Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Extrapolation based interval approach to compute the frequency responses of non-rational transfer functions with nonlinear parametric dependencies

Nataraj S. V. Paluri and Shanta Sondur

Abstract— **We consider the problem of computing the frequency response of a non-rational transfer function having uncertainties in the parameters. A complete characterization of the frequency plots of such a transfer function is provided and an efficient and guaranteed algorithm for computing the envelope of the Bode plot using *interval analysis* (IA) is devised. In particular, it is shown that the range enclosure property of the *interval analysis* can be used to compute the Bode envelopes of the non-rational transfer function with parametric dependencies. The range enclosures computed using IA are guaranteed to contain the minima and maxima of the function over a given box. The IA is combined with uniform subdivision technique to obtain the range enclosures of desired accuracy. We also show, how the well known Richardson extrapolation technique helps in accelerating the convergence process of the algorithm.**

Keywords: Bode plot, Extrapolation, Frequency response, Interval analysis, Linear systems, Robust control, Uncertain systems

I. Introduction

The frequency domain analysis of systems is an important topic in control theory. Powerful graphical tools exist in classical control, such as the Nyquist plot, Bode plots and Nichols chart. These have been widely used to evaluate the frequency domain behavior of system and also for designing lead-lag type controllers through classical-like robust feedback synthesis approaches, see, for example, [1]–[3].

In this paper, we address the problem of computation of the envelope of Bode frequency response of a non-rational transfer function with non-linear parametric uncertainties varying over a box.

Consider a system transfer function $\mathrm{g}(s,x)$, where s is the Laplace variable and $x \in \mathbb{R}^n$ is a vector of system parameters. The magnitude and phase angle functions for $\mathrm{g}(s,x)$ are denoted as

$$f_M(x) = |\mathrm{g}(s = j\omega, x)|\,;\; f_P(x) = \arg\{\mathrm{g}(s = j\omega, x)\}$$

where, ω is the frequency. Suppose there is uncertainty in the system parameters, so that x assumes values in a given box $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ with components $\mathbf{x}_j = \left[\underline{x}_j, \overline{x}_j\right]$. Assume that $f_M(x)$ and $f_P(x)$ are bounded on $\mathbf{x}$. Let the range of $f_M(x)$ and $f_P(x)$ over $\mathbf{x}$ be denoted as range $(f_M(x), \mathbf{x})$ and range $(f_P(x), \mathbf{x})$. Then, the Bode magnitude and phase envelopes M^* and P^* are defined as

$$M^* = \left[\underline{M}^*, \overline{M}^*\right]; P^* = \left[\underline{P}^*, \overline{P}^*\right],$$

$$\underline{M}^* = \inf\{\text{range}(f_M(x), \mathbf{x})\}, \overline{M}^* = \sup\{\text{range}(f_M(x), \mathbf{x})\}$$

$$\underline{P}^* = \inf\{\text{range}(f_P(x), \mathbf{x})\},\ \overline{P}^* = \sup\{\text{range}(f_P(x), \mathbf{x})\}$$

There have been several studies [4]–[10] on the computation of the frequency response of control systems under different types of parametric dependencies such as affine, multi-linear, polynomial, etc. However, there is a lack of tools in the literature to construct the Bode envelopes for the general non-linear type of parametric dependencies. Following [11], [12], an interval analysis approach based on the so-called natural interval extensions (NIE) (of the magnitude and phase functions) and a subdivision strategy can, however, be used to generate the Bode envelopes of non-rational transfer functions having non-linear parametric dependencies. Any overestimation can be tightened to the desired accuracy by successively partitioning the domain using suitable subdivision factors. In spite of this good feature, the method often converges too slowly, as the convergence is *linear* in the domain widths. Thus, it often requires excessive computational effort to produce Bode envelopes of desired accuracy.

Here, we propose an algorithm to compute the Bode envelopes of non-rational transfer functions by combining the NIE with uniform subdivision and *extrapolation.* When an extrapolation procedure is combined with the natural interval extension and subdivision, the speed of convergence can be greatly accelerated. Consequently, we can compute Bode envelopes of the desired accuracy more quickly and with much less effort.

Briefly, the proposed method to construct the magnitude envelope at a given ω is as follows. Using natural interval extension of the magnitude function and geometrically increasing subdivision factors N, we construct two separate sequences: one sequence is that of the lower bounds that converge to the infimum of the envelope, and the other sequence is that of the upper bounds that converge to the supremum of the envelope. These sequences are extrapolated to their respective limits, and *intervals* enclosing the envelope are constructed using Brezinski's error control criterion [13].

Nataraj S. V. Paluri is with Faculty of Systems and Control Engineering Group, Room No. 101, ACRE Building, Indian Institute of Technology, Bombay, Mumbai 400 0076, INDIA nataraj@sc.iitb.ac.in

Shanta Sondur is a Research scholar of Systems and Control Engineering Group, Room No. 101, ACRE Building, Indian Institute of Technology, Bombay, Mumbai 400 0076, INDIA shanta@sc.iitb.ac.in

The idea applies in an analogous fashion to the phase envelope construction.

The outline of the paper is as follows. In section II, we discuss the basics of interval analysis. In section III, we review the extrapolation techniques and error control. In section IV, we present the proposed algorithm. In section V, we demonstrate the effectiveness of the proposed method on a vehicle clutch system example. Finally, in section VI we draw the conclusions of the work.

II. Interval Analysis Preliminaries

This section collects some preliminaries of interval analysis [11]. A real interval is a closed and bounded set of real numbers $\mathbf{x} = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \,|\, \underline{x} \leq x \leq \overline{x}\}$ where $\underline{x}$ and $\overline{x}$ are the lower and upper endpoints of the interval $\mathbf{x}$. The set of all real intervals is denoted by $\mathbb{IR}$. Two intervals are equal if their corresponding endpoints are equal. The intersection of two intervals $\mathbf{x}$ and $\mathbf{y}$ is empty, $\mathbf{x} \cap \mathbf{y} = \Phi$, if either $\underline{x} > \overline{y}$ or $\underline{y} > \overline{x}$. Else, the intersection of $\mathbf{x}$ and $\mathbf{y}$ is again an interval $\mathbf{x} \cap \mathbf{y} = \left[\max\left(\underline{x}, \underline{y}\right), \min\left(\overline{x}, \overline{y}\right)\right]$. The set inclusion: $\mathbf{x} \subseteq \mathbf{y}$ if and only if $\underline{y} \leq \underline{x}$ and $\overline{x} \leq \overline{y}$. Further, the width of an interval is defined as wid $\mathbf{x} = \overline{x} - \underline{x} \geq 0$, and its midpoint is mid $\mathbf{x} = \frac{1}{2}\left(\underline{x} + \overline{x}\right)$.

An interval vector is a vector whose elements are intervals and is denoted by $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ with components $\mathbf{x}_k = [\underline{x}_k, \overline{x}_k]$. The set of all l-dimensional real interval vectors is denoted by $\mathbb{IR}^l$. The relations $\in, =$, and $\subseteq$, union, intersection, and mean are all defined component wise. 'Box' is synonym for an interval vector.

Definition 1: (Natural interval extension) A natural interval extension of f on the box $\mathbf{x}$ is obtained by replacing, in the expression for f, all occurrences of real x_i with intervals $\mathbf{x}_i$ and all real operations with the corresponding interval operations. The natural interval extension of f on $\mathbf{x}$ is written as $f(\mathbf{x})$. By the inclusion property of interval arithmetic, range $(f, \mathbf{x}) \subseteq f(\mathbf{x})$.

We define uniform subdivision of the interval vector $\mathbf{x}$ using the subdivision factor N, as:

$$\begin{aligned} \mathbf{x}_{i,j} &= \left[\underline{\mathbf{x}}_i + (j-1) \text{ wid } \mathbf{x}_i/N,\ \underline{\mathbf{x}}_i + j\text{wid } \mathbf{x}_i/N\right], \\ & \quad j = 1, 2, \ldots, N \\ \mathbf{x}_i &= \bigcup_{j=1}^{N} \mathbf{x}_{i,j}, \\ \mathbf{x} &= \bigcup_{j_i=1}^{N} \left(\mathbf{x}_{1,j_1},\ \mathbf{x}_{2,j_2}, \ldots, \mathbf{x}_{l,j_l}\right) \end{aligned}$$

Let wid $E(\mathbf{y}) :=$ wid $f(\mathbf{y})-$wid range$(f, \mathbf{y})$ denote the excess width (or overestimation) of $f(\mathbf{y})$ for $\mathbf{y} \subseteq \mathbf{x}$. Then,

Theorem 2: wid $E(\mathbf{y}) \leq L$ wid $\mathbf{y}$, for all boxes $\mathbf{y} \subseteq \mathbf{x}$, where L is a constant that depends only on f.

Proof: See [11]. ■

The above theorem says that the overestimation of the natural interval extension goes down at least *linearly* in the domain width. That is, the natural interval extension has a linear or first order convergence property. This is the major drawback of using the natural interval extension with subdivision technique - it requires a large number of subdivisions and time to produce results of high accuracy, see also [12].

One way to overcome these drawbacks is to accelerate the convergence process of the sequences generated by the above subdivision method. Extrapolation methods (equivalently, convergence acceleration methods or sequence transformations) are popularly used for accelerating the convergence process of sequences [14]. Extrapolation methods basically transform the original sequence into another one which converges to the limit more quickly. Therefore, we combine the above method with extrapolation to overcome the aforesaid limitations. A brief description of the extrapolation process is given next.

III. Extrapolation Process and Error Control

Amongst the various extrapolation methods [15], perhaps the most popular and widely used method is the Richardson extrapolation process. Let $K \in \mathbb{N}$, $\rho \geq 2$, and $\{S_n\}$ be the sequence to be accelerated. The Richardson extrapolation process can be given as

$$T_0^{(j)} = S_j,\ j = 0, 1, \ldots, K$$

$$T_k^{(j)} = T_{k-1}^{(j)} + \frac{\left(T_{k-1}^{(j)} - T_{k-1}^{(j-1)}\right)}{\left(\rho^k - 1\right)}, \begin{cases} k = 1, 2, \ldots, K, \\ j = k, \ldots, K. \end{cases}$$

The sequences $\left\{T_k^{(j)}\right\}$ computed above can be arranged in a two-dimensional array called the Romberg Table, denoted $[T]^k$. The k^{th} column of the Romberg Table is referred to as the $(k-1)^{th}$ *extrapolated* column. Details of the Richardson extrapolation process are in [15].

The sequences $\left\{T_k^{(j)}\right\}$ obtained through the Richardson extrapolation process can converge to the true limit, from either side of the limit. This is the typical property observed in the extrapolation process. The extrapolation process does not give any guarantee that the extrapolation on the lower bound sequence will again result in a lower bound on the infimum of the magnitude envelope (the same also holds true for the upper bound sequence). Hence, it is necessary to have error estimate for the entries in the Romberg Table. Among the many error estimation methods [14]–[16], we adopt the error control criterion proposed in [14] to generate intervals that enclose the *true* envelope.

A. Brezinski's Error Control criterion

Brezinski's theorem on error control [13] explains how to construct a sequence of intervals containing the unknown limit of the sequence under consideration. Let $\{S_n\}$ be the sequence under consideration. Let S be the limit of the sequence $\{S_n\}$. Let $\{T_n\}$ and $\{V_n\}$ be two other sequences obtained by applying the Richardson extrapolation process to $\{S_n\}$. Suppose the sequence $\{T_n\}$ converges faster than $\{S_n\}$, and $\{V_n\}$ converges faster than $\{T_n\}$, both to the same limit S. Thus, $\{S_n\}, \{T_n\}$, and $\{V_n\}$ can be successive columns of the Romberg Table.

Let $b \in R$, called as the Brezinski's factor. Define $V_n(b) = V_n - b(V_n - T_n), n \in N$ and construct the *interval*

$$J_n(b) = \begin{bmatrix} \min\left(V_n(b),\ V_n(-b)\right), \\ \max\left(V_n(b),\ V_n(-b)\right) \end{bmatrix}$$

Theorem 3: If $T_n - S = o(S_n - S)$ and if $V_n - S = o(T_n - S)$ then $\forall b \neq 0, \exists N : \forall n \geq N,\ S \in J_n(b)$. Moreover $V_n(\pm b) - S = o(S_n - S)$.

Proof: See [13]. ■

Remark 3.1: Brezinski has pointed out a fundamental practical point [13]: "Under some assumptions, the theorem given above says that for all n greater than N, S belongs to some interval. However, such a N is not known without adding supplementary assumptions. Such an N has been attained if the interval at the step $n+1$ is contained in the interval obtained at the step n, whatever $n \geq N$ may be. This is a good test for having attained this N".

IV. Proposed Algorithm for Bode Envelopes

In the proposed method, at any given ω we obtain intervals enclosing the magnitude (or phase) envelope using the natural interval extension and uniform subdivision, for a sequence of geometrically increasing subdivision factors N. Then, we construct two separate sequences from the lower and upper endpoints of the obtained intervals, and using the Richardson extrapolation process, extrapolate the sequences to their respective limits (which are the infimum and supremum of the magnitude envelope). This produces the Romberg tables for the infimum and supremum of the magnitude envelopes. Next, we apply Brezinski's error control criterion to the Romberg tables, and generate the so-called Brezinski's tables of intervals. Finally, from the Brezinski's tables, we construct a table of intervals that approximate the magnitude (or phase) envelope at ω. The sequences of intervals in this table converge (column-wise) increasingly faster to the envelope than those generated using only the natural interval extension and subdivision.

A. Sequence Generation

Using the natural interval extension and uniform subdivision method, we first construct two separate sequences of lower and upper bounds on the infimum and supremum of the Bode envelope. Then, we obtain the Romberg tables for the infimum and supremum by extrapolating these sequences separately to their respective limits. We denote the sequences of lower and upper bounds as $\left\{A_0^{(j)}\right\}_{j=0}^{K}$, $\left\{B_0^{(j)}\right\}_{j=0}^{K}$.

B. Romberg Tables

Having constructed the sequences of lower and upper bounds, we apply the Richardson extrapolation process and obtain the respective Romberg tables by executing Algorithms Romberg_inf and Romberg_sup.

ALGORITHM FOR ROMBERG TABLE OF THE INFIMUM:

$[A]^K = \text{Romberg\_inf}\left(\left\{A_0^{(j)}\right\}_{j=0}^{K},\ K\right)$

Inputs: The sequence of lower bounds $\left\{A_0^{(j)}\right\}_{j=0}^{K}$, and the number K of columns in the Romberg Table.

Output: The Romberg Table $[A]^K$ containing the extrapolated sequences.

```
BEGIN Algorithm
  1) Set $T_0^{(j)} = A_0^{(j)}, j = 0, 1, \ldots, K$.
  2) Construct the Romberg Table for the range infimum,
     using the Richardson extrapolation process in (III):
     $A_k^{(j)} = A_{k-1}^{(j)} + \frac{A_{k-1}^{(j)} - A_{k-1}^{(j-1)}}{(2^k - 1)}, \left\{ \begin{array}{l} k = 1, 2, \ldots, K, \\ j = k, \ldots, K. \end{array} \right.$
  3) RETURN the Romberg Table
     $[A]^K = \left\{A_k^{(j)}, k = 0, 1, \ldots, K, j = k, \ldots, K\right\}$
END Algorithm
```

We can have a similar algorithm *Romberg_sup* based on $\left\{B_0^{(j)}\right\}_{j=0}^{K}$ to generate the Romberg Table $[B]^K$ of extrapolated sequences for the supremum (the description is omitted here).

C. Brezinski's error control

Based on Theorem 3 and Remark 3.1, we can have an algorithm to construct the so-called Brezinski's tables of intervals for the infimum and supremum of the magnitude envelope. From the Brezinski's tables, we construct a final table of intervals approximating the magnitude envelope - the entries of the table are intervals converging to the magnitude envelope, with increasingly higher orders of convergence.

ALGORITHM FOR APPROXIMATION OF INFIMUM AND SUPREMUM

$\left[[C_{\inf}]^K, [D_{\sup}]^K\right] = \text{App\_env\_infsup}\left([A]^K \text{ and } [B]^K\right)$

Input: The Romberg Tables $[A]^K$ and $[B]^K$, and a value for Brezinski's factor $b \in \mathbb{R}$.

Output: The Tables $\left[[C_{\inf}]^K, [D_{\sup}]^K\right]$ containing the approximating intervals for infimum and supremum of the magnitude envelope.

```
BEGIN Algorithm
  1) Set $k = 0$.
  2) From the Romberg Table $[A]^K$, construct Brezinski's
     table $[C]^K$ of intervals for the infimum as follows:
     $V_{k+2}^{(j)}(b) = A_{k+2}^{(j)} - b\left(A_{k+2}^{(j)} - A_{k+1}^{(j)}\right),$
     $\left\{ \begin{array}{l} k = 0, 2, \ldots, K-2, \\ j = k+2, \ldots, K. \end{array} \right.$
     $C_{k+2}^{(j)} = \begin{bmatrix} \min\left(V_{k+2}^{(j)}(+b), V_{k+2}^{(j)}(-b)\right), \\ \max\left(V_{k+2}^{(j)}(+b), V_{k+2}^{(j)}(-b)\right) \end{bmatrix},$
     $\{\ k = 0, 2, \ldots, K-2,\ j = k+2, \ldots, K.\ \}.$
```

```
3) Similarly, from the Romberg Table $[B]^K$, construct Brezinski's Table $[D]^K$ of intervals for supremums.
4) Check for nestedness of the intervals in Table $[C]^K$ and form another table $[C_{\inf}]^K$ with these nested interval. Do likewise for the intervals in $[D]^K$ and with nested interval form the Table $[D_{\sup}]^K$. (note: nestedness is checked column-wise, for consecutive intervals in each column).
5) RETURN $\left[[C_{\inf}]^K, [D_{\sup}]^K\right]$.
```

END Algorithm

D. MAIN ALGORITHM

Based on the above developments, we can now propose an algorithm to compute the magnitude envelope at a given ω to a specified accuracy. The algorithm for the phase envelope is completely analogous and so we omit its description here.

ALGORITHM FOR MAGNITUDE ENVELOPE GENERATION **:**

$$[Env\_approx]^K = \text{Mag\_env}[\text{g}(s,x), \mathbf{x}, \omega, \varepsilon]$$

Inputs: An expression for the transfer function $\text{g}(s,x)$, the parameter box $\mathbf{x}$, the frequency ω, and an accuracy parameter ε.

Outputs: The Table $[Env\_approx]^K$ containing the intervals approximating the magnitude envelope. The interval at the bottom-right of this table encloses the envelope with the desired accuracy.

BEGIN Algorithm

```
1) Set $j \leftarrow 0$ and initialize list $\mathcal{L} \leftarrow \{\mathbf{x}\}$
2) Set $A_0^{(j)} = \underline{f_M(\mathbf{x})}, B_0^{(j)} = \overline{f_M(\mathbf{x})}$
3) (**Iterative part**) Start a new iteration:
   a) Subdivision phase: Pick up the boxes from the list, bisect them in all direction to get subboxes and put them back into the same list $\mathcal{L}$.
   b) Set $j \leftarrow j+1$ and $l_r \leftarrow$ length of $\mathcal{L}$.
   c) Set $N_j \leftarrow 2^j$.
   d) Pick all the boxes $\mathbf{x}_i, i = 1, 2, \ldots, l_r$ from $\mathcal{L}$, and obtain the range enclosure $f_{M_{(N_j)}}(\mathbf{x})$
      $$\begin{aligned} f_{M_{(N_j)}}(\mathbf{x}) &= \left[\underline{f_{M_{(N_j)}}(\mathbf{x})}, \overline{f_{M_{(N_j)}}(\mathbf{x})}\right] \\ &= \bigcup_{i=1}^{l_r} f_{M_{(i)}}(\mathbf{x}_i) \end{aligned}$$
   e) Set
      $A_0^{(j)} \longleftarrow \underline{f_{M_{(N_j)}}(\mathbf{x})}, B_0^{(j)} \longleftarrow \overline{f_{M_{(N_j)}}(\mathbf{x})}$
   f) Set $K \leftarrow j$
   g) (Generate Romberg table for lower bound of the magnitude envelope ). Execute the algorithm
      *Romberg_inf* $[A]^K =$
      $Romberg\_\inf\left(\left\{A_0^{(j)}\right\}_{j=0}^K, K\right)$
   h) (Generate Romberg table for upper bound of the magnitude envelope ). Execute the algorithm
      *Romberg_sup* $[B]^K =$
      $Romberg\_\sup\left(\left\{B_0^{(j)}\right\}_{j=0}^K, K\right)$
   i) (Compute Brezinski's Tables of approximate intervals for the infimum and supremum of the magnitude envelope). Execute the algorithm
      App_env_infsup $\left[[C_{\inf}]^K, [D_{\sup}]^K\right] =$
      $App\_env\_\inf\sup\left([A]^K \text{ and } [B]^K\right)$
   j) (Termination criterion: Check the width of the approximating intervals)
      IF $\left(\left(wid\,[C_{\inf}]^K \text{ and } wid\,[D_{\sup}]^K\right) < \varepsilon\right)$
      THEN GO TO step 4.
   k) Cutoff test:
      i) Set $c_{\max} = \min\left\{\overline{f_{M_{(i)}}(\mathbf{x}_i)}\right\}$, and $c_{\min} = \max\left\{\underline{f_{M_{(i)}}(\mathbf{x}_i)}\right\}$, $i = 1, \ldots, l_r$.
      ii) IF $\underline{f_{M_{(i)}}(\mathbf{x}_i)} > c_{\min}$ and $\overline{f_{M_{(i)}}(\mathbf{x}_i)} < c_{\max}$ THEN discard the box $\mathbf{x}_i$ from the list $\mathcal{L}$ from further processing ELSE put them back into the list $\mathcal{L}$.
4) Construct intervals whose lower and upper end-points are the corresponding entries in the tables $[C_{\inf}]^K, [D_{\sup}]^K$, satisfying the desired accuracy (cf. step no. 3j).
5) Lastly, construct a Romberg-like Table of intervals $[Env\_approx]^K$ based on the intervals of the above step. These intervals approximate the magnitude envelope at $\omega$ with increasing accuracy as we go column-wise in the table. The interval at the bottom-right of this table encloses the envelope with the desired accuracy.
```

END Algorithm

V. ILLUSTRATIVE EXAMPLE - VEHICLE CLUTCH SYSTEM

The automatic control of vehicle start-up using a friction clutch is a subject of continuing interest for vehicle manufacturers. An effective automatic control system can be designed based on the Bode frequency response plots of the system.

The proposed algorithm for Bode envelopes is demonstrated on the vehicle clutch system with large parametric uncertainties [17]. We shall demonstrate the algorithm to construct only the Bode magnitude envelope (the Bode phase envelope can be constructed in an analogous fashion, and so the description is omitted here). The magnitude function relating the input clutch position to the output transmission speed is

$$f_M(x) = 10\log\left\{\frac{\left(65.61\left(x_1 - x_2\omega^2\right)\right)^2 + (24734.97\omega)^2}{(-1157.39\omega)^2 + \left(3.07\left(x_1 - \frac{\omega^2}{\frac{1}{65.61} + \frac{1}{x_2}}\right)\right)^2}\right\}$$
$$+ 20\log\left(\frac{x_3}{(65.61 + x_2)\omega}\right)$$

The physical parameters are the axle shaft spring rate (x_1), vehicle inertia (x_2), and the ratio between change in clutch torque and change in clutch position (x_3). These have the following large parameter uncertainties

$$x_1 \in [58000, 115000]\,,\ x_2 \in [1400, 11000]\,,\ x_3 \in [100, 800]\,.$$

Suppose the Bode magnitude envelopes of this system are required to be constructed to an accuracy $\varepsilon = 10^{-10}$ at the frequencies $\omega \in \{1, 10, 30, 50, 60, 70, 80, 90, 100, 200, 300\}$

Executing the proposed algorithm at the various frequencies, we obtain the magnitude envelopes (the magnitude plot is not shown here due to the lack of space but is available from the author). Let us discuss in detailthe obtained results at some frequency, say for instance, at $\omega = 10$. Table I reports the intervals $[Env\_approx]^K$ generated with the proposed method at this ω. The table also gives the overestimation associated with each interval $[Env\_approx]^K$, computed as

$$\text{wid } [Env\_approx]^K - \text{ wid } M^*.$$

The first column of Table I gives the uniform subdivision factor N, the second column ($k = 0$) shows the intervals obtained with the natural interval extension and the uniform subdivision, the third column ($k = 1$) gives the intervals for the first extrapolated column, and the subsequent columns give the intervals $[Env\_approx]^K$ for the extrapolated columns $k = 2, \ldots, K$. The bracketed entries in Table I give the overestimation associated with each interval $[Env\_approx]^K$.

From Table I, we see that the intervals generated by the extrapolation process are of accuracy that increases columnwise. The intervals generated by the proposed algorithm *enclose* the magnitude envelope, eventually up to the desired high accuracy, thus generating very sharp enclosures of the magnitude envelope.

We also compare the computational time required by the proposed algorithm with that of the interval global optimization routine *Globsol* [18]. The required accuracy is now set as $\varepsilon = 10^{-10}$. Both the methods are executed on a SUN FIRE 280 R machine with dual 750 MHz UltraSparc III processors and 2 GB RAM. While *Globsol* takes a total computational time of 37.26 minutes to compute the Bode magnitude envelopes to this accuracy, the proposed algorithm takes 27.88 minutes for the same.

VI. Conclusions

The proposed algorithm based on extrapolation proves to be an efficient tool for computing the Bode envelopes for uncertain transfer function with general parametric uncertainty. The application of Extrapolation technique also can be applied to interval optimization algorithms, to accelerate the speed of convergence, to the global minima or maxima. This is currently being explored.

TABLE I

Range enclosures for the Bode magnitude envelopes and the corresponding overestimation obtained with the proposed techniques for the vehicle clutch example.

N	k=0
32	[-31.0173, 10.1195] (1.9e+0)
64	[-30.2461, 9.9993] (9.7e-1)
128	[-29.8332, 9.9398] (4.9e-1)
256	[-29.6192, 9.9102] (2.5e-1)
512	[-29.5102, 9.8955] (1.3e-1)
1024	[-29.4551, 9.8881] (6.3e-2)
2048	[-29.4275, 9.8844] (3.2e-2)
	k=1
64	[-30.2461, 9.9993] (9.7e-1)
128	[-29.8332, 9.9398] (4.9e-1)
256	[-29.6192, 9.9102] (2.5e-1)
512	[-29.5102, 9.8955] (1.3e-1)
1024	[-29.4551, 9.8881] (6.3e-2)
2048	[-29.4275, 9.8844] (3.2e-2)
	k=2
128	[-29.42031, 9.881109] (2.1e-2)
256	[-29.40511, 9.880792] (5.5e-3)
512	[-29.40109, 9.880714] (1.4e-3)
1024	[-29.40005, 9.880695] (3.6e-4)
2048	[-29.39979, 9.880691] (8.9e-5)
	k=3
256	[-29.4000428, 9.880690518] (3.4e-4)
512	[-29.3997474, 9.880689714] (4.6e-5)
1024	[-29.3997079, 9.880689615] (5.9e-6)
2048	[-29.3997028, 9.880689602] (7.5e-7)
	k=4
512	[-29.399705228, 9.880689601624] (3.2e-6)
1024	[-29.399702235, 9.880689600477] (2.2e-7)
2048	[-29.399702033, 9.880689600407] (1.4e-8)
	k=5
1024	[-29.3997020354, 9.8806896004038] (1.6e-08)
2048	[-29.3997020198, 9.8806896004030] (5.5e-10)

References

[1] S. P. Bhattacharyya, H. Chapellat, and L. H. Keel, *Robust Control - The Parametric Approach.* Prentice Hall, New York, 1995.

[2] I. M. Horowitz, *Quantitative Feedback Design Theory (QFT).* QFT Publications, Boulder, Colorado, USA, 1993.

[3] N. Tan and D. P. Atherton, "New approach to assessing the effects of parametric variations in feedback loops," *IEE Proc.-Control Theory Appl.*, vol. 150, no. 2, pp. 101–111, March 2003.

[4] A. C. Bartlett, A. Tesi, and A. Vicino, "Frequency response of uncertain systems with interval plants," *IEEE Transactions, Automatic Control*, vol. 38, no. 6, pp. 929–933, 1993.

[5] A. C. Bartlett, "Nyquist, Bode and Nichols plots of uncertain systems," *Proceedings of American Control Conference*, pp. 2033–2036, 1993.

[6] ——, "Computation of frequency response of systems with uncertain parameters: a simplification," *International Journal of Controls*, vol. 57, no. 6, pp. 1293–1309, 1993.

[7] M. Fu, "Computing the frequency response of linear systems with parametric perturbations," *Systems and Control Letters*, vol. 15, pp. 45–52, 1990.

[8] B. R. Barmish, *New Tools for Robustness of Linear Systems.* Macmillan Publishing Co., New York, NY, 1994.

[9] T. E. Djaferis, *Robust control design: a polynomial approach.* Kluwer Academic Publishers, Boston, 1995.

[10] J. Ackermann, *Robust control: systems with uncertain physical parameters.* Springer-Verlag, 1993.

[11] R. E. Moore, *Methods And Applications of Interval Analysis.* SIAM, Philadelphia, 1979.

[12] P. S. V. Nataraj and J. J. Barve, "Reliable computation of frequency response plots for non-rational transfer functions to prescribed accuracy," *Reliable Computing*, vol. 9, pp. 373–389, 2003.

[13] C. Brezenski, "Error control in convergence acceleration processes," *IMA J. Numerical Analysis*, vol. 3, pp. 65–80, 1983.

[14] C. Brezenski and M. R. Zaglia, *Extrapolation Methods: Theory and Practice.* North-Holland, Amsterdam, 2002.

[15] A. Sidi, *Practical Extrapolation Methods*. Cambridge University Press, Cambridge, 2003.

[16] G. Walz, *Asymptotics and Extrapolation*. Akademie Verlag, Berlin, 1996.

[17] W. Chen and D. J. Ballance, “Plant template generation for uncertain plants in QFT,” *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, vol. 121, pp. 359–364, 1999.

[18] R. B. Kearfott, *Rigorous Global Search: Continuous Problems.* Kluwer Academic Publishers, Dordrecht, 1996.