An Improved Algorithm for Set Inversion using Interval Analysis with Application to Control System

P. S. V. Nataraj
Professor
Systems and Control Engineering
IIT Bombay, Mumbai - 400 076, India
nataraj@sc.iitb.ac.in

Manoj M. Deshpande
Research Scholar
Systems and Control Engineering
IIT Bombay, Mumbai - 400 076, India
mmdeshpande@ee.iitb.ac.in

Abstract— **We present an algorithm to characterize the set $S = \{x \in \Re^l : f(x) > 0\} = f^{-1}(]0, \infty[^m)$ in the frame work of set inversion using interval analysis. The proposed algorithm improves on the algorithm of Jaulin et. al. The improvement exploits powerful Hansen's method for solving systems of nonlinear inequalities. We test and compare the performance of the proposed and existing algorithms in characterizing the domain for the robust stability. The results of the testing show that the proposed algorithm is computationally efficient and encloses the solution more sharply than the existing algorithm, requires less memory space and iterations.**

I. Introduction

Let f be a nonlinear function from $\Re^l$ to $\Re^m$. Let Y be a subset of $\Re^m$. Then, the problem of set inversion can be posed as the characterization of

$$S = \{x \in \Re^l : f(x) \in Y\} = f^{-1}(Y) \tag{1}$$

In this work, we address the problem of characterizing S defined by a set of nonlinear inequalities

$$S = \{x \in \Re^l : f(x) > 0\} = f^{-1}(]0, \infty[^m) \tag{2}$$

Such problem arise, for instance, in robust stability analysis of feedback control system [1]. Jaulin et al. proposed an algorithm for solving the above problem via interval analysis. The algorithm has also been successfully tested in applications, such as characterization of robust stability domains [1], parameter and state estimation [2] and robotics [3].

This problem can be considered as a problem of finding feasible points or region for nonlinear inequalities. There are number of methods that are concerned with the feasible points of nonlinear inequalities. Those are gradient or gradient based methods [6],trust-region and/or penalty function algorithms [7]. None of these methods can guarantee to correctly find the feasible points.

In this work, we propose some improvements to the the algorithm of Jaulin et. al. for characterization of set S via set inversion. The proposed improvement is based on powerful Hansen's method for solving systems of nonlinear inequalities.

We test and compare the performance of the proposed and existing algorithms in characterizing the domain for robust stability. The results of the testing show that the proposed algorithm is computationally efficient and encloses the solution more sharply than existing algorithm- meaning that it deletes certainly infeasible region.

II. Solving Nonlinear Inequalities by Set Inversion via Interval Analysis

A. Interval Notations

Let $\Re$ be the set of reals, $x \in \Re^l, \mathbf{X} \subseteq \Re^l$ be an axis-aligned parallelepiped (also called as a box). Let $I(\mathbf{X})$ be the set of all boxes contained in $\mathbf{X}$. Interval is represented as $\mathbf{X} = [\underline{x}, \overline{x}]$, where $\underline{x}$ is infimum of $\mathbf{X}$ and $\overline{x}$ is supremum of $\mathbf{X}$. Let the width of $\mathbf{X}$ be defined as $w(\mathbf{X}) = max\mathbf{X} - min\mathbf{X}$ if $\mathbf{X} \subseteq I\Re$, and as $w(\mathbf{X}) = max\{w(\mathbf{X}_1), ..., w(\mathbf{X}_l)\}$, if $\mathbf{X} \subseteq I(\Re^l)$. Let the midpoint of $\mathbf{X}$ be defined as $m(\mathbf{X}) = (min(\mathbf{X}) + max(\mathbf{X}))/2$ if $\mathbf{X}$, and as $m(\mathbf{X}) = m(\mathbf{X}_1), .., m(\mathbf{X}_l)$, if $\mathbf{X} \in$.

Let $\bar{f}(\mathbf{X})$ denote the range of a function f on $\mathbf{X}$. A function $F : I(\mathbf{X}) \rightarrow I(\Re)$ is said to be inclusion function for f, if $\bar{f}(\mathbf{Y}) \subseteq F(\mathbf{Y})$ for all $\mathbf{Y} \in I(\mathbf{X})$. A natural inclusion function form is obtained from the expression for f by replacing all occurrence of x_i with $\mathbf{X}_i$ and all real operations with the corresponding interval operations.

B. Existing Algorithm

Let $\mathbf{X}^0 \subseteq \Re^l$ be the initial feasible box. The algorithm of Jaulin et. al.[1][4] encloses the portion of S contained in $\mathbf{X}^0$, between two partitions K_{in}, certainly feasible boxes and K_{out} in the sense that $K_{in} \subseteq \mathbf{X}^0 \cap S \subseteq K_{out}$ where, $K_{out} = K_{in} \cup K_\epsilon$ and K_ϵ is a list of all indeterminate boxes. The algorithm of Jaulin et. al. is as follows:
Inputs: The initial box $\mathbf{X}^0$, natural inclusion function F,

and an accuracy parameter ε_x.
Outputs: A list K_{in} of all boxes guaranteed to belong to S, and list $K_{out} = K_{in} \cup K_\epsilon$.
BEGIN Algorithm

1. Initialize $\mathbf{X} = \mathbf{X}^0, K_{in} = \{\}, K_{out} = \{\}$ list $L = \{\mathbf{X}\}$.

2. Remove the first box $\mathbf{X}$ from list L and evaluate $F_i(\mathbf{X})$.

3. If $inf(F_i(\mathbf{X})) > 0$ for all $i = 1, ..., m$ then store $\mathbf{X}$ in the listK_{in} and K_{out}, go to step 7.

4. If $sup(F_i(\mathbf{X})) \leq 0$ for any $i = 1, ..., m$ then go to step 7.

5. If $w(\mathbf{X}) \leq \varepsilon_x$, then deposit $\mathbf{X}$ in the list K_{out} and go to step 7.

6. Bisect $\mathbf{X}$ in maximum width coordinate direction, getting boxes $\mathbf{V}^1, \mathbf{V}^2$ such that $\mathbf{X} = \mathbf{V}^1 \cup \mathbf{V}^2$. Deposit these sub boxes in list L.

7. If the list L is empty, EXIT algorithm. Else go to step 2.

END Algorithm.

III. Proposed Algorithm

A. The Hansen's Method

Consider the problem of one nonlinear inequality [5], namely,

$$g(\mathbf{x}) \leq 0, \quad \mathbf{x} \in \mathbf{X}, \tag{3}$$

where $\mathbf{X}$ is a parallelepiped box with each component of x bounded by $\underline{x}_i$ from below and by $\overline{x}_i$ from above. Here $\underline{x}_i$ and $\overline{x}_i$ are predetermined. Applying the mean value theorem to the function $g(x)$, we have

$$g(x) = g(y) + (x-y)^T \nabla g(\xi),$$

where $\nabla g(\xi)$ is the vector of the first-order partial derivatives of the function $g(x)$ at the point $\xi \in X$. Since $\nabla g(\xi) \in \nabla g(\mathbf{X})$, (3) can be rewritten as

$$g(y) + (x-y)^T \nabla g(\mathbf{X}) \leq 0, \quad x \in X, \tag{4}$$

where $\nabla g(\mathbf{X})$ is the interval vector of the first-order derivatives of $g(x)$ computed within the box $\mathbf{X}$. Because use of (4) may risk of losing feasible points of (3), Hansen [5] propose to use

$$g(y) + (x-y)^T \nabla g(\mathbf{X}) > 0, \quad x \in \mathbf{X}, \tag{5}$$

in order to reduce the size of X or to bound the feasible point set. Since all the points x that satisfy (5) are certainly not feasible, no feasible points of the original problem (3) will be eliminated by deleting the solutions to (5) from $\mathbf{X}$ [5].

Since the linear interval inequality (5) is involved with n variables, we cannot solve it for the certainly infeasible points of (3) simultaneously for all the components to branch, say x_i of x with the longest length of side or the largest difference between the upper and lower bounds. Thus we can rewrite (5) as follows:

$$g(y) + \sum_{j=1, j\neq i}^{n} (\mathbf{X}_j - y_j) \acute{g_j}(\mathbf{X}) + (x_i - y_i) \acute{g_i}(\mathbf{X}) > 0. \tag{6}$$

Since all the derivatives in (6) are computed with X, their interval widths may be too large. In order to sharpen the interval bounds for the derivatives, we can also use the following alternative inequality:

$$\begin{aligned} g(y) + \sum_{j=1, j\neq i}^{n} (\mathbf{X}_j - y_j) \acute{g_j}(\mathbf{X}_1, ..., \mathbf{X}_j, y_{j+1}, ..., y_n) \\ + (x_i - y_i) \acute{g_i}(\mathbf{X}_1, ..., \mathbf{X}_i, y_{i+1}, ..., y_n) > 0, \end{aligned} \tag{7}$$

(see [5]). (6) (7) can be symbolically rewritten as follows:

$$\mathbf{U} + \mathbf{V}t > 0, \tag{8}$$

where $\acute{g_k}(.)$ is the derivative interval of $g(x)$ with respect to $x_k (k = 1, 2, ..., n)$,

$$\begin{aligned} \mathbf{U} &= g(y) + \sum_{j=1, j\neq i}^{n} (\mathbf{X}_j - y_j) \acute{g_j}(\mathbf{X}) = [\underline{u}, \overline{u}], \\ \mathbf{V} &= \acute{g_i}(\mathbf{X}) = [\underline{v}, \overline{v}] \end{aligned}$$

for the case of (6), and

$$\begin{aligned} \mathbf{U} &= g(y) + \sum_{j=1, j\neq i}^{n} (\mathbf{X}_j - y_j) \acute{g_j}(\mathbf{X}_1, ..., \mathbf{X}_j, y_{j+1}, ..., y_n), \\ \mathbf{V} &= \acute{g_i}(\mathbf{X}_1, ..., \mathbf{X}_i, y_{i+1}, ..., y_n) \end{aligned}$$

for the case of 7, and

$$t = x_i - y_i.$$

Let the solution set of t in (8) be denoted by $\mathbf{T}_c$, namely,

$$\mathbf{T}_c = \{t : u + vt > 0, u \in \mathbf{U}, v \in \mathbf{V}\}.$$

By direct analysis of (8) using the interval division rules(see e.g., [5],[8]), we obtain

$$\mathbf{T}_c = \begin{cases} [-\underline{u}/\overline{v}, \infty) & \text{if } \ \underline{u} > 0, \underline{v} \geq 0 \ \text{ and } \ \overline{v} > 0, \\ [-\underline{u}/\underline{v}, \infty) & \text{if } \ \underline{u} \leq 0 \ \text{ and } \ \underline{v} > 0, \\ (-\infty, -\underline{u}/\overline{v}] & \text{if } \ \underline{u} \leq 0 \ \text{ and } \ \overline{v} < 0, \\ (-\infty, -\underline{u}/\underline{v}] & \text{if } \ \underline{u} > 0, \underline{v} < 0 \ \text{ and } \ \overline{v} \leq 0, \\ [-\underline{u}/\overline{v}, -\underline{u}/\underline{v}] & \text{if } \ \underline{u} > 0 \ \text{ and } \ \underline{v} < 0 < \overline{v}, \\ (-\infty, \infty) & \text{if } \ \underline{u} > 0 \ \text{ and } \ \underline{v} = \overline{v} = 0, \\ (empty \ \ set) & \text{if } \ \underline{u} \leq 0 \ \text{ and } \ \underline{v} \leq 0 \leq \overline{v}. \end{cases} \tag{9}$$

It is interesting to note that the solution $\mathbf{T}_c$ is independent of the upper bound of $\mathbf{U}$; this nevertheless not obvious either in its original form (8) or from the definition of $\mathbf{T}_c$.

We can now compute the complement set of $\mathbf{T}_c$, denoted by $\mathbf{T}$, as follows:

$$\mathbf{T} = \{t : t \in (-\infty, \infty)\,\text{and}\,t \notin \mathbf{T}_c\}.$$

$$\mathbf{T} = \begin{cases} (-\infty, -\underline{u}/\overline{v}] & \text{if } \underline{u} > 0, \underline{v} \geq 0 \text{ and } \overline{v} > 0, \\ (-\infty, -\underline{u}/\underline{v}] & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} > 0, \\ [-\underline{u}/\overline{v}, \infty) & \text{if } \underline{u} \leq 0 \text{ and } \overline{v} < 0, \\ [-\underline{u}/\underline{v}, \infty) & \text{if } \underline{u} > 0, \underline{v} < 0 \text{ and } \overline{v} \leq 0, \\ [-\infty, -\underline{u}/\overline{v}] \cup [-\underline{u}/\underline{v}, \infty] & \text{if } \underline{u} > 0 \text{ and } \underline{v} < 0\overline{v}, \\ (-\infty, \infty) & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} \leq 0 \leq \overline{v}, \\ (empty\ \ set) & \text{if } \underline{u} > 0 \text{ and } \underline{v} = \overline{v} = 0 \end{cases} \tag{10}$$

It should be pointed out that although all the points in $\mathbf{T}_c$ are certainly infeasible, this does not mean that all the points in $\mathbf{T}$ are feasible. In fact, as the complement set of $\mathbf{T}_c$, $\mathbf{T}$ will generally contain many (but certainly not all) infeasible points satisfying (8).

With the solution set $\mathbf{T}$ of (10), we can then compute the reduced point set(s) of $\mathbf{X}$. The new interval of $\mathbf{X}$ along the coordinate axis x_i is given as follows:

$$\mathbf{X}_i^n \quad = \quad \mathbf{X}_i \cap (\mathbf{T} + y_i),$$

where $\mathbf{X}_i^n$is the new interval for the component x_i. If $\mathbf{T}$ consists of two parts, say $\mathbf{T}_{i1}$ and $\mathbf{T}_{i2}$, we can then fathom $\mathbf{X}$ into two disconnected boxes, which are computed by replacing $\mathbf{T}$ with the respective intervals, namely,

$$\mathbf{X}_{i1}^n = \mathbf{X}_i \cap (\mathbf{T}_{i1} + y_i), \tag{11}$$

$$\mathbf{X}_{i2}^n = \mathbf{X}_i \cap (\mathbf{T}_{i2} + y_i), \tag{12}$$

The procedure to eliminate certainly infeasible points in this section is repeated for $i = 1, 2, ..., n$ and supposed to result in m subboxes. Choose one of these boxes of reduced size, say the one with longest side. Replace X with this new subbox and then repeat the procedure to further eliminate certainly infeasible points.

B. Proposed Algorithm

The proposed algorithm uses Hansen's method explained above to reject some of the certainly infeasible boxes. In proposed algorithm Hansen's inequality test is applied before bisection. Since the Hansen's method is described for the inequality $g(x) \leq 0$, we have to pose the given inequalities accordingly. This method discard large portion of the box and box get shrunk. Thus reduces the search region

Algorithm

Input:Initial box $\mathbf{X}^0$, natural inclusion function F, accuracy parameter ε_x

Outputs: A list K_{in} of all boxes guaranteed to belong to S, and list $K_{out} = K_{in} \cup K_\epsilon$.

BEGIN Algorithm

1. Initialize $k = 0, \mathbf{X} = \mathbf{X}^0, K_{in} = \{\}, K_{out} = \{\}$ list $L = \{\mathbf{X}\}$, set to be inverted, $Y =]-\infty, 0[$ or $]-\infty, 0]$.

2. Remove the first box $\mathbf{X}$ from list L and evaluate $F_i(\mathbf{X})$.

3. If $sup(F_i(\mathbf{X}_k)) < 0$ for all $i = 1, ..., m$ then $K_{in} = K_{in} \cup X_k$ and $K_{out} = K_{out} \cup X_k$, go to step 9.

4. If $inf(F_i(\mathbf{X}_k)) \geq 0$ for any $i = 1, ..., m$ then go to step 9.

5. If $w(\mathbf{X}_k) \leq \varepsilon_x$, then deposit $\mathbf{X}_k$ in the list $K_{out} = K_{out} \cup \mathbf{X}_k$ and go to step 9.

6. Call Hansen inequality test for the boxes where $w(\mathbf{X}_k) > \varepsilon_x$.

7. If the box resulted from Hansen's inequality test is empty go to step 9.

8. Bisect $\mathbf{X}_k$ in maximum width coordinate direction, getting boxes $\mathbf{V}^1$, $\mathbf{V}^2$ such that $\mathbf{X}_k = \mathbf{V}^1 \cup \mathbf{V}^2$. Deposit these sub boxes in list L.

9. If the list L is empty, EXIT algorithm. Else go to step 2.

END Algorithm

IV. Result and Discussion

We considered a test example for comparing the performance of the existing and proposed algorithm. Both algorithms were programmed and tested with fort Fortran95 interval compiler of Sun Microsystems [10]. Boxes generated by programme were plotted with the help of MATLAB6.5 [9].

To compare the performance of proposed and existing algorithm we chose following performance metric:

- Number of iteration
- Maximum list length
- Number of indeterminate boxes, K_ε
- Number of solution boxes, K_{in}

Example 1. [1] Consider a characteristic polynomial $P(s) = s^3 + \sin(p_1p_2)s^2 + p_1^2 s + p_1p_2$, for its asymptotic stability, Routh table translate it into

$$\begin{aligned} (p_1p_2) &> 0, \\ (\sin(p_1p_2)) &> 0, \\ p_1^2 \sin(p_1p_2) - p_1p_2 &> 0. \end{aligned}$$

with search domain $[-3, 3] \times [-3, 3]$ and $\varepsilon_x = 0.05$. Characterizing set P of all p_1, p_2 is set inversion problem as $P = f^{-1}(]0, \infty[)^3$.

Example 2. [1]Consider polynomial

$$P(s) = s^4 + (2z + \frac{1}{T})s^3 + (\frac{2z-6}{T} + 1)s^2 - \frac{1}{T}s - \frac{1}{T}$$

for its asymptotic stability, Routh table translate it into

$$-\frac{1}{T} > 0,$$

$$(2z + \frac{1}{T}) > 0,$$

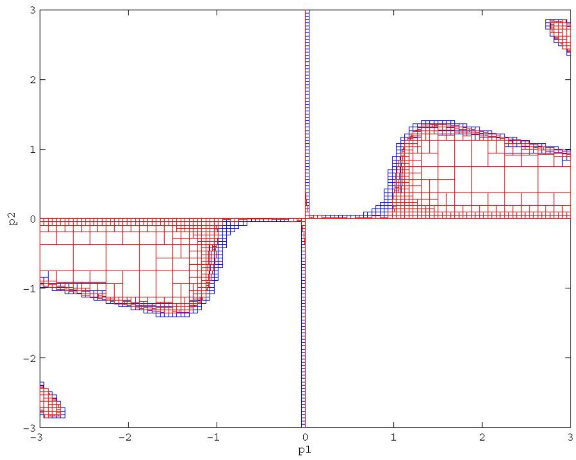

Fig. 1. Total solution,K_{out}- red with proposed algorithm and blue with existing algorithm for example 1.

TABLE I
PERFORMANCE COMPARISON OF ALGORITHMS WITH ACCURACY PARAMETER $\varepsilon_x = 0.05$ FOR EXAMPLE 1.

Performance metric	Existing Algorithm [1]	Proposed Algorithm
Total solution boxes K_{out}	1088	1004
Indeterminant boxes K_ε	732	634
Iteration	3151	2167
Max. list length	1053	472

$$(2z + \frac{1}{T})(\frac{2z-6}{T} + 1) + \frac{1}{T} > 0,$$

$$(-\frac{1}{T})(2z + \frac{1}{T})(\frac{2z-6}{T} + 1) - \frac{1}{T^2} + \frac{1}{T}(2z + \frac{1}{T})^2 > 0.$$

with search domain $[-10, 10] \times [-10, 10]$ *and* $\varepsilon_x = 0.05$. Characterizing set of all (T, z) is set inversion problem as $(T, z) = f^{-1}(\,]0, \infty[^4)$.

Example 3. [4] Let X be the set of all x's in R^2 that satisfy

$$e^{x_1} + e^{x_2} \in [10, 11]$$
$$e^{2x_1} + e^{2x_2} \in [62, 72]$$

TABLE II
PERFORMANCE COMPARISON OF ALGORITHMS WITH ACCURACY PARAMETER $\varepsilon_x = 0.05$ FOR EXAMPLE 2.

Performance metric	Existing Algorithm [1]	Proposed Algorithm
Total solution boxes K_{out}	3903	3521
Indeterminant boxes K_ε	2583	2192
Iteration	3589	2111
Max. list length	11625	9243

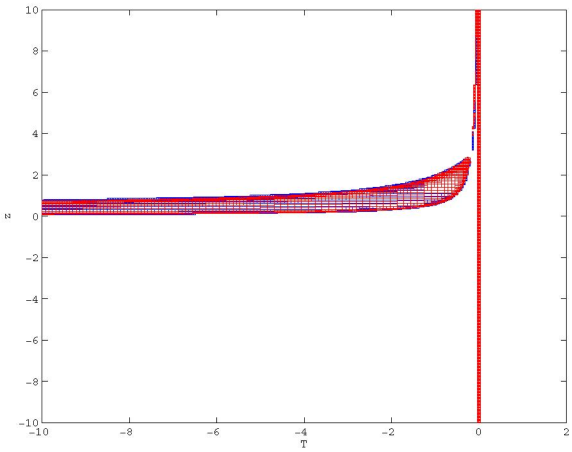

Fig. 2. Total solution,K_{out}- red with proposed algorithm and blue with existing algorithm for example 2.

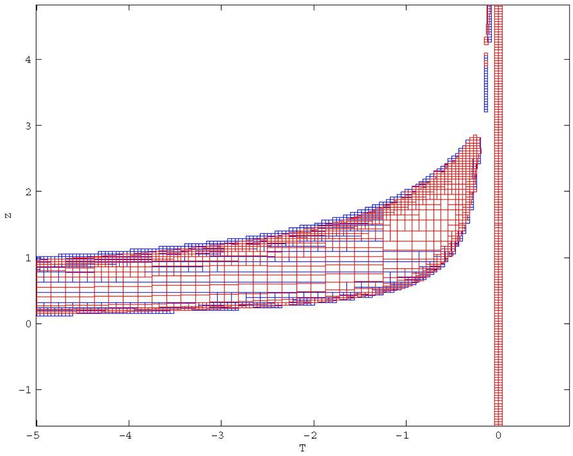

Fig. 3. Zoomed total solution,K_{out}- red with proposed algorithm and blue with existing algorithm for example 2.

characterizing X is set inversion problem as $X = f^{-1}([10, 11] \times [62, 72])$ for $X = [0, 3] \times [0, 3]$ and $\varepsilon_x = 0.01$.

Overall summary of the performance of the proposed algorithm is as follows:
1. No portion of feasible region is lost i.e. entire feasible region is captured.
2. Larger portion of certainly infeasible region is rejected.
3. Hence sharper solution set is obtained.
4. Number of iteration is 34% less.
5. Maximum list length is 51 % less.

V. Conclusion

The proposed algorithm uses Hansen's inequality test to reject some of the certainly infeasible boxes. It was demonstrated through a example that the proposed algorithm is computationally efficient and gives sharper solution than existing algorithm in less than a second. Moreover, it is noteworthy that the proposed algorithm saves memory by reducing maximum list length and it also reduces number of function calls by reducing number of iterations.

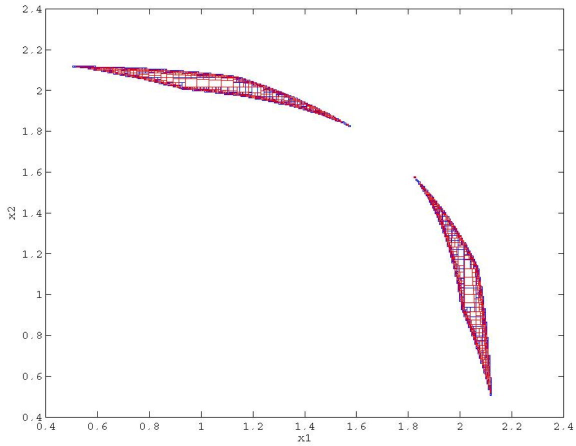

Fig. 4. Total solution, K_{out}- red with proposed algorithm and blue with existing algorithm example 3.

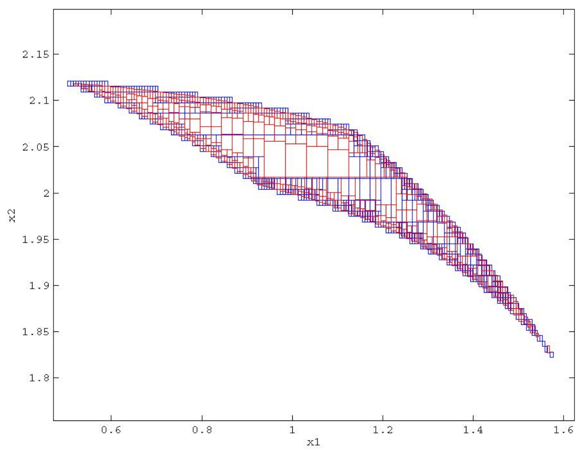

Fig. 5. Zoomed total solution, K_{out}- red with proposed algorithm and blue with existing algorithm example 3.

TABLE III

PERFORMANCE COMPARISON OF ALGORITHMS WITH ACCURACY PARAMETER $\varepsilon_x = 0.01$ FOR EXAMPLE 3.

Performance metric	Existing Algorithm [1]	Proposed Algorithm
Total solution boxes K_{out}	1343	1109
Indeterminant boxes K_ε	874	688
Iteration	4345	2253
Max. list length	1313	438

References

[1] E. Walter and L. Jaulin, "Guaranteed characterization of stability domain via set inversion," *IEEE Transaction on Automatic Control*, vol. 39, No. 4, pp. 123–137, 1993.

[2] L. Jaulin and E. Walter, " Set Inversion via Interval Analysis for Nonlinear Bounbed- Error Estimination," *Automatica*, vol. 29,No. 4, pp. 1053–1064, 1993.

[3] L. Jaulin and E. Walter, "Guaranteed Tuning, with Application to Robust Control and Motion Planning," *Automatica*, vol. 32, No. 8, pp. 1217–1221, 1996.

[4] L. Jaulin, M. Kieffer, O. Didrit and E. Walter *Applied Interval Analysis,* Springer, 2002.

[5] E. R. Hansen *Global Optimization using Interval Analysis,* New York: Marcel Dekkar, 1992.

[6] H. Ratschek, J. Rokne *New Computer Methods for Global Optimization,* Ellis Horwood, Chichester, 1988.

[7] J. E. Dennis, K. Williamson *A Trust-region approach to nonlinear systems of equalities and enequalities,* SIAM J. Optim. 9 (1999) 291-315.

[8] R. E. Moore *Methods and Applications of Interval Analysis,* Philadelphia, PA: SIAM, 1979.

[9] MathWorks Inc., *MATLAB user guide, version 6.5,* MA, USA, 2004.

[10] Sun Microsystems, *Forte Fortran 95 user manual,* Palo Alto, CA, USA, 2001.