On the Computation of Value Sets of Multi-Liner Functions

Yuzo Ohta
Kobe University
ohta@cs.kobe-u.ac.jp

and

Masaaki Nagahara
Kyoto University
nagahara@i.kyoto-u.ac.jp

Abstract: In this paper, the issue of computing a good estimate of the value set of a characteristic polynomial $f(s, q)$ is addressed, where q denotes a vector of uncertain parameters belonging to a box and f is a multi-linear function of q. If $f(s, p)$ is a totally decomposed expression, then a good estimate of the value set can be computed very fast by using Non-convex Polygon Interval Arithmetic. In this paper, an efficient method to compute good estimate of the value set is proposed when f is not a totally decomposed expression.

I. Introduction

The concept of the value set is very useful in analyzing and designing robust control systems [1] – [5]. Therefore, many efforts have been done to compute the value set or an estimate of it [4], [6] – [10].

Since we confine ourselves to the problem of computing a good estimate of the value set at a given frequency, we consider a function $f(q)$ from $\mathbf{R}^m$ into $\mathbf{C}$, where q is the vector of uncertain parameters and belongs to a given m-dimensional box denoted by Q.

If $f(q)$ is a totally decomposed expression (TDExp), that is, each q_i appears exactly once in $f(q)$ [1], then we can compute good estimate of the value set very fast by using Non-convex Polygon Interval Arithmetic (NPIA) [10] [2]. However, if $f(q)$ is not a TDExp, then the estimate obtained by using NPIA might be much larger than the value set.

Let $x(q)$ and $y(q)$ be real and imaginary parts of $f(q)$ and define the Jacobian matrix $H(q)$ of functions x and y by

$$H(q) = \left[h^1(q) \;\; h^2(q) \;\; \cdots \;\; h^m(q) \right], \tag{1}$$

$$h^k(q) = \begin{bmatrix} \frac{\partial x}{\partial q_k}(q) \\ \frac{\partial y}{\partial q_k}(q) \end{bmatrix} \quad k = 1, 2, \cdots, m. \tag{2}$$

At page 146 of [4], it is said, *"The construction of the value set for m parameters requires again mapping the edges. $\cdots$ the points $[q_1 \; q_2 \cdots q_m]^T$ for which $H(q)$ has rank < 2 have to be determined. $\cdots$ The solution vectors of this system of equations contribute to the boundary of the value set. These vectors are in general not easy to describe. $\cdots$ In most of practical cases gridding the whole Q-box is more effective than solving the system of equations."*

In this paper, under a mild condition, we show that tracing points satisfying $\operatorname{rank} H(q) < 2$ is much more efficient than gridding the whole Q-box. Moreover, when $f(q)$ has sub-functions which are TDExps, we propose a method to reduce computing time.

Notation: In this paper, for integers i and j such that $i \leqq j$, $[i..j]$ denotes the set of integers $\{i, i+1, \cdots, j\}$. $\mathbf{R}$ and $\mathbf{C}$ denote the set of all real and complex numbers, respectively. For a set $V \subseteq \mathbf{C}^m$, ∂V, $\operatorname{int} V$, $\operatorname{ri} V$ and $\operatorname{outbd} V$ denote the boundary of V, the interior of V, the relative interior of V and the closed set surrounded by the outer boundary of V (see Fig. 1), respectively. For a finite set V, $|V|$ denotes the cardinal of V, that is, $|V| = n$ if $V = \{v_i | \; i \in [1..n]\}$. For a set $\{q^i \in \mathbf{C}\}_{i=1}^m$ and a region $V \subseteq \mathbf{C}$, $\operatorname{conv}[q^1, q^2, ..., q^m]$ and $\operatorname{conv}[V]$ denote the convex hull of $\{q^i \in \mathbf{C}\}_{i=1}^m$ and V. For a given m polytope Q, $\mathcal{F}_r(Q)$ denotes the set of all r-faces of Q.

II. Value Set and its Boundary

Let us consider a function $f : Q \subseteq \mathbf{R}^m \to \mathbf{C}$, where $Q = [0, 1]^m$ and f is twice continuously differentiable in a open set including Q. For convenience, we denote real and imaginary parts of $f(q)$ by $x(q)$ and $y(q)$, respectively. The value set of f is defined by

$$f(Q) = \{f(q) = x(q) + \jmath y(q) \mid q \in Q\}. \tag{3}$$

and the issue we consider is to compute $\operatorname{outbd} f(Q)$ [3]. If we compute $\partial f(Q)$, we get $\operatorname{outbd} f(Q)$ by computing $\operatorname{outbd}[\partial f(Q)]$. Therefore, computing $\partial f(Q)$ plays the main role in computing $\operatorname{outbd} f(Q)$.

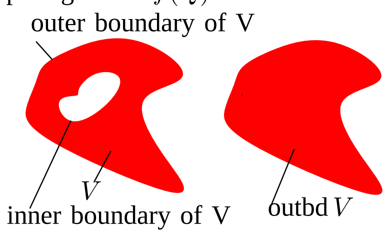


Fig. 1 A set V and $\operatorname{outbd} V$.

[1]TDExp corresponds to the totally decomposable tree structured decomposition [9]. Note that interval polynomials, functions corresponding to GKT [2] and linear functions $f(q)$ of q have TDExp's.

[2]NPIA is the arithmetic defined on the set of all polygons in the complex plane.

[3]We will explain later the reason why we compute $\operatorname{outbd} f(Q)$ not $f(Q)$ itself.

As we have already quoted, it is said that the point q whose image locates at the boundary of the value set is a point in an edge or satisfies $\operatorname{rank} H(q) < 2$, where $H(q)$ is defined in (1) [4].

Lemma 1: For any $F_r^k \in \mathcal{F}_r([0,1]^m)$, we define $J(F_r^k) \subseteq [1..m]$ and $I(F_r^k) = [1..m] \backslash J(F_r^k)$ such that

$$F_r^k = \{q \in \mathbf{R}^m \mid q_i = \sigma_i \in \{0,1\}, \ \ i \in I(F_r^k), \\ q_j \in [0,1], \ \ j \in J(F_r^k)\}. \tag{4}$$

Let $r \geqq 2$ and let $J(F_r^k) = \{j_1, j_2, ..., j_r\}$. For each $q \in F_r^k$, let us define a submatrix $H(q; J(F_r^k))$ of $H(q)$ by

$$H(q; J(F_r^k)) = [h^{j_1}(q) \ \ \cdots \ \ h^{j_r}(q)], \tag{5}$$

Assume that $h^{j_1}(q) \neq 0$. Then the following three conditions are equivalent.

1) $\operatorname{rank} H(q; J(F_r^k)) < 2$

2) $\operatorname{rank} [h^{j_1}(q) \ \ \ h^{j_\ell}(q)] < 2$ for all $\ell \neq 1$, that is,

$$\psi(q; F_r^k) = \begin{bmatrix} \psi_1(q; F_r^k) \\ \vdots \\ \psi_{r-1}(q; F_r^k) \end{bmatrix} = 0, \quad q \in F_r^k \tag{6}$$

where $\psi_\ell(q; F_r^k) = \det[h^{j_1}(q) \ h^{j_{\ell+1}}(q)]$.

3) For each $j_\ell \in J(F_r^k)$, $\ell \neq 1$, there exists a real number $\gamma_\ell(q)$ such that $h^{j_\ell}(q) = \gamma_\ell(q) h^{j_1}(q)$.

Lemma 1 means that, to examine the condition that $\operatorname{rank} H(q;\ J(F_r^k)) < 2$, we need not examine that all 2×2-subdeterminants of $H(q;\ J(F_r^k))$ vanish; it suffices to examine particular $r-1$ subdeterminants. The point $q \in F_r^k$ satisfying $\operatorname{rank}\ H(q;\ J(F_r^k)) < 2$ is the solution of (6).

Theorem 1: Let $Q = [0,1]^m$ and let $F_r^k \in \mathcal{F}_r(Q)$. Define the set $\Omega(F_r^k)$ by

$$\Omega(F_r^k) = \{q \in F_r^k \mid \psi(q; F_r^k) = 0\}. \tag{7}$$

Then, we have

$$\partial f(Q) \ \subseteq \ \left[\bigcup_{F_r^k \in \mathcal{F}_r(Q),\ 2 \leqq r \leqq m} f(\Omega(F_r^k)) \right] \bigcup \left[\bigcup_{F_1^k \in \mathcal{F}_1(Q)} f(F_1^k) \right]. \tag{8}$$

In (8), some of $\Omega(F_r^k)$ may be empty, and in this case we understand that $f(\Omega(F_r^k))$ is empty.

In general, $q \in \Omega(F_r^k)$, that is, $\operatorname{rank} H(q; J(F_r^k)) < 2$ does not imply that $\operatorname{rank} H(q) < 2$ since $J(F_r^k) \subseteq [1..m]$. This is the reason why the statement in [4] is ambiguous.

To illustrate the idea to compute $\{\Omega(F_r^k)\}$ and $\operatorname{outbd} f(Q)$, let us consider the case when $m = 3$. In Fig. 2,

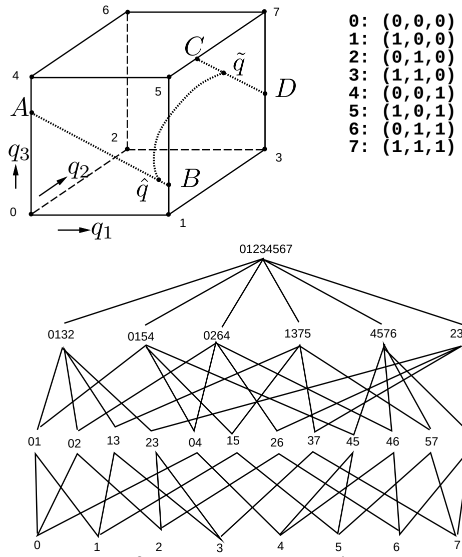


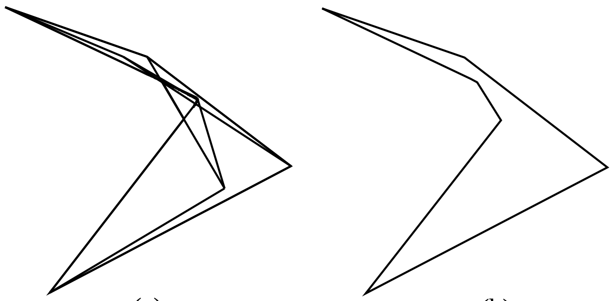
Fig. 2 $AB = \Omega(F_2^2)$ and $CD = \Omega(F_2^4)$ are line segments. $\Omega(F_3^1) = \{\chi_3^{1,1}\}$ is a curve connecting $\hat{q}$ and $\tilde{q}$.

$Q = F_3^1 = [0,1]^3$ and the face graph of Q are shown. At the face graph shown in Fig. 2, Q is represented as 01234567. The face graph gives the information about the direct inclusion relations between faces. Any $F_r^k \in \mathcal{F}_r([0,1]^m)$ has $2r$ subfaces $\mathcal{F}_{r-1}(F_r^k) = \{F_{r-1}^{k_\ell}\}_{\ell=1}^{2r}$, and ∂F_r^k consist of these subfaces. For example, ∂F_3^1 consists of six 2-faces, F_2^1 = 0132, F_2^2 = 0154, $\cdots$, and F_2^6 = 2376. Since f is a multi-linear function of q, $\Omega(F_2^k)$, if it is not empty, is a line segment which is easily obtained. Moreover, according to our experience, we expect that $\Omega(F_r^k)$ consists from curves $\{\chi_r^{k,j}\}$, which has end points at ∂F_r^k. Two $\Omega(F_2^k)$'s are shown: one is the line segment $AB = \Omega(F_2^2)$ and the other is the line segment $CD = \Omega(F_2^4)$. One $\Omega(F_3^1) = \{\chi_3^{1,1}\}$ is shown. The endpoint of $\chi_3^{1,1}$ are $\hat{q} \in \Omega(F_2^2)$ and $\tilde{q} \in \Omega(F_2^4)$. A method to compute $\hat{q}$, $\tilde{q}$ and a method to compute $\chi_3^{1,1}$ from $\hat{q}$ is proposed later.

To get $\operatorname{outbd} f(Q)$, we compute

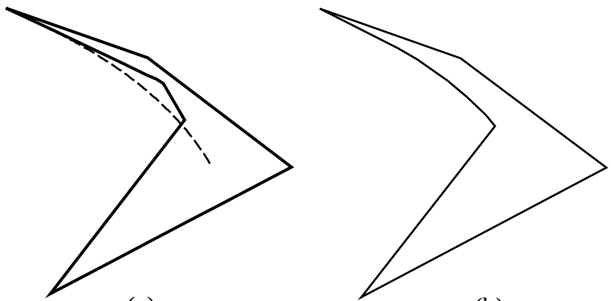
$$\begin{aligned} V_1 &= \operatorname{outbd} \left[\bigcup_{F_1^k} f(F_1^k) \right] \\ V_2 &= \operatorname{outbd} \left[V_1 \bigcup f(\Omega(F_2^1)) \right] \\ V_3 &= \operatorname{outbd} \left[V_2 \bigcup f(\Omega(F_2^2)) \right] \\ \operatorname{outbd} f(Q) &= \operatorname{outbd} \left[V_3 \bigcup f(\Omega(F_3^1)) \right]. \end{aligned}$$

In Fig. 3, an example of computing $V_1 = \operatorname{outbd} [\bigcup_{F_1} f(F_1^k)]$ is shown. Also, in Fig. 4, an example of computing $V_2 = \operatorname{outbd} \left[V_1 \bigcup f(\Omega(F_2^1)) \right]$ is shown. The algorithm to compute V_1 is given in [10] and the algorithm to compute V_2 is also given in [10].



(a) (b)

Fig. 3 (a) $\{f(F_1^k)\}$ and (b) $V_1 = \text{outbd}\,[\bigcup_{F_1} f(F_1^k)]$.



(a) (b)

Fig. 4 (a) V_1 and $f(\Omega(F_2^1))$ are drawn by solid line and dashed line, respectively, and (b) $\text{outbd}\,[V_1 \bigcup f(\Omega(F_2^1))]$.

The outline of the algorithm to compute an estimate of outbd $f(Q)$ is the following procedure GetBoundary, in which we assume $\hat{\Omega}(F_r^k)$ is implemented as a list of curves.

procedure GetBoundary

Step 1. $V := \text{outbd}\,[\{f(F_1^k)\}]$;
Step 2. for $F_2^k \in \mathcal{F}_2(Q)$ do begin
Step 3. compute $\hat{\Omega}(F_2^k)$;
Step 4. $V := \text{outbd}\,[V \cup f(\hat{\Omega}(F_2^k))]$;
end;
Step 5. if there is no $\Omega(F_2^k) \neq \emptyset$ then return V;
Step 6. for $r \in [3..m]$ do begin
Step 7. for $F_r^k \in \mathcal{F}_r(Q)$ do begin
Step 8. $\Omega(F_r^k)$:=GetOmega(F_r^k);
Step 9. $V := \text{outbd}\,[V \cup \{f(\Omega(F_r^k))\}]$;
end;
Step 10. if there is no $\Omega(F_r^k) \neq \emptyset$ then return V;
end;
Step 11. return V;

procedure GetOmega(F_r^k)

Step 1. $\Omega(F_r^k) := \emptyset$;
Step 2. for $F_{r-1}^{k_\ell} \in \mathcal{F}_{r-1}(F_r^k)$ do
append GetCurves$(r-1, k_\ell)$ to $\Omega(F_r^k)$;
Step 3. return $\Omega(F_r^k)$;

procedure GetCurves$(r-1, k_\ell)$

Step 1. $\chi := \emptyset$; $\Gamma := \emptyset$;
Step 2. find all $\hat{q}^j \in \Omega(F_{r-1}^{k_\ell})$ such that $\psi(\hat{q}^j; F_r^k) = 0$ and append it to Γ;
Step 3. for $\hat{q}^j \in \Gamma$ do compute $\chi_r^{k,j}$ and append it to χ;
Step 4. return χ;

At this point we need some remarks.

1) At Step 2 of th GetCurves$(r-1, k_\ell)$, we search a point $\hat{q}^j$ satisfying

$$\psi(\hat{q}^j; F_r^k) = 0, \quad \hat{q} \in \chi_{r-1}^{k_\ell,j}. \tag{9}$$

Since $\chi_{r-1}^{k_\ell,j}$ is given by a list of a finite number of points in real implementation, there might be no $\hat{q}^j \in \chi_{r-1}^{k_\ell,j}$ satisfying (9). To get the first element $\hat{q}^j$ of $\chi_r^{k,j}$, we solve

$$\psi(\hat{q}^j; F_r^k) = 0, \quad \hat{q} \in F_{r-1}^{k_\ell}. \tag{10}$$

To solve (10), we apply Newton method with the initial vector $q^0 \in \chi_{r-1}^{k_\ell,j}$. Since Newton method has the local convergent area and the quadratic convergence property, it is reasonable to suppose that (10) has no solution in a neighborhood of q^0 and quit Newton iteration if iteration number exceed several number or approximate solution goes out from the neighborhood of q^0. Then, we select next $q^0 \in \chi_{r-1}^{k_\ell,j}$ for another initial vector. Therefore, this is the one parameter search, and, hence, it can be done easily.

2) At Step 3 of GetCurves$(r-1, k_\ell)$, we compute $\chi_r^{k,j}$, which is executed by solving

$$\psi(q; F_r^k) = 0, \quad q \in F_r^k, \tag{11}$$

where F_r^k is an r-face such that $F_{r-1}^{k_\ell} \in \mathcal{F}_{r-1}(F_r^k)$.

Note $\psi(q;\, F_r^k) = 0$ consists of $r-1$ equations and has r variables. Therefore, we choose a variable, say $q_{\hat{i}}$, as a parameter and solve (11) for the remaining variables, that is, when the last point, say q^n, in $\chi_r^{k,j}$ is given, we will solve

$$\psi(q^{n+1}; F_r^k) = 0, \quad q \in F_r^k, \quad q_{\hat{i}}^{n+1} = q_{\hat{i}}^n + \Delta, \tag{12}$$

where $\hat{i} \in J(F_r^k)$. When $n = 0$, that is, $q^0 = \hat{q}^j$, then $\hat{i} = J(F_r^k) - J(F_{r-1}^{k_\ell})$. If $\hat{q}_{\hat{i}}^j = 0$, then $\Delta > 0$; if $\hat{q}_{\hat{i}}^j = 1$, then $\Delta < 0$. Needless to say, we must choose Δ so that $0 \leqq q_{\hat{i}}^n + \Delta \leqq 1$. To compute the solution q^{n+1} of (12), we apply Newton method with the initial vector $q^{n+1,0}$ satisfying $q_i^{n+1,0} = q_i^n$ if $i \neq \hat{i}$ and $q_{\hat{i}}^{n+1,0} = q_{\hat{i}}^n + \Delta$. The obtained solution q^{n+1} of (12) is appended to $\chi_r^{k,j}$ as the last point. We continue this process until we reach ∂F_r^k.

Before we reach ∂F_r^k, it might happen that (12) has no solution, which means $|\Delta|$ is too large, and we reset $\Delta :=$

$\Delta/2$ when $|\Delta|$ is not too small. If $|\Delta|$ is very small, we need to change $\hat{i}$. The new $\hat{i}$ is selected so that

$$|q_{\hat{i}}^n - q_{\hat{i}}^{n-1}| \geqq |q_i^n - q_i^{n-1}| \quad \forall\, i \in J(F_r^k),$$

and determine the sign of Δ so that $\Delta(q_{\hat{i}}^n - q_{\hat{i}}^{n-1}) > 0$.

To apply Newton method, it is needed that $(r-1)\times(r-1)$ submatrix of $\frac{\partial\psi}{\partial q}(q; F_r^k)$ is nonsingular. and, hence, if we face the situation that $\frac{\partial\psi}{\partial q}(q; F_r^k)$ is singular, we quit to execute GetBoundaryand we fail to compute an estimate of outbd $f(Q)$.

Moreover, we supposed that $\Omega(F_r^k)$ consists from curves $\{\chi_r^{k,j}\}$, which has end points at ∂F_r^k. But, it is not obvious that this is true or not. The following result gives a partial answer for this issue.

Theorem 2: For each $F_r^k \in \mathcal{F}_r([0,1]^m$, $r \geqq 2$, and for each $q \in \Omega(F_r^k)$, we assume that $h^{j_1}(q) \neq 0$ and that any $(r-1)\times(r-1)$ submatrix of $\frac{\partial\psi}{\partial q}(q; F_r^k)$ is nonsingular, where $\psi(q; F_r^k)$ is given by (6).

Given $r \geqq 2$. If there is a $q^* \in$ ri F_r^k such that $\psi(q^*; F_r^k) = 0$, then there exist a continuous function $\chi_r^{k,1} : [0,1] \to F_r^k$ and a constant $\eta^* \in (0,1)$ such that $q^* = \chi_r^{k,1}(\eta^*)$ and

$$\begin{cases} \chi_r^{k,1}(0), \chi_r^{k,1}(1) \in \partial F_r^k, \\ \chi_r^{k,1}(\eta) \in \text{ri } F_r^k \quad \forall\, \eta \in (0,1) \\ \psi(\chi_r^{k,1}(\eta); F_r^k) = 0 \quad \forall\, \eta \in [0,1] \end{cases} \tag{13}$$

Theorem 2 implies that $\Omega(F_r^k)$ consists of curves $\{\chi_r^{k,j}\}$ and that endpoints of $\chi_r^{k,j}$ are located in ∂F_r^k, and, hence, if conditions of Theorem 2 hold, then the proposed method works effectively.

To examine the usefulness of the proposed method, we consider f is given by

$$f(q) = \sum_{(k_1,k_2,k_3)\in\{0,1\}^3} C_{k_1k_2k_3} q_1^{k_1} q_2^{k_2} q_3^{k_3} \tag{14}$$

We generate 600 examples of $f(q)$ by generating $C_{k_1k_2k_3}$ using drand48, which generate pseudo-random numbers. For each $f(q)$, we compute estimate of outbd $f(Q)$ by applying the proposed method. $f(\Omega(F_3^k))$ appears at outbd $f(Q)$ for only 10 examples.

Fig. 5 shows a typical example. In this case, outbd $f(Q) =$ outbd $[\{f(F_1^k)\}]$. At $q^i = [q_1, q_2, q_3]^T$, $q_j = \ell_j/10$, $\ell_j = 0, 1, \cdots, 10$, $j = 1, 2, 3$, we compute $f(q^i)$ and show it in Fig. 5. When readers see these points, readers might consider that outbd $f(Q)$ is more complicated. But we examined that the estimate of outbd $f(Q)$ by our method gives the exact outbd $f(Q)$ by applying finer gridding.

In Fig. 6, there are 12 $f(F_1^k)$'s, which are shown by solid lines, 4 $f(\Omega(F_2^k))$'s, which are shown by broken lines, and 1 $f(\Omega(F_3^k))$, which is shown by a dotted line. outbd $[\{(F_1^k)\}]$ is shown by a thick solid polygon. Two of $f(\Omega(F_2^k))$'s appears at outbd $f(Q)$ partially. Endpoints $f(\Omega(F_3^k))$ are shown by $\circ$'s.

We also consider 500 multilinear functions of 4 variables. In this case, some $f(\Omega(F_r^k))$, $r \geqq 3$, appears at outbd $f(Q)$ for only 16 examples (for 13 examples of them, $f(\Omega(F_r^k))$, $r \geqq 3$ are overlapped by $f(\Omega(F_2^{k'})))$. We have similar results for multilinear functions of 5 or 6 variables.

Moreover, we emphasize that the computing time of the proposed method is incomparably small than that of gridding method.

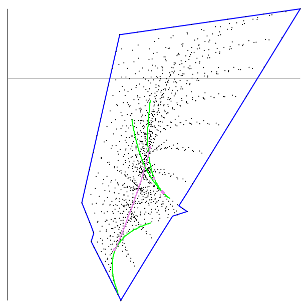


Fig. 5

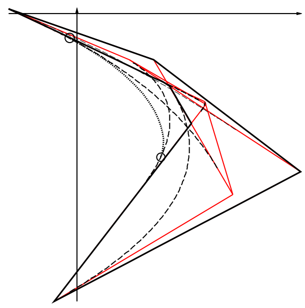


Fig. 6

III. Further Improvement

In this section, we consider to further reduce the computing time of the method we proposed in the previous section when $f(q)$ has sub-functions which are TDExps.

Suppose that $q \in \mathbf{R}^m$ is given by $q = [q_1^T\ q_2^T\ \cdots\ q_{\tilde{m}}^T]^T$, $q_\ell \in Q_\ell = [0,1]^{m_\ell}$, and f is given by

$$f(q) = \tilde{f}(z), \quad z_\ell = f_\ell(q_\ell), \quad \ell \in [1..\tilde{m}]$$

where f_ℓ is a TDExp and $\tilde{f} : \mathbf{C}^{\tilde{m}} \to \mathbf{C}$ is a multi-linear function of $z = [z_1\ z_2\ \cdots\ z_{\tilde{m}}]^T$. Let $Z_\ell = f_\ell(Q_\ell)$, $\ell \in [1..\tilde{m}]$, and $Z = Z_1 \times Z_2 \times \cdots \times Z_{\tilde{m}}$.

First of all we show the following:

Theorem 3: If z be arbitrary point of Z such that $z_\ell \in$ int Z_ℓ for some ℓ, then

$$\tilde{f}(z) \in \text{int } \tilde{f}(Z). \tag{15}$$

Therefore, we have

$$\begin{aligned} \partial f(Q) &= \partial\tilde{f}(Z) \\ &\subseteq \partial\{\tilde{f}(z) \mid z_\ell \in \partial Z_\ell \quad \forall\, \ell \in [1..\tilde{m}]\}. \end{aligned} \tag{16}$$

By Theorem 3, it is enough to consider the image of "boundary" of Z as long as we consider $\partial f(Q)$.

Since f_ℓ is a TDExp, a polygon P_ℓ satisfying

$$\text{outbd } f_\ell(Q_\ell) \subseteq P_\ell \subseteq \mathcal{N}(\text{outbd } f_\ell(Q_\ell); \varepsilon)$$

is computed very fast by using NPIA[10], where $\mathcal{N}(V;\varepsilon)$ denotes an ε neighborhood of V.

In the following, we set $Z_\ell = P_\ell \supseteq f_\ell(Q_\ell)$. Suppose that Z_ℓ is an n_ℓ-gon. Let $\{z_\ell^j\}_{j=1}^{n_\ell}$ be the set of all nodes of Z_ℓ, and $\{L_{Z_\ell}^{i_\ell} = \text{conv}\,[z_\ell^{i_s},\, z_\ell^{i_t}],\ i_s = 1, 2, \cdots, n_\ell\}$ be the set of all edges of P_ℓ, where $i_t = i_s + 1$ if $i_s < n_\ell$ and $i_t = 1$ if $i_s = n_\ell$.

Let $\mathcal{I} = \{I = (i_1,\, i_2,\, ...,\, i_{\tilde{m}}) \mid i_\ell \in [1..n_\ell],\ \ell \in [1..\tilde{m}]\}$. Selecting $I \in \mathcal{I}$ means that we consider a vector z, whose ℓ-th component z_ℓ is on the i_ℓ-th edge $L_{Z_\ell}^{i_\ell}$ of the ℓ-th polygon. Since any $L_{Z_\ell}^{i_\ell} = \text{conv}\,[z_\ell^{i_s},\ z_\ell^{i_t}]$ can be written as $L_{Z_\ell}^{i_\ell} = \{t_\ell z_{i_s}^\ell + (1-t_\ell) z_{i_t}^\ell \mid t_\ell \in [0,\ 1]\}$, for each $I \in \mathcal{I}$ we define a complex function $g(\cdot; I) : \mathbf{R}^{\tilde{m}} \to \mathbf{C}$ by

$$\begin{aligned} g(t;I) \quad &= \quad \tilde{f}(z), \quad z = [z_1 \ \ z_2 \ \ \cdots \ \ z_{\tilde{m}}]^T, \\ & \qquad z_\ell = t_\ell z_\ell^{i_s} + (1-t_\ell) z_\ell^{i_t} \in L_{Z_\ell}^{i_\ell} \\ & \qquad t_\ell \in [0,1], \quad \ell \in [1..\tilde{m}]. \end{aligned} \tag{17}$$

Then, obviously we have,

$$\partial f(P) = \bigcup_{I \in \mathcal{I}} \partial g([0,1]^{\tilde{m}}; I). \tag{18}$$

By applying Theorems 1, 2 and GetBoundaryfor $f(\cdot) = g(\cdot; I)$, we can compute $\partial g([0,1]^{\tilde{m}}; I)$. It requires that to apply GetBoundary $|\mathcal{I}|$ times, where $|\mathcal{I}| = \prod_{\ell=1}^{\tilde{m}} n_\ell$.

A typical example of $f_\ell(q_\ell)$ is

$$q_\ell \in \mathbf{R}^3 \quad f_\ell(q_\ell) = s^2 q_{\ell,3} + s q_{\ell,2} + q_{\ell,1}$$

and Z_ℓ is rectangular. In this case, $\tilde{m} = m/3$, $n_\ell = 4$, $|\mathcal{I}| = 4^{\tilde{m}}$.

Roughly speaking, computing time of GetBoundaryis proportional to the number of faces of $[0,1]^m$, which is $\sum_{r=1}^m {}_mC_r 2^{m-r}$. Note that the ratio

$$\frac{\sum_{r=1}^m \ {}_mC_r 2^{m-r}}{\sum_{r=1}^{\tilde{m}} \ {}_{\tilde{m}}C_r 2^{\tilde{m}-r} 4^{\tilde{m}}} \approx e^{4\tilde{m}/18+1.5} \quad \tilde{m} = 1, 2, ..., 7$$

and, hence, computing Z_ℓ and considering $\tilde{f}(z)$ is useful to reduce the computing time.

Let $F_r^k(I)$ be the k-th r-face of the $\tilde{m}$ polytope $[0,1]^{\tilde{m}}$ corresponding to $I \in \mathcal{I}$ and

$$f(\mathcal{E}_Z) = \bigcup_{F_1^k(I) \in \mathcal{F}_1([0,1]^{\tilde{m}}),\ I \in \mathcal{I}} g(F_1^k(I); I)$$

and

$$f(\mathcal{V}_Z(I)) = \bigcup_{z = F_0^k(I) \in \mathcal{F}_0([0,1]^{\tilde{m}})} g(z; I).$$

To reduce the computing time much more, we propose the following:

procedure CheckConvHull

Step 1. V := outbd $f(\mathcal{E}_Z)$;
Step 2. for each $I \in \mathcal{I}$ do begin
  if conv $f(\mathcal{V}_Z(I)) \subseteq V$ then continue;
Step 3. else V := GetBoundary$\cup\, V$;
  end;
Step 4. return V;

In our experience, by using CheckConvHull, we can reduce computing time about $1/10$.

IV. Concluding Remark

In this paper, we derived two basic results for computing the boundary of value sets (Theorems 1 and 3), and proposed a method computing a good estimate of value sets. At the present, we have no systematic method to check the condition in Theorem 2, but at least numerical testing we have no example that the image of any gridding point locates outside of the region computed using GetBoundaryas long as we can examine (for each $m \in [3..6]$ we generate at least 500 sets of coefficient of multi-linear functions by using the function drand48 generating pseudo-random numbers).

The computing time of the proposed method is incomparably small than that of gridding method. Moreover, we have a polygon, not just a set of points which are computed by gridding, and it is very useful in applications, for example it is easy to check the value set includes 0 or not.

References

[1] B. Barmish, *New Tools for Robustness of Linear Systems*, Macmillan (1994).
[2] S. P. Bhattacharyya, H. Chapellat and L. H. Keel: *Robust Control: The Parametric Approach*, Prentice Hall (1995).
[3] I. Horowitz, *Synthesis of Feedback Systems*, Academic Press 1963.
[4] J. Ackermann A. Bartlett, D. Kaesbauer, W. Sienel and R. Steinhauser, *Robust Control Systems with Uncertain Physical Parameters*, Springer-Verlag, Berlin (1993).
[5] T. E. Djaferis, *Robust Control Design: A Polynomial Approach*, Kluwer Academic Pub. (1995).
[6] J. Ackerman and W. Sienel, "On the computation of value sets for robust stability analysis," *Proc. of 1st European Control Conf.* pp.1345–1350 (1991).
[7] P. O. Gutman, C. Baril and L. Neumann, "An algorithm for computing value sets of uncertain transfer functions in fractional form", *IEEE Trans. on Auto. Contr.*, pp.1268-1273 (1994).
[8] Y. Ohta, L. Gong and H. Haneda, "Polygon interval arithmetic and interval evaluation of value sets of transfer functions," *IEICE Trans. on Fundamentals*, Vol. E77-A, No. 6, pp.1033-1042 (1994).
[9] B. Barmish, J. Ackerman and H. Hu, "The tree structured decomposition: a new approach to robust stability analysis," *Proc. Conf. on Information Sciences and Systems held at Princeton* (1990).
[10] Y. Ohta, "Non-convex Polygon Interval Arithmetic as a Tool for the Analysis and Design of Robust Control Systems," J. of

Reliable Computing, Special Issue on Applications to Control, Signals, Vol. 6, No.3, pp.247-279 (2000).
[11] L. Bears, *Introduction to Topology*, New York University (1945-1955).
[12] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).
[13] A. S. B. Holland, *Complex Function Theory*, Elsevier North Holland (1980).

APPENDIX

Proof of Theorem 1.

To prove Theorem 1 we need the following:

Lemma A.1: Let $F_r^k \subseteq Q$ be an $r - face$, where $r \geqq 2$. If $q^1 \in$ ri F_r and if $\operatorname{rank} H(q^1; \mathcal{T}_r) = 2$, then $f(q^1) \in$ ri $f(F_r)$.

(Proof) Let us consider a Function $\tilde{f} : \mathbf{R}^r \to \mathbf{R}^r$ given

$$\tilde{f}(q) = [x(q) \;\; y(q) \;\; q_{i_3} \;\; \cdots q_{i_r}]^T$$

where $i_3, \cdots, i_r$ are chosen so that $\det \tilde{f}'(q^1) \neq 0$, where $\tilde{f}'(q^1)$ is the Jacobian matrix of $\tilde{f}$ at $q = q^1$. Then, by Inverse Function Theorem [12], $\tilde{f}$ is a local homeomorphism in a neighborhood of q^1, and, hence, it is an open mapping because of the domain invariance theorem [11] (a continuous local homeomorphism in a finite dimensional space is an open mapping). Therefore, we have $\tilde{f}(q^1) \in$ ri $f(F_r)$.

Proof of Theorem 1.

Since $Q = F_m^1$, by Lemma A.1, we have

$$\partial f(Q) \subseteq f(\Omega_{m-1}^1) \, \cup \, \partial f(\partial Q)$$

Note that

$$\partial Q = \bigcup_{F_{m-1}^k \in \mathcal{F}_{m-1}(Q)} F_{m-1}^k,$$

and by Lemma A.1, if $q \in$ ri F_r^k then $f(q) \in$ int $f(F_r^k)$, and, hence, we have

$$\partial f(\partial Q) \subseteq \bigcup_{F_{m-1}^k \in \mathcal{F}_{m-1}(Q)} [f(\Omega_{m-1}^k) \, \cup \, \partial f(\partial F_{m-1}^k)].$$

Repeating this process, we finally have (8). ∎

Proof of Lemma 1.

It is easy to see that 1) $\Longrightarrow$ 2) and that 3) $\Longrightarrow$ 1). We show that 2) $\Longrightarrow$ 3). 2) means that $\tilde{h}^{j_1} = [h_1^{j_1}(x) \; -h_2^{j_1}(x)]^T \in \mathbf{R}^2$ and $[h_1^{j_\ell}(x) \;\; h_2^{j_\ell}(x)]^T \in \mathbf{R}^2$ is orthogonal. On the other hand, $\tilde{h}^{j_1}$ and $[h_1^{j_1}(x) \;\; h_2^{j_1}(x)]^T$ is orthogonal, and, hence, we have 3). This completes the proof. ∎

Proof of Theorem 2.

Let us consider the case when $r = m - 1$, $k = 1$. Other cases can be proved in a quite similar way.

From the assumption, there exists a $\hat{j} \in J_m^1 = [1..m]$ such that the $(m-1)\times(m-1)$ matrix $M(q)$ obtained by removing the $\hat{j}$-th column $\gamma(q)$ from the Jacobian matrix $\frac{\partial \psi}{\partial q}(q; J_{m-1}^1)$ of $\psi(q;\, J_{m-1}^1)$ is nonsingular. In the following, without loss of generality, we assume that $\hat{j} = m$ and let

$$q = \begin{bmatrix} \xi \\ \eta \end{bmatrix}, \quad \frac{\partial \psi}{\partial q}(q) = [M(q) \;\; \gamma(q)].$$

Since $M(q)$ and $\gamma(q)$ are continuous and since $[0,1]^m$ is compact, there exists a constant $\Delta > 0$ such that

$$\|M(q)^{-1}\| \, |\gamma(q)| \leqq \Delta \quad \forall \, q \in [0,1]^m.$$

Let $q^1 = q^*$. By Implicit Function Theorem, there are neighborhoods B_ξ^1 of ξ^1 and $B_\eta^1 = (\alpha_1,\, \beta_1) \subseteq [0,1]$ of η^1, where $\alpha_1 < \eta^1 < \beta_1$, such that for any $\eta \in [\alpha_1,\, \beta_1]$ the equation $\psi(q; J_m^1) = \psi((\xi,\eta); J_m^1) = 0$ has a unique solution $\xi(\eta) \in \overline{B_\xi^1}$ and that $\xi : B_\eta^1 \to \mathbf{R}^{m-1}$ is continuously differentiable and satisfies

$$\frac{\partial \xi}{\partial \eta}(\eta) = M^{-1}(\chi(\eta))\gamma(\chi(\eta)), \quad \chi(\eta) = \begin{bmatrix} \xi(\eta) \\ \eta \end{bmatrix}.$$

We will show that

$$\exists \, \tilde{\eta} \in (\eta^1, 1] : \; \chi(\tilde{\eta}) \in \partial[0,1]^m. \tag{A.1}$$

If $\beta_1 \geqq 1$ then there exists $\tilde{\eta} \in [\eta^1, \beta_1]$ such that (A.1) holds. Therefore, we assume for any $\eta \in [\eta^1, \beta_1]$ $\chi(\eta) \in$ int $[0,1]^m$.

Set $i = 2$. Suppose that we repeat the following as long as (A.1) does not hold:

At $q^i = \chi(\beta_{i-1})$, apply Implicit Function Theorem to have B_ξ^i of ξ^i and $B_\eta^i = (\alpha_i,\, \beta_i) \subseteq [0,1]$ of η^i, then increment i.

Then, the above repetition ends for a finite i (in this case, (A.1) holds), or we have a monotone increasing infinite sequence $\{\beta_j\}$ and

$$\chi(\eta) \in \text{ int } [0,1]^m \quad \forall \, \eta \in [\beta_{i-1}, \beta_i] \subseteq [0,1].$$

Then, $\tilde{\beta} = \lim_{j\to\infty} \beta_j \leqq 1$, and for any $k < j$ we have

$$\begin{aligned} |\xi(\beta_j) - \xi(\beta_k)| &= \left| \int_{\beta_k}^{\beta_j} \frac{\partial \xi}{\partial \beta}(\beta) d\beta \right| \\ &= \left| \sum_{\ell=k}^{j-1} \int_{\beta_\ell}^{\beta_{\ell+1}} J^{-1}(\chi_\ell(\beta))\gamma(\chi_\ell(\beta)) d\beta \right| \\ &\leqq \Delta |\beta_j - \beta_k|, \end{aligned}$$

and, hence, $\{\xi(\beta_k)\}$ is a Cauchy sequence and it converges. If we set $\chi(\tilde{\beta}) = \lim_{k\to\infty} \chi(\beta_k)$, then we have $\psi(\chi(\tilde{\beta}); J_m^1) = 0$ by the continuity of ψ.

If $\chi(\tilde{\beta}) \in \partial[0,1]^m$, then we have (A.1) for $\tilde{\eta} = \tilde{\beta}$.

If it is not, (i.e., $\chi(\tilde{\beta}) \in$ int $[0,1]^m$), then $\tilde{\beta} < 1$ and we apply Implicit Function Theorem at $\chi(\tilde{\beta})$ and have B'_ξ

and (α', β'), $\beta' > \tilde{\beta}$, which contradict to the construction of $\{\beta_j\}$ and definition of $\tilde{\beta}$.

Therefore, the above repetition must finish in finite times, and (A.1) holds.

Similarly , we can show that there is an $\hat{\eta} < \eta^*$ such that $\chi(\hat{\eta}) \in \partial[0, 1]^m$.

Finally, by an affine transformation mapping $[\hat{\eta}, \tilde{\eta}]$ into $[0, 1]$, we have the conclusion. ∎

Proof of Theorem 3.

Let i be an index such that $z_i \in$ int Z_i and let $\tilde{Z}$ = $\{\tilde{z} \in Z \mid \tilde{z}_j = z_j \; j \neq i, \; \tilde{z}_i \in Z_i\}$. Obviously $z \in \tilde{Z}$ and there is a relative open set $G \subseteq \tilde{Z}$ such that $z \in G$. We consider a function $\hat{f} : \tilde{Z} \rightarrow$ **C** defined by $\hat{f}(z)$ = $\tilde{f}(z)$. Then by Open-Mapping Theorem ([13], p.225), $\hat{f}(G) = \tilde{f}(G)$ is an open set, and, hence, we have (15). From (15), we easily have (16). This completes the proof. ∎