Computer Algebra for Exact Complex Stability Margin Computation

Nainn-Ping Ke*

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2563, U.S.A.
email: nke@eudoxus.usc.edu

Abstract

As previous results, multivariable stability margin (k_M) problem can be formulated as solving polynomial systems by using symbolic computation and stratified Morse theory. Once the solutions are found, the stability margin problem can be easily solved. For complex k_M problem, no matter how many uncertainties, there is only one one-dimensional polynomial system which needs to be solved in order to find all singularities to determine whether the boundary of Horowitz template intercept the origin or not. The objective of this paper is to describe how to use Groebner Basis method to solve this polynomial system. Due to the continuity property of complex μ, numerical solutions are good enough for complex μ computation. In addition, we can sample this one-dimensional polynomial system into several zero-dimensional polynomial systems. There are many efficient algorithm to solve these zero-dimensional polynomial systems. Therefore, we have an efficient way of singularity related method to compute exact complex k_M.

keywords: Groebner basis, symbolic computation, robustness, stability.

1 Introduction

Symbolic Computation or Computer Algebra can be defined as computation with symbols representing mathematical objects and with numbers [11, 17]. Computer Algebra Systems allows us to do programming in Symbolic computation environment, e.g., Maple, Mathematica, Macsyma, Axiom, Reduce, etc. In particular, symbolic computation contains a lot of algorithms and methods for dealing with multivariate polynomials and corresponding polynomial systems. In addition, Computer Algebra Systems have been applied to solve various control problems in recent years [12, 25]. The most notable control application of symbolic computation is to solve nonlinear control problems [12, 25]. Another known control application of symbolic computation is applied Quantifier Elimination theory to do multi-objective feedback design [13].

Groebner Basis method was introduced by Bruno Buchberger in 1965 [1, 5, 7, 10, 29]. Given a finite set of multivariate polynomials over a field, a new set of polynomials can be found by Buchberger's algorithm, called a Groebner Basis, which will be used to obtain the solutions of this polynomial system. This method has been extensively studied, developed and has been implemented on many Computer Algebra Systems.

Robust control refers to the control of a plant with significant uncertainty by using a fixed controller and is a main branch of modern control theory [4, 6, 18]. Stability robustness analysis is a basic and fundamental topic in robust control. Robustness margin, k_M, is a measure of the stability analysis of a feedback system. However, it is a known result that robustness margin problem is NP-hard. Thus, pursuing the exact calculation of stability robustness margin is a tough job for control engineers. Grid method is a popular way to compute k_M which be considered as "Exact" computation [6]. However, the dimension of the uncertainty space in robust stability analysis is normally high, for example, 20. If we choose 20 sampling points in each uncertainty, it means that 20^{20} evaluations must be made. This task is beyond the ability of any current supercomputer. Therefore, the motivation of this research is to find an alternative way to compute k_M without using the grid method and it has the same accuracy.

Safonov and Athans introduced multivariable stability margin (k_M) [26]. On the other hand, a framework for dealing with structured uncertainties is the "structured singular value" μ which was introduced by Doyle [14]. μ is the reciprocal of k_M. The standard method is to use the convex hull of the template to compute k_M instead of the template. Many methods related to convexity have been proposed to compute the upper bound of complex μ [9, 14]. Moreover, many Toolboxes use with MATLAB to compute the upper bound of complex μ in numerical environment also available [3, 9]. However, some examples have been used to show the results of these methods could be too conservative. On the contrary, our method is to use the boundary of the template to compute k_M, not the convex hull [19, 20, 21, 22]. Hence the results of our method could

*Recently, the author joined Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109-8099.

be very accurate if we can find the solutions of the corresponding polynomial system [23].

As previous discussion, grid method is hard to apply to compute complex μ with many uncertainties. The main goal of this research is to find an efficient way to compute accurate complex μ and do not have to wait forever like grid method do for a typical complex μ problem. Real k_m problem has been considered as a solving polynomial system problem by Ke [21]. Complex k_m problem has first been reformulated as a solving a set of algebraic equations by Ke [22]. Later, this result was extended to a set of polynomial system [23]. However, this polynomial system is one-dimensional. To solve this polynomial system is a tough job. Fortunately, if we want to have the same accuracy as grid method, we can grid one of the equations and obtain several zero-dimensional polynomial systems. Because the corresponding polynomial systems are zero-dimensional, several existing algorithms provide efficient ways to compute the solutions. Hence, we have an efficient way of singularity related method to compute exact complex μ. We will use two examples to demonstrate how to apply the methods and algorithms of Groebner Basis for complex μ computation.

2 Problem Formulation

A standard stability margin problem is shown in Figure 1. Let $f : \mathbb{D} \times \Omega \ni (q, \omega) \longmapsto det(I + L(j\omega)\Delta) \in \mathbb{C}$ be Nyquist Map, where $\mathbb{D}$ is the space of uncertain parameters (e.g., the Kharitonov cube or a polydisc) and Ω is frequency spectrum. $L(j\omega) \in \mathbb{C}^{n\times n}$ is the stable loop transfer matrix. q is the vector of parametric uncertainty which can be real or complex. The Horowitz template is $\mathcal{N}_\omega = f(\mathbb{D}, \omega)$ [18, 19, 20, 21]. The maximum uncertainty bound, k_M, such that the system is stable for all $q \in \mathbb{D}$ with $||q||_\infty \leq k_M$ is called the robustness margin of a control system. In the above $||q||_\infty \stackrel{\text{def}}{=} \max_i\{| q_i |\}$ where $| q_i |$ denotes the absolute value of q_i for real uncertainties and the modulus of q_i for complex uncertainties. If we fix frequency, the k_M problem becomes to find the smallest solutions, $||\Delta||$, of $f_\omega = det(I + L\Delta) = 0$ which is a polynomial equation of unknowns. Intuitive, k_M is achieved when the boundary of the template intercepts $0 + j0$. Hence, we start from a small k to check if its image cover the origin or not. If yes, we set $k_M = k$. If not, we increase size k and check again. Figure 2 attempts to depict this method.

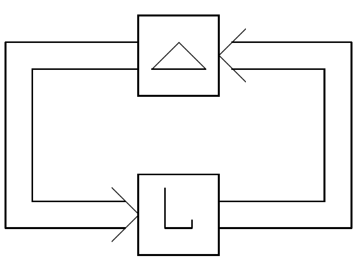

Figure 1: A standard robust stability problem

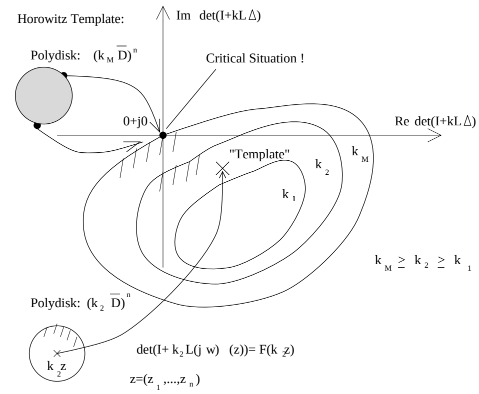

Figure 2: Horowitz template approach

3 Mathematical Background

As a subset of N of $\mathbb{R}^{\mathbb{K}}$, ∂N, the boundary of N, is defined as $\partial N = \bar{N} - Int(N)$. The closure of N, $\bar{N}$, is the intersection of all closed sets of $\mathbb{R}^{\mathbb{N}}$ covering N, while the interior of N, $int(N)$, is the union of all open sets contained in N.

A critical point q^* of a smooth map f_ω defined over a smooth manifold is a point where the rank of the Jacobian drops [18, 21]. Therefore, the number of real constraints for a smooth critical point in the case of n real uncertainties is equal to the number of the real variables minus one (n-1). The boundary of the Horowitz template is contained in the image of the singularity set.

A zero-dimensional polynomial system has a finite number of roots [1]. The solutions of one-dimensional polynomial system is topological equivalent to a curve. For complex k_M computation, our corresponding polynomial system is one-dimensional. To solve one-dimensional polynomial system is not a easy job.

Therefore, we have to transform it to zero-dimensional polynomial systems in order to find the solutions.

4 Groebner Basis Method

Groebner basis method is a major technique to compute the solutions of polynomial systems [1, 5, 7, 10, 11, 17]. Given a finite set of multivariate polynomials over a field, a new set of polynomials can be found by Buchberger's algorithm, called a Groebner basis, which will be used to obtain the solution of the polynomial system. This method has been implemented on many Computer Algebra Systems, e.g., Mathematica, Maple, Axiom, Reduce, Macsyma, etc. Before to compute Groebner basis, we have to set an admissible term ordering first. There are many different admissible term ordering, e.g. lexicographic order, total order, degree lexicographic order, degree reverse lexicographic order.

The Groebner basis has the following properties:

1. Every polynomial ideal has a Groebner basis.
2. Given a set of polynomials, the Groebner basis of this polynomial set still has the same solutions.
3. Fixing a term ordering, every non-zero ideal has a unique reduced Groebner basis.

The way to apply Groebner basis method is as follows. First, find the Groebner basis with lexicographic order. The Groebner basis with lexicographic order can be easily used to compute the solutions of the polynomial system due to "elimination property". The use of the elimination property is as follows: There is a univariate polynomial in the Groebner basis with lexicographic order. Hence, we can easily solve this univariate polynomial. Then, by substituting the solutions to the Groebner basis, we obtain other univariate polynomials. By the same procedure, we can obtain all solutions of the original polynomial system. This is similar to Gauss's elimination method in the linear cases. Therefore, if we have the Groebner basis with lexicographic order, we can easily obtain the solutions of the polynomial system. However, the complexity of Groebner basis of polynomial system of degree d in n variables with lexicographic order is much higher than other ordering. When the number of solution is finite, the complexity is $d^{\mathcal{O}(n^3)}$. From the complexity point of view, the best ordering is the degree reverse lexicographic ordering. The computation of the Groebner basis by this ordering of a polynomial system is d^{n^2}. This complexity will decrease to d^n if the solutions at infinity are finite in number as well. Hence, there are several methods to compute Groebner basis with lexicographic ordering by using basis conversion to save the computation time [15]. Therere, we want first to compute a Groebner basis with other ordering, and then to convert this basis to a Groebner basis with lexicographic order. How to obtain a Groebner basis with lexicographic order by basis conversion is under active development. FGLM algorithm which is known basis conversion algorithm has been implemented in Maple [15]. FGLM only can be applied on zero-dimensional polynomial system. Groebner Walk algorithm, another basis conversion algorithm, has implemented in Mathematica [7, 28].

For complex μ computation, we are dealing with zero-dimensional polynomial systems. In addition, complex μ is continuous [19, 20]. Hence, the numerical solutions are accurate enough for complex μ computation. The computation time for constructing the corresponding Groebner basis with lexicographic order is normally fast. For Example 2, the computation time to obtain a Groebner basis is less than one second in Mathematica [8].

5 Complex k_M Computation

Uncertainty space in k_M problem is non-manifold [19, 20, 22, 23]. Due to the difficulty in dealing with the non-manifold, the concept of stratified spaces is preferred. The idea for searching the singularity of the complex uncertainty space is as follows: Complex uncertainty space for 2-uncertainties is similar to a solid torus which is a manifold with boundary. It can be decomposed into two smooth manifolds, i.e., solid torus $=$ open torus $\cup$ torus . Hence, the Morse theory can be applied on both manifolds for singularities. In addition, due to lemma 1 of 1982 Doyle's paper or Open Mapping Theorem, the image of the distinguished boundary is first to intercept the origin not the open torus [14, 19]. Therefore, only the singularities on the distinguished boundary, i.e., $\mathbb{T}^2$, are useful for k_M computation. We can used the **polar form** to represent the boundary of any size of polydisc with $z_i = ke^{j\phi_i}$, where k is the size of the polydisc, a fixed number. For example, in case of 2 complex uncertainties; the distinguished boundary of the polydisc, $S^1 \times S^1$, is a 2-dimensional manifold which can be expressed as $f_\omega^\phi = f_\omega(\phi_1, \phi_2)$. The Jacobian becomes

$$J^h_{f_\omega^\phi} = \begin{pmatrix} \frac{\partial \Re f_\omega^\phi}{\partial \phi_1} & \frac{\partial \Re f_\omega^\phi}{\partial \phi_2} \\ \frac{\partial \Im f_\omega^\phi}{\partial \phi_1} & \frac{\partial \Im f_\omega^\phi}{\partial \phi_2} \end{pmatrix}.$$

The constraint for the critical points is $\frac{\partial \Re f_\omega^\phi}{\partial \phi_1}\frac{\partial \Im f_\omega^\phi}{\partial \phi_2} - \frac{\partial \Im f_\omega^\phi}{\partial \phi_1}\frac{\partial \Re f_\omega^\phi}{\partial \phi_2} = 0$. For the general case, i.e. m uncertainties: $f_\omega^\phi = f_\omega(\phi_1, \phi_2, \cdots, \phi_m)$ The Jacobian becomes

$$J^h_{f_\omega^\phi} = \begin{pmatrix} \frac{\partial \Re f_\omega^\phi}{\partial \phi_1} & \frac{\partial \Re f_\omega^\phi}{\partial \phi_2} & \cdots & \frac{\partial \Re f_\omega^\phi}{\partial \phi_m} \\ \frac{\partial \Im f_\omega^\phi}{\partial \phi_1} & \frac{\partial \Im f_\omega^\phi}{\partial \phi_2} & \cdots & \frac{\partial \Im f_\omega^\phi}{\partial \phi_m} \end{pmatrix} \tag{1}$$

From the definition of critical point, the constraints are

$$\left.\begin{matrix} \frac{\partial \Re f_\omega^\phi}{\partial \phi_1}\frac{\Im f_\omega^\phi}{\partial \phi_2} - \frac{\partial \Im f_\omega^\phi}{\partial \phi_1}\frac{\Re f_\omega^\phi}{\partial \phi_2} = 0 \\ \cdots \\ \frac{\partial \Re f_\omega^\phi}{\partial \phi_1}\frac{\Im f_\omega^\phi}{\partial \phi_m} - \frac{\partial \Im f_\omega^\phi}{\partial \phi_1}\frac{\Re f_\omega^\phi}{\partial \phi_m} = 0 \end{matrix}\right\} \tag{2}$$

Therefore, the number of constraints for critical points in the case of m uncertainties is equal to $m-1$. Thus, no matter how many uncertainties, there is only one polynomial system needs to be solved.

6 An Algorithm

In Figure 2, we already sketch an algorithm for complex k_M computation. Now, we apply the results of previous sections to obtain an algorithm for compute k_M as follows:

Input f_ω.

Output k_M.

Initialization Given initial k.

1. Substitute z_i by polar form $(ke^{j\phi_i})$ to represent the boundary of polydisc.
2. Differentiate Nyquist map by ϕ_i to obtain Jacobian.
3. Construct the corresponding polynomial system by using equation 2.
4. Find the solutions of the polynomial system.
5. Use the Nyquist map to find the boundary of the Horowitz template. If the boundary of the template intercepts the origin, $k_M = k$ and stop. Otherwise, increase k and go to step 3.

In this algorithm, how to solving the polynomial systems are the major task. Once we find all real solutions of the polynomial systems, k_M can be computed easily.

7 Examples and Discussion

The following examples are used to demonstrate how to formulate k_M problems as solving polynomial systems by using Symbolic Computation. In addition, we will show how to construct the Groebner basis. These examples has been implemented in Mathematica and Maple.

Example 1 *Consider the Nyquist Map with complex uncertainties for fixed frequency* $f_\omega = z_1 * z_2 + z1 + 2$. *From the previous discussion, this problem can be reformulated as* $f_\omega^\phi = e^{j\,\phi_1} * e^{j\,\phi_2} + e^{j\,\phi_1} + 2$, *where* $e^{j\,\phi} = cos(\phi) + j * sin(\phi)$. *We can use function "diff" to differentiate* f_ω^ϕ *in Maple to obtain the Jacobian:*

$$J = \begin{pmatrix} -sin(\phi_1+\phi_2) - sin\phi_1 & -sin(\phi_1+\phi_2) \\ cos(\phi_1+\phi_2) + cos\phi_1 & cos(\phi_1+\phi_2) \end{pmatrix}$$

Hence, the constraint is $-sin\phi_1 * cos(\phi_1+\phi_2) + sin(\phi_1+\phi_2) * cos\phi_1 = 0$. *In addition, the hidden constraints are* $sin\phi_1^2 + cos_1^2 = 1$ *and* $sin\phi_2^2 + cos\phi_2^2 = 1$ *We can easily use function "solve" to solve this equation in MATLAB with Symbolic Math Toolbox. However, for more uncertainties, we are not able to solve this kind of the trigonometric equations. Hence, we should construct the corresponding polynomial system as follows: Use function "subs" in Maple to substitute* $sin\phi_1$ *by* s_1 *and substitute* $cos\phi_1$ *by* c_1. *Use the similar substitutions for other trigonometric functions. Then, the corresponding polynomial system becomes*

$$\left.\begin{array}{c} -s_1 * (c_1 * c_2 - s_1 * s_2) + (s_1 * c_2 + c_1 * s_2) * c_1 = 0 \\ s_1^2 + c_1^2 = 1 \\ s_2^2 + c_2^2 = 1 \end{array}\right\} \quad (3)$$

This is a one-dimensional polynomial system. We can grid on one equation and get zero-polynomial systems. For example, we can set s_1 *as 0 and* c_1 *as 1 to eliminate second equation of equation 3. Hence, we obtain the constraints, .* $s_2 = 0$ *and* $s_2^2 + c_2^2 = 1$. *This is a zero-dimensional polynomial systems. Similarly, we sample the circle* $s_1^2 + c_1^2 = 1$ *and obtain the corresponding polynomial systems. The simulation results for the template and the boundary are shown in 3 and 4, respectively.*

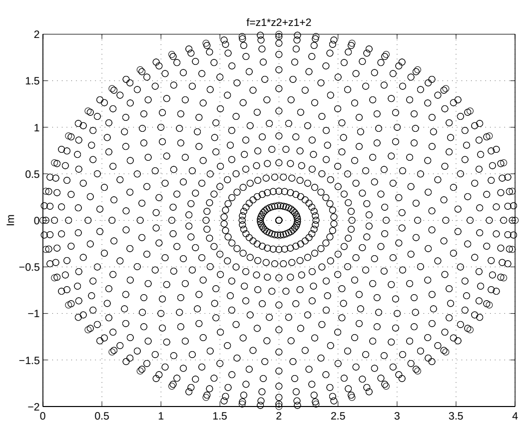

Figure 3: Template of fw=z1*z2+z1+2

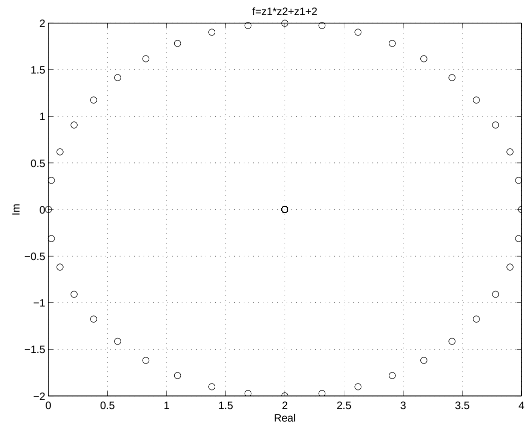

Figure 4: Boundary of f=z1*z2+z1+2

Exmaple 2 *Consider the Nyquist map* $f_\omega = (1+j) * z_1 * z_2 * z_3 + z_1 + z_2 + z_3 + 4$. *where* $\mid z_i \mid \le 1$. *This problem can be reformulated as* $f_\omega^\phi = (1+j) * e^{j\,\phi_1} * e^{j\,\phi_2} * e^{j\,\phi_3} + e^{j\,\phi_1} + e^{j\,\phi_2} + e^{j\,\phi_3} + 4$. *Then, the constraints are* $cond_1(\phi_1,\phi_2,\phi_3) = \frac{\partial \Re f_\omega}{\partial \phi_1} * \frac{\partial \Im f_\omega}{\partial \phi_2} - \frac{\partial \Re f_\omega}{\partial \phi_2} * \frac{\partial \Im f_\omega}{\partial \phi_1} = 0$ *and* $cond_2(\phi_1,\phi_2,\phi_3) = \frac{\partial \Re f_\omega}{\partial \phi_1} * \frac{\partial \Im f_\omega}{\partial \phi_3} - \frac{\partial \Re f_\omega}{\partial \phi_3} * \frac{\partial \Im f_\omega}{\partial \phi_1} = 0$. *Following the same procedure of Example 1, we can find the constraint polynomial equations are*

$$\left.\begin{array}{l} (-s_1 * c_2 * c_3 + s_1 * s_2 * s_3 - c_1 * c_2 * s_3 - c_1 * s_2 * c_3 + \\ s_1 * c_2 * s_3 + s_1 * s_2 * c_3 - c_1 * c_2 * c_3 + c_1 * s_2 * s_3 - \\ s_1) * (-c_1 * s_2 * c_3 - c_1 * c_2 * s_3 + s_1 * s_2 * s_3 - \\ s_1 * c_2 * c_3 + c_2 - c_1 * s_2 * s_3 + c_1 * c_2 * c_3 - \\ s_1 * s_2 * c_3 - s_1 * c_2 * s_3) - (-c_1 * s_2 * c_3 - \\ c_1 * c_2 * s_3 + s_1 * s_2 * s_3 - s_1 * c_2 * c_3 + c_1 * s_2 * s_3 - \\ c_1 * c_2 * c_3 + s_1 * s_2 * c_3 + s_1 * c_2 * s_3 - s_2) * \\ (-s_1 * c_2 * c_3 + s_1 * s_2 * s_3 - c_1 * c_2 * s_3 - c_1 * s_2 * c_3 + \\ c_1 - s_1 * c_2 * s_3 - s_1 * s_2 * c_3 + c_1 * c_2 * c_3 \\ -c_1 * s_2 * s_3) = 0, \\ (-s_1 * c_2 * c_3 + s_1 * s_2 * s_3 - c_1 * c_2 * s_3 - c_1 * s_2 * c_3 + \\ s_1 * c_2 * s_3 + s_1 * s_2 * c_3 - c_1 * c_2 * c_3 + c_1 * s_2 * s_3 \\ -s_1) * (-c_1 * c_2 * s_3 - c_1 * s_2 * c_3 - s_1 * c_2 * c_3 \\ +s_1 * s_2 * s_3 + c_3 + c_1 * c_2 * c_3 - c_1 * s_2 * s_3 - \\ s_1 * c_2 * s_3 - s_1 * s_2 * c_3) - (-c_1 * c_2 * s_3 - c_1 * s_2 * c_3 \\ -s_1 * c_2 * c_3 + s_1 * s_2 * s_3 - c_1 * c_2 * c_3 + c_1 * s_2 * s_3 + \\ s_1 * c_2 * s_3 + s_1 * s_2 * c_3 - s_3) * (-s_1 * c_2 * c_3 \\ +s_1 * s_2 * s_3 - c_1 * c_2 * s_3 - c_1 * s_2 * c_3 + c_1 - s_1 * c_2 * \\ s_3 - s_1 * s_2 * c_3 + c_1 * c_2 * c_3 - c_1 * s_2 * s_3) = 0, \\ s_1^2 + c_1^2 = 1, \\ s_2^2 + c_2^2 = 1, \\ s_3^2 + c_3^2 = 1 \end{array}\right\} \quad (4)$$

If we can set $c_1 = 1$ *and* $s_1 = 0$, *we obtain the following zero-dimensional polynomial system:*

$$\left.\begin{array}{l} -s_2^2 * s_3 - s_2^2 * c_3 - s_2 * s_3 + s_2 * c_3 + s_2 \\ -c_2^2 * s_3 - c_2^2 * c_3 + c_2 * s_3 + c_2 * c_3 = 0, \\ -s_2 * s_3^2 - s_2 * s_3 - s_2 * c_3^2 + s_2 * c_3 \\ -c_2 * s_3^2 + c_2 * s_3 - c_2 * c_3^2 + c_2 * c_3 + s_3 = 0, \\ -1 + s_2^2 + c_2^2 = 0, \\ -1 + s_3^2 + c_3^2 = 0 \end{array}\right\} \quad (5)$$

The reduced Groebner basis method with lexicographic order $s_2 > c_2 > s_3 > c_3$ is

$$\left.\begin{array}{l} 64*s_2+32*c_2+760*c_3^4-124*c_3^3-793*c_3^2- \\ 2*c_3+127, \\ 5*c_2^2-2*c_2-5*c_3^2+2*c_3, \\ 8*c_2*c_3-8*c_2-200*c_3^4+100*c_3^3+167*c_3^2- \\ 42*c_3-25, \\ 64*s_3-124*c_3^3-793*c_3^2+30*c_3+127+ \\ 760*c_3^4, \\ -47*c_3^3-11*c_3^3+3+11*c_3+4*c_3^4+40*c_3^5 \end{array}\right\} \tag{6}$$

We can see this Groebner basis is in "triangular" form. Once this Groebner basis is obtain, the solutions can be easily found. If we set $c_1 = cos(\pi/8)$ and $s_1 = sin(\pi/8)$, we obtain another set of zero-dimensional polynomial. Once all solutions are found, we can use Eq. 2 to obtain the Horowitz template. The simulation results for the template and the boundary are shown in Figure 5 and Figure 6, respectively. From Figure 5 and Figure 6, we know the number of the solutions is much less than the number of grid points in grid method.

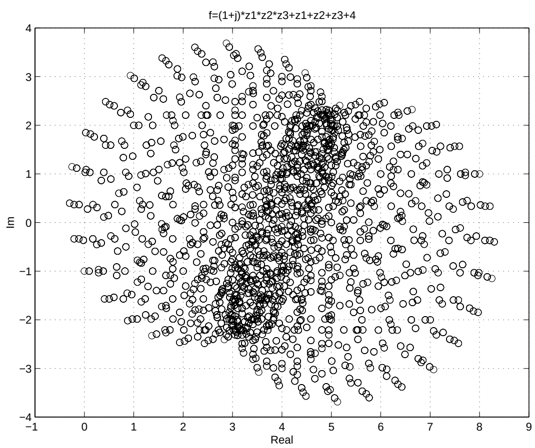

Figure 5: Template of fw=(1+j)z1z2z3+z1+z2+z3+4

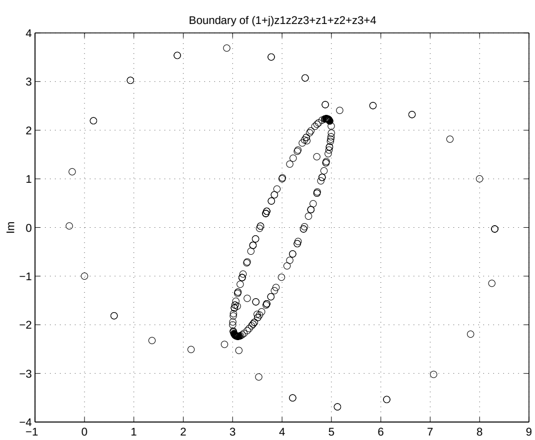

Figure 6: Boundary of fw=(1+j)z1z2z3+z1+z2+z3+4

From Example 1 and Example 2, we have an impression that we will face much larger polynomial systems if we add one more uncertainty and a few terms in Nyquist map. However, the reasons for us to expect that there will be a good algorithm to compute complex k_M are the followings:

1. Only several zero-dimensional polynomial systems need to be solved for complex k_M.
2. The solutions which are real, and Morse critical points, and lie between -1 and 1 are useful.

Based on the earlier discussion, it is hard to implement our approach to compute k_M under the pure numeric computing environment. The reasons are the followings:

1. It needs to partially differentiate the Nyquist map relative to q_i or ϕ_i.
2. It needs to change and/or substitute variables.
3. It needs to solve polynomial systems.

8 Conclusions

Hence, it has shown that the robustness margin problem can be formulated as solving polynomial systems by Symbolic Computation. The main task of this approach is to find the solutions which are the Morse critical points. However, due to the high dimension of the uncertainty space, for example, 20 uncertainties, which could be a real world problem, the corresponding polynomial system is huge, for example, 38 equations for 20 complex uncertainties. Fortunately, the number of the solutions in this case is finite. Hence, "Can we solve these kind of huge polynomial systems ?" and "What is the limit number of uncertainties which we can compute its exact complex k_M?" becomes a main issue of our future research. Beside Groebner basis method, homotopy continuation method will be another main techniques to tackle our problems [2, 24].

For a simple example as Example 2, we already face a large polynomial system. Hence, one could ask "Is this method too complicated to implement ?" However, from the theory of computation, robust stability problem can be reduced to solving polynomial system, that means robust stability problem is easier than solving polynomial system problem [16, 27]. Therefore, if we can solve the corresponding polynomial systems efficiently we can solve complex μ problem. Especially, we are dealing with zero-dimensional polynomial system which is considered as an easy job by some computer scientists and mathematician. In addition, we could use the boundary behavior to improve our method in the future. Moreover, due to the continuity property of complex μ, numerical solutions are good enough for complex μ computation. Hence, we have a good way to compute exact complex μ.

Acknowledgements

I am grateful to Prof. B. Buchberger, Research Institute for Symbolic Computation, Johannes Kepler University of Linz and Prof. Q.-N. Tran, Department of Computer Science, Lamar University for many invaluable discussions and suggestions. I also like to thank

Prof. E. A. Jonckheere and Prof. M. G. Safonov, EE-Systems, USC, for many helpful discussions and encouragement. In addition, I like to thank Prof. J. Doyle, CDS, Cal. Tech., for many useful comments.

References

[1] W. W. Adams and P. Loustaunau, *An Introduction to Groebner Basis*, AMS, 1994.

[2] E. L. Allgower and K. Georg, *Numerical Continuation Methods, an Introduction*, Springer, Berlin, 1990.

[3] G. Balas, J. C. Doyle, K. Glover, A. Packard and R. Smith, *μ-Analysis and Synthesis Toolbox*, MUSYN Inc. and The MathWorks, Inc, 1991.

[4] B. R. Barmish, *New Tools for Robustness of Linear System*, Macmillan, 1994.

[5] T. Becker and V. Weispfenning, *Groebner Basis - A Computational Approach to Commutative Algebra*, Springer-Verlag, 1993.

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

[7] B. Buchberger and F. Winkler, *Groebner Bases: Theory and Applications*, Cambridge University Press, 1998.

[8] B. Buchberger, private communication, 2000.

[9] R. Y. Chiang and M. G. Safonov, *Robust Control Toolbox*, The MathWorks, Inc., 1992.

[10] D. Cox, J. Little and D. O'Shea, *Ideal, Varieties, and Ideal - An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer-Verger, 1992.

[11] J. H. Davenport, Y. Siret and E. Tournier, *Computer Algebra - Systems and Algorithms for Algebraic Computation*, Academic Press, New York, 1993.

[12] B. de Jager, *The Use of Symbolic Computation in Nonlinear Control: Is It Viable?*, IEEE Trans. Automat. Contr., vol. 40, pp. 84–89, 1995.

[13] P. Y. Dorato, W. Yang and C. Abdallah, *Robust Multiobjective Feedback Design by Quantifier Elimination*, J. Symbolic Computation, **24**, pp. 153–159, 1997.

[14] J. C. Doyle, *Analysis of Feedback Systems with Structured Uncertainty*, IEE Proc., vol. 129, pp. 242–250, 1982.

[15] J. C. Faugére, P. Gianni, D. Lazard and T. Mora, *Efficient Computation of Zero-Dimensional Groebner Basis by Change of Ordering*, J. Symbolic Computation, **16**, pp. 329–334, 1993.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, W. H. Freeman, New York, 1979.

[17] K. O. Geddes, S. R. Czapor and G. Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, Boston, 1992.

[18] E. A. Jonckheere, *Algebraic and Differential Topology of Robust Stability*, Oxford University Press, Oxford, New York, 1997.

[19] E. A. Jonckheere and N. P. Ke, *Complex-Analytic Theory of the μ-Function*, Proc. American Control Control (ACC), pp.366-371, Albuquerque, New Mexico, 1997; also in J. Math. Anal. Appl. **237**, 201–239,1999.

[20] E. A. Jonckheere and N. P. Ke, *Real versus Complex Robustness Margin Continuity as a Smooth versus Holomorphic Singularity Problem*, Proc. 36th IEEE Conf. Decision and Control, pp. 3266–3271, San Diego, CA; also in J. Math. Anal. Appl. **237**, pp. 541–572,1999.

[21] N. P. Ke, *An Algorithm of Symbolic Computation for Stability Margin*, Proc. ACC , pp. 2304–2308, Philadelphia, Pennsylvania, 1998.

[22] N. P. Ke, *A New Method to Compute Complex Robustness Margin*, Proc. IFAC'99 World Congress, vol. G, pp. 37–42, Beijing, P. R. China, 1999.

[23] N. P. Ke, *Symbolic and Algebraic Computation in Robust Stability Analysis*, Booklet of Posters' Abstract, E. V. Zima and M. O. Rayes (eds.), pp. 27–30, ACM/SIGSAM International Symposium on Symbolic and Algebraic Computation, Vancouver, Canada, 1999.

[24] T. Y. Li, *Numerical Solution of Multivariate Polynomial Systems by Homotopy Continuation Methods*, Acta Numerica, 1997.

[25] N. Munro (ed.), *Symbolic Methods in Control System analysis and Design*, IEE, London, U. K., 1999.

[26] M. G. Safonov and M. Athans, *A Multiloop Generalization of the Circle Criterion for Stability Margin Analysis*, IEEE Trans. Automat. Control, AC-26, pp. 415–422, 1981.

[27] M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, Boston, 1997.

[28] Q.-N. Tran and F. Winkler (ed.), the special issue of the Journal of Symbolic Computation on Groebner Bases, to appear, 2000.

[29] F. Winkler, *Polynomial Algorithm in Computer Algebra*, Springer-Verlag, 1996.