2,\ldots,n \qquad (1)$$

$$ceq_q(z) = 0, q = 1,2,\ldots,m$$

The branch-and-prune structure is frequently utilized in practice for the resolution of limited global optimization issues [2]. This structure is utilized by a number of interval approaches [3][4] for the purpose of locating the local minimum. This paper present B-spline methods for addressing nonconvex NLP problem in control system. Where f objective polynomial function and c_i, c_{eq_j} are *polynomial* constraints functions. Polynomial B-splines are constructed using the original power-form polynomial [5][6]. After then, the bounds on the range of f and c_i, c_{eq_j} using minimum and maximum coefficients values in B-spline expansion.

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

The proposed method has four advantages: (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).

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

2.1 Univariate polynomial

Lets consider univariate polynomial case first, (3)

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

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 $m \geq d$ 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, $\boldsymbol{w}$, and denoted as $\Omega_m(\mathrm{J}, \boldsymbol{w})$, and $\boldsymbol{w}$ is given as,

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

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

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

$$\Omega_q(I, \boldsymbol{w}) := \{\Omega \in C^{q-1}(I) : \Omega | [z_j, z_{j+1}] \in N_q, j = 0, \cdots, s-1\}. \qquad (5)$$

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 $z_j := a + ju$ for $j \in \{-q, \cdots, -1\} \cup \{s + 1, \cdots, s + q\}$,

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

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

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

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

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

where

$$\Delta_q(z) := \frac{1}{q!}\sum_{i=0}^{q+1}(-1)^i \binom{q+1}{v} (z - v)_+^q, \qquad (9)$$

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

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

where

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

and

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

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. $\sum_{j=1}^{s-1} B_j^q(z) = 1.$

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

$$z^l := \sum_{r=-q}^{v-1} \pi_r^{(l)} B_r^q(z),\ l = 0, \cdots, q, \tag{13}$$

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

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

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

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

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),

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

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

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

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

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

we can write (18) as

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

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

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

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 $\mathbf{y} \subseteq I$. The derivative of $F(x)$ in direction x_r is represented by equation (21).

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

If $\mathbf{w}$ is a knot vector then partial derivative $F_r'(\mathbf{y})$ gives the bound of the range enclosure for the derivative of F with respect to $\mathbf{y}$. 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,

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

2.3 B-Spline Range Enclosure Property

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

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

$$\bar{g}(\mathbf{y}) \subseteq \mathrm{D}(\mathbf{y}) = [\min \mathrm{D}(\mathbf{y}), \max \mathrm{D}(\mathbf{y})]. \quad (24)$$

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

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 $\mathbf{y}$. Let

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

the box that has to be consider for domain subdivison in the rth direction $(1 \le r \le v)$. It results in two subboxes $\mathbf{y}_A$ and $\mathbf{y}_B$ as follows

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

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

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

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 $D_o(\mathbf{y})$, $D_{g_i}(\mathbf{y})$ and $D_{h_j}(\mathbf{y})$, 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. $\tilde{e} = max\ D_o(\mathbf{y})$.

Step 6: The next step is to zero out all of the components of a flag vector designated as $F := (F_1, \dots, F_p, F_{p+1}, \dots, F_{p+q}) = (0, \dots, 0)$. 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 in the box $\mathbf{y}$, i.e. $c_i(y) \leq 0$ for $y \in \mathbf{y}$. If such is the case, there is no requirement to verify it once again $c_i(y) \leq 0$ for all other subbox $\mathbf{y}_0 \subseteq \mathbf{y}$. The same can be said about c_{eq_j}. We make use of flag vector in order to manage this information $F = (F_1, \dots, F_p, , F_{p+1}, \dots, F_{p+q})$ 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 a running list $\mathcal{L}$ assigned with the item $\mathcal{L} \leftarrow \{\mathbf{y}, D_o(\mathbf{y}), D_{g_i}(\mathbf{y}), D_{h_j}(\mathbf{y}), F\}$, and a list of possible solutions $\mathcal{L}^{sol}$to the empty list.

Step 8: Place items in descending order of $(min\ D_o(\mathbf{y}))$ 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 $\{\mathbf{y}, D_o(\mathbf{y}), D_{g_i}(\mathbf{y}), D_{h_j}(\mathbf{y}), F\}$, and discard it's entry from $\mathcal{L}$.

Step 10: Implement cut-off test 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 $\{\mathbf{y}, D(\mathbf{y})\}$ be the item that is being processed at the moment, in which case $\tilde{e} \leq min\ D(\mathbf{y})$. Then, this item surely the global minimizer cannot be contained and must be discard the item $\{\mathbf{y}, D_o(\mathbf{y}), D_{g_i}(\mathbf{y}), D_{h_j}(\mathbf{y}), F\}$ if $min\ D_o(\mathbf{y}) > \tilde{p}$ and return to Step 9.

Step 11: Decision on subdivision. If

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

then augment the item $\{\mathbf{x}, \min D_0(\mathbf{x})\}$ 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 $\mathbf{y}$ at midpoint. It results into two subboxes $\mathbf{y}_1$ and $\mathbf{y}_2$ such that $\mathbf{y} = \mathbf{y}_1 \cup \mathbf{y}_2$.

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 $\mathbf{y}_r$ and get range enclosure $\mathbb{D}_o(\mathbf{y}_r)$, $\mathbb{D}_{g_i}(\mathbf{y}_r)$, and $\mathbb{D}_{h_j}(\mathbf{y}_r)$ for for f and constraints $(c_i \,\&\, c_{eq_j})$.
3. Consider $\tilde{e}_{local} = min(\mathbb{D}_o(\mathbf{y}_r))$.
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 $\mathbb{D}_{g_i}(\boldsymbol{b}_r) \leq 0$ 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 global minimum to current minimum estimate 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 $min\, D_o(\mathbf{x}) = \hat{e}$.

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

IV. TEST 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. Consider $G_p(s)$ as plant transfer functions and $G_c(s)$ as controller transfer functions. The fundamental characteristic equation of a control system is

$$det(I - G_p(s)G_c(s)) = 0.$$

Now, let us take into consideration the presence of uncertainty in parameters, where $\boldsymbol{q}$ represents the vector of unknown parameters. Consider $G_p(s, \boldsymbol{q})$ and $G_c(s, \boldsymbol{q})$ as transfer functions for the plant and controller respectively, exhibiting a certain level of uncertainty. The equation that represents the characteristics of the system under consideration, taking into account the associated uncertainties, is expressed as follows:

$det(I - G_p(s,\boldsymbol{q})G_c(s,\boldsymbol{q})) = 0.$

Polynomial form of above determinant can be expressed in following form.

$F(s,\boldsymbol{q}) = a_n(\boldsymbol{q})s^n + a_{n-1}(\boldsymbol{q})s^{n-1} + \ldots + a_1(\boldsymbol{q})s + a_0(\boldsymbol{q}).$

The coefficients $a_i(\boldsymbol{q}), i = 0, \ldots, n$ are polynomial functions that involve more than one variable. One definition of a stability margin k_m is,

$k_m(j\omega) = \inf\{k: F(j\omega, \mathbf{b}(k)) = 0, \forall\, \mathbf{b} \in Q\}.$

If $k_m \geq 1$, then the stability margin is robust. Obtaining a stable solution with relation to a linear system that has a characteristic equation $F(j\omega, \boldsymbol{q})$, is a optimization problem that takes the form of the following.

$\min_{b_i, z \geq 0, \omega \geq 0} z$

s.t. $F(j\omega, \mathbf{b}) = 0,$

$F(j\omega, \mathbf{b}) = 0,$

$b_i^N - \triangle\, b_i^- z \leq b_i \leq b_i^N + \triangle\, b_i^+ z,\ i = 1, \ldots, n,$

The point q^N is considered stable in the presence of unknown parameters, whereas the estimated limits $\triangle\, q_i^+,\ \triangle\, q_i^-$ represent the range of possible values [1]. The aforementioned issue may be classified as a constrained global optimization problem, specifically considering multivariate polynomial functions.

In this problem, the global minimum must be determined in such case, the safety margin would be overestimated. An overestimate may lead to the incorrect conclusion that a certain system is stable when it is not stable [1]; this is an error. For the purpose of ensuring that the local minimum of k is, in fact, discovered, it is necessary to employ a well-established global optimization strategy. This capability is illustrated by the accompanying example.

Example : Examine the closed-loop system's l_∞ stability margin The issue with global optimization is presented by

$\min z$

s.t. $q_1^4 q_2^4 - q_1^4 - q_2^4 q_3 = 0,$

$1.4 - 0.25z \leq q_1 \leq 1.4 + 0.25z,$

$1.5 - 0.20z \leq q_2 \leq 1.5 + 0.20z,$

$0.8 - 0.20z \leq q_3 \leq 0.8 + 0.20z.$

There are four continuous variables q_1, q_2, q_3, and z. With one and six, c_{eq} and c_i constraints. The suggested approach has an accuracy of 10^{-6}, and results are z = 1.0899, as the global minimum and $q_1 = 1.1275, q_2 = 1.282, q_3 = 1.018$ as a global minimizer.

These findings are consistent with what was described in [1]. The amount of time necessary to find a solution to this issue is 58.85 seconds.

V. CONCLUSION

In this study, we have put forward a novel approach for addressing the robust stability analysis issue. Our suggested method involves the use of a constrained global optimization algorithm. Specifically, we employ the inclusion function in the form of a polynomial B-spline to establish bounds on the range of the multivariate nonlinear polynomial function. Proposed approach does not need the use of linearization and relaxing methods, but it is capable of solving the issue with the desired level of precision.

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 Bernstein 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.