数式処理 *J.JSSAC (2007)*
Vol. 13, No. 2, pp. 3 - 25
奨励賞論文

Development of Parametric Robust Control Design Toolbox

Noriko HYODO *

AlphaOmega Inc.

(RECEIVED 2007/1/15)

Abstract

A lot of important control system design problems are regarded as parametric and non-convex optimization problems. Parametric robust control design toolbox solves these robust control problems and visualize the feasible parameter regions via a parameter space approach based on symbolic-numeric computation. This paper shows a method of solving robust control system design problems by using symbolic-numeric computation and description of parametric robust control design toolbox.

1 Introduction

Control system design is to find out feasible parameters to be designed for which a target system satisfies given control design specifications. A lot of important control system design problems are regarded as parametric and non-convex optimization problems, and of course to solve these problems only by using numeric computation is difficult. Recently, there has been an increasing interest in the application of computer algebra to control system analysis and design.

We have been developing the parametric robust control design toolbox. This toolbox solves the robust control design problems by parameter space approach based on symbolic-numeric computation, and visualize the feasible parameter regions of robust control design problems. By using symbolic computation, this toolbox solves the robust control problems exactly. This toolbox uses quantifier elimination(QE) by symbolic computation.

2 Robust controller design by a parameter space approach

A parameter space approach is known to control community as an effective method for robust control synthesis and multi-objective design of fixed-structure controllers. Multi-objective robust control problem is one of main concerns in control theory. But it often the case that these robust

*noriko@a2z.co.jp

control problems are regarded as nonlinear and non-convex optimization problems. We have proposed a method with a software tool for parametric robust control synthesis by symbolic-numeric computation which can solve above problems. The method is based on a parameter space approach accomplished by using QE. This section shows the algorithm, methods and solving procedures of parameter space approach that are used in parametric robust control design toolbox.

2.1 Design procedure

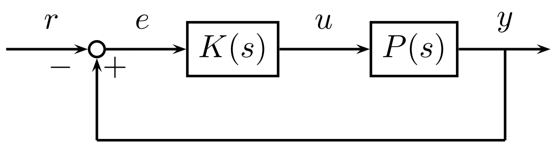

Fig. 1: A feedback control system

Now let us consider a feedback control system as shown in Fig.1, and propose a scheme for fixed-structure robust controller synthesis based on *a parameter space approach* as follows:

1. According to the characteristics of the plant and also design requirements, determine the structure of a class of controllers $\mathcal{K}$ and select the design parameters in $\mathcal{K}$. For instance, when the PI-controller $K(s) = k + \frac{m}{s}$ is chosen, k and m are the parameters to be designed.
2. Reduce the given specifications ϕ_i to the equivalent first-order formulas ψ_i.
3. Compute the admissible regions of the design parameters for all specifications ϕ_i by applying QE to the first-order formulas ψ_i derived from ϕ_i.
4. Superpose the admissible regions in the parameter space. Then we can take appropriate parameters from the intersections by considering other specifications.

In Step 2, most of important design specifications for robust control such as frequency restricted H_∞ norm constraints, stability (gain/phase) margin and stability radius specifications, and pole location requirements can be recast as sign definite condition (1) by using simple symbolic computations [6],[8]. This is beneficial to achieve Step 3 efficiently.

In Step 3, a specialized QE algorithm using Sturm-Habicht sequence [2], which is more efficient than a general QE algorithm based on cylindrical algebraic decomposition, can be applied for solving a sign definite condition [1]. By using specialized QE, nonlinear and non-convex problems could be solved exactly. QE-based method provides us an exact and whole feasible regions of the design parameters.

2.2 Sign definite condition

In this toolbox, selected specifications are reduced to the equivalent first-order formulas that is called sign definite condition(SDC). This is a definition of sign definite condition(see [6] for details).

Definition 1
A function $f(x) : \mathbb{R} \mapsto \mathbb{R}$ is sign definite in the interval $x \in [a, b], a < b$, denote $f(x) \in \mathrm{N}_0[a, b]$ hereafter, if $f(x)$ preserves the sign in the interval, or does not cross zero in the interval.

Then SDC $f(x) \in \mathrm{N}_0[a, b]$ can be transformed to the condition $f(z) \in \mathrm{N}_0[0, \infty]$ by the bilinear transformation $z = -(x - a)/(x - b)$. Put simply, $f(z) \in \mathrm{N}_0[0, \infty]$ means

$$\forall z > 0,\ f(z) > 0 \tag{1}$$

where $f(z)$ is an univariate polynomial with parametric coefficients.

The QE-based approach can uniformly and efficiently deal with the most of important design specifications for robust control by reducing specifications into SDC. Though some of the specifications are non-convex constraints, we can deal with such non-convex cases exactly and also parametrically by using QE.

2.2.1 Design examples

We propose some examples to show our design procedure for parametric controller design problems.

Example 2 (frequency restricted H_∞-norm constraints)
We consider a feedback control system as shown in Fig.1 with plant $P(s)$ and PI-controller $C(s)$ where

$$P(s) = \frac{1}{s-1},\ \ C(s) = k + \frac{m}{s}. \tag{2}$$

Our aim is to obtain parameters k and m which satisfy the following properties:

- the sensitivity function $S(s)$ and the complementary sensitivity function $T(s)$ satisfy the following frequency restricted norm constraints:

$$\|S(s)\|_{[0,\omega_s]} < \gamma_s, \tag{3}$$
$$\|T(s)\|_{[\omega_t,\infty]} < \gamma_t, \tag{4}$$

  where

$$S(s) = \frac{1}{1 + P(s)C(s)}, \tag{5}$$
$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}. \tag{6}$$

Based on the procedure shown in [6], the norm constraints (3) and (4) can be recast as SDCs.

$$f_s(x) = x^2 + \frac{(2m\gamma_s^2 - (k-1)^2\gamma_s^2 + 1)x + m^2\gamma_s^2}{-1 + \gamma_s^2} \in \mathrm{N}_0[0, \omega_s^2], \tag{7}$$
$$f_t(x) = x^2 + (2m - (k-1)^2 + \frac{k^2}{\gamma_t^2})x + m^2(1 - \frac{1}{\gamma_t^2}) \in \mathrm{N}_0[\omega_t^2, \infty]. \tag{8}$$

Then the SDC (7) and (8) are reduced by a bilinear transformation

$$\begin{aligned} f_s(z) = \gamma_s^4 z^3 &+ (\frac{((k-1)^2\gamma_s^2 - 2m\gamma_s^2 - 1)\omega_s^4 + m^2\gamma_s^2}{\gamma_s^2 - 1} - \gamma_s^4\omega_s^2)z^2 \\ &- \frac{((k-1)^2\gamma_s^2 - 2m\gamma_s^2 - 1)\omega_s^2 + 2m^2\omega_s^2\gamma_s^2}{\gamma_s^2 - 1}z + \frac{m^2\omega_s^4\gamma_s^2}{\gamma_s^2 - 1} \in \mathrm{N}_0[0, \infty], \end{aligned} \tag{9}$$

$$f_t(z) = z^2 + (2m - (k-1)^2 + \frac{k^2}{\gamma_t^2} - 2\omega_t)z + \omega_t^4 - \omega_t^2(2m - (k-1)^2 + \frac{k^2}{\gamma_t^2}) + m^2(1 - \frac{1}{\gamma_t^2}) \in \mathrm{N}_0[0, \infty]. \tag{10}$$

By using QE to solve SDC (9) and (10), we obtain feasible regions. Then we can get the feasible regions of controller parameters so that the system (2) satisfies the mixed sensitivity specification.

Example 3 (Gain margin constraint)
Gain margin constraint can be reduced to a sign definite condition as follows (see [8]). Consider a transfer function $G(s)$ and decompose $G(j\omega)$ as

$$G(j\omega) = \frac{g_r(\omega) + jg_j(\omega)}{d(\omega)} \tag{11}$$

where $g_r(\omega)$, $g_j(\omega)$ and $d(\omega)$ are polynomials in ω.

As shown in [8], $G(s)$ holds the gain margin (γ_m, γ^M) iff the following system of equations

$$\begin{cases} f_1(\omega, t) &= g_r(\omega) - d(\omega)t = 0 \\ f_2(\omega) &= g_j(\omega) = 0 \end{cases}$$

is not satisfied in $\omega \in \mathbb{R}, t \in [-1/\gamma_m, -1/\gamma^M]$. One can obtain a polynomial $f_g(t)$ by eliminating ω from f_1, f_2. Then the condition that $G(s)$ holds the gain margin (γ_m, γ^M) can be reduced to a sign definite condition, that is, $f_g(t)$ is sign definite in $t \in [-1/\gamma_m, -1/\gamma^M]$. By bilinear transformation, we can get the $f_g(z) \in [0, \infty]$.

Example 4 (Phase margin constraint)
Phase margin constraint can be reduced to a sign definite condition as follows (see [8]). As the in the case of gain margin constraint, consider a transfer function $G(s)$ and decompose $G(j\omega)$ (see (11)).

As shown in [8], $G(s)$ holds the phase margin ϕ iff the following system of equations

$$\begin{cases} f_1(\omega) &= g_r^2(\omega) + g_j^2(\omega) - d^2(\omega) = 0 \\ f_2(\omega, t) &= g_r(\omega) - d(\omega)t = 0 \end{cases}$$

is not satisfied in $\omega \in \mathbb{R}, t \in [-1, cos(-\pi + \phi)]$. One can obtain a polynomial $f_p(t)$ by eliminating ω from f_1, f_2. Then the condition that $G(s)$ holds the phase margin ϕ can be reduced to a sign definite condition, that is, $f_p(t)$ is sign definite in $t \in [-1, cos(-\pi + \phi)]$. By bilinear transformation, we can get the $f_p(z) \in [0, \infty]$.

Example 5 (Pole location(Wedge shape))
For the wedge shape region, $x + jy \in \overline{D}$ can be expressed as

$$\begin{cases} x &= \omega \\ y &= m(\omega - t) \end{cases}, \omega \in R, t \in [b, \infty].$$

Let us consider how to assign all roots of a characteristic polynomial $p(s)$ in the specified region $D \in C$. This is equivalent to

$$p(s) \neq 0, \ \ \forall s \in \overline{D}$$

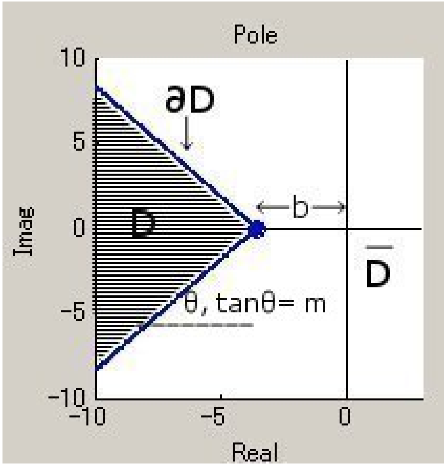

Fig. 2: Pole location(wedge shape)

where $\overline{D}$ denotes the complementary set of D in C. In this case, $p(s)$ is called D-stable. Then, the pole location problem is stated as follows:

$$\begin{cases} p_r(x,y) &= 0 \\ p_j(x,y) &= 0 \end{cases}, p(s) = p_r(x,y) + jp_j(x,y)$$

do not hold in $s = x + jy \in \overline{D}$.

As shown in [8], $p(s)$ has all roots in D of which complementary set $\overline{D}$

$$\begin{cases} P_r(\omega,t) &= 0 \\ P_j(\omega,t) &= 0 \end{cases} where \begin{cases} P_r(\omega,t) &= p_r(x(\omega,t), y(\omega,t)) \\ P_j(\omega,t) &= p_j(x,y) \end{cases}$$

is not satisfied in $\omega \in R, t \in [b,\infty]$. One can obtain a polynomial $f_w(t)$ by eliminating ω from P_r, P_j. Then the condition that all roots of $p(s)$ are in the wedge shape region can be reduced to a sign definite condition, that is, $f_w(t)$ is sign definite in $t \in [b,\infty]$. By bilinear transformation, we can get the $f_w(z) \in [0,\infty]$.

2.3 Solving SDC by a specialized QE

This section briefly sketches a special QE method based on the Sturm-Habicht sequence for the SDC (see [1] for details), which plays a key rôle in this paper.

Definition 6 (L. Gonzàlez et al. [3])

Let P, Q be polynomials in $\mathbb{R}[x]$ and write $P = \sum_{k=0}^{n} a_k x^k$, $Q = \sum_{k=0}^{m} b_k x^k$, where n, m are non-negative integers. For $i = 0, 1, \ldots, \ell = \min(n,m)$, the subresultant associated to P, n, Q and m of index i is defined by $Sres_i(P,n,Q,m) = \sum_{j=0}^{i} M_j^i(P,Q)x^j$, where $M_j^i(P,Q)$ is the determinant of the matrix composed by the columns $1, 2, \ldots, n+m-2i-1$ and $n+m-i-j$ in the matrix

$$s_i(P,n,Q,m) := \overbrace{\begin{pmatrix} a_n & \ldots & a_0 & & \\ & \ddots & & \ddots & \\ & & a_n & \ldots & a_0 \\ b_m & \ldots & b_0 & & \\ & \ddots & & \ddots & \\ & & b_m & \ldots & b_0 \end{pmatrix}}^{n+m-i} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} m-i \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} n-i \end{matrix}$$

Let $v = n+m-1$ and $\delta_k = (-1)^{\frac{k(k+1)}{2}}$. The Sturm-Habicht sequence associated to P and Q is then defined as the list of polynomials $\{SH_j(P,Q)\}_{j=0,\ldots,v+1}$ given by

$$\begin{aligned} SH_{v+1}(P,Q) &= P\,, \\ SH_v(P,Q) &= P'Q\,, \\ SH_j(P,Q) &= \delta_{v-j} \cdot Sres_j(P,p,P'Q,v) \\ &\qquad \text{for } j = 0,1,\ldots,v-1\,, \end{aligned}$$

where $P' = \frac{dP}{dx}$. In particular, $\{SH_j(P,1)\}_{j=0,\ldots,v+1}$ is called the Sturm-Habicht sequence of P. Here it is simply denoted by $\{SH_j(P)\}$.

The Sturm-Habicht sequence can be used for real root counting, just like the Sturm sequence. Moreover it has favourable properties in terms of specialization and computational complexity (see [3, 4] for details).

Theorem 7 (González-Vega et al. [4])
Let $P(x) \in \mathbb{R}[x]$ and

$$\{g_0(x), \ldots, g_k(x)\}$$

be a set of polynomials obtained from $\{SH_j(P(x))\}$ by deleting the identically zero polynomials. Let $\alpha,\beta \in \mathbb{R} \cup \{-\infty,+\infty\}$ and $\alpha < \beta$. Define $W_{SH}(P;\alpha)$ as the number of sign variations on $\{g_0(\alpha), \ldots, g_k(\alpha)\}$. Then, $W_{SH}(P;\alpha,\beta) := W_{SH}(P;\alpha) - W_{SH}(P;\beta)$ gives the number of real roots of P in $[\alpha,\beta]$.

Denote the principal j-th Sturm-Habicht coefficient of $SH_j(f)$, i.e., the coefficient of degree j of $SH_j(f)$, by $st_j(f)$ and a constant term of $SH_j(f)$ by $ct_j(f)$ for all j. Then,

$$\begin{aligned} W_{SH}(f;0,+\infty) &= W_{SH}(f;0) - W_{SH}(f;+\infty) \\ &= V\Big(\{ct_n(f), \ldots, ct_0(f)\}\Big) - V\Big(\{st_n(f), \ldots, st_0(f)\}\Big), \end{aligned} \qquad (12)$$

where $V\Big(\{a_i\}\Big)$ stands for the number of sign changes over the sequence $\{a_i\}$.

The SDC holds if and only if both $W_{SH}\,(f;0,+\infty) = 0$ and $st_n(f) > 0$ hold.

Hence an equivalent condition to the SDC can be obtained through the following *combinatorial procedure*:

1. Consider all (at most) 3^{2n-1} possible sign combinations over the polynomials $ct_i(f)$, $st_i(f)$ since $ct_0(f) = st_0(f)$, $st_n(f) > 0, st_{n-1}(f) > 0$.

2. Choose all sign conditions that satisfy $W_{SH}(f; 0, +\infty) = 0$ by (12).
3. Construct semi-algebraic sets generated by $ct_i(f)$, $st_i(f)$ for the selected sign conditions and combine them as a union.

The obtained condition is of the form of a union of semi-algebraic sets, so called *disjunctive normal form*. (The obtained result is expected to contain many empty sets. Some impossible sign combinations can be pruned beforehand (see [1]).)

3 Parametric robust control design toolbox

Our parametric robust control design toolbox is a GUI-based parametric robust control toolbox. And this toolbox is also based on symbolic quantifier elimination cooperating with numerical simulation.

To use Maple/MATLAB as a platform has the advantage that Maple packages are automatically incorporated into MATLAB by using "MATLAB Extended Symbolic Math Toolbox". The QE solver used in our toolbox is a Maple package called "SyNRAC", which is a symbolic-numeric toolbox for solving real algebraic constraints [9]. Our toolbox provides visualization facilities (of bode diagram, nyquist plot and pole location) for

- open-loop analysis, and
- controller synthesis.

Those are shown by using numerical computation.

3.1 General appearance

Current version of our toolbox supports controller synthesis in terms of following specifications:

- H_∞ norm constraints (sensitivity/complementary sensitivity functions)
- Hurwitz stability
- stability (gain/phase) margin specification
- pole location requirement

This toolbox can deal with not only a single-objective controller synthesis but also multi-objective controller synthesis among the above specifications based on a parameter space approach accomplished by quantifier elimination.

And current version of our parametric robust control design toolbox has four windows:

- **Main window** (Fig.3(1))
  has some edit field, controller, plant, specifications, and parameters.
- **Control synthesis window** (Fig.3(2))
  shows Bode diagram (for sensitivity function and complementary sensitivity function) and Nyquist plot and Pole/Zero Location for closed-loop transfer function, and users change specifications to manipulate graphic objects on the control synthesis window.
- **Open-loop window** (Fig.3(3))
  shows Bode diagram and Pole/Zero Location for open-loop transfer function.

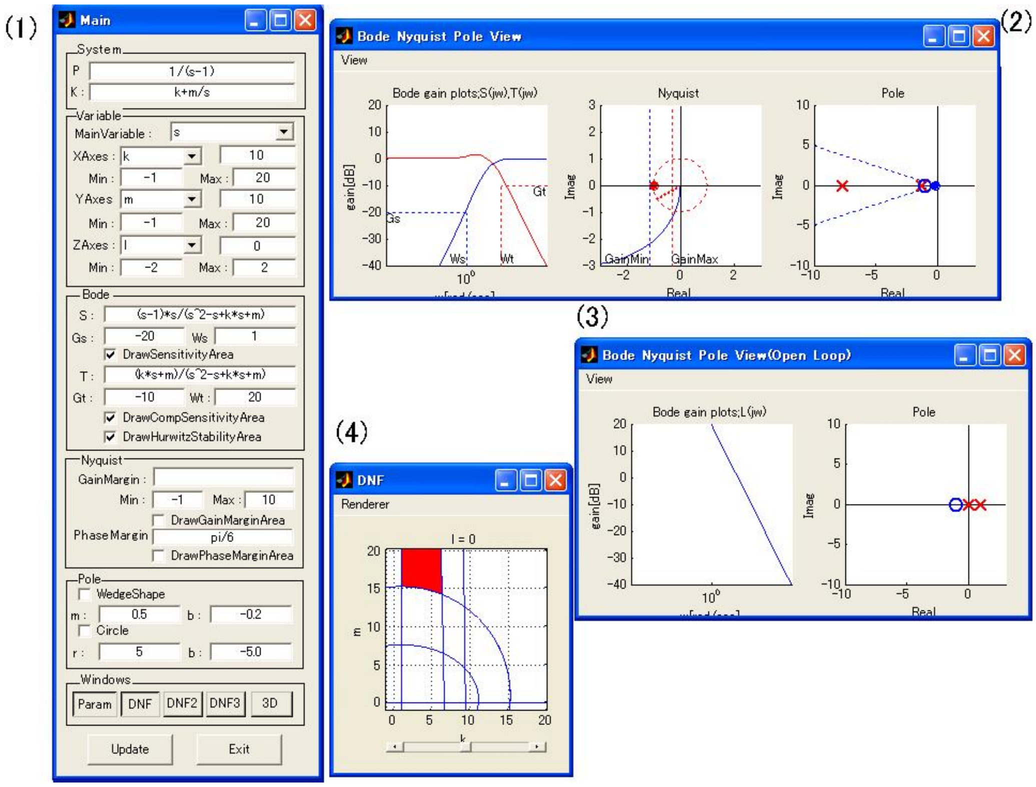

Fig. 3: A screen shot of parametric robust control toolbox

- **Parameter space window** (Fig.3(4))
  shows the regions of feasible parameters that satisfy specifications.

Our toolbox can be easily handled by users via the above mentioned windows. Basic operating procedure is shown as follows:

1. Select and push the “Windiws" buttons located at bottom of Main window which you want to show on main window.
2. Determine the structure of the controller and plant, and type the controller and plant into the boxes marked “System" at the top of main window.
3. Select the specifications which you want to investigate, check the boxes corresponding to your selection, and specify/input the concrete specifications.
4. Push the “Update" button, you can get feasible regions.

In Step 2, plant and controller structure can be specified with main window.

In Step 3, users can select a number of specifications. And there are two ways to set specifications, users can use both input methods. One method is to edit text box on main window, and another method is to move graphic objects on parameter window by using a mouse.

In Step 4, feasible regions are computed and shown on parameter region window. The toolbox shows feasible region which satisfy each specifications. Fig.3(4) is shown a superposition of feasible regions of sensitivity, complementary sensitivity and Hurwitz stability.

If users click or drag with the mouse on parameter region window, parameter values are reflected in system, users can look see the behavior of bode and nyquist diagram and pole assignment on parameter window.

Above description is about using method of parametric robust control design toolbox. Fig.4 is a system and execution flow of parametric robust control design toolbox.

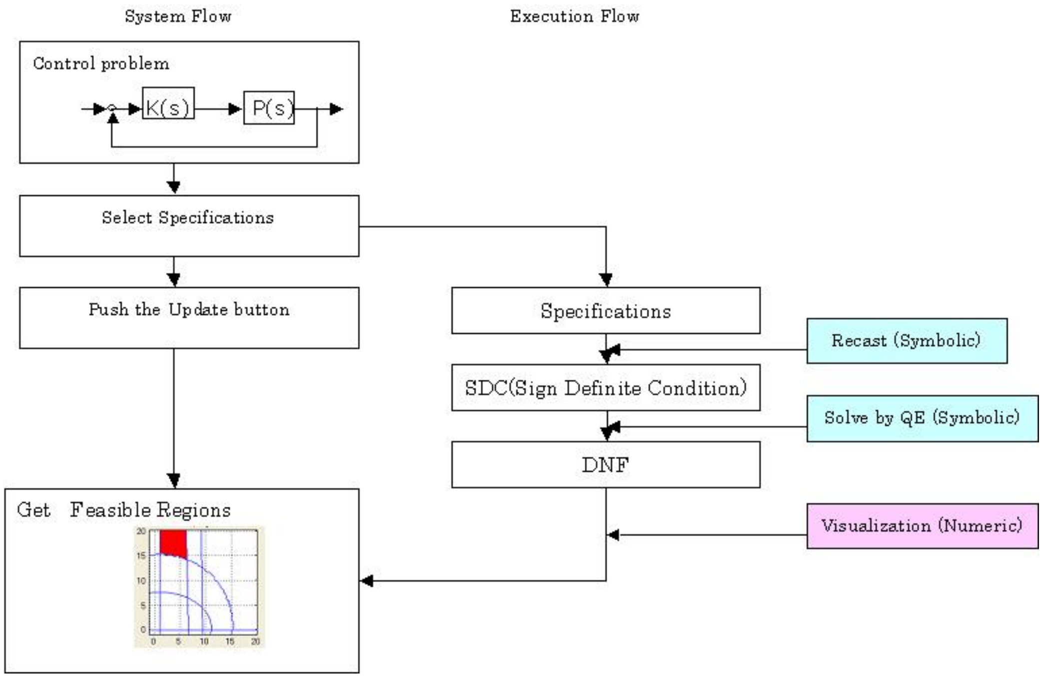

Fig. 4: System and execution flow

3.2 Optional functions

3.2.1 3-D view

If controller has 3 parameters, *e.g.*,PID-controller, the toolbox displays 3-D figure and cross-sectional views of feasible regions. Fig.5 is an example of 3-D view and Fig.6 is cross-sectional views at each axis of all feasible parameters for satisfying H_∞ norm specification(sensitity). As in case of 2 parameters system, users can click or drag with the mouse on parameter region window, parameter values are reflected in system, users can look see the behavior of bode and nyquist diagram and pole assignment on parameter window. Cross-sectional views are movable by slide bar.

3-D view windows has some check boxes.

- show whole region

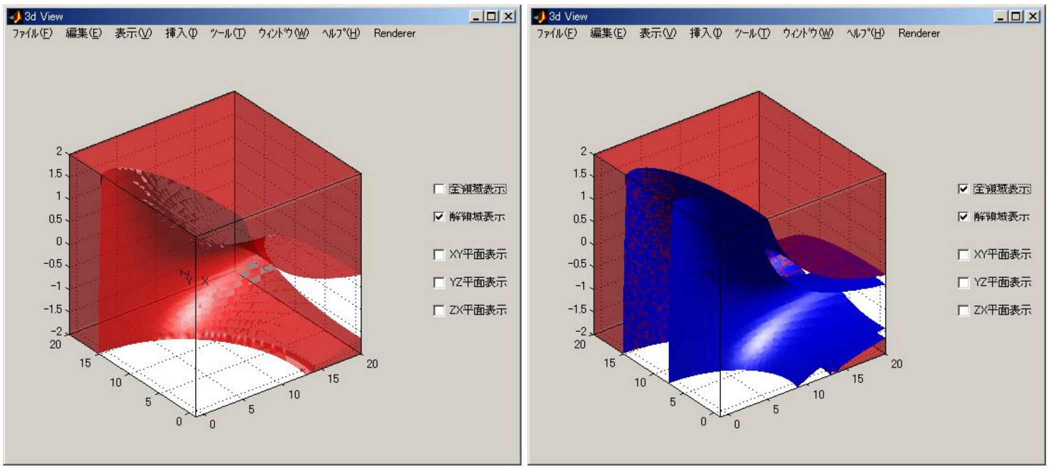

Fig. 5: A screen shot of parametric robust control toolbox

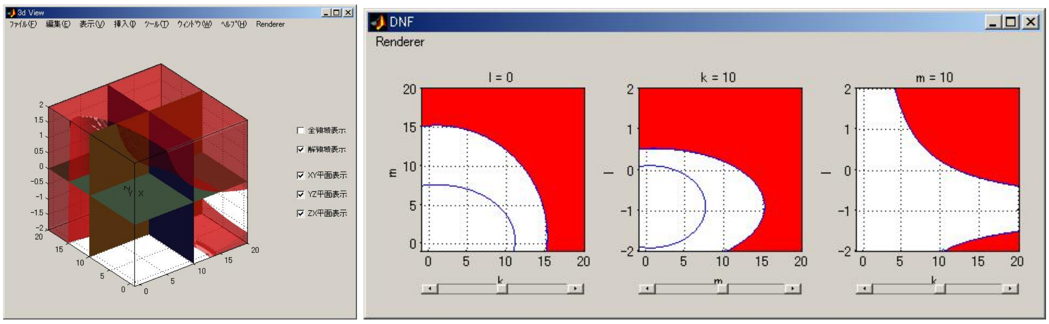

Fig. 6: A screen shot of parametric robust control toolbox

- show only feasible region

Left side of Fig.5 shows feasible region of phase margin specification, and right side one is view of whole region.

- show XY plane
- show YZ plane
- show ZX plane

And left side of Fig.6 are snapshots that each plane is shown. The XY, YZ, ZX planes in 3-D view window work with slide bars on parameter region window(right side of Fig.6).

3.2.2 Approximate feasible parameter regions

Parametric robust control design toolbox gives us exact feasible regions of parameters since it solves control design problems by using QE. But QE-based approach is expensive and can not solve large size problems in a reasonable amount of time. Hence, in such cases, it would be reasonable to employ a numerical method to obtain an approximate result. Here we employ a *validated numerical method* to solve a SDC approximately (but with guarantee) using interval arithmetics.

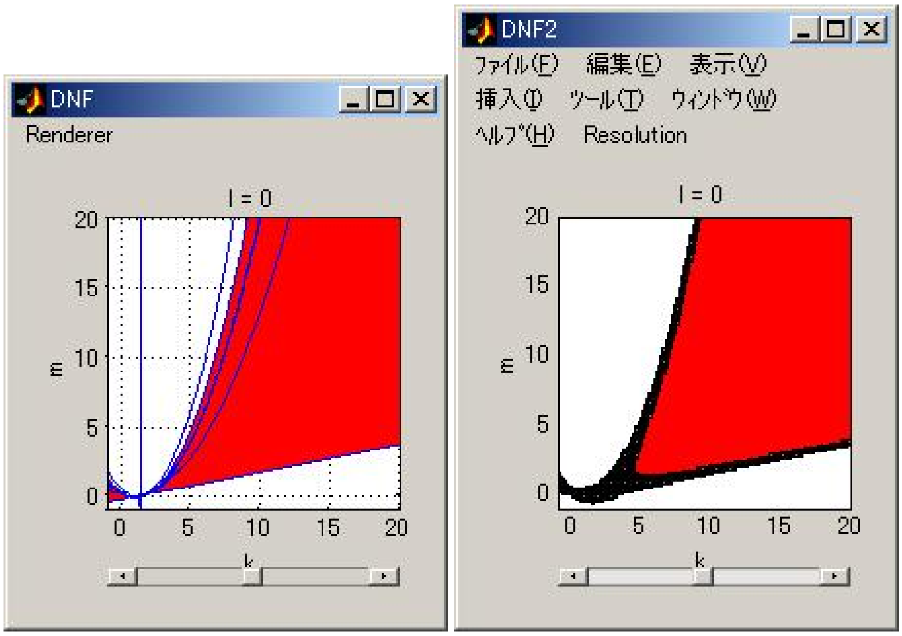

Fig. 7: A screen shot of parametric robust control toolbox

The algorithm is shown as follows:

Algorithm 8

- Given:
  - specification ϕ
  - box B
- Find:
  An element of the set {T, F, U}
  - T implies that ϕ is true for all elements of B
  - F implies that ϕ is false for all elements of B
  - U undecided

We consequently obtain three kinds of regions in a parameter space by using Algorithm8,

- Red : regions which satisfy the given SDC,
- White : regions which does not satisfy the given SDC,
- Black : undecided regions,

which are composed by a set of boxes. One can first specify accuracy (fineness of a smallest box) when we use this function. In other words, this is a *box decomposition* of a parameter space with respect to a given SDC to the specified accuracy. See the right side figure of Fig.7 for example.

4 Conclusion

We have been developing a parametric robust control design toolbox that is based on MATLAB and for robust parametric control via a parameter space approach based on symbolic-numeric computation. By using specialized QE, nonlinear and non-convex problems could be solved exactly, users can get feasible parameter areas instead of feasible parameter point. This toolbox can treat not only single objective problems but also multi-objective problems, we can check the feasible regions that is superposed feasible regions of each problem. And our toolbox can be easily handled because the toolbox is a GUI-based toolbox, and can show the feasible parameter areas by visualization. For those reason, this toolbox is very helpful in education and actual engineering fields.

5 Acknowledgments

This work has been supported in part by CREST of JST(Japan Science and Technology Agency). And I am deeply grateful to Mr.Myunghoon Hong, Mr.Hitoshi Yanami, Mr.Hirokazu Anai and Prof. Shinji Hara for their cooperation.

A Appendix

In this section, we show the detailed usage of parametric robust control design toolbox.

A.1 How to use the toolbox

A.1.1 Basic function

Current version of parametric robust control design toolbox has four windows:

- **Main window** (Fig.8(1))
  has some edit field, controller, plant, specifications, and parameters.
- **Control synthesis window** (Fig.8(2))
  shows Bode diagram (for sensitivity function and complementary sensitivity function) and Nyquist plot and Pole/Zero Location for closed-loop transfer function, and users change specifications to manipulate graphic objects on the control synthesis window.
- **Open-loop window** (Fig.8(3))
  shows Bode diagram and Pole/Zero Location for open-loop transfer function.
- **Parameter space window** (Fig.8(4))shows the regions of feasible parameters that satisfy specifications.

  Basic operating procedure is shown as follows:

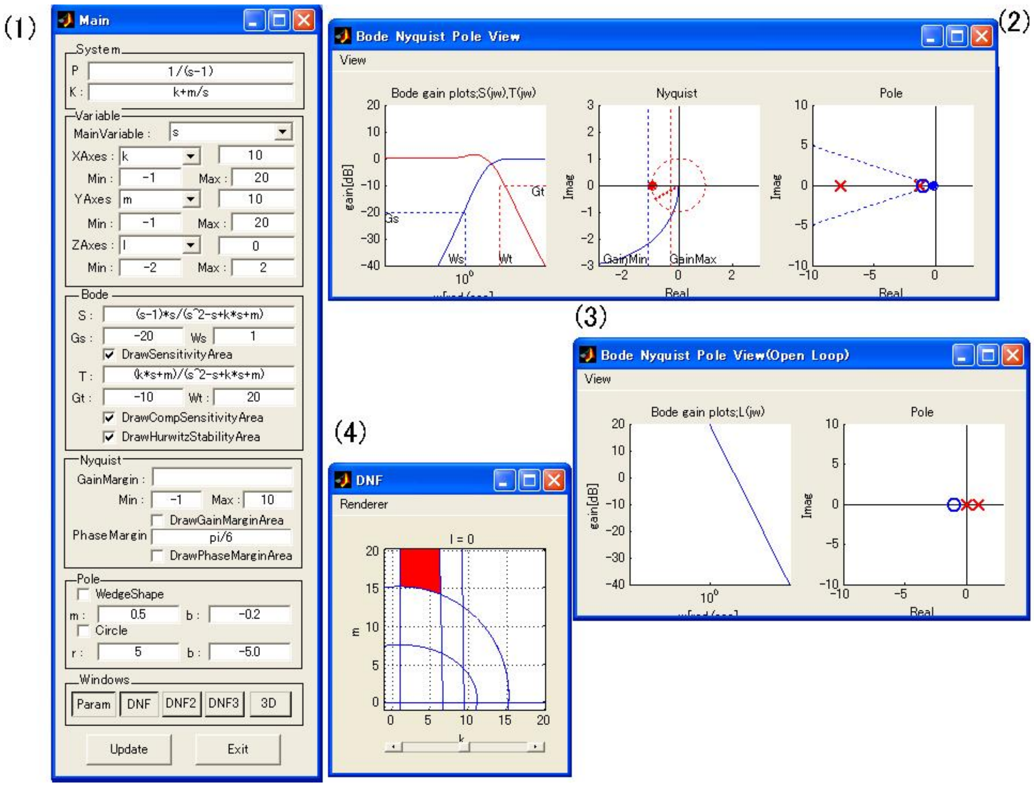

Fig. 8: Snapshot of parametric robust control toolbox

1. Select and push the “Windiws" buttons located at bottom of Main window which you want to show on main window.
2. Determine the structure of the controller and plant, and type the controller and plant into the boxes marked “System" at the top of main window.
3. Select the specifications which you want to investigate, check the boxes corresponding to your selection, and specify/input the concrete specifications.
4. Push the “Update" button, you can get feasible regions.

In Step 2, plant and controller structure can be specified with main window “System". In Step 3, users can select a number of specifications. And there are two ways to set specifications, users can use both input methods. One method is to edit textbox on main window, and another method is to move graphic objects on control synthesis window by using a mouse. In Step 4, feasible regions are computed and shown in red on parameter space window. If user select multiple specifications, the toolbox superposes each feasible region which satisfy each specifications. Fig.8(4) is shown a superposition of feasible regions of sensitivity, complementary sensitivity and Hurwitz stability.

If users click or drag with the mouse on parameter space window, parameter values are reflected in system, users can look see the behavior of bode diagram and nyquist plot and pole/zero location on control synthesis window.

A.2 Description of window and how to use each window

A.2.1 Main window

Fig.8(1) is main window of parametric robust control toolbox. Users can do as follows on main window.

- Input plant and controller functions.
- Select specifications.
- Select windows that users want look see.

A.2.2 System section

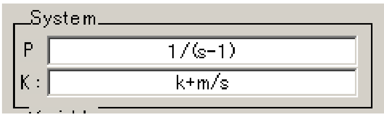

Fig. 9: System section

System section(Fig.9) is for inputting plant and controller functions. Users can use variable s and parameters k, m, l.

A.2.3 Variable section

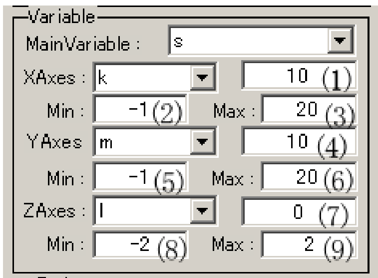

Fig. 10: Variable section

Variable section(Fig.10) is for assign variables and parameter area, and show/assign parameter values that are selected. This toolbox show feasible areas with k on x-axis, m on y-axis and l on z-axis.

1. Value of parameter k that is specified now on the parameter space window. This value is editable by not only keyboard input but also to click or drag on the parameter space window.
2. Minimum number of area of x-axis of parameter space window.

3. Maximum number of area of x-axis of parameter space window.
4. Value of parameter m that is specified now on the parameter space window. This value is editable by not only keyboard input but also to click or drag on the parameter space window.
5. Minimum number of area of y-axis of parameter space window.
6. Maximum number of area of y-axis of parameter space window.
7. Value of parameter l that is specified now on the parameter space window. This value is editable by not only keyboard input but also to click or drag on the parameter space window.
8. Minimum number of area of z-axis of parameter space window.
9. Maximum number of area of z-axis of parameter space window.

A.2.4 Bode section

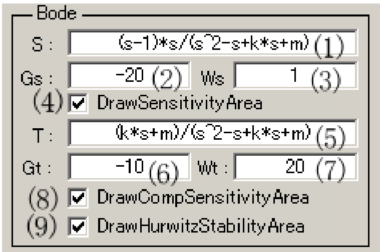

Fig. 11: Bode section

Bode section(Fig.11) is for input/show information that is shown on bode diagram. Descriptions of each items follow.

1. Sensitivity function $S(s)$ which is designed. Users can not change this section because it is computed automatically.
2. Gain constraint γ_s which is applied to sensitivity function $S(s)$. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
3. Max value of frequency ω_s which is applied to sensitivity function $S(s)$. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
4. Check this checkbox if you want to get parameter feasible region of sensitivity function.
5. Complementary sensitivity function $T(s)$ which is designed. Users can not change this section because it is computed automatically.
6. Gain constraint γ_t which is applied to complementary sensitivity function $T(s)$. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).

7. Minimum value of frequency ω_t which is applied to complementary sensitivity function $T(s)$. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
8. Check this checkbox if you want to get parameter feasible region of complementary sensitivity function.
9. Check this checkbox if you want to get parameter feasible region of Hurwitz stability of control synthesis.

A.2.5 Nyquist section

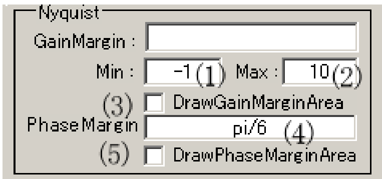

Fig. 12: Nyquist section

Nyquist section(Fig.12) is for input/show information that is shown on nyquist plot. Descriptions of each items follow.

1. Minimum value of gain margin which is applied to the control synthesis. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
2. Maximum value of gain margin which is applied to the control synthesis. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
3. Check this checkbox if you want to get parameter feasible region of gain margin of control synthesis.
4. Value of phase margin which is applied to the control synthesis. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).
5. Check this checkbox if you want to get parameter feasible region of phase margin of control synthesis.

A.2.6 Pole section

Pole section(Fig.13) is for input/show information that is shown on pole/zero location. Descriptions of each items follow.

1. Check this checkbox if you want to get parameter feasible region that satisfies pole assignment of wedge shape.
2. Slope of the WedgeShape. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).

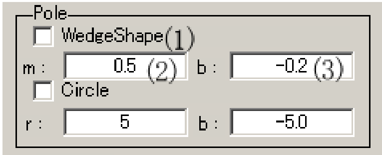

Fig. 13: Pole section

3. Distance from origin to WedgeShape. This value is editable by not only keyboard input but also to operate GUI object on the control synthesis window(Fig.8(2)).

A.2.7 Windows section

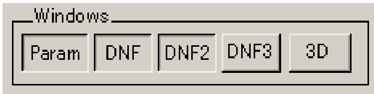

Fig. 14: Windows section

Windows section is for opening the windows. “Param" button opens the control synthesis window. “DNF" button opens the parameter space window. “DNF2" button opens the approximate feasible parameter regions window. “DNF3" button opens the fast parameter space window that draws by symbolic computation and evaluation of sample point. “3D" button opens the 3-D view window.

A.2.8 Buttons

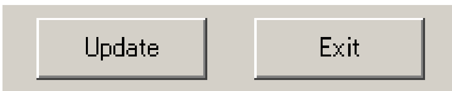

Fig. 15: Buttons

Those buttons are used to operate the “Main" window. If “Update" button is pushed, this toolbox compute the specifications. “Exit" button is for closing the parametric robust control design toolbox.

A.2.9 Control synthesis window

Fig.8(2) is the Control synthesis window. Users can set specifications by operating GUI objects, and can look see the behavior of control synthesis.

A.2.10 Bode diagram

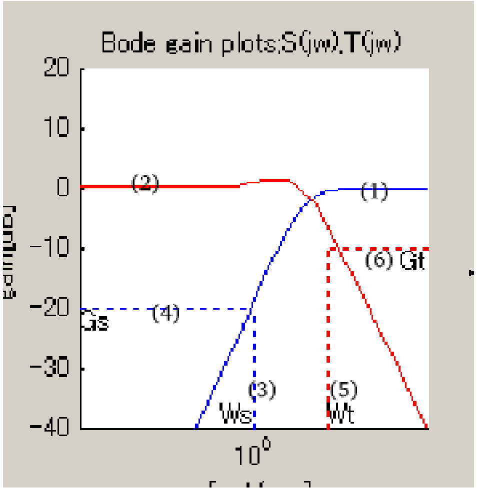

Fig. 16: Bode diagram

Bode diagram object is for setting the norm constraint of sensitivity and complementary sensitivity function, and show those bode diagrams. Sensitivity function and related specification is drawn in blue, and complementary sensitivity function and related specification is drawn in red. Description of each object are follow.

1. Bode diagram of sensitivity function $S(s)$.
2. Bode diagram of complementary sensitivity function $T(s)$.
3. Max value of frequency ω_s which is applied to sensitivity function $S(s)$. This object is movable with mouse dragging.
4. Gain constraint γ_s which is applied to sensitivity function $S(s)$. This object is movable with mouse dragging.
5. Gain constraint γ_t which is applied to complementary sensitivity function $T(s)$. This object is movable with mouse dragging.
6. Gain constraint γ_t which is applied to complementary sensitivity function $T(s)$. This object is movable with mouse dragging.

A.2.11 Nyquist plot

Nyquist plot(Fig.17) shows nyquist locus of control synthesis and users can set gain/phase margin on this window.

1. Locus of open-loop transfer function $L(j\omega)(0 < \omega)$ on complex number plane.
2. Minimum value of gain margin. This object is movable with mouse dragging.

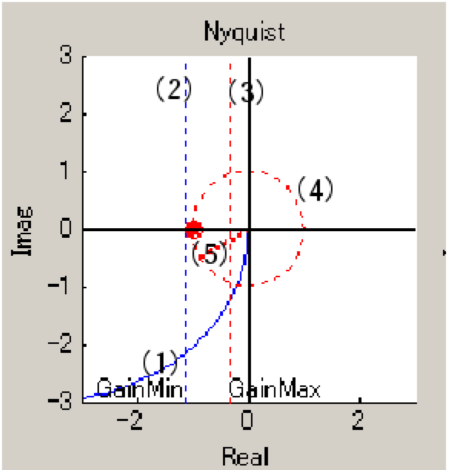

Fig. 17: Nyquist plot

3. Maximum value of gain margin. This object is movable with mouse dragging.
4. Unit circle.
5. Minimum value of phase margin. This object is movable with mouse dragging.

A.2.12 Pole/Zero location

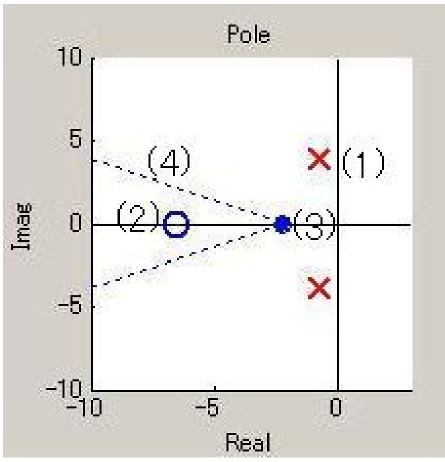

Fig. 18: Pole assignment

Pole assignment(Fig.18) shows behaviors of pole and zero of control synthesis on complex number plane.

1. Pole of control synthesis. If pole are in right half plane, control synthesis become unstable.
2. Zero of control synthesis. It negate pole of control synthesis.
3. Distance from origin to WedgeShape. This object is movable with mouse dragging.
4. Slope of the WedgeShape. This object is movable with mouse dragging.

A.2.13 Parameter space window

Parameter space window(Fig.19) shows feasible regions which satisfy the specifications that selected on the main window.

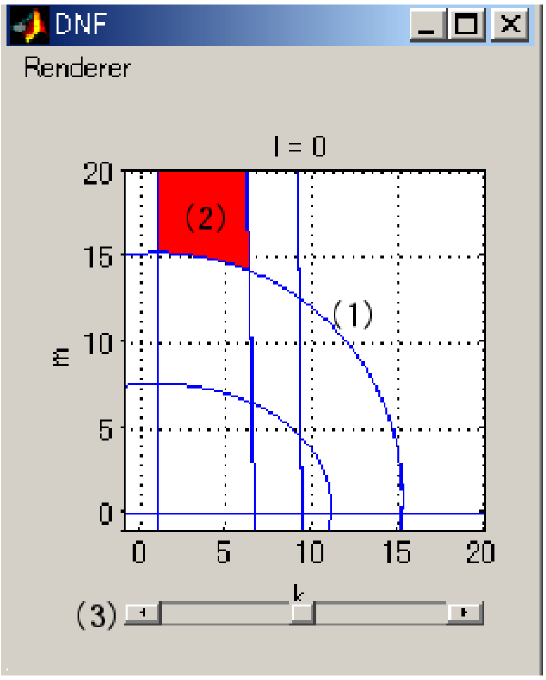

Fig. 19: Parameter space window

1. Border lines from specifications which are selected by user on main window.
2. Feasible parameter region which satisfy all selected specifications. This region colors in red.
3. Slide bar for changing parameter value which is interfaced with 3D view. This object is movable with mouse clicking/dragging.

If users click or drag with the mouse on parameter region window, parameter values are reflected in system, users can look see the behavior of control synthesis on control synthesis window.

A.2.14 Optional functions

A.2.15 3-D view

If controller has 3 parameters, *e.g.*,PID-controller, the toolbox displays 3-D figure and cross-sectional views of feasible regions. The Fig.20 is an example of 3-D view and Fig.21 is cross-sectional views at each axis of all feasible parameters. Users can use this function to put “3D" button on main window. 3-D view window has some checkboxes, users can look see following

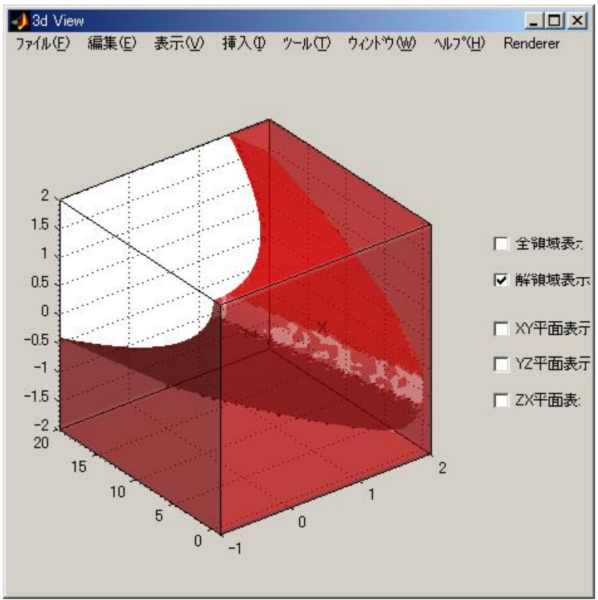

Fig. 20: 3-D view window

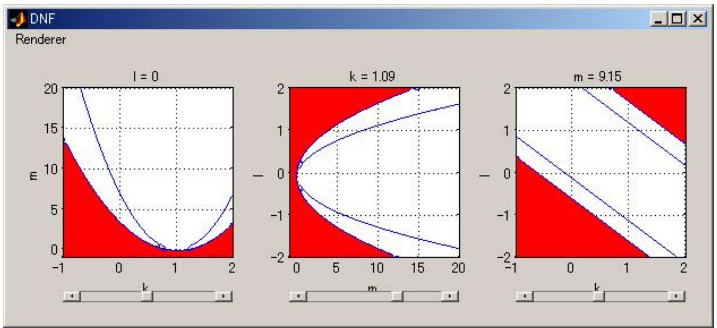

Fig. 21: parameter space window

functions to check the boxes.

1. show whole region

2. show only feasible region
3. show XY plane
4. show YZ plane
5. show ZX plane

And “Tools" which is menu of 3-D view window has some functions that operate 3-D objects, “Zoom in", “Zoom out", “3-D rotation", “Remove camera", etc. As in the case of 2 parameters system, users can click or drag with the mouse on parameter region window(right side of 20), parameter values are reflected in system, users can look see the behavior of bode and nyquist diagram and pole assignment on parameter window. Cross-sectional views are movable by slide bar.

A.2.16 Approximate feasible parameter regions

Parametric robust control design toolbox us exact feasible regions of parameters since it solves control design problems by using QE. But QE-based approach is expensive and can not solve large size problems in a reasonable amount of time. Hence, in such cases, it would be reasonable to employ a numerical method to obtain an approximate result. Here we employ a *validated numerical method* to solve a SDC approximately (but with guarantee) using interval arithmetics. Users can use this function to put “DNF2" button on main window.

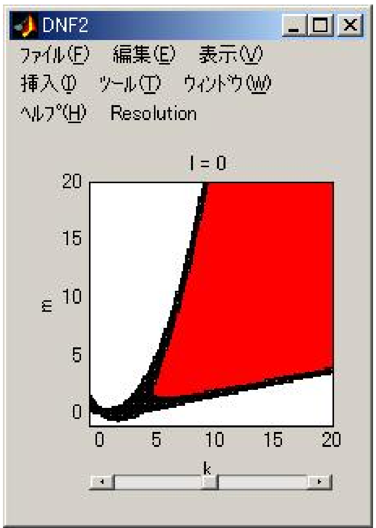

Fig. 22: Approximate feasible parameter regions

Users can select a drawing resolution “low", “mid", “high" from “Resolution" menu. Same as parameter space window, red area satisfy the selected specifications, and white area does not satisfy the selected specifications. The area that painted in black is unclear area that satisfy or not satisfy the specification under selected resolution. And the function that users can look see the behavior of control synthesis is not implemented in this function.

References

[1] H. Anai and S. Hara. Fixed-structure robust controller synthesis based on sign definite condition by a special quantifier elimination. In *Proceedings of American Control Conference 2000*, pages 1312–1316, 2000.

[2] H. Anai, H. Yanami, K. Sakabe, and S. Hara, "Fixed-structure robust controller synthesis based on symbolic-numeric computation: design and algorithms with a CACSD toolbox," Proceedings of CCA/ISIC/CACSD 2004 (Taipei, Taiwan), pp 1540–1545, 2004.

[3] L. Gonzàlez, H. Lombardi, T. Recio, and M.-F. Roy. Sturm-Habicht sequence. In *Proceedings of the ACM SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, pages 136–146. ACM-Press, 1989.

[4] L. González-Vega, T. Recio, H. Lombardi, and M.-F. Roy. Sturm-Habicht sequences determinants and real roots of univariate polynomials. In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 300–316. Springer, Wien, New York, 1998.

[5] K. Sakabe, H. Yanami, H. Anai and S. Hara, "MATLAB toolbox for robust control synthesis by symbolic computation" *Proceedings of SICE Annual Conference 2004 (Sapporo, Japan)*, pp. 1968–1973, 2004.

[6] S.Hara, T.Kimura and R.Kondo, "Parameter space design for H_∞ control(in Japanese)," *Trans. of SICE*, vol.27,no.6,pp. 714–716, 1991.

[7] S.Hara, T.Kimura and R.Kondo. "H_∞ control system design by a parameter space approach" *In Proceedings of MTNS-91(Koube, Japan)*, pp. 287–292, 1991.

[8] T.Kimura and S.Hara. "A Robust Control System Design by a Parameter Space Approach Based on Sign Definite Condition" *In Proceedings of Korean Automatic Control Conference(KACC)*, pp. 1533–1538, 1991.

[9] H.Anai and H.Yanami. "SyNRAC: A Maple-package for solving real algebraic constraints" *In Proceedings of International Workshop on Computer Algebra Systems and their Applications(CASA) 2003 (Saint Petersburg, Russian Federation), P.M.A. Sloot et al. (Eds.):ICCS 2003, LNCS2657.*Springer, pp. 828–837, 2003.

[10] Noriko Hyodo, Myunghoon Hong, Hitoshi Yanami, Hirokazu Anai, Shinji Hara. "Development of a MATLAB toolbox for parametric robust control — new algorithms and functions —" *In Proceedings of SICE Annual Conference 2006 (Busan, Korea)*, pp. 2856–2861, 2006.