SYNTHESIS OF CONTROLLERS FOR NON-MINIMUM PHASE AND UNSTABLE SYSTEMS USING NON-SEQUENTIAL MIMO QUANTITATIVE FEEDBACK THEORY

A Thesis

by

CHEN-YANG LAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2004

Major Subject: Mechanical Engineering

SYNTHESIS OF CONTROLLERS FOR NON-MINIMUM PHASE AND UNSTABLE SYSTEMS USING NON-SEQUENTIAL MIMO QUANTITATIVE FEEDBACK THEORY

A Thesis

by

CHEN-YANG LAN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Suhada Jayasuriya (Chair of Committee)	Alexander Parlos (Member)
Shankar P. Bhattacharyya (Member)	Dennis L. O'Neal (Interim Head of Department)

May 2004

Major Subject: Mechanical Engineering

ABSTRACT

Synthesis of Controllers for Non-minimum Phase and Unstable Systems Using Non-sequential MIMO Quantitative Feedback Theory. (May 2004)

Chen-yang Lan, B.S., National Tsing Hua University,
Hsinchu, Taiwan
Chair of Advisory Committee: Dr. Suhada Jayasuriya

Considered in this thesis is multi-input multi-output (MIMO) systems with non-minimum phase (NMP) zeros and unstable poles where some of the unstable poles are located to the right of the NMP zeros. In the single-input single-output (SISO) case such systems pose serious difficulties in controller synthesis for performance and stability. In spite of the added degrees of freedom, the MIMO case also poses difficulties as has been experienced in the stabilization of the X-29 aircraft. When using the MIMO QFT technique the synthesis of the multivariable problem starts by considering the diagonal entries, $(P^{-1})_{ii} = 1/q_{ii}$, derived from the plant transfer function matrix P that are used to develop a controller. In effect the design problem is reduced to several MISO designs with the diagonal entries serving as equivalent SISO plants. Developed in this thesis is a transformation scheme that can be used to condition the resulting equivalent SISO plants so that the difficult problem of NMP zeros lying to the left of unstable poles is avoided. It is accomplished by introducing two transformation matrices M , N so that a new plant $P_1 = M^{-1}PN$, and a controller $G_1 = N^{-1}GM$, where entries of both M , N can be either constants or polynomials of s . Thus, it is proposed that P_1 be used instead of the original plant P when carrying out a MIMO QFT design. For example, if we have unstable poles or NMP zeros in q_{ii} obtained from P we can define a new set $\hat{q}_{ii}$ where each $\hat{q}_{ii}$ has a desirable stable and/or minimum phase (MP) structure. All that one has to do then is, if feasible, to determine non-singular matrices M and N such that $P_1^{-1} = N^{-1}P^{-1}M$ and $(P_1^{-1})_{ii} = 1/\hat{q}_{ii}$. Examples illustrate the use of the proposed transformation.

To my father and mother.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor, Professor Suhada Jayasuriya, for his support and fruitful discussions during my course of study at Texas A&M University. It was a great opportunity for me to investigate an exciting control problem. His ability to solve basic engineering problems and his practical knowledge in dealing with the control systems motivated and guided me throughout my research activities.

I would also like to extend my thanks to my graduate committee members Dr. Alexander Parlos and Dr. Shankar P. Bhattacharyya for their interest and valuable guidance during the research.

My thanks go to Mr. Murray L. Kerr for sharing his precious experience in QFT and his valuable discussions. My thanks go also to all my other friends for their company which has enriched my graduate life.

I am thankful to my parents, Mr. Min-chia Lan and Mrs. Su-fen Kao for their immeasurable support throughout my life. Without their love, patience and support, I would have never come this far.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF SYMBOLS

SYMBOLS

s	Laplace domain variable
ω	Frequency
$P(s)$	Plant transfer function
P_{ij}	Entries of $P(s)$
z_{ij}	Numerator of P_{ij}
p_{ij}	Denominator of P_{ij}
P_0	Nominal plant
$F(s)$	Prefilter
$G(s)$	Controller
g_i	Diagonal entries of a controller
$\Phi(s)$	Pole polynomial of a multi -variable system
$Z(s)$	Zero polynomial of a multi -variable system
$R(t)$	Reference input
$Y(t)$	Output
$U(s)$	Plant input
P^{-1}	Inverse plant transfer function matrix
$Q(s)$	Reciprocal matrix of P^{-1}
q_{ii}	Diagonal entries of $Q(s)$, equivalent SISO plant for NS MIMO QFT

SYMBOLS

ω_c	Cross-over frequency
P_e	Equivalent loop transmission function for Singular G Method
H	Single feedback element to stabilize P_e
$T(s)$	Closed-loop transfer function matrix
$S(s)$	Sensitivity transfer function matrix
ω_b	Bandwidth
r_0	Low frequency gain
ϕ	Phase margin
M , N	Transforming matrices
$P_1(s)$	Transformed Plant
$G_1(s)$	Controller for transformed plant
$F_1(s)$	Prefilter for transformed plant
θ_i	Enforced polynomials
r_i	Assigned stable roots

CHAPTER I

INTRODUCTION

1.1 Introduction

Quantitative Feedback Theory (QFT) is an effective design methodology to solve the control design problem in a quantitative manner for systems with uncertainty. It can be used to solve single-input single-output (SISO) systems and multi-input multi-output (MIMO) systems (Fig. 1.1). In the SISO case the design steps have been well understood. In the MIMO environment, QFT can be categorized into two approaches: (I) non-sequential (NS) method ([1], [2]), and (II) sequential method ([3]).

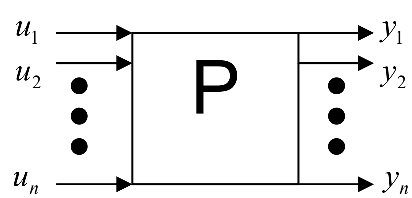

Fig. 1.1 A $n \times n$ MIMO plant

In NS MIMO QFT a $n \times n$ design problem (Fig. 1.2) is first converted into a set of equivalent multi-input single-output (MISO) problems (Fig. 1.3), which indeed are n separate SISO equivalent plants. Also translated are the robust performance (RP) and robust stability (RS) specifications of the original $n \times n$ system into RP and RS specifications for those n SISO plants. The SISO QFT method is then executed for each derived SISO design problem. If those n SISO design problems can be successfully completed, then it follows from the Schauder's fixed-point mapping theorem ([1]) that the MIMO closed-loop uncertain system's RP specifications are satisfied. Moreover,

---

This thesis follows the style of *Journal of Dynamic Systems, Measurement, and Control*.

from a NS MIMO QFT robust stability theorem ([4]) if the MIMO system also satisfies a necessary and sufficient existence condition, then the MIMO closed-loop uncertain system is guaranteed to be robustly stable. The basic QFT design procedure synthesizes a diagonal controller $G(s)$ and a prefilter $F(s)$.

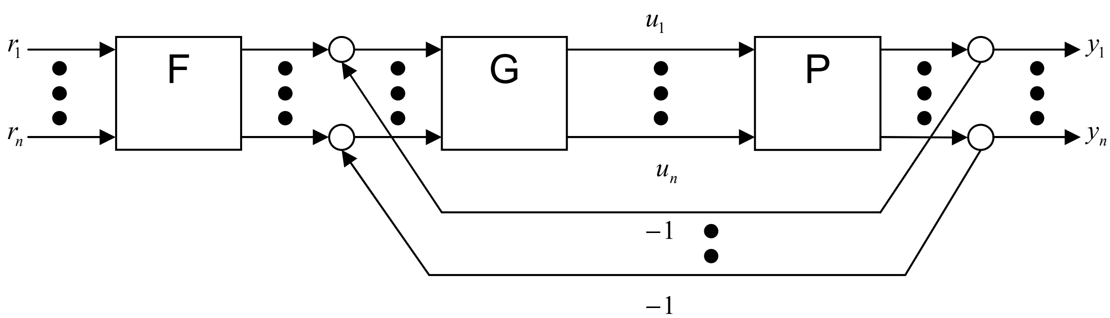

Fig. 1.2 MIMO feedback structure

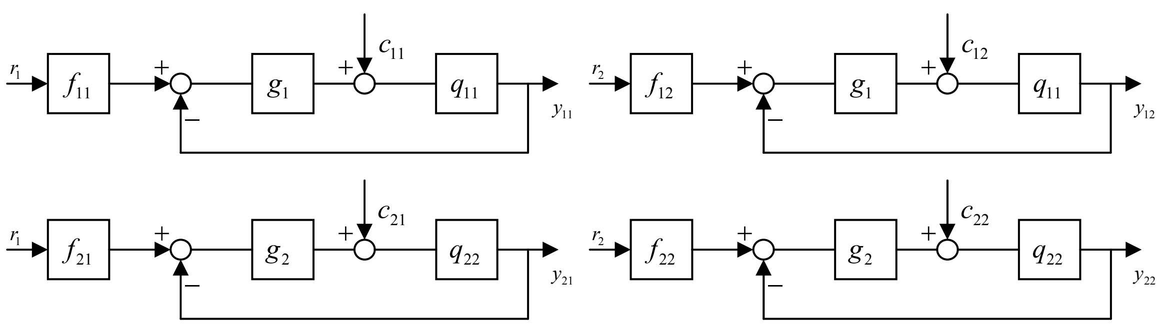

Fig. 1.3 Effective MISO loops for 2×2 system

In summary, if a $n \times n$ uncertain system satisfies a necessary and sufficient existence condition, then the success of that MIMO design problem depends on whether or not the equivalent n SISO design problems can be successfully completed.

1.2 Motivation

For a SISO control design problem, when the plant has unstable poles lying to the right of non-minimum phase (NMP) zeros, i.e. right-half-plane (RHP) dipoles, the SISO QFT design problem may not be solvable. The reason for this is that in the frequency domain design a NMP zero gives a maximum allowable loop cross-over frequency (ω_c) whilst an unstable pole requires a minimum ω_c. Thus, if the NMP zero is closer to the imaginary axis than the unstable pole, there may not exist a controller to stabilize that SISO system. This will be further addressed in chapter Ⅱ.

From a robust stability theorem, in order to guarantee that a controller stabilizes the whole plant family in a SISO uncertain design problem we require the controller to stabilizes at least one plant (usually the nominal plant) in the plant family, and (ii) the zero exclusion condition be satisfied with no imaginary axis pole-zero cancellations. This is the same as requiring that the nominal plant be stabilized without its nominal loop transfer function penetrating the U-contour with no right-half-plane pole-zero cancellation between the controller and the plant for the SISO QFT design problem.

In the MIMO problem if one of the equivalent SISO systems generated from NS MIMO QFT has a RHP dipole for the entire plant family then we may not be able to successfully complete this SISO QFT design problems. Consequently, it follows from the NS MIMO QFT robust stability theorem that the closed-loop MIMO system will not be stabilized.

In MIMO systems, however the role played by NMP zeros are not exactly the same as that of SISO systems. Sometimes a minimum phase (MP) and stable MIMO plant could end up having at least one of the SISO equivalent systems unstable. Even a NMP stable MIMO system could lead to a NMP and unstable plant among its equivalent SISO systems. In these cases, re-numbering the inputs and/or outputs is worth a try and may give a different structure for the equivalent SISO system, i.e. from unstable to stable and/or from NMP to MP. Thus, it is possible to transform the equivalent SISO systems from an apparent un-stabilizable situation to a stabilizable one.

For example (concocted), consider the nominal MIMO plant:

$$P(s)=\frac{\begin{bmatrix}-10(s-5) & 50(s+1)\\ 25(s+2) & -5(s-10)\end{bmatrix}}{6s(4s+15)} \tag{1.1}$$

which is stable and has no MIMO zeros. The pole polynomial and the zero polynomial for this MIMO plant are, respectively,

$$\Phi(s)=s(4s+15) \tag{1.2}$$

$$Z(s)=1 \tag{1.3}$$

From the basic QFT transformations it is easily found that the equivalent plant matrix (Q-matrix) for this plant is:

$$Q(s)=\begin{bmatrix}\dfrac{1}{0.1(s-10)} & \dfrac{1}{s+1}\\ \dfrac{1}{0.5(s+2)} & \dfrac{1}{0.2(s-5)}\end{bmatrix} \tag{1.4}$$

It is then easily seen that the diagonal entries 11 and 22 of Q, which are the equivalent SISO plants to be used in the NS MIMO QFT design, are unstable while the original plant (equation (1.2)) is stable.

Now, if we re-number the inputs so that $U=\begin{bmatrix}u_2\\ u_1\end{bmatrix}$ then the plant becomes:

$$P^{'}(s)=\frac{\begin{bmatrix}50(s+1) & 10(s-5)\\ -5(s-10) & 25(s+2)\end{bmatrix}}{6s(4s+15)} \tag{1.5}$$

And the corresponding new Q-matrix is:

$$Q^{'}(s)=\begin{bmatrix}\dfrac{1}{0.5(s+2)} & \dfrac{1}{0.2(s-5)}\\ \dfrac{1}{0.1(s-10)} & \dfrac{1}{s+1}\end{bmatrix} \tag{1.6}$$

We note that the new entries 11 and 22 are stable.

The example above shows that by re-numbering the inputs the equivalent SISO nominal plants for a NS MIMO QFT design problem may be transformed from being unstable to stable. Consequently, the minimum ω_c requirement necessary for stabilizing unstable systems is removed.

Furthermore, the effect of control directions allows an added degree of freedom for stabilizing a MIMO system possessing a dipole. This is not the same as in SISO design problems, because SISO problems have only one control direction. This RHP dipole MIMO system is not always possible to be stabilized with NS MIMO QFT since this kind of MIMO systems always leads to some of the equivalent systems having a dipole for all plants in the family.

A real life example such as the longitudinal flight control of the X-29 ([5]), which is a 2×2 RHP dipole system with uncertainty, falls into this category and cannot be stabilized using the standard NS MIMO QFT. The X-29 was however robustly stabilized by Walke using the so-called "Singular G Method" ([5]) proposed by Horowitz. This is a case in point which shows that although a NMP and unstable MIMO system may not be stabilized by the standard NS MIMO QFT method it does not necessarily mean that such difficult MIMO systems can not be stabilized. It is worth pointing out, however, that although it appears possible to stabilize the X-29 with other competing robust control methods no design has been made with the required stability specifications. In other words, this class of problems is extremely difficult to solve and remains an unsolved issue.

The presence of dipoles in a MIMO system is not the same as a dipole in a SISO system because the directions can play a significant role in MIMO systems. The standard NS MIMO QFT assumes or considers only diagonal controllers whereas it is entirely possible to use a fully populated controller. The restriction to diagonal controllers obviously limits the ability of NS MIMO QFT to deal with systems having dipoles.

1.3 Singular G Method

The term "Singular G Method" ([5]) has been coined because of the nature of compensation employed. The resulting controller is singular. It was proposed by Horwitz as an approach to stabilize unstable and/or NMP systems. In addition it also allows development of a prefilter, which permits the system to achieve a desired tracking quality. However, this method does not directly synthesize a robust controller. Indeed, it deals with each plant in the family separately when one tries to get rid of the RHP dipole situation and then apply QFT to stabilize a SISO equivalent plant, P_e.

The Singular G Method unlike the NS MIMO QFT uses the plant matrix directly. Its block diagram is as depicted in Figure 1.4. The method is next explained by means of a 2×2 plant. Here, H is a single feedback element rather than a matrix and r is also a single command input. Hence,

$$u_1 = g_1[H(-k_1 y_1 - k_2 y_2)] + f_1 r \tag{1.7}$$

$$u_2 = g_2[H(-k_1 y_1 - k_2 y_2)] + f_2 r \tag{1.8}$$

It follows that

$$Y = PU = P\left[-H\begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix} Y + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} r\right] \tag{1.9}$$

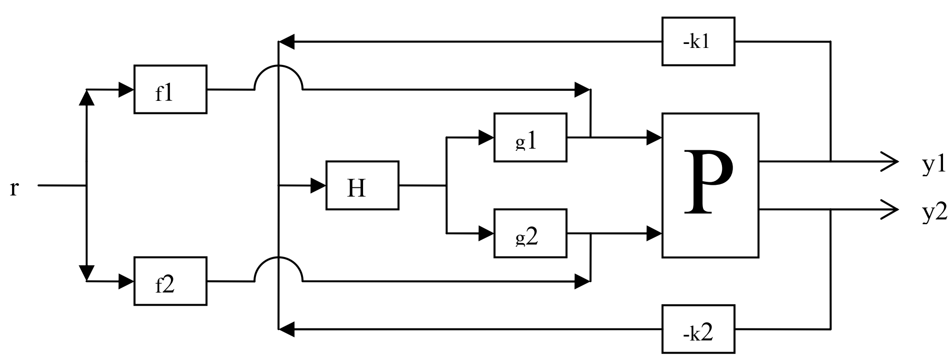

Fig. 1.4 Singular G block diagram

The controller and prefilter are defined as follows:

$$G(s) = H\begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix} \tag{1.10}$$

$$F(s) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{1.11}$$

In case there is no RHP pole-zero cancellations, the stability of the closed-loop system is determined by the poles of the sensitivity function which comes from the $\det(I+PG)$.

$$\det(I+PG)=1+H\{(P_{11}k_1g_1)+(P_{12}k_1g_2)+(P_{21}k_2g_1)+(P_{22}k_2g_2)\}=1+HP_e \qquad (1.12)$$

Hence, the first objective is to choose k_1, k_2, g_1, and g_2 such that the equivalent plant, P_e, has no RHP zeros. This is done by applying the Routh Stability Criterion. To simplify the calculations let:

$$g_1 = a(g_2) \qquad (1.13)$$

$$k_2 = b(k_1) \qquad (1.14)$$

If $P_{ij} = z_{ij}/d$, i.e. all entries of the plant have a common denominator, one obtains:

$$\det(I+PG)=1+g_2k_1\left(H/d\right)(az_{11}+z_{12}+abz_{21}+bz_{22}) \qquad (1.15)$$

Here,

$$az_{11}+z_{12}+abz_{21}+bz_{22} = A_1s^m + A_2s^{m-1}+\cdots+A_{m+1} \qquad (1.16)$$

and,

$$A_i = f(a,b)=c_1a+c_2ab+c_3b+c_4,\ c_i = \text{constant}. \qquad (1.17)$$

The values of a and b are then determined by the first column of the Routh array. Since these functions are typically nonlinear, the quickest way to solve it is to write a code which iteratively substitutes values into the functions and checks for the proper signs. (**Remark:** a and b can also be functions of s.)

After acquiring a and b, one stabilizes the SISO equivalent plant, P_e, by selecting a H with the values of g_2 and k_1 set to one. Since the equivalent plant now is just an unstable plant without any NMP zeros, it can be stabilized with just a large gain. After selecting an H, one might distribute H back to g_2 and k_1 so that $H=1$.

After distribution, the new g_i s and k_i s are denoted as $g_i^{'}$ s and $k_i^{'}$ s and the new equivalent plant as $P_e^{'}$. One can obtain the transfer function from r to y_1 and to y_2 respectively as below:

$$t_1 = \left[f_1 P_{11} + f_2 P_{12} + \left(g_2^{'} f_1 - g_1^{'} f_2 \right) k_2^{'} \left(\det(P) \right) \right] \Big/ \left(1 + P_e^{'} \right) \tag{1.18}$$

$$t_2 = \left[f_1 P_{21} + f_2 P_{22} + \left(g_1^{'} f_2 - g_2^{'} f_1 \right) k_1^{'} \left(\det(P) \right) \right] \Big/ \left(1 + P_e^{'} \right) \tag{1.19}$$

The prefilter is then determined from matching a desired transfer function model to transfer functions in equation (1.18) and (1.19).

$$T_i = n_i(s) \Big/ p_i(s) = t_i \tag{1.20}$$

1.4 Objectives of the Thesis

In this thesis, re-examined are the issues raised by the longitudinal flight control of X-29 since it has not been stabilized by NS MIMO QFT. Moreover, compared are the features of the “Singular G Method” with the transformed NS MIMO QFT methodology developed in this research.

Hence, one of the objectives of this research is to analyze the way in which NS MIMO QFT design functions are obtained leading to the n SISO equivalent plants and develop a novel way of obtaining those n SISO equivalent systems such that those n equivalent SISO design problems will be free of RHP dipoles. Then it will be possible to execute SISO QFT designs on the equivalent plants without having to deal with unnecessary bandwidth constraints. A straight forward procedure for applying the transformed NS MIMO QFT is also developed.

1.5 Thesis Overview

Chapter I briefly introduces the NS MIMO QFT and the “Singular G Method”. A general description of the challenges faced is given and the goal of this thesis is described.

Chapter II explains the case of RHP dipoles and its effects on SISO systems both from a root loci and frequency domain design view point. This chapter details the problems encountered when applying the NS MIMO QFT to a system possessing RHP dipoles.

Chapter III gives a detailed description of the transformed NS MIMO QFT. A way to obtain the transformation matrices is provided, and a procedure to apply the transformed NS MIMO QFT is developed.

In Chapter IV, a concocted example is solved with both the transformed NS MIMO QFT and the “Singular G Method”. The features of both methods are illustrated and compared.

Chapter V brings in the X-29 longitudinal flight control problem. A general description of this aircraft, and its control surfaces, as well as the model used in this thesis is given. The transformed NS MIMO QFT is applied to this longitudinal flight control problem in this chapter.

Chapter VI concludes this thesis with a summary and recommendations for further research into both the design method and the X-29 design problem.

CHAPTER II

RIGHT HALF PLANE DIPOLES IN SISO SYSTEMS

2.1 Introduction

The difficulty in designing a controller for a SISO RHP dipole system is illustrated in this chapter. This is illustrated through the root loci of RHP dipole systems. Also, the reasons that a NMP zero limits the bandwidth and an unstable pole requires a minimum bandwidth are explained.

2.2 NMP SISO Systems and Unstable SISO Systems

A SISO system with zeros in the RHP is called a NMP system. A system of this type may be stabilized by a small gain. This can be explained from the root loci of the system. As the feedback gain increases, the closed-loop poles go away from the locations of open-loop poles and approach the locations of open-loop zeros and asymptotes. Hence, a small gain keeps the closed-loop poles away from the NMP zero location and stabilizes the closed-loop system. For example, consider a system with two RHP zeros (at 1 and 60) and three stable poles (at -5, -10, and -100). Its transfer function and root loci are as in equation (2.1) and Figure 2.1.

$$P(s) = \frac{0.8333s^2 - 50.83s + 50}{0.01s^3 + 1.15s^2 + 15.5s + 50} \qquad (2.1)$$

A SISO unstable system has poles in the RHP. A system of this type always requires a large gain to stabilize. This can also be explained from the root loci of the system. As the feedback gain increases, the closed-loop poles go away from the locations of open-loop poles and approach the locations of open-loop zeros and asymptotes. A large gain is required to drive the closed-loop poles away from their open-loop locations, especially the RHP ones, so that the closed loop may be stabilized. However, this is not absolutely essential because there might be some asymptotes pointing to the unstable region. Thus, a system may go unstable as the gain is increased or it may not be stabilized with a simple gain. Consider as an example a system with two zeros (at -1 and -5) and three poles (at 3, -7, and -13). Its transfer function and root loci are as in equation (2.2) and Figure 2.2.

$$P(s) = \frac{(s+1)(s+5)}{(s-3)(s+7)(s+13)} \tag{2.2}$$

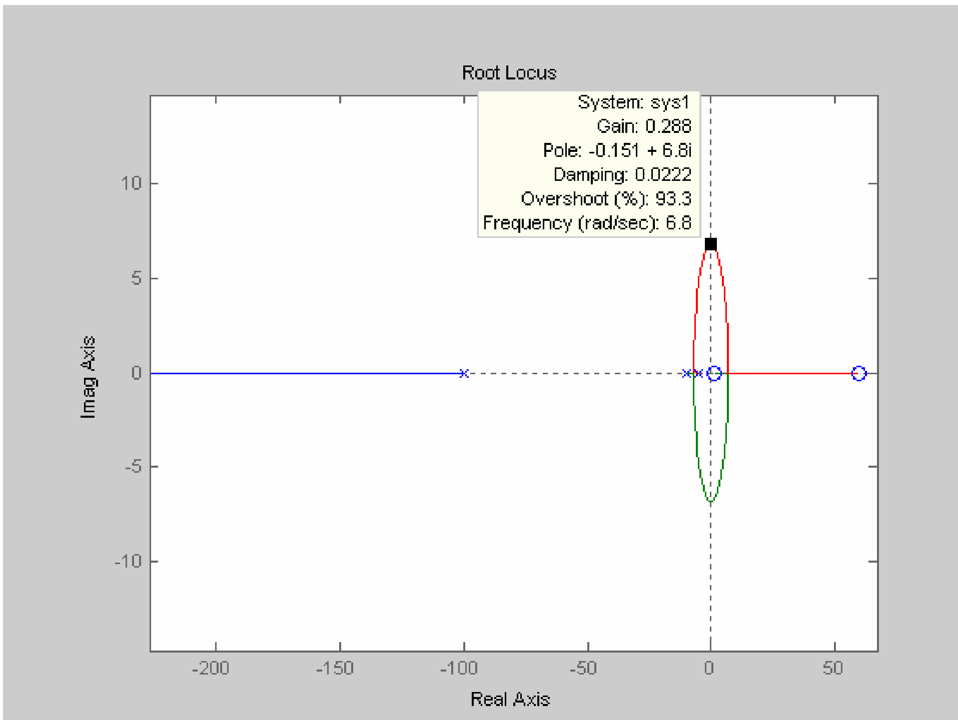

Fig. 2.1 Root Loci for the NMP system example

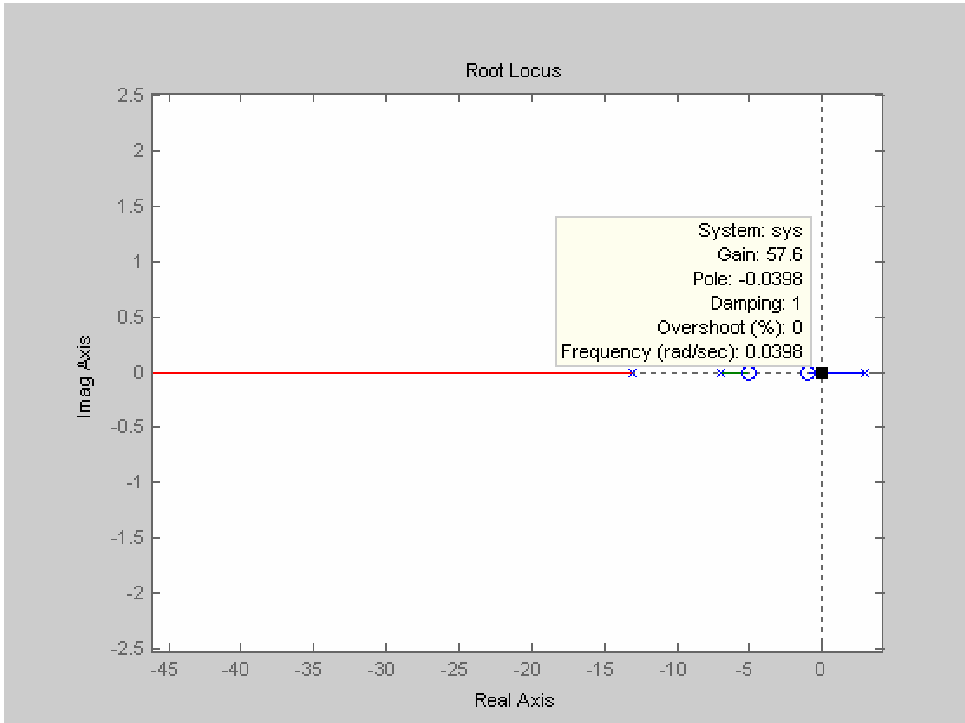

Fig. 2.2 Root Loci for the unstable system example

2.3 Root Loci for RHP Dipole Systems

When there is an unstable pole lying to the right of a NMP zero, this pair of pole and zero is called a RHP dipole. A RHP dipole system is extremely difficult or may even be impossible to stabilize.

As discussed earlier, a NMP zero requires a small gain to stabilize while an unstable pole requires a large gain. If these two acceptable gain regions do not overlap, then there will be no solution. This is exactly the situation with a RHP dipole system. This can be easily illustrated on the root loci of a RHP dipole system. Clearly, one of the poles will eventually go to the NMP zero location and will always be the unstable one lying to the right of that NMP zero. In other words, the NMP zero of a dipole blocks the unstable pole's path to LHP. For a system with a NMP zero at 1 and an unstable pole at 2, its root loci plot is shown in Figure 2.3.

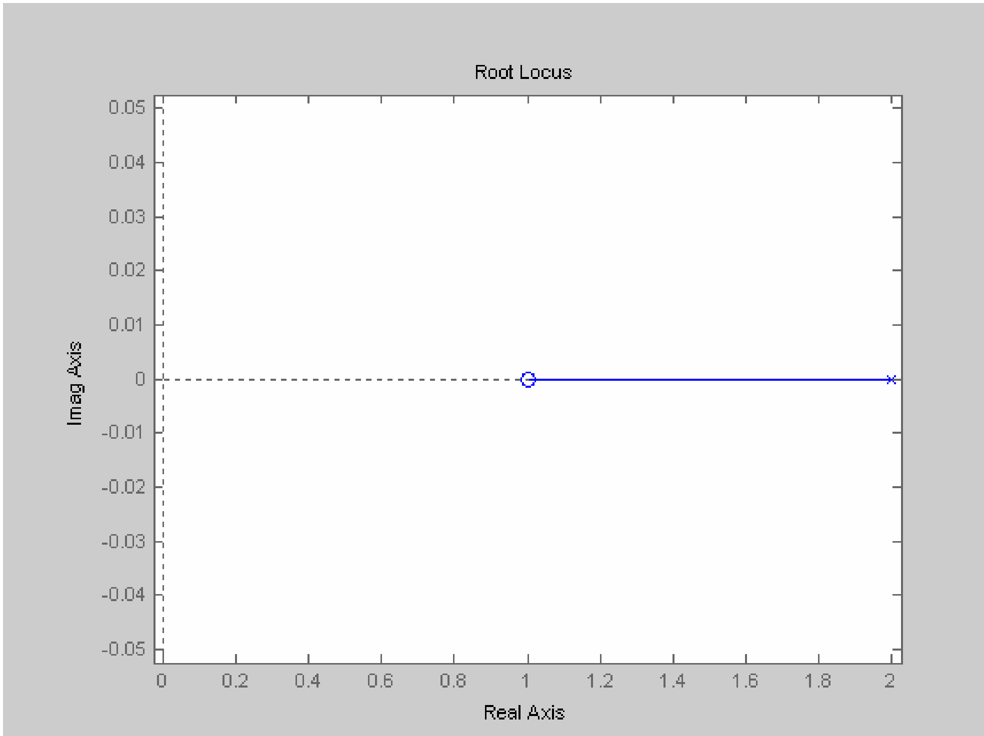

Fig. 2.3 Root Loci of the dipole example

One approach which can be adopted to resolve this situation is to insert an extra pole in between the zero and pole of a dipole. By doing so, one might have a chance to break up a dipole and make those two unstable poles shift in a circular manner. Thus, the closed-loop system may achieve conditional stability. As an example, consider a system with one zero (at 1) and three poles (at -5 and 2). The dipole is broken up after inserting an extra pole at 1.5 as shown in Figure 2.4.

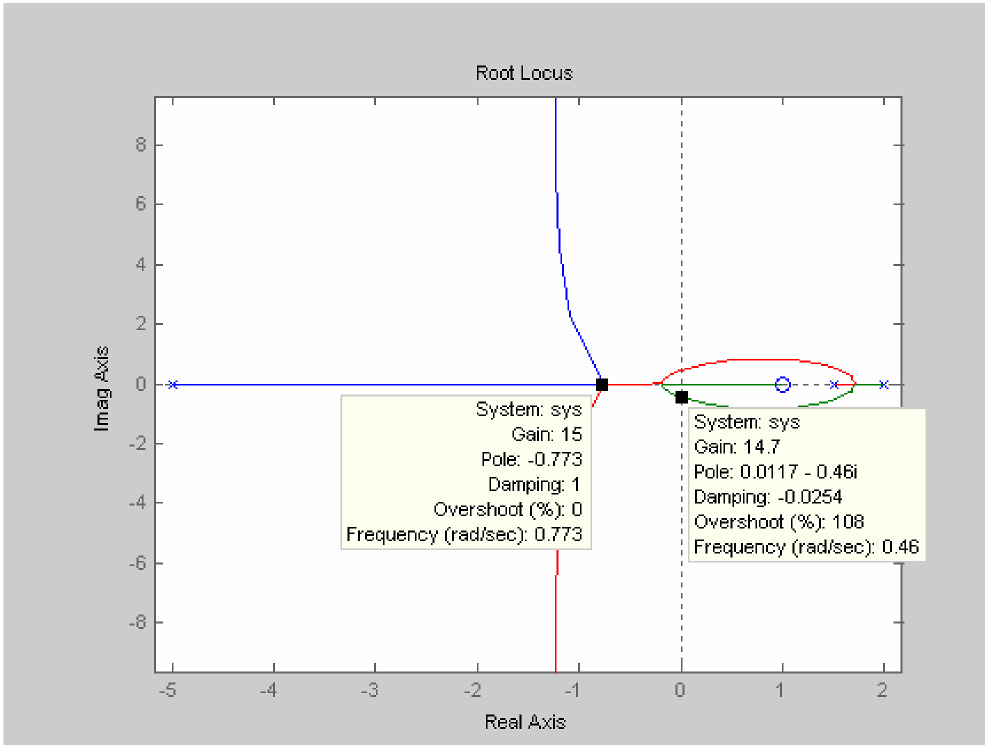

Fig. 2.4 Example of dipole broken up

However, this approach may not always work: In situations that the pole and zero in a dipole are very close to each other, the dipole is very close to the imaginary axis, or there are other dynamics on LHP that will resist the unstable poles moving toward left, this trick may fail. An example is shown in Figure 2.5 of a system possessing one zero (at 1) and three poles (at 2, -5, and -10). The inserted unstable pole is at 1.5.

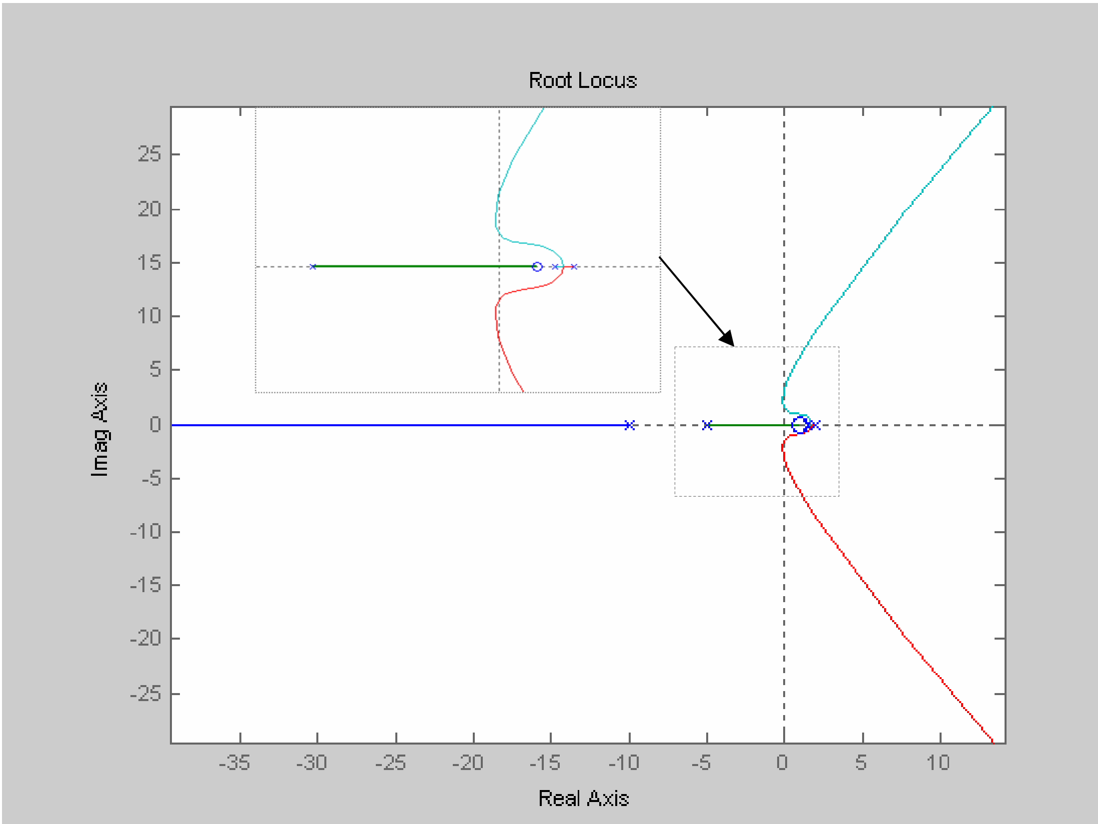

Fig. 2.5 A failed case

2.4 Frequency Domain Design for RHP Dipole Systems

As mentioned in Chapter I, a NMP zero sets a maximum achievable bandwidth (ω_b) on a system while an unstable pole requires a minimum bandwidth. This poses a serious difficulty while loop shaping the system, and even make it impossible to find a solution.

2.4.1 Effect of NMP zeros

In order to illustrate the constraint posed by a NMP zero, a measurement called gain-bandwidth product ($r_0 \times \omega_b$) is introduced and depicted in Figure 2.6. This product is the feedback benefit obtained from a loop. If the slope at the cross-over frequency (ω_c) is fixed, changing the gain r_0 and the bandwidth ω_b will affect the value of this product but its maximum is limited by the ω_c.

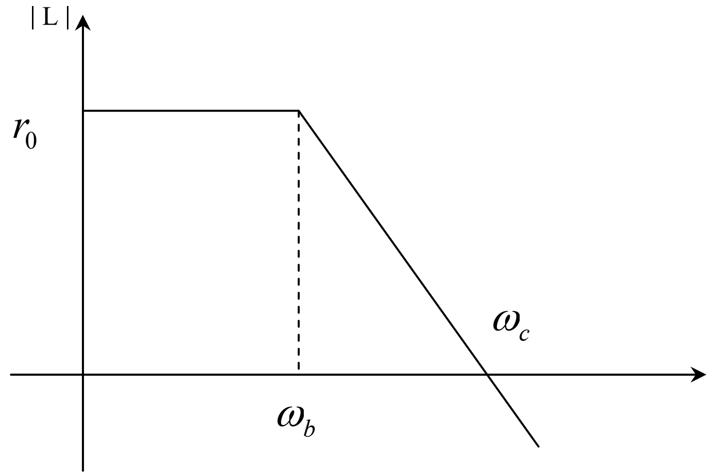

Fig. 2.6 Magnitude plot of a loop

Suppose a NMP system can be described as in equation (2.3). The minimum phase part of the system and the NMP zero are separated out as $P_m(s)$ and $(s-a)$ respectively as in equation (2.3).

$$P_{nmp}(s) = P_m(s)(s-a) \tag{2.3}$$

This can then be written as the following equation (2.4).

$$P_{nmp}(s) = (s+a)P_m(s)\left(\frac{s-a}{s+a}\right) = (s+a)P_m(s)B(s) \tag{2.4}$$

Here, $B(s)=\left(\dfrac{s-a}{s+a}\right)$ is an all-pass function.

Suppose that one obtains a controller $G(s)$ after design. The loop transmission can then be expressed as in equation (2.5).

$$L_{nmp}(s)=G(s)(s+a)P_m(s)B(s)=L_m(s)B(s) \tag{2.5}$$

$L_m(s)$ is the minimum phase part of the loop transmission. Let the phase angle at the cross-over frequency ω_c be $-\Pi$. This means the phase margin is zero. The phase of $L_{nmp}(s)$ at $\omega_c=a$ is then:

$$\angle L_{nmp}(ja)=\angle L_m(ja)+\angle B(ja)=-\Pi \tag{2.6}$$

This implies:

$$\angle L_m(ja)=-\frac{\Pi}{2} \tag{2.7}$$

From Bode Integral, $\dfrac{\Pi}{2}\dfrac{dL}{d\omega}\cong phase$, an approximation of the slope can be obtained.

$$\frac{dL}{d\omega}=\frac{2}{\Pi}\times phase=-1=-20\,{}^{dB}\!/_{decay} \tag{2.8}$$

Hence the gain-bandwidth product is known from equation (2.9).

$$\frac{20\log r_0-0}{\log\frac{\omega_b}{a}}=-20 \implies r_0\omega_b=a \tag{2.9}$$

If one does the same calculation for a bigger or smaller cross-over frequency, the gain-bandwidth product will decrease in both cases. Therefore, the feedback benefit is restricted to less or equal to a by a NMP zero at $s=a$. As a result, at a fixed low frequency gain the bandwidth is limited.

2.4.2 Effect of unstable poles

As for an unstable pole case, the system is described as in equation (2.10). The unstable pole is first split out from the stable part $P_{stable}(s)$. Then, an all-pass function is introduced, $B^{'}(s)=\left(\dfrac{s+a}{s-a}\right)$.

$$P_{unstable}(s) = \frac{P_{stable}(s)}{(s-a)} = \left(\frac{P_{stable}(s)}{(s+a)}\right)\left(\frac{s+a}{s-a}\right) = \left(\frac{P_{stable}(s)}{(s+a)}\right)B'(s) \qquad (2.10)$$

Once a controller is synthesized, the loop transmission can be expressed as equation (2.11), where $L_{stable}(s)$ is the stable part of the loop transmission.

$$L_{unstable}(s) = \left(\frac{G(s)P_{stable}(s)}{s+a}\right)B'(s) = L_{stable}(s)B'(s) \qquad (2.11)$$

In order to have a stable closed-loop system, $L_{unstable}(s)$ requires having one counter-clockwise (CCW) encirclement on the Nyquist plot. Considering path ① and path ② in Figure 2.7, obviously path ① is the type that is needed. Moreover, it is not possible for $L_{stable}(s)$ to provide any CCW encirclement on the Nyquist plot since all its poles lay on the LHP. Clearly, such a encirclement can be obtained only from the all-pass function, $B'(s) = \left(\frac{s+a}{s-a}\right)$.

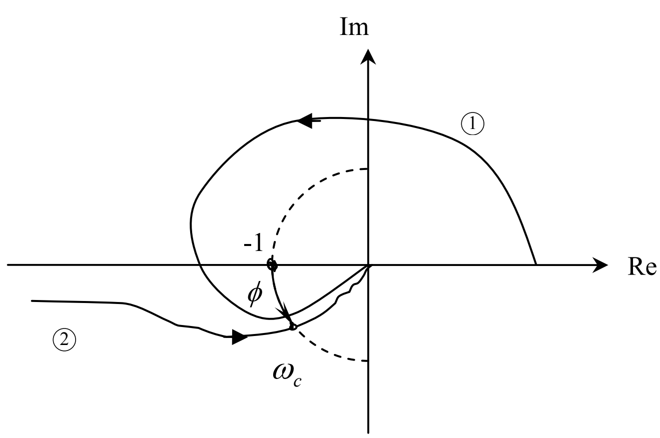

Fig. 2.7 Nyquist plot

Further suppose that $|L_{unstable}(j\omega)|$ and hence $|L_{stable}(j\omega)|$ are monotonic decreasing with respect to ω. Then, the value of $\angle L_{stable}(j\omega) - \angle L_{stable}(0)$ is always negative. From Figure 2.7, equation (2.12) is obtained.

$$\angle L_{unstable}(j\omega_c) - \angle L_{unstable}(0) = \phi \tag{2.12}$$

Also, we know that:

$$\angle L_{unstable}(j\omega) = \angle L_{stable}(j\omega) + \angle B'(j\omega) \tag{2.13}$$

and it follows that

$$\angle L_{unstable}(j\omega_c) - \angle L_{unstable}(0) = \angle L_{stable}(j\omega_c) + \angle B'(j\omega_c) - \angle L_{stable}(0) - \angle B'(0) \tag{2.14}$$

Since $\angle L_{stable}(j\omega) - \angle L_{stable}(0)$ is negative and $\angle B'(0) = \Pi$, one obtains the inequality (2.15).

$$\angle L_{unstable}(j\omega_c) - \angle L_{unstable}(0) < \angle B'(j\omega_c) - \Pi \tag{2.15}$$

Hence, it can be derived as inequality (2.16).

$$\angle B'(j\omega_c) > \Pi + \phi \tag{2.16}$$

Answers for some ϕ values are in equation (2.17).

$$\begin{aligned} \phi &= \frac{\Pi}{4} \Rightarrow \omega_c > \frac{a}{2} \\ \phi &= \frac{\Pi}{2} \Rightarrow \omega_c > a \end{aligned} \tag{2.17}$$

As a result, for a desired phase margin ϕ a minimum cross-over frequency is required, and hence a minimum bandwidth.

In the case of a RHP dipole plant, the NMP zeros from the dipoles set an upper limit on the bandwidth as illustrated in 2.4.1 and the unstable poles from the dipoles set a lower limit on the bandwidth as illustrated in this section. Therefore, for systems with an unstable pole lying to the right of a NMP zero it may not be possible to find a solution.

2.5 Summary

The difficulty posed by a RHP dipole system in design has been explained in this chapter. First, this problem was illustrated from a root loci design viewpoint. In this part, the root loci plots of a NMP system, an unstable system, and a RHP dipole system were examined and compared through examples. As a result, the NMP zero in a RHP dipole blocks the path of the unstable pole to the stable region. Then, the same difficulty is illustrated through a frequency domain argument. The effect of a NMP zero and the effect of an unstable pole in frequency domain design are explained. Subsequently, the combination of these two effects is the effect of a RHP dipole. As a result, the NMP zero sets an upper limit on the bandwidth and the unstable pole sets a lower limit on the bandwidth. Hence, a RHP dipole system poses an extremely difficult design problem.

CHAPTER III

TRANSFORMED NS MIMO QFT

3.1 Introduction

This chapter explains the transformed NS MIMO QFT method and how this transformation scheme is applied with NS MIMO QFT to handle RHP dipole systems. Also provided is a procedure for determining a set of transformation matrices M and N which yield desirable pole-zero locations in the equivalent SISO plants to facilitate the NS MIMO QFT design methodology. Only one feasible approach is considered here although there are number of other approaches that may be pursued.

3.2 Matrix Transformation

Consider the basic governing equations for reference $R(t)$ to output response $Y(t)$ given by

$$T = (I + PG)^{-1} PGF \;, \tag{3.1}$$

from which one has the usual equation

$$(I + PG)T = PGF \tag{3.2}$$

and

$$(P^{-1} + G)T = GF \;. \tag{3.3}$$

In the non-sequential MIMO QFT procedure, one typically considers P^{-1} to define equivalent plant sets q_{ii} where

$$(P^{-1})_{ij} = \left[\frac{1}{q_{ij}} \right]. \tag{3.4}$$

Introduce two non-singular matrices M, N (Fig. 3.1) such that

$$(I + MM^{-1}PNN^{-1}G)T = PNN^{-1}GF \;. \tag{3.5}$$

It then follows that

$$(I + M^{-1}PNN^{-1}GM)M^{-1}T = M^{-1}PNN^{-1}GMM^{-1}F\,. \tag{3.6}$$

The equation (3.6) can now be written as follows

$$(I + P_1G_1)T_1 = P_1G_1F_1 \tag{3.7}$$

where

$$P_1 = M^{-1}PN\ ,\ G_1 = N^{-1}GM\ ,\ T_1 = M^{-1}T\ ,\ F_1 = M^{-1}F\,. \tag{3.8}$$

It is noted that equation (3.8) is in exactly the same form as the standard equation with P, G, F replaced by P_1, G_1, F_1. Consequently, one could derive all the required QFT MIMO design equations with these new variables.

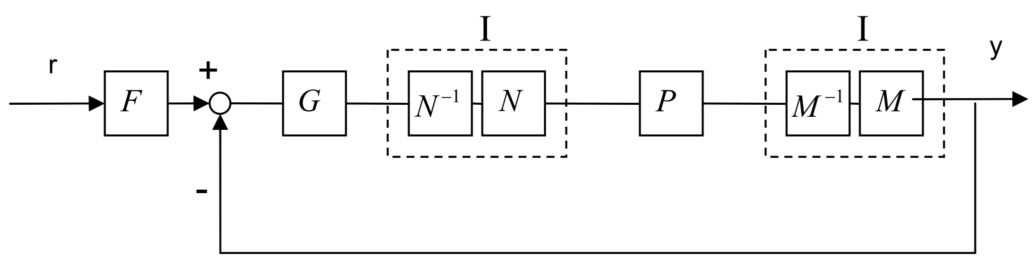

Fig. 3.1 Transformation

From the preceding development it is clear that one can use the flexibility offered by M and N to be taken advantage of to design the MISO loops. Finally, when a design is obtained one can do an inverse transformation to obtain the controller for the original system as

$$G = NG_1M^{-1}\,. \tag{3.9}$$

The only remaining thing to prove then is that the closed-loop stability of $\det(I + P_1G_1)$ implies the stability of the original closed-loop $\det(I + PG)$. This follows immediately by noting that

$$\det(I + P_1G_1) = \det(I + M^{-1}PNN^{-1}GM) = \det(I + M^{-1}PGM) = \det\{M^{-1}(I + PG)M\} \tag{3.10}$$

which is equal to $\det(I + PG)$.

3.3 Selecting the Matrices

Provided here is a procedure to obtain a suitable pair of the transformation matrices. Although it is an option to use both M and N matrices, it is better to start by using only N or M because selecting M and N together is more complex than just a single M or N.

3.3.1 Plant set-up

Considered here is a 2×2 MIMO plant of the form

$$P=\begin{bmatrix} \frac{z_{11}}{p_{11}} & \frac{z_{12}}{p_{12}} \\ \frac{z_{21}}{p_{21}} & \frac{z_{22}}{p_{22}} \end{bmatrix}, \tag{3.11}$$

where z_{ij} and p_{ij} are polynomials of s.

The determinant of P is:

$$\det P=\frac{z_{11}\cdot z_{22}\cdot p_{12}\cdot p_{21}-z_{12}\cdot z_{21}\cdot p_{11}\cdot p_{22}}{p_{11}\cdot P_{22}\cdot p_{12}\cdot P_{21}}=\frac{k_1\cdot f(s)\cdot \mathrm{Z}(s)}{k_2\cdot f(s)\cdot \Phi(s)}=\frac{k\cdot \mathrm{Z}(s)}{\Phi(s)} \tag{3.12}$$

Where $f(s)=(s-f_1)\cdot(s-f_2)\cdot(s-f_3)\cdots$ is the common factor of $(p_{12}\cdot p_{21}-p_{11}\cdot p_{22})$ and $(p_{11}\cdot p_{22}\cdot p_{12}\cdot p_{21})$. $Z(s)$ is the zero polynomial of the 2×2 system and $\Phi(s)$ is the pole polynomial of the 2×2 system.

3.3.2 The standard Q matrix

In the non-sequential design methodologies the MIMO design problem is decomposed into n MISO design problems with n SISO equivalent plants. The equivalent plant matrix Q for the plant defined in 3.3.1 is given by:

$$Q=\begin{bmatrix}\dfrac{p_{22}\det P}{z_{22}} & \dfrac{p_{12}\det P}{z_{12}}\\ \dfrac{p_{21}\det P}{z_{21}} & \dfrac{p_{11}\det P}{z_{11}}\end{bmatrix} \tag{3.13}$$

$$=\begin{bmatrix}\dfrac{z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22}}{z_{22}p_{11}p_{12}p_{21}} & -\dfrac{z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22}}{z_{12}p_{11}p_{22}p_{21}}\\ -\dfrac{z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22}}{z_{21}p_{11}p_{12}p_{12}} & \dfrac{z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22}}{z_{11}p_{22}p_{12}p_{21}}\end{bmatrix}$$

$$=\begin{bmatrix}\dfrac{kZ(s)p_{22}}{z_{22}\Phi(s)} & \dfrac{kZ(s)p_{12}}{z_{12}\Phi(s)}\\ \dfrac{kZ(s)p_{21}}{z_{21}\Phi(s)} & \dfrac{kZ(s)p_{11}}{z_{11}\Phi(s)}\end{bmatrix}$$

The diagonal entries are the equivalent plants to be used.

3.3.3 Using only the N matrix transformation

Suppose $N=\begin{bmatrix}n_{11} & n_{12}\\ n_{21} & n_{22}\end{bmatrix}$ where N^{-1} exists and n_{ij} are polynomials of s. Defining $P_{1n} = P\cdot N$, the resulting Q matrix then is:

$$Q_{1n}=\begin{bmatrix}q_{n11} & q_{n12}\\ q_{n21} & q_{n22}\end{bmatrix} \tag{3.14}$$

$$=\begin{bmatrix}\dfrac{(n_{11}n_{22}-n_{12}n_{21})(z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22})}{(n_{12}z_{21}p_{22}-n_{22}z_{22}p_{21})p_{11}p_{12}} & -\dfrac{(n_{11}n_{22}-n_{12}n_{21})(z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22})}{(n_{12}z_{11}p_{12}-n_{22}z_{12}p_{11})p_{22}p_{21}}\\ -\dfrac{(n_{11}n_{22}-n_{12}n_{21})(z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22})}{(n_{11}z_{21}p_{22}-n_{21}z_{22}p_{21})p_{11}p_{12}} & \dfrac{(n_{11}n_{22}-n_{12}n_{21})(z_{11}z_{22}p_{12}p_{21}-z_{12}z_{21}p_{11}p_{22})}{(n_{11}z_{11}p_{12}-n_{21}z_{12}p_{11})p_{22}p_{21}}\end{bmatrix}$$

In the diagonal entries, let

$$\theta_1 = n_{12}z_{21}p_{22}-n_{22}z_{22}p_{21} \tag{3.15}$$

and

$$\theta_2 = n_{11}z_{11}p_{12}-n_{21}z_{12}p_{11}. \tag{3.16}$$

The objective for systems with right half plane dipoles is to select n_{ij} such that the dipole in the diagonal entries of Q disappears. In order to accomplish that, one can select a target q_{nii} to be as follows:

a. q_{n11} is MP but unstable, q_{n22} is NMP but stable and all roots of $\left(n_{11}n_{22} - n_{12}n_{21}\right)$ are stable, if the multivariable unstable poles show up in q_{n11}.

b. q_{n11} is NMP but stable, q_{n22} is MP but unstable and all roots of $\left(n_{11}n_{22} - n_{12}n_{21}\right)$ are stable, if the multivariable unstable poles show up in q_{n22}.

c. Furthermore, one could have q_{nii} to be NMP and unstable as long as the NMP zero is lying to the right of the unstable poles. This allow $\left(n_{11}n_{22} - n_{12}n_{21}\right)$ to have unstable roots which are lying to the right of q_{nii}'s unstable pole.

Usually, cases a. and b. are preferred since they are very easily stabilized by just applying a large gain to the unstable one and a small gain to the NMP one.

Regardless of which case one employs, matrix N cannot possess any RHP root which is the same as a MIMO system unstable pole. The reason for this is that the unstable roots of N will show up as the controller's NMP zeros, since $G = N \cdot G_{1n}$. Thus, it results in RHP pole-zero cancellations between G and the original P.

Remark: In case both q_{n11} and q_{n22} have a dipole, one will need to remove the unstable pole from either q_{n11} or q_{n22}. There are two possible ways the situation may be resolved. One way is to produce an unstable root from $\left(n_{11}n_{22} - n_{12}n_{21}\right)$, which could cancel the unstable pole. However, this violates the requirement that the N matrix can not possess any right-half-plane root which is same as the MIMO system unstable poles. The other way is to assign a common denominator for n_{12} and n_{22} such that this common denominator can cancel the unstable pole in q_{n11}. (Same as for q_{n22}) Nevertheless, by doing so $\left(n_{11}n_{22} - n_{12}n_{21}\right)$ will also have that common denominator (the unstable pole). Thus, the effort to achieve this in this manner is futile. This suggests that to get around the unstable pole one needs to use both M and N matrices.

Let $z_{ji}p_{jj} = \left(a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0\right)$ and $z_{jj}p_{ji} = \left(b_m s^m + b_{m-1}s^{m-1} + \cdots + b_0\right)$ in equation (3-15) and (3-16). When selecting n_{ij} s from θ_1 and θ_2, one can start with $n_{ij} = c_u s^u + c_{u-1}s^{u-1} + \cdots + c_0$ and $n_{jj} = d_v s^v + d_{v-1}s^{v-1} + \cdots + d_0$, and use the formulae below.

$$\pm n_{ij}z_{ji}p_{jj} \mp n_{jj}z_{jj}p_{ji}$$

$$= \pm\left(c_u s^u + c_{u-1}s^{u-1} + \cdots + c_0\right)\left(a_n s^n + \cdots + a_0\right) \mp \left(d_v s^v + d_{v-1}s^{v-1} + \cdots + d_0\right)\left(b_m s^m + \cdots + b_0\right)$$

$$= K\left(s - r_1\right)\left(s - r_2\right)\cdots\left(s - r_t\right) \tag{3.17}$$

Here, some r_i are specified for the target q_{nii} and the rest of them are free variables that can be chosen but have to be stable roots. For example, one may need to assign r_1 to cancel the NMP zero. Thus, the total number of roots, t, is the biggest number among $n+u$, $v+m$ and the number of the specified roots. This is also the highest order for the polynomial $\pm n_{ij}z_{ji}p_{jj} \mp n_{jj}z_{jj}p_{ji}$.

Normally, one has to search for a solution from constant n_{ij} to order 1-polynomials, and then to order 2 polynomials, etc. At the same time, one also needs to select those free r_i's such that the resulting N matrix satisfies the requirements.

In fact, an easier way is to argue the orders, u and v, to be high enough such that the number of variables matches the number of the polynomial's coefficients, $t+1$. Then solve the variables c_i 's and d_i 's by Gauss-elimination assuming those free r_i 's are known. After that, one can specify those free r_i 's such that the resulting N matrix satisfies the requirements. However, the determinant of the N matrix is a nonlinear function of those free r_i's. The quickest way to solve is to write a code which iteratively substitutes values into the functions and checks for the proper signs. Sometimes, one can just arbitrary select those free r_i's as a first attempt.

The trick here is that whenever the order of both n_{ij} is increased by one, the order t of the polynomial is only increased by one so that a sufficient number of independent coefficients appear to match a desired polynomial. In particular, one obtains two more variables by adopting this procedure. Thus, it is possible to set up enough variables for Gauss-elimination.

3.3.4 Using only M matrix transformation

Using M matrix transformation is similar to the use of the N matrix. Suppose

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \tag{3.18}$$

where $M^{-1} = \begin{bmatrix} \hat{m}_{11} & \hat{m}_{12} \\ \hat{m}_{21} & \hat{m}_{22} \end{bmatrix}$ exists and $\hat{m}_{ij}$ can be constants or polynomials.

With the transformation $P_{1m} = M^{-1} \cdot P$ the resulting Q matrix is:

$$Q_{1m} = \begin{bmatrix} \dfrac{(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(\hat{m}_{21}z_{12}p_{22} - \hat{m}_{22}z_{22}p_{12})p_{11}p_{21}} & -\dfrac{(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(\hat{m}_{11}z_{12}p_{22} - \hat{m}_{12}z_{22}p_{12})p_{11}p_{21}} \\ -\dfrac{(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(\hat{m}_{21}z_{11}p_{21} - \hat{m}_{22}z_{21}p_{11})p_{12}p_{22}} & \dfrac{(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(\hat{m}_{11}z_{11}p_{21} - \hat{m}_{12}z_{21}p_{11})p_{12}p_{22}} \end{bmatrix} \tag{3.19}$$

Then one follows the same procedure as described earlier for the N matrix to select the M matrix with the controller

$$G = G_{1m} \cdot M^{-1}. \tag{3.20}$$

3.3.5 Using both N and M transformation matrices

In case using a single matrix fails, one can try to use both M and N. It is more powerful and also more complex. The main difference between using single matrix and both matrices is that there no p_{ij} s are left unaffected at diagonal entries of Q. In some cases where q_{n11} and q_{n22} (or, q_{m11} and q_{m22}) are both unstable, using both matrices appears to be the only way to accomplish the job. The reason is one cannot cancel the unstable pole at diagonal entries of Q by selecting entries of a single matrix. Moreover, the complexity of using both matrices grows as the number of entries and thus the number of coefficients increases, and also the coupled effect of $n_{ij} \cdot m_{ij}$ when the Q matrix is formed.

Although one can assume all entries of M and N to be polynomials in s and then try to select their coefficients as in using a single matrix, it is suggested that one arbitrarily assigns one of the matrices to be a specified constant matrix or assumes one of the matrices with arbitrary but constant variable entries, and then try to find another s-polynomial matrix.

Using the transformation $P_{1mn} = M^{-1} \cdot P \cdot N$, the resulting Q matrix is:

$$Q_{1mn} = \begin{bmatrix} q_{mn11} & q_{mn12} \\ q_{mn21} & q_{mn22} \end{bmatrix} \tag{3.21}$$

$$q_{mn11} = \frac{(n_{11}n_{22} - n_{12}n_{21})(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(n_{12}\hat{m}_{21}z_{11}p_{22}p_{12}p_{21} + n_{12}\hat{m}_{22}z_{21}p_{11}p_{22}p_{12} + n_{22}\hat{m}_{21}z_{12}p_{11}p_{22}p_{21} + n_{22}\hat{m}_{22}z_{22}p_{11}p_{12}p_{21})}$$

$$q_{mn12} = -\frac{(n_{11}n_{22} - n_{12}n_{21})(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(n_{12}\hat{m}_{11}z_{11}p_{22}p_{12}p_{21} + n_{12}\hat{m}_{12}z_{21}p_{11}p_{22}p_{12} + n_{22}\hat{m}_{11}z_{12}p_{11}p_{22}p_{21} + n_{22}\hat{m}_{12}z_{22}p_{11}p_{12}p_{21})}$$

$$q_{mn21} = -\frac{(n_{11}n_{22} - n_{12}n_{21})(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(n_{11}\hat{m}_{21}z_{11}p_{22}p_{12}p_{21} + n_{11}\hat{m}_{22}z_{21}p_{11}p_{22}p_{12} + n_{21}\hat{m}_{21}z_{12}p_{11}p_{22}p_{21} + n_{21}\hat{m}_{22}z_{22}p_{11}p_{12}p_{21})}$$

$$q_{mn22} = \frac{(n_{11}n_{22} - n_{12}n_{21})(\hat{m}_{11}\hat{m}_{22} - \hat{m}_{12}\hat{m}_{21})(z_{11}z_{22}p_{12}p_{21} - z_{12}z_{21}p_{11}p_{22})}{(n_{11}\hat{m}_{11}z_{11}p_{22}p_{12}p_{21} + n_{11}\hat{m}_{12}z_{21}p_{11}p_{22}p_{12} + n_{21}\hat{m}_{11}z_{12}p_{11}p_{22}p_{21} + n_{21}\hat{m}_{12}z_{22}p_{11}p_{12}p_{21})}$$

Here too, we select n_{ij} and $\hat{m}_{ij}$ such that the dipole disappears at diagonal entries of Q. Thus, $G = N \cdot G_{1mn} \cdot M^{-1}$.

Following the procedures outlined in the preceding section, we can find a set of M and N matrices such that the transformed Q matrix is more suitable for design. It should be noted that we have given only one possible way of finding M and N matrices. Many more M and N sets are possible.

3.4 Design Procedure for the Transformed NS MIMO QFT

It is straightforward to apply the NS MIMO QFT design procedure to the transformed system. This procedure is illustrated with the flow chart of Figure 3.2.

One starts from obtaining a TFM for the plant and also getting familiar with the MIMO poles and zeros, i.e. the properties of the system. Then verify that the diagonal dominance and the necessary and sufficient condition are satisfied before deciding to use QFT. After that, one derives the Q-matrix from the inverse plant P^{-1}. If one of the equivalent SISO plant possesses any RHP dipole, then use the transformed NS MIMO QFT. Otherwise, one can just simply apply the standard NS MIMO QFT.

In order to find the matrices pair, one can use the method provided in the preceding section. The entries of the matrices are solved for from augmented polynomials assuming free r_i s are known. Free r_i s are the roots of the stable polynomial. Then one searches for a proper set of free r_i s such that the roots of the matrix determinant are all in the open LHP.

Once the transformation matrices are obtained, one applies them to the original plant and gets the transformed plant. The nominal Q-matrix from the transformed plant should now be free of the RHP dipoles. Hence, one can do a quick check by assigning a large/small gain to the unstable/NMP equivalent SISO plant and then look at the nominal closed-loop stability of the original plant.

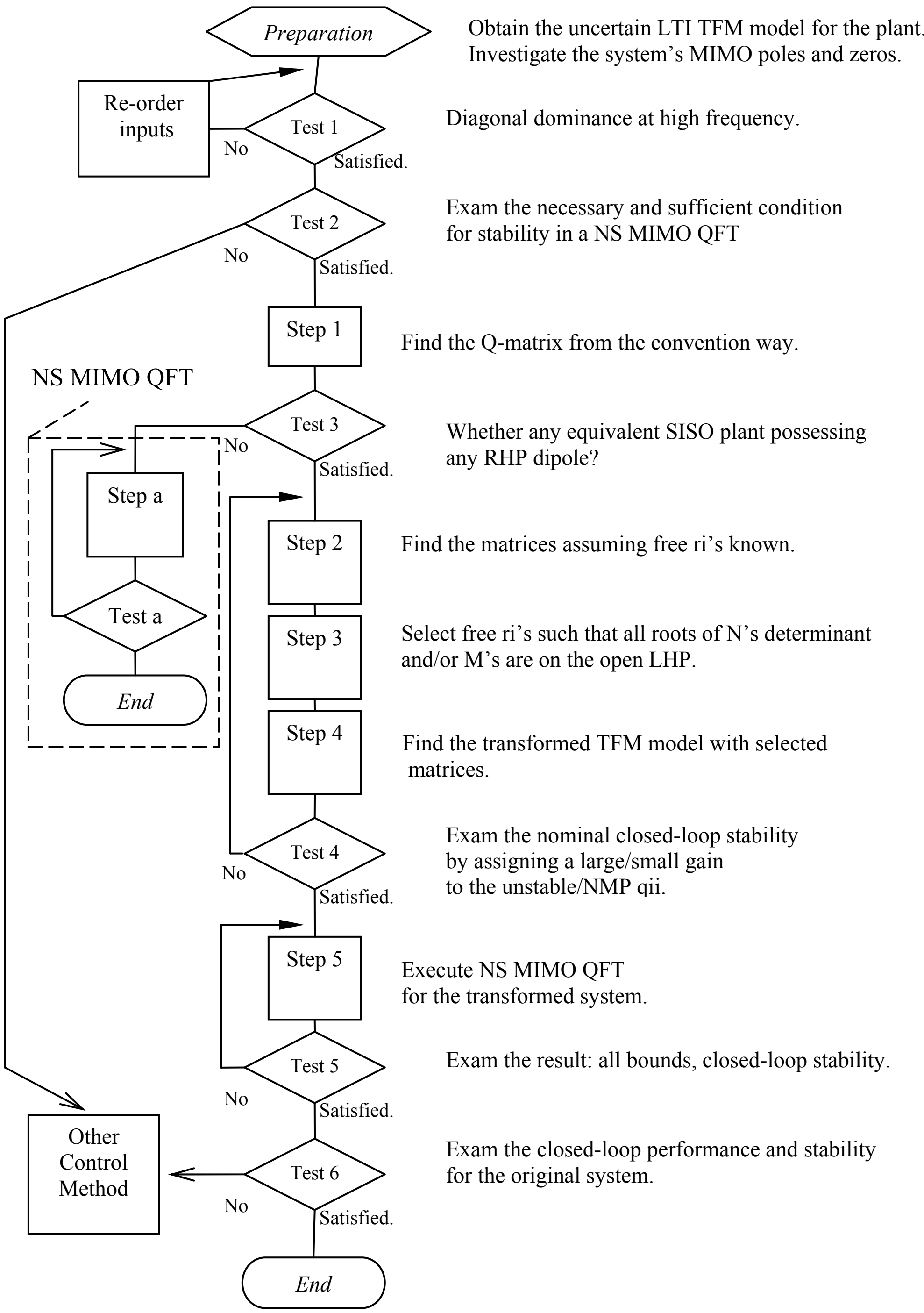

Fig. 3.2 Flow chart visualization for transformed NS MIMO QFT

If everything goes as expected in executing the preceding steps then one proceeds with the NS MIMO QFT on the transformed plant (step5 and test 5) and synthesizes a diagonal controller G_1. Finally, G_1 is inverse transformed to find G and verify the RS and RP specifications of the original plant with G.

3.5 Summary

The transformed NS MIMO QFT is illustrated in this chapter. The way in which those two transformation matrices function is explained as well. Then an approach to determine a set of transformation matrices is provided. With this set of transformation matrices, the equivalent SISO plants will have desirable pole-zero locations. Hence, the standard NS MIMO QFT design methodology can be successfully executed on the transformed plant. Finally, the straightforward procedure to apply the transformed scheme is illustrated with a flow chart.

CHAPTER IV

ILLUSTRATIVE EXAMPLE

4.1 Introduction

In this chapter, the transformed NS MIMO QFT methodology is demonstrated through a concocted control problem. The Singular G Method is also used to solve the concocted problem. Both methods and the results obtained are compared.

4.2 Problem Statement

The considered example is a control problem of a 2×2 LTI uncertain plant which possesses a RHP dipole. The 2×2 LTI uncertain plant P is as described below.

$$P=\begin{bmatrix}\frac{k}{s+a} & \frac{-k}{s+a}\\ \frac{k}{s+a} & \frac{k}{s-2}\end{bmatrix},\quad \text{where } 1\le k\le 2 \text{ and } 1\le a\le 2. \tag{4.1}$$

It is required to design a (fully populated / diagonal) controller G such that for all plants in the family the closed-loop system is robustly stable. Robust performance specifications are not considered in this example as the objective here is to demonstrate the stabilization of a system with RHP dipoles.

The nominal plant is chosen as the specific plant corresponding to $k=1$, and $a=1$. Thus,

$$P_0=\begin{bmatrix}\frac{1}{s+1} & \frac{-1}{s+1}\\ \frac{1}{s+1} & \frac{1}{s-2}\end{bmatrix} \tag{4.2}$$

, and

$$Det(P_0)=\frac{(2s-1)}{(s-2)(s+1)^2}. \tag{4.3}$$

It is obvious that there is a dipole in each member of the plant family since

$$Det(P) = \frac{k^2(2s+a-2)}{(s+a)^2(s-2)} \cdot \quad (4.4)$$

Consequently using the standard NS MIMO QFT it is impossible or extremely difficult to solve this problem.

4.3 Using the Transformed NS MIMO QFT

The procedure developed in chapter Ⅲ is applied to this example.

4.3.1 Selecting the transformation matrices

The first objective here is to determine M and N matrices based on the nominal plant so that the resulting two SISO equivalent nominal plants do not possess any dipoles.

If one tries to use the conventional NS MIMO QFT on the nominal plant P_0, then the resulting Q-matrix is:

$$Q_0 = \begin{bmatrix} \frac{(2s-1)}{(s+1)^2} & \frac{(2s-1)}{(s+1)(s-2)} \\ \frac{-(2s-1)}{(s+1)(s-2)} & \frac{(2s-1)}{(s+1)(s-2)} \end{bmatrix} \quad (4.5)$$

It is noticed that the entry q_{011} is N.M.P and stable, but that entry q_{022} has a dipole.

Suppose we let M be the identity matrix and apply only the N matrix transformation to the system such that $P_n = P \times N$ where

$$N = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}. \quad (4.6)$$

All entries of N can be constants or polynomials of s and N^{-1} must exist.

With this choice, the nominal equivalent plant matrix (Q-matrix) is:

$$Q_{0n} = \begin{bmatrix} \frac{(n_{11}n_{22} - n_{12}n_{21})(2s-1)}{((n_{12}+n_{22})s - 2n_{12} + n_{22})(s+1)} & \frac{-(n_{11}n_{22} - n_{12}n_{21})(2s-1)}{(n_{12}-n_{22})(s+1)(s-2)} \\ \frac{-(n_{11}n_{22} - n_{12}n_{21})(2s-1)}{((n_{11}+n_{21})s - 2n_{11} + n_{21})(s+1)} & \frac{(n_{11}n_{22} - n_{12}n_{21})(2s-1)}{(n_{11}-n_{21})(s+1)(s-2)} \end{bmatrix}. \quad (4.7)$$

From above, we wish to select n_{12} and n_{22} such that q_{0n11} will be N.M.P. and stable, and n_{11} and n_{21} such that q_{0n22} will be M.P and unstable.

Consequently, we will enforce the following constraints.

$$\left(n_{11}n_{22}-n_{12}n_{21}\right)=f(s),\ \text{where f(s) is stable} \tag{4.8}$$

$$\theta_1=\left(n_{12}+n_{22}\right)s-2n_{12}+n_{22}\quad be\quad Stable \tag{4.9}$$

$$\theta_2=\left(n_{11}-n_{21}\right)=\left(2s-1\right)\quad \text{cancels the unstable zero} \tag{4.10}$$

From (4-10), we let $\begin{cases} n_{11}=as+b \\ n_{21}=cs+d \end{cases} \Rightarrow \begin{cases} a-c=2 \\ b-d=-1 \end{cases}$

Thus, arbitrarily choose $\begin{cases} a=3 & b=2 \\ c=1 & d=3 \end{cases}$

From (4-9), we require $\begin{cases} n_{12}+n_{22}>0 \\ n_{22}-2n_{12}>0 \end{cases}$.

Hence we choose $\begin{cases} n_{12}=1 \\ n_{22}=3 \end{cases}$

As a result, $N=\begin{bmatrix} 3s+2 & 1 \\ s+3 & 3 \end{bmatrix}$ and $Det(N)=8s+3$. This leads to the following Q given by

$$Q_{0n}=\begin{bmatrix} \dfrac{(8s+3)(2s-1)}{(4s+1)(s+1)} & \dfrac{0.5(8s+3)(2s-1)}{(s+1)(s-2)} \\ \dfrac{-(8s+3)}{(s+1)(2s+1)} & \dfrac{(8s+3)}{(s+1)(s-2)} \end{bmatrix}. \tag{4.11}$$

Now the transformed plant is in a manageable form. It is easy to stabilize the two diagonal entries defining the two SISO equivalent plants with a small gain for q_{0n11} and a large gain for q_{0n22}. For verification, we simply choose

$$G_{0n}=\begin{bmatrix} 0.3 & 0 \\ 0 & 20 \end{bmatrix}. \tag{4.12}$$

Therefore, the controller used for the original nominal plant is:

$$G_0=N\times G_{0n}=\begin{bmatrix} 0.9s+0.6 & 20 \\ 0.3s+0.9 & 60 \end{bmatrix}. \tag{4.13}$$

The resulting closed-loop TFM and sensitivity TFM are:

$$T_0 = (I + P_0G_0)^{-1} \times (P_0G_0)$$
$$= \frac{\begin{bmatrix} 3(2s-1)(s^2+159s+58) & -400(s+1)(s-2) \\ 3(2s-1)(2s+1)(s+1) & 20(88s^2+44s+1) \end{bmatrix}}{16(s+0.0072)(s+0.4751)(s+108.9551)} \quad (4.14)$$

$$S_0 = (I + P_0G_0)^{-1}$$
$$= \frac{\begin{bmatrix} 10(s^2+79s+18)(s+1) & 400(s+1)(s-2) \\ -3(2s-1)(2s+1)(s+1) & (16s+7)(s+1)(s-2) \end{bmatrix}}{16(s+0.0072)(s+0.4751)(s+108.9551)} \quad (4.15)$$

The nominal closed-loop 2×2 system with controller G_0 and plant P_0 is stable since the denominator in S_0 is stable and there are no RHP pole-zero cancellations between P_0 and G_0.

4.3.2 The transformed problem

We now apply the chosen M and N to our original problem and transform it to a new design problem.

Using the transformation $P_1 = M^{-1} \times P \times N$ the Transformed Problem is:

$$P_1 = \begin{bmatrix} \frac{k(2s-1)}{s+a} & \frac{-2k}{s+a} \\ \frac{k(4s^2+(a-1)s+3a-4)}{(s+a)(s-2)} & \frac{k(4s-2+3a)}{(s+a)(s-2)} \end{bmatrix} \text{ where } 1 \le k \le 2 \text{ and } 1 \le a \le 2 \quad (4.16)$$

We wish to design a controller $G_1 = \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}$ such that for all P_1 the closed-loop system is robustly stable.

It is worth noting that in this transformed NS MIMO QFT design problem, the nominal equivalent SISO plants are free from any dipoles. Thus, one can easily stabilize the nominal plant (such as G_{0n} in 4.3.1).

4.3.3 Applying the standard NS MIMO QFT

We now perform the standard NS MIMO QFT design on the transformed design problem. The plant templates for each loop are shown in Fig. 4.1 and Fig. 4.2.

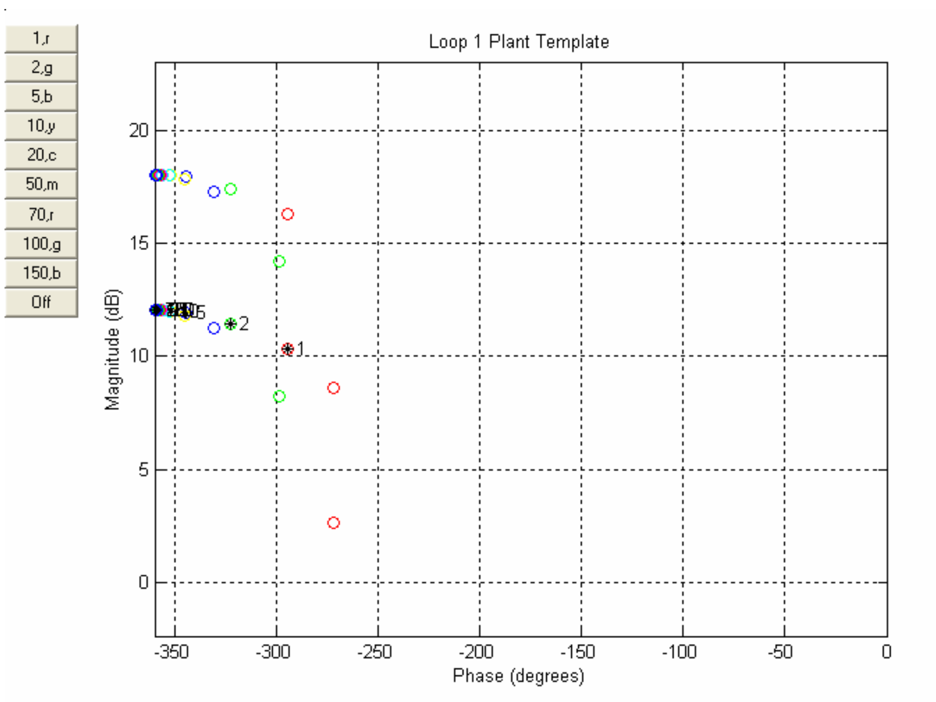

Fig. 4.1 Loop 1 plant template

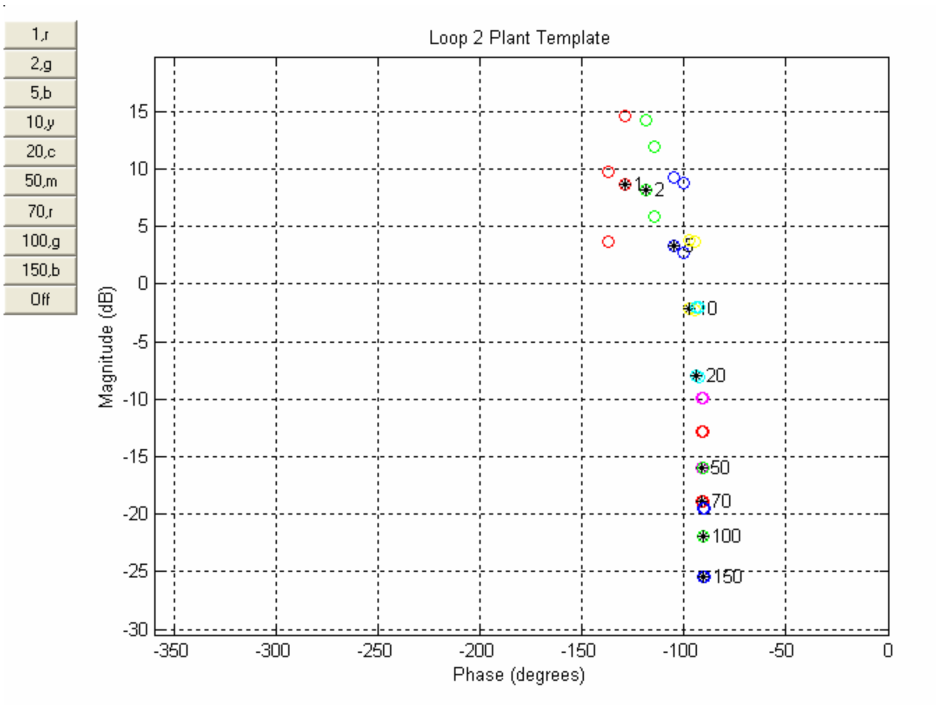

Fig. 4.2 Loop 2 plant template

The chosen controller for loop 1 (equivalent SISO plant 1/ 11 entry of the Q matrix) is

$$g_1 = \frac{0.12}{s+1} \tag{4.17}$$

and for loop 2 (equivalent SISO plant 2/ 22 entry of the Q matrix) is

$$g_2 = \frac{81600}{(s+17)(s+800)} \,. \tag{4.18}$$

Thus,

$$G_1 = \begin{bmatrix} \frac{0.12}{s+1} & 0 \\ 0 & \frac{81600}{(s+17)(s+800)} \end{bmatrix} . \tag{4.19}$$

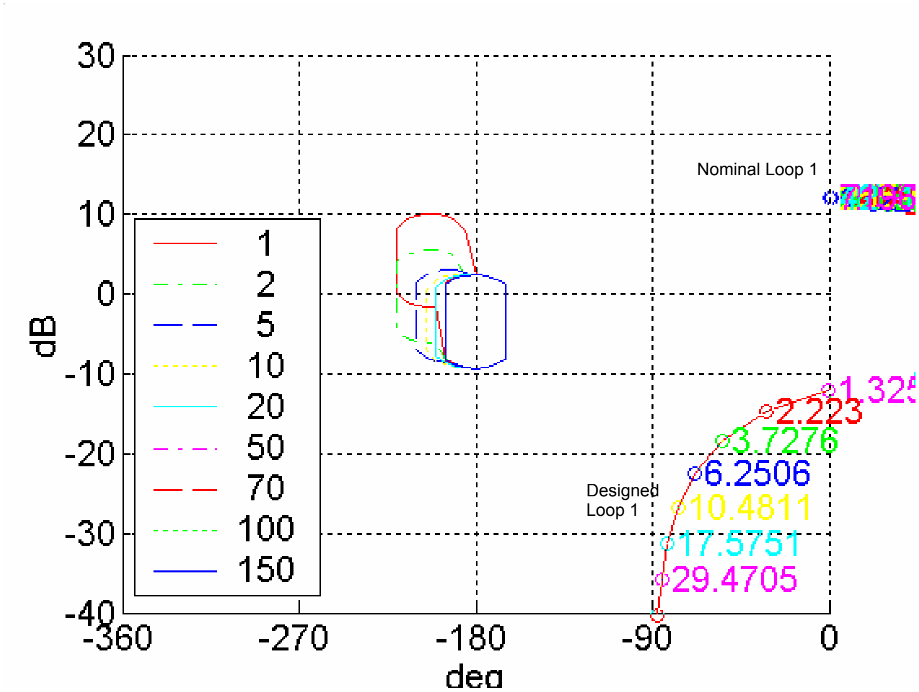

Fig. 4.3 Loop-shaping result for loop 1

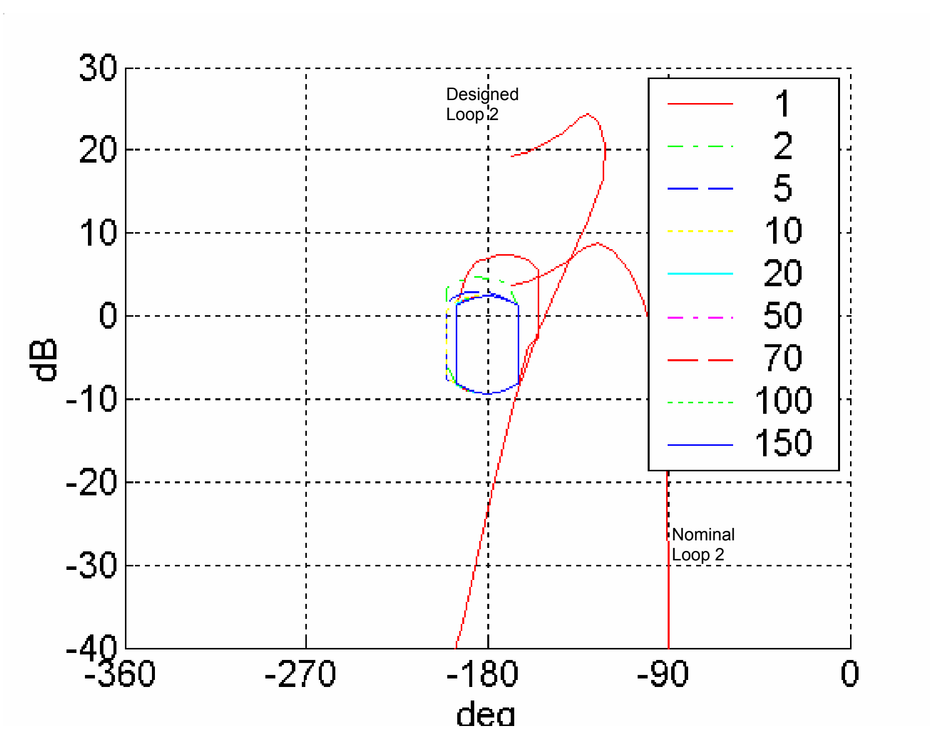

Fig. 4.4 Loop-shaping result for loop 2

As seen from loop-shaping results (Figure 4.3 and Figure 4.4), loop 1 can be easily stabilized by a small gain and loop2 can be easily stabilized by a large gain. The U-contour here was arbitrarily assigned for design in the transformed problem. However, we did not specify any robust performance (RP) requirements, especially the cross-coupling RP specifications. Cross-coupling RP specifications will show up as disturbances in loop 1 and loop 2. Therefore, we have to use higher (lower) gain in loop 1(loop 2) design. In other words, the RP specification is more critical than the arbitrarily assigned robust stability specification. Indeed, if the RP specifications and the RS specification are satisfied, the closed-loop system's robust stability is guaranteed by the NS MIMO QFT robust stability theorem (Theorem 1, [4]). Finally, we add some high frequency poles to the controller so that the controller has limited bandwidth.

4.3.4 Inverse transformation and closed-loop stability verification

We now check the effect of the final design on the original system. The real controller for our original uncertain plant is given by,

$$G = N \times G_1 = \begin{bmatrix} \frac{3(3s+2)}{25(s+1)} & \frac{81600}{(s+17)(s+800)} \\ \frac{3(s+3)}{25(s+1)} & \frac{244800}{(s+17)(s+800)} \end{bmatrix} \tag{4.20}$$

We now calculate some closed-loop sensitivity transfer function matrices for checking stability.

At $k = 1$ and $a = 1$, the denominator of the sensitivity TFM is:

$$25s^6 + 20456s^5 + 365243s^4 + 8512838s^3 + 21024878s^2 + 9892052s + 707200 =$$
$$25(s+800.5194)(s+2.1038)(s+0.5049)(s+0.087)(s+7.5124+18.0506i)(s+7.5124-18.0506i)$$

At $k = 1$ and $a = 2$, the denominator of the sensitivity TFM is:

$$25s^6 + 20506s^5 + 406124s^4 + 9217975s^3 + 35571104s^2 + 37708804s + 13763200 =$$
$$25(s+800.5196)(s+2.9953)(s+0.7031+0.3166i)(s+0.7031-0.3166i)$$
$$(s+7.6595+18.0965i)(s+7.6595-18.0965i)$$

At $k=2$ and $a=1$, the denominator of the sensitivity TFM is:

$$25s^6+20462s^5+370136s^4+16747076s^3+51005531s^2+20545754s+625600=$$
$$25(s+801.0372)(s+2.745)(s+0.4368)(s+0.0331)(s+7.1139+27.1189i)(s+7.1139-27.1189i)$$

At $k=2$ and $a=2$, the denominator of the sensitivity TFM is:

$$25s^6+20512s^5+411023s^4+17457100s^3+79901108s^2+82598608s+30246400=$$
$$25(s+801.0375)(s+3.6796)(s+0.6339+0.3433i)(s+0.6339-0.3433i)$$
$$(s+7.2476+27.1537i)(s+7.2476-27.1537i)$$

Roots of those denominators are all negative. Indeed, the poles of the sensitivity TFM is plotted with various plant parameters as in Figure 4.5 and all the poles lie in the open LHP. Moreover, there is no pole-zero cancellation between P and G. Thus, the uncertain plant is stabilized with the controller G.

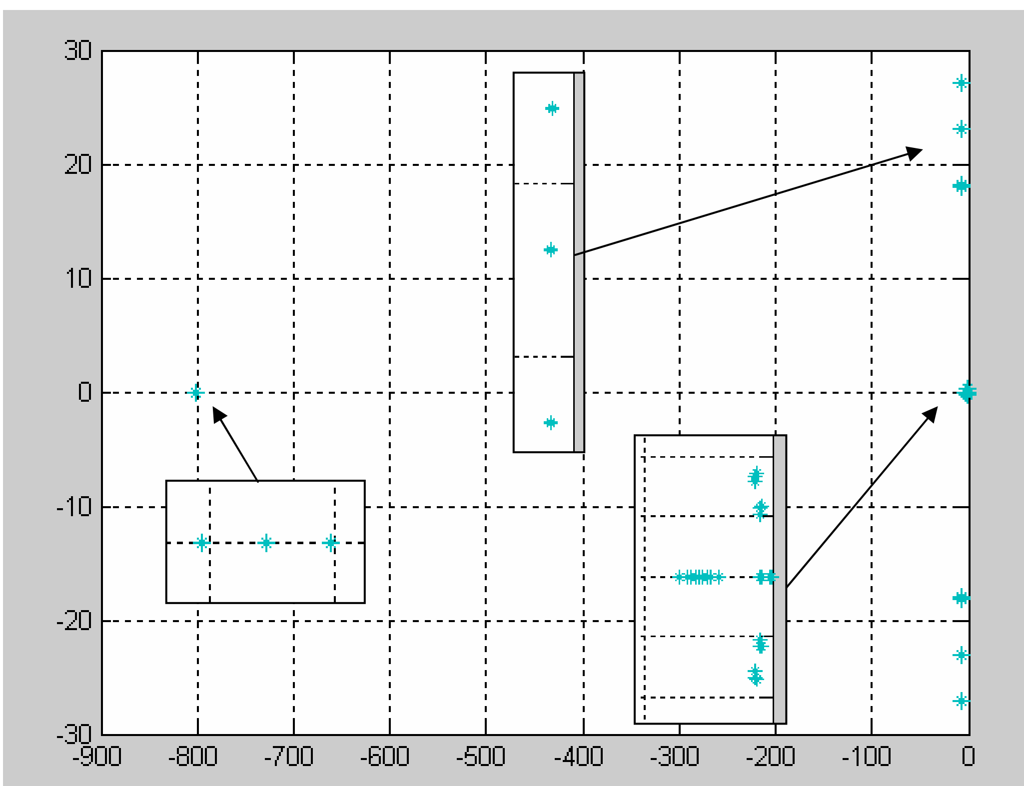

Fig. 4.5 Poles of sensitivity TFM with various parameters

The singular value plot of this sensitivity TFM is also provided in Figure 4.6. The system has both sensitivity reduction and sensitivity gain in the low frequency depending on the directions. This is because of the RHP dipole in the plant. Moreover, an impulse response simulation for the closed-loop system is shown as in Figure 4.7. As expected, all channels decay to almost zero after certain time.

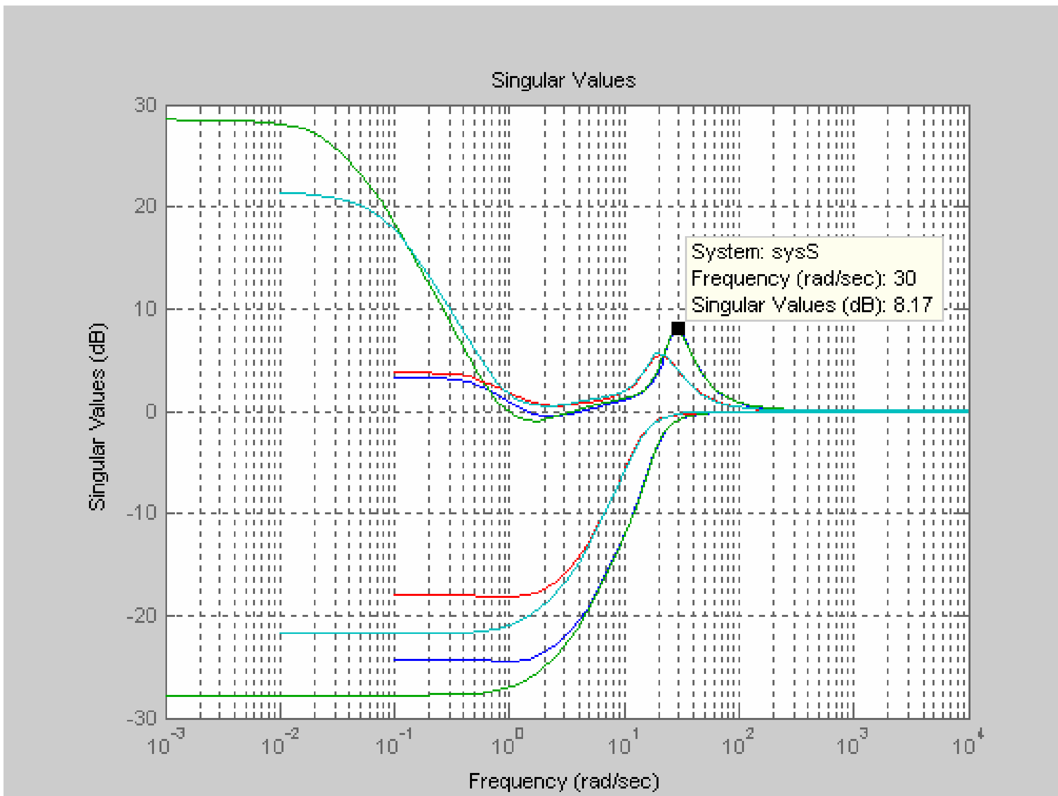

Fig. 4.6 Singular value plot of sensitivity TFM with various parameters

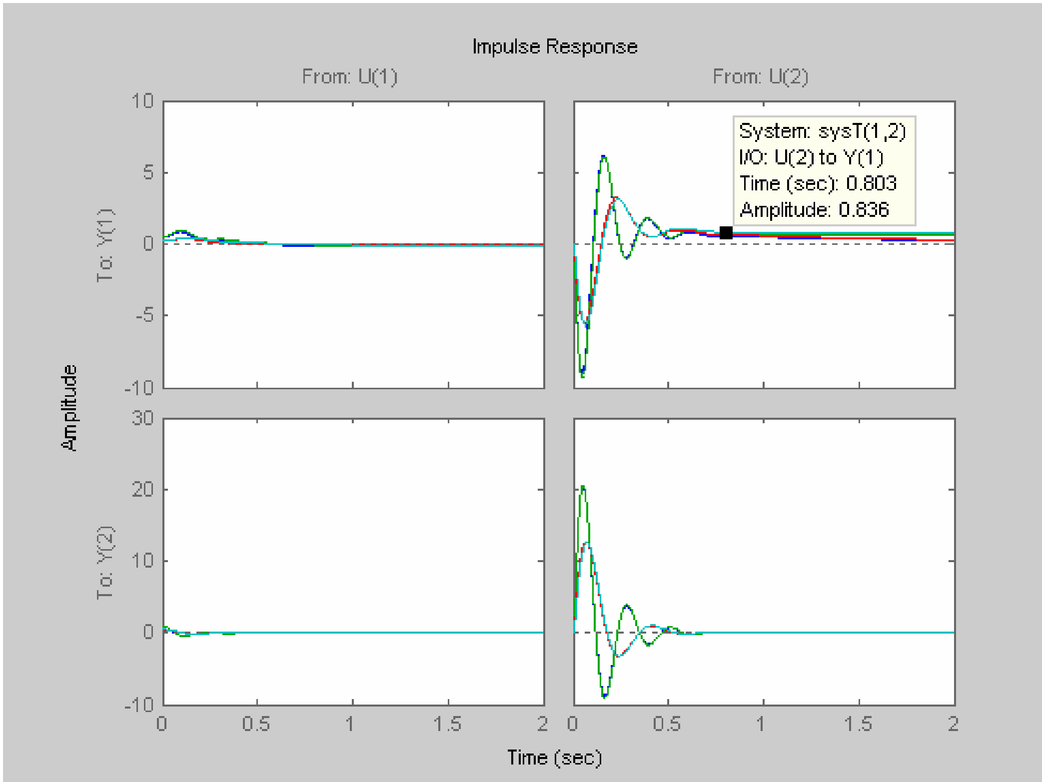

Fig 4.7 Closed-loop TFM impulse response simulation with various parameters

4.4 Using Singular G Method

In this section, we apply the Singular G Method, which has been described in Chapter Ⅰ, to the example problem. Since in the example problem it does not ask for any performance, we do not need to worry about the prefilter design or just simply set it to be a vector $\begin{bmatrix}1\\1\end{bmatrix}$. Then first step is to find the equivalent plant, which from equation (1.12) is:

$$P_e = k\left[\frac{k_1 g_1}{s+a} + \frac{-k_1 g_2}{s+a} + \frac{k_2 g_1}{s+a} + \frac{k_2 g_2}{s-2}\right] \tag{4.21}$$

Similarly, we let $g_1 = b(g_2)$ and $k_2 = c(k_1)$ to simplify the calculation. Therefore, we determine the closed-loop stability from equation (1.15), which results in equation (4.22).

$$1+HP_e = 1+kH\left[\frac{bk_1g_2}{s+a}+\frac{-k_1g_2}{s+a}+\frac{bck_1g_2}{s+a}+\frac{ck_1g_2}{s-2}\right]$$

$$=1+k_1g_2\left(\frac{kH}{(s+a)(s-2)}\right)[(bc+b+c-1)s+(2-2b-2bc+ac)] \qquad (4.22)$$

From this point on, we see k_1g_2H as a controller to be designed to the equivalent plant as:

$$P_e=\frac{k[(bc+b+c-1)s+(2-2b-2bc+ac)]}{(s+a)(s-2)} \qquad (4.23)$$

Here, we try to make the equivalent plant minimum phase with proper choice of b and c. This can be searched with a computer program since it is a nonlinear function. One pair of working results is $b=0.5$ and $c=0.4$.

The next step is to find the controller, k_1g_2H, which stabilizes the equivalent plant. Let $H^{'}=k_1g_2H$. We then enforce that the denominator of the closed-loop equivalent plant, $\frac{P_eH^{'}}{1+P_eH^{'}}$, has all its roots at the open LHP. This is the same as asking equation (4.24) has all its roots at the open LHP.

$$kH^{'}[[(bc+b+c-1)s+(2-2b-2bc+ac)]]+(s+a)(s-2)$$
$$=s^2+\left(kH^{'}(bc+b+c-1)+a-2\right)s+\left(kH(2-2b-2bc+ac)-2a\right) \qquad (4.24)$$

Thus, we find out that if $H^{'}\geq 10$ the closed-loop equivalent system is robustly stable. Indeed, this step can be accomplished with SISO QFT as well. Suppose we have $H^{'}=15$. Then the resulting controller is:

$$G=H\begin{bmatrix} b & bc \\ 1 & c \end{bmatrix}=\begin{bmatrix} 7.5 & 3 \\ 15 & 6 \end{bmatrix} \qquad (4.25)$$

We can then obtain the sensitivity TFM of the uncertain plant with the controller G.

$$S=\begin{bmatrix} \frac{2[s^2+(a+9k-2)s+(6ka-2a-6k)]}{2s^2+(3k-4+2a)s+(18k+12ka-4a)} & \frac{6[k(s-2)]}{2s^2+(3k-4+2a)s+(18k+12ka-4a)} \\ \frac{-15[k(3s+2a-2)]}{2s^2+(3k-4+2a)s+(18k+12ka-4a)} & \frac{(2s+2a-15k)(s-2)}{2s^2+(3k-4+2a)s+(18k+12ka-4a)} \end{bmatrix}$$

(4.26)

The roots for the denominator of this sensitivity TFM are plotted in Figure 4.8 with various plant parameters. As seen from the figure, all roots are in the open LHP. Moreover, there is no pole-zero cancellation between the controller and the plant. Therefore, we can conclude that the closed-loop system is robustly stable.

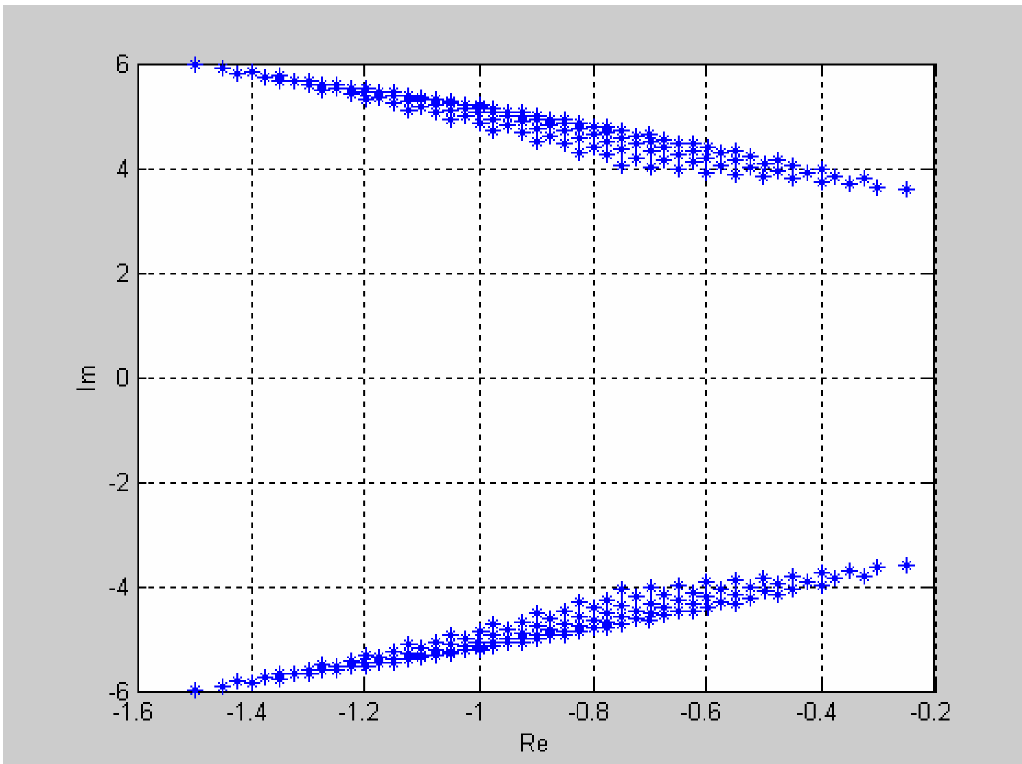

Fig 4.8 Poles of the sensitivity TFM with various parameters

The singular value plot of this sensitivity TFM is also provided in Figure 4.9 and all singular values are the same for different parameters. Also the system has both sensitivity reduction and sensitivity gain in the low frequency. However, the sensitivity gain is relatively small compared with the result from transformed NS MIMO QFT. This implies that the Singular G Method does not shape the loop corresponding to NMP zero.

Moreover, an impulse response simulation for the closed-loop system is shown as in Figure 4.10. As expected, all channels decay to almost zero after certain time. If one simply compares the impulse responses from both methods, the result from the transformed NS MIMO QFT has faster response and smaller amplitude.

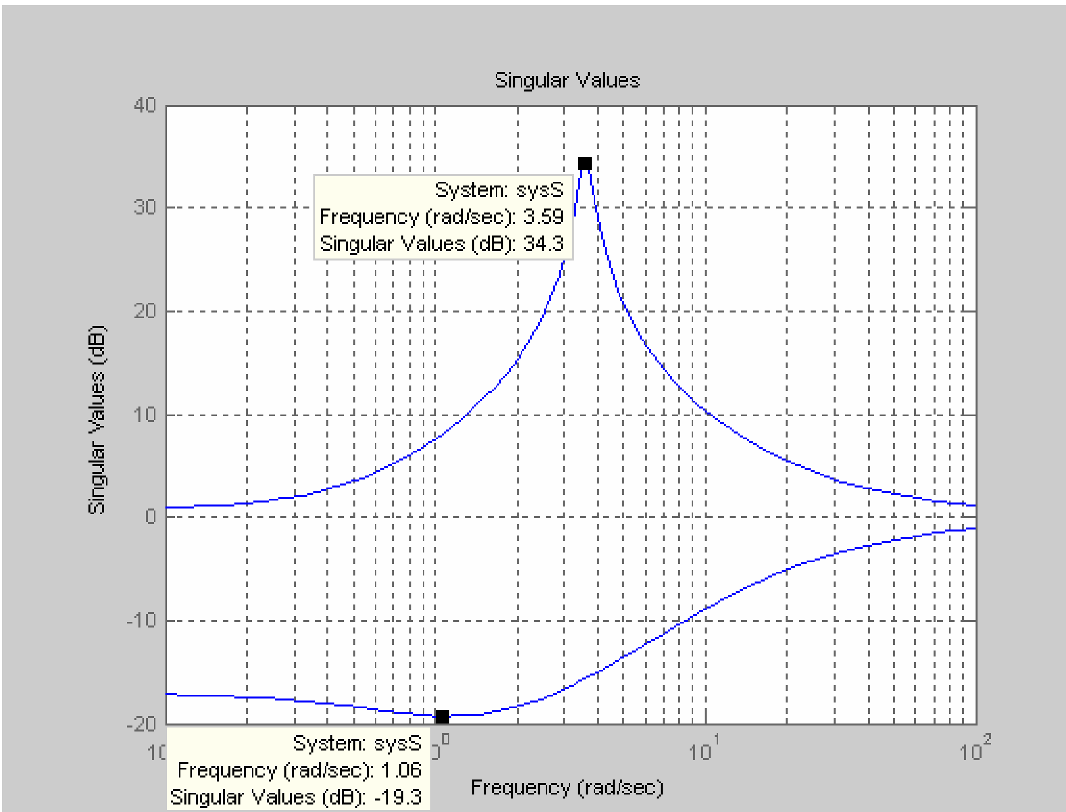

Fig 4.9 Singular value plot of the sensitivity TFM for various parameters

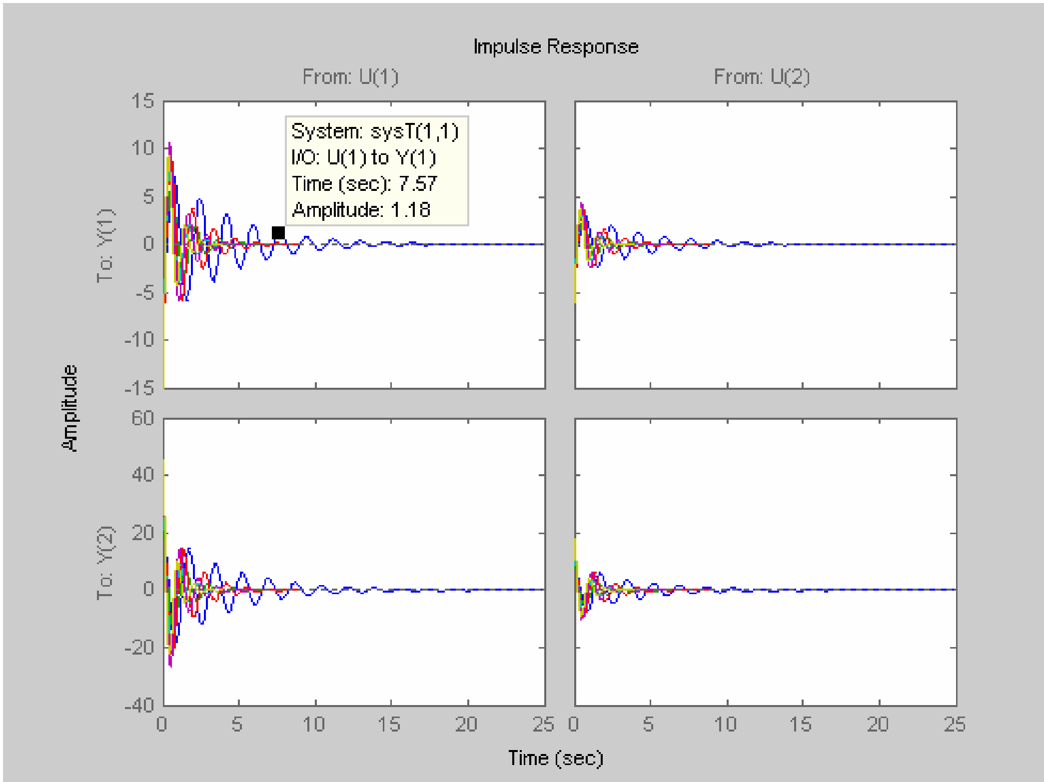

Fig. 4.10 Closed-loop TFM impulse simulation with various parameters

4.5 Comparison

From Figure 1.4 the block diagram from Singular G Method can be redrawn as the block diagram in Figure 4.11. In using the Singular G Method as introduced in chapter I, r is treated as a single reference input. In this case, the controller, G, can not be moved to the feed-forward path right before P since G^{-1} does not exist, and hence one can not get the unity feedback closed-loop structure as in Figure 1.2. Also c_i s are treated as a sensor noise disturbance. However, if one takes a different view point and sees c_i s as the reference inputs with unity feedback, the Singular G Method indeed is tracking an input disturbance, r. In fact, this view point is more similar to the normal control structure and the result for closed-loop stability is the same no matter whether one treats r or c_i s as the reference input as long as there is no RHP pole-zero cancellation between P and G.

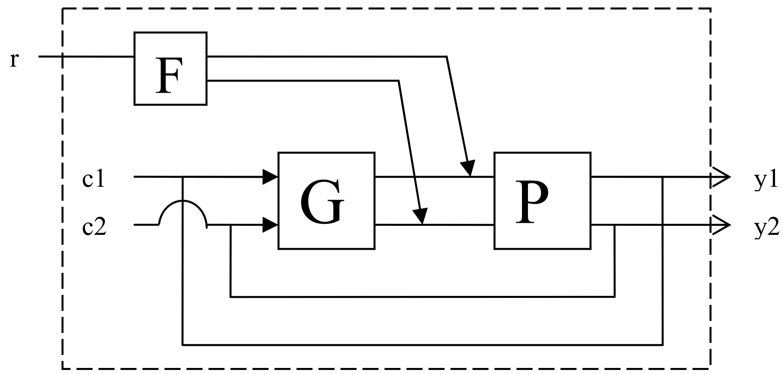

Fig 4.11 Closed-loop system from Singular G Method

Although the results from the Singular G Method do stabilize the plant, one obvious drawback is that both outputs are tracking the same reference input. Hence, the outputs are not independent. Actually, even though one sees c_i s as the reference input, the outputs are still dependent on each other and the linear combination of c_i s. The reason for this is that the closed-loop TFM from outputs to the c_i s is singular.

The controller, G, from the Singular G Method can be expressed as a diagonal matrix, D, pre-multiplied with a singular matrix, N_s.

$$G_s = H\begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix} = \begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix} H = \begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix}\begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} = N_s D \quad (4.27)$$

Here,

$$D = H \times I = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} \quad (4.28)$$

and,

$$N_s = \begin{bmatrix} k_1 g_1 & k_2 g_1 \\ k_1 g_2 & k_2 g_2 \end{bmatrix}. \quad (4.29)$$

Therefore, one can modify the block diagram in Figure 4.11 to the one shown in Figure 4.12.

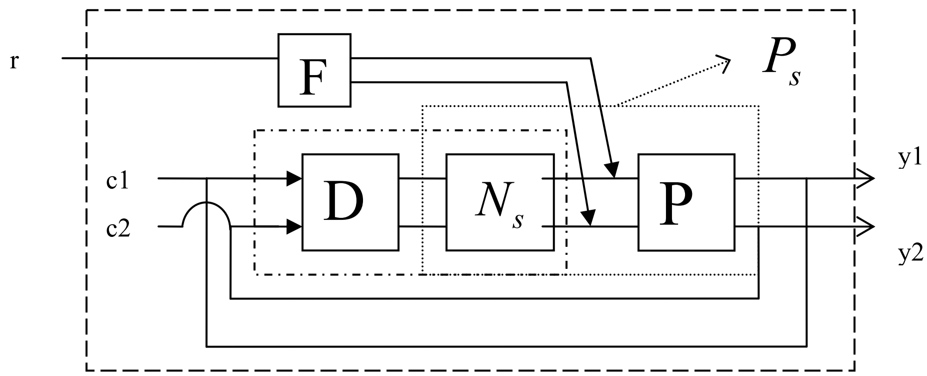

Fig 4.12 Modified block diagram for Singular G Method

The structure of the Singular G Method is then revealed. For stability, if one sees the c_i s as the reference inputs, the Singular G Method can be stated as: find a singular transforming matrix N_s such that a transformed singular plant, $P_s = P \times N_s$, is stabilized from a diagonal controller D which has the same diagonal entries, H. This H is determined from stabilizing a SISO equivalent plant, P_e. The reason is that the closed-loop stability of this transformed singular plant is determined from the poles of its sensitivity function, and then from equation (4.30) this is the same as the poles of the SISO equivalent plant's closed-loop system.

$$\det(I + DP_s) = 1 + HP_e \tag{4.30}$$

Here, the choice of the singular matrix N_s should release the RHP dipole of the SISO equivalent plant by eliminating the RHP zero from the original plant, P. Moreover, all entries of N_s can be just constants or/and polynomials of s. Finally, the controller is obtained from the diagonal controller D by multiply it with the singular transforming matrix N_s.

Therefore in the design of a controller to stabilize the plant, the Singular G Method, when one sees the c_i s as the reference inputs, can be compared to transformed NS MIMO QFT using only N matrix, which results in a block diagram as Figure 4.13. The transformed plant is $P_n = P \times N$. Then one applies the NS MIMO QFT to this transformed plant and obtains a diagonal controller G_n. At last, the real controller is found by inverse transforming the diagonal controller G_n. Here, the transforming matrix N is selected such that the transformed plant has desired structures in the nominal equivalent SISO plants for NS MIMO QFT.

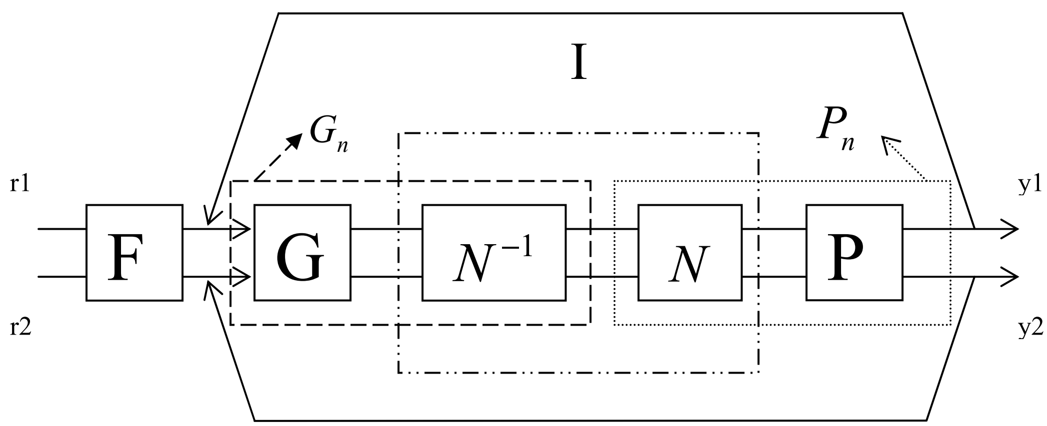

Fig 4.13 N matrix transformation for NS MIMO QFT

Following the above view point of the Singular G Method, the effectiveness of these two methods in stabilizing a plant are then compared in three ways: similarities, differences, and difficulties.

i Similarities:

Both methods use a transforming matrix to condition their equivalent SISO plant(s) for QFT design and then obtain a diagonal controller, D or G_n, from the transformed plant. Both methods result in a fully populated controller.

ii Differences:

The transforming matrix, N_s, is singular in the Singular G Method while it, N , has to be non-singular in the transformed NS MIMO QFT. In the case of a 2X2 plant: After transforming, the Singular G Method obtains a singular transformed plant and from this transformed plant directly gets a SISO equivalent plant for SISO QFT design. In transformed NS MIMO QFT, two equivalent SISO plants are obtained from the inverse of the transformed plant and both the closed-loop systems of these two equivalent SISO plants have to be stable after applying SISO QFT design. The result from the Singular G Method has one degree-of-freedom since the outputs are dependent. In contrast, the result from the NS MIMO QFT still has independent control of the outputs. The resulting controller from the Singular G Method is singular.

iii Difficulties:

Both methods need to solve nonlinear equations when choosing the transforming matrices. This is usually done with the help of a computer program.

Consequently, it is safe to say that Singular G Method is an extreme case in applying the transforming scheme.

CHAPTER V

SYNTHESIS OF A CONTROLLER FOR THE LONGITUDINAL FLIGHT CONTROL OF X-29 AIRCRAFT USING TRANSFORMED NS MIMO QFT

5.1 Introduction

In this chapter, the transformation scheme developed is applied to the X-29 longitudinal flight control problem. The procedure if executed on the basis of the X-29 plant which violates the diagonal dominance condition the resulting controller would fail to stabilize the plant. After switching the order of the inputs, the diagonal dominance is satisfied. However, a proper set of transforming matrices for this plant with switched inputs has not been found yet due to the time consuming nature of iteration. Thus, the complete resolution of the design was not achieved. Nevertheless, a full structure to solve this flight control problem is outlined and explained in the following sections.

5.2 Plant Model

The Grumman forward swept wing demonstrator aircraft, the X-29, is designed to demonstrate the forward swept wing technology. This aircraft is known for its high maneuverability and open-loop instability.

The plant mode 1 is originally from Walke's thesis ([5]) at the Air Force Institute of Technology. It was derived in state space form and then simplified into a state space form of 2 inputs, 2 outputs, and 4 states for longitudinal flight. This also means the system has 4 poles. Since a TFM model is required for NS MIMI QFT, it is obtained from MATLAB and has a minimal realization as the state space model.

$$P = \begin{bmatrix} \dfrac{z_{11}}{p_{11}} & \dfrac{z_{12}}{p_{12}} \\ \dfrac{z_{21}}{p_{21}} & \dfrac{z_{22}}{p_{22}} \end{bmatrix}, \; p_{11} = p_{12} = p_{21} = p_{22} = \Phi(s) \tag{5.1}$$

The longitudinal flight model of X-29 is described by four flight conditions: (0.9 Mach, sea level), (0.4 Mach, sea level), (0.7 Mach, 15K) and (0.9 Mach, 50K) ([5]). The corresponding plant transfer function matrix (TFM) for those flight conditions is described below:

Flight Condition 1 (0.9 Mach, sea level):

$$P_1 = \frac{\begin{bmatrix} .1172s^4 + .4216s^3 + 65.4671s^2 + 4.1325s - .0008 & .3481s^4 + .3005s^3 - 70.3135s^2 - 4.2903s + .0007 \\ .5577s^3 + 2.1318s^2 + .1352s & -.3264s^3 - 2.2732s^2 - .1391s \end{bmatrix}}{s^4 + 5.5906s^3 - 69.9868s^2 - 4.3334s - .3775} \tag{5.2}$$

Flight Condition 2 (0.4 Mach, sea level):

$$P_2 = \frac{\begin{bmatrix} .0091s^4 + .0193s^3 + 1.155s^2 + .0048s - .0003 & .071s^4 + .0241s^3 - 1.4211s^2 - .0041s + .0003 \\ .0858s^3 + .0848s^2 + .0013s & -.0502s^3 - .1031s^2 - .0009s \end{bmatrix}}{s^4 + 1.5182s^3 - 9.1427s^2 - 0.1456s - 0.0969} \tag{5.3}$$

Flight Condition 3 (0.7 Mach, 15K):

$$P_3 = \frac{\begin{bmatrix} .021s^4 + .0435s^3 + 4.2258s^2 + .0491s - .0002 & .145s^4 + .053s^3 - 7.4361s^2 - .077s + .0002 \\ .1589s^3 + .1863s^2 + .0028s & -.118s^3 - .3252s^2 - .004s \end{bmatrix}}{s^4 + 1.8445s^3 - 25.8258s^2 - .3460s - .0902} \tag{5.4}$$

Flight Condition 4 (0.9 Mach, 50K):

$$P_4 = \frac{\begin{bmatrix} .0044s^4 + .0042s^3 + .601s^2 + .0056s - .000014 & .0437s^4 + .0053s^3 - .7493s^2 - .0052s + .00001 \\ .0548s^3 + .0231s^2 + .0004s & -.0364s^3 - .0277s^2 - .0002s \end{bmatrix}}{s^4 + .6068s^3 - 7.3938s^2 - .2146s - .0489} \tag{5.5}$$

It can be noticed that all of the TFMs have at least one NMP zero at the origin and one unstable pole, i.e., the plant in the family is always NMP and unstable.

The Q-matrices for all four flight conditions obtained from the standard NS MIMO QFT methodology are as follows:

Flight Condition 1 (0.9 Mach, sea level):

$$Q_1 = \begin{bmatrix} \frac{0.71198(s+0.05899)(s+0.003335)}{(s+6.903)(s+0.06172)} & \frac{0.66759s(s+0.05899)(s+0.003335)}{(s+14.62)(s-13.82)(s+0.06116)(s-0.0001553)} \\ \frac{0.41669(s+0.05899)(s+0.003335)}{(s+3.758)(s+0.0645)} & \frac{-1.9828s(s+0.05899)(s+0.003335)}{(s+0.06333)(s-0.0001867)(s^2+3.534s+558.4)} \end{bmatrix} \quad (5.6)$$

Flight Condition 2 (0.4 Mach, sea level):

$$Q_2 = \begin{bmatrix} \frac{0.13042(s+0.01874)(s-0.01078)}{(s+2.046)(s+0.008433)} & \frac{0.092171s(s+0.01874)(s-0.01078)}{(s+4.645)(s-4.308)(s+0.01593)(s-0.01304)} \\ \frac{0.076329(s+0.01874)(s-0.01078)}{(s+0.9733)(s+0.01503)} & \frac{-0.72156s(s+0.01874)(s-0.01078)}{(s+0.01776)(s-0.01364)(s^2+2.126s+127.3)} \end{bmatrix} \quad (5.7)$$

Flight Condition 3 (0.7 Mach, 15K):

$$Q_3 = \begin{bmatrix} \frac{0.21626(s+0.01331)(s-0.0006885)}{(s+2.743)(s+0.01245)} & \frac{0.17599s(s+0.01331)(s-0.0006885)}{(s+7.341)(s-6.986)(s+0.01288)(s-0.002521)} \\ \frac{0.16059(s+0.01331)(s-0.0006885)}{(s+1.157)(s+0.01542)} & \frac{-1.2152s(s+0.01331)(s-0.0006885)}{(s+0.01434)(s-0.002717)(s^2+2.06s+201.2)} \end{bmatrix} \quad (5.8)$$

Flight Condition 4 (0.9 Mach, 50K):

$$Q_4 = \begin{bmatrix} \frac{0.070156(s+0.00935)(s-0.001167)}{(s+0.7542)(s+0.006622)} & \frac{0.058445s(s+0.00935)(s-0.001167)}{(s+4.197)(s-4.083)(s+0.008558)(s-0.001655)} \\ \frac{0.046672(s+0.00935)(s-0.001167)}{(s+0.4047)(s+0.01654)} & \frac{-0.57784s(s+0.00935)(s-0.001167)}{(s+0.01136)(s-0.002035)(s^2+0.9399s+135.9)} \end{bmatrix} \quad (5.9)$$

Since all diagonal entries of four Q-matrices have a RHP dipole they are extremely difficult if not impossible to be stabilized with SISO QFT and hence the MIMO plant can not be stabilized with NS MIMO QFT.

5.3 SVD for the Plant TFM and Transformed Plant TFM

The reason for this step is that a plant has to satisfy a necessary and sufficient condition such that the closed-loop robust stability can be guaranteed after a successful design is obtained from NS QFT. From this we know that one class of systems, which violates this requirement, is systems with pinned zero input and output directions. This is examined in this step.

Therefore, the singular value decomposition at zeros for the original plants and the transformed plants are preformed, and then the inner dot products of their input and output directions are examined. Fortunately, both plants are fine.

5.4 Finding the N Matrix

In this section, we try to find a proper N matrix and express all entices of the N matrix as functions of r_i s.

We first assume that

$$N = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}. \tag{5.10}$$

Then, we have the transformed plant from $P_{1_n} = P \cdot N$ and hence the diagonal entries of the transformed Q matrix.

$q_{an11} =$

$$\frac{-0.6285\left(\mathrm{n}_{11}\mathrm{n}_{22} - \mathrm{n}_{21}\mathrm{n}_{12}\right)(s+0.05899)(s+0.003335)}{\left(-0.8827\mathrm{n}_{22} + 1.508\mathrm{n}_{12}\right)s^2 + \left(-6.148\mathrm{n}_{22} + 5.765\mathrm{n}_{12}\right)s - 0.3760\mathrm{n}_{22} + 0.3656\mathrm{n}_{12}} \tag{5.11}$$

$q_{an22} =$

$$\frac{-2.447\left(\mathrm{n}_{11}\mathrm{n}_{22}-\mathrm{n}_{21}\mathrm{n}_{12}\right)s\left(s+0.05899\right)\left(s+0.003335\right)}{\left(1.234\mathrm{n}_{11}+3.666\mathrm{n}_{21}\right)s^4+\left(4.440\mathrm{n}_{11}+3.164\mathrm{n}_{21}\right)s^3+\left(689.4\mathrm{n}_{11}-740.5\mathrm{n}_{21}\right)s^2+ \left(43.52\mathrm{n}_{11}-45.18\mathrm{n}_{21}\right)s-0.008148\mathrm{n}_{11}+0.007037\mathrm{n}_{21}} \tag{5.12}$$

We further let:

h1 = 0.8827;

h2 = 6.148;

h3 = 0.3760;

k1 = 1.508;

k2 = 5.765;

k3 = 0.3656;

and,

h4 = 1.23426005835549;

h5 = 4.439643755774853;

h6 = 6.894486424004172e + 002;

h7 = 4.351989959207325e + 001;

h8 = 8.148125087765034e - 003;

k4 = 3.665920872982476;

k5 = 3.164236031744317;

k6 = 7.404873433734643e + 002;

k7 = 4.518171440648887e + 001;

k8 = 7.036714625689909e - 003;

Then, we have the two polynomials:

$$\theta_1 = \left(-h_1 n_{22} + k_1 n_{12}\right)s^2 + \left(-h_2 n_{22} + k_2 n_{12}\right)s + \left(-h_3 n_{22} + k_3 n_{12}\right) \quad (5.13)$$

$$\theta_2 = \left(h_4 n_{11} + k_4 n_{21}\right)s^4 + \left(h_5 n_{11} + k_5 n_{21}\right)s^3 + \left(h_6 n_{11} - k_6 n_{21}\right)s^2 + \left(h_7 n_{11} - k_7 n_{21}\right)s + \left(-h_8 n_{11} + k_8 n_{21}\right) \quad (5.14)$$

Our goal here is to make q_{an11} M.P. but unstable and q_{an22} N.M.P. but stable. Therefore, we enforce:

the determinant of N matrix $= n_{11}n_{22} - n_{12}n_{21}$ is stable

$\theta_1 = (s - 6.066) \times$ a stable polynomial

$\theta_2 =$ a stable polynomial

From the order of θ_1, we then assume:

$$n_{12} = as + b; \quad (5.15)$$

$$n_{22} = cs + d; \quad (5.16)$$

and impose:

$$\theta_1 = (s - r_1)(s - r_2)(s - 6.066)\,; \tag{5.17}$$

where r_1 and r_2 are stable.

Similarly, we assume:

$$n_{11} = es^3 + fs^2 + gs + h\,; \tag{5.18}$$

$$n_{21} = ks^3 + ls^2 + os + r\,; \tag{5.19}$$

and then impose:

$$\theta_2 = (s - r_3)(s - r_4)(s - r_5)(s - r_6)(s - r_7)(s - r_8)(s - r_9); \tag{5.20}$$

where r_3, r_4, r_5, r_6, r_7, r_8, and r_9 are stable.

Finally, we solve (a,b,c,d,e,f,g,h,k,l,o,r) in terms of ($r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9$) from a Matlab command. We then have the N matrix with coefficients in terms of r_i s but we still have one more requirement to fulfill.

5.5 Selecting Free r_i s

This is the most time consuming and uncertain step. One easier way to do is to express the determinant of the N matrix in terms of r_i s. Since this determinant is a power 4 s-polynomial, we can match its coefficients to be the same as a polynomial with all roots in LHP and thus solve r_i s. However, r_i s are all stable (negative) and the coefficients of the determinant is nonlinear polynomial of r_i s. Therefore, even with 8 variables it is difficult to find a solution for sure. Consequently, an iterative program to search for a set of r_i s so that the determinant of the N matrix is stable was developed. A set of solutions is as given below:

$r_1 = -5;$
$r_2 = -3;$
$r_3 = -1;$
$r_4 = -2;$
$r_5 = -3;$
$r_6 = -2.5;$
$r_7 = -1.5;$
$r_8 = -0.01;$
$r_9 = -0.001;$

With this set of r_i s, the roots of the N matrix determinant are:

$-4.576096298076095e+000 + 4.184772240753429e+000i$
$-4.576096298076095e+000 - 4.184772240753429e+000i$
$-3.426404274657646e-002 + 1.153335696657026e-001i$
$-3.426404274657646e-002 - 1.153335696657026e-001i$

Thus, the coefficients of the N matrix are:

$a = -8.376009523223778e+002;$
$b = -5.784774387580804e+003;$
$c = -1.432291144790173e+003;$
$d = -5.381735341628663e+003;$
$e = 2.068406613167494e-001;$
$f = 1.840636113498792e+000;$
$g = 7.205764157500426e+000;$
$h = -1.937881730487596e-001;$
$k = 2.031426370332519e-001;$
$l = 1.685275398750232e+000;$
$o = 6.623270023616939e+000;$
$r = -1.924208023993863e-001;$

5.6 The Transformed Nominal Plant and Its Closed-loop Stability

The transformed nominal equivalent SISO systems are:

$$q_{an11} = \frac{-0.6285(-126.1s^4 - 1162.8s^3 - 4930s^2 - 349.0s - 70.19)(s+0.05899)(s+0.003335)}{(s+3)(s+5)(s-6.066)}$$

(5.21)

$$q_{an22} = \frac{-2.447(-126.1s^4 - 1162.8s^3 - 4930s^2 - 349.0s - 70.19)s(s+0.05899)(s+0.003335)}{(s+1)(s+2)(s+3)(s+3)(s+5)(s+2.5)(s+1.5)(s+0.01)(s+0.001)}$$

(5.22)

As we expected the q_{an11} is M.P. but unstable and the q_{an22} is N.M.P. but stable. This allows q_{an11} to be very easily stabilized by a large gain. For q_{an22}, the N.M.P. zero is just a differentiator so it can be stabilized by gain =1. Let

$$G_n = \begin{bmatrix} 1.8e+004 & 0 \\ 0 & 1 \end{bmatrix}, \qquad (5.23)$$

which stabilize q_{an11} and q_{an22}.

We then find $G = N * G_n$ and apply G back to the original nominal plant. The sensitivity TFM has all its roots of all denominators in the open LHP. Moreover, there is no RHP pole-zero cancellation between G and the nominal plant P_a. Therefore, the nominal closed-loop system is stable.

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \qquad (5.24)$$

$$s_{11} = \frac{2.5884e-006\,(s+10.22)\,(s-6.066)\,(s+0.4373)\,(s+0.01016)}{(s+0.1206)\,(s+0.008163)\,(s+1.59e-005)\,(s^2+0.006717s+0.02215)\,(s^2+9.15s+38.4)}$$

$$s_{12} = \frac{0.0011276(s+11.6)^5\,(s-6.066)\,(s+0.1702)\,(s-0.1139)\,(s-0.001081)}{(s+11.59)^4\,(s+0.1206)\,(s+0.008163)\,(s+1.59e-005)\,(s^2+0.006717s+0.02215)\,(s^2+9.15s+38.4)}$$

$$s_{21} = \frac{-0.0016683\,s\,(s+0.007423)\,(s^2+0.1284s+0.04)\,(s^2+9.153s+38.46)}{(s+0.1206)\,(s+0.008163)\,(s+1.59e-005)\,(s^2+0.006717s+0.02215)\,(s^2+9.15s+38.4)}$$

$$s_{22} = \frac{0.0032298\,(s+2.965)\,(s+2.606)\,(s+1.754)\,(s+1.7)\,(s+0.9749)\,(s+0.01092)\,(s+1.68e-005)}{(s+0.1206)\,(s+0.008163)\,(s+1.59e-005)\,(s^2+0.006717s+0.02215)\,(s^2+9.15s+38.4)}$$

5.7 Synthesize a Diagonal Controller for the Transformed Problem

In this step, we first transform the original system into the transformed system by $P_n = P * N$ for all four flight conditions. Then, jump back to 5.3 and make sure that the zero direction is not pinned. Otherwise, we have to find another N matrix. Hopefully, we are fine in our case.

Then, we apply the standard NS MIMO QFT to the transformed plant family and add a stability specification to it. The plant templates are shown as in Fig. 5.1 and Fig. 5.2. Then, we synthesize a diagonal controller by loop shaping the two equivalent SISO systems without penetrating the U-contour as in Fig. 5.3 and Fig. 5.4. We notice that the nominal plant is very easily stabilized by assigning a large and a small gain. From the concocted example, we also know that it might require larger gain (or smaller; depend on the equivalent q_{nii}) to take care of the disturbance, which is from the off diagonal entries of Q_n, so that the closed-loop system will remain stable.

After we decide a controller, we calculate the transformed closed-loop sensitivity TFM and check their denominator for stability. Unfortunately, no matter how large (or small) the feedback gain is assigned, the closed-loop is unstable except for the nominal. The reason is that the diagonal dominance requirement at high frequency is violated in both the original system and transformed system.

Indeed, we have to synthesize a controller so that the transformed closed-loop system is stable. Only in that case, a stable closed-loop system for the original plant family can be expected.

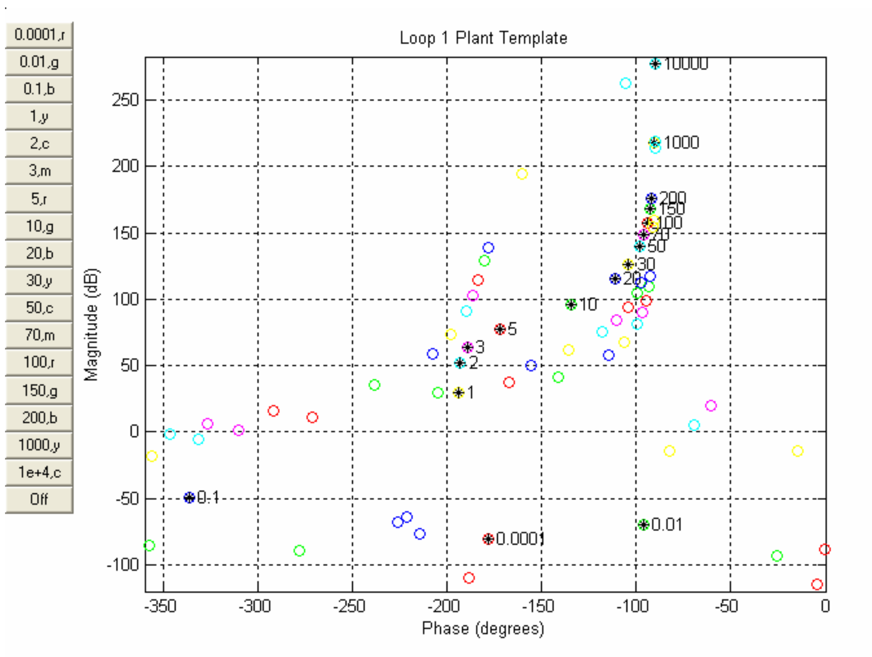

Fig. 5.1 Plant template for q_{n11}

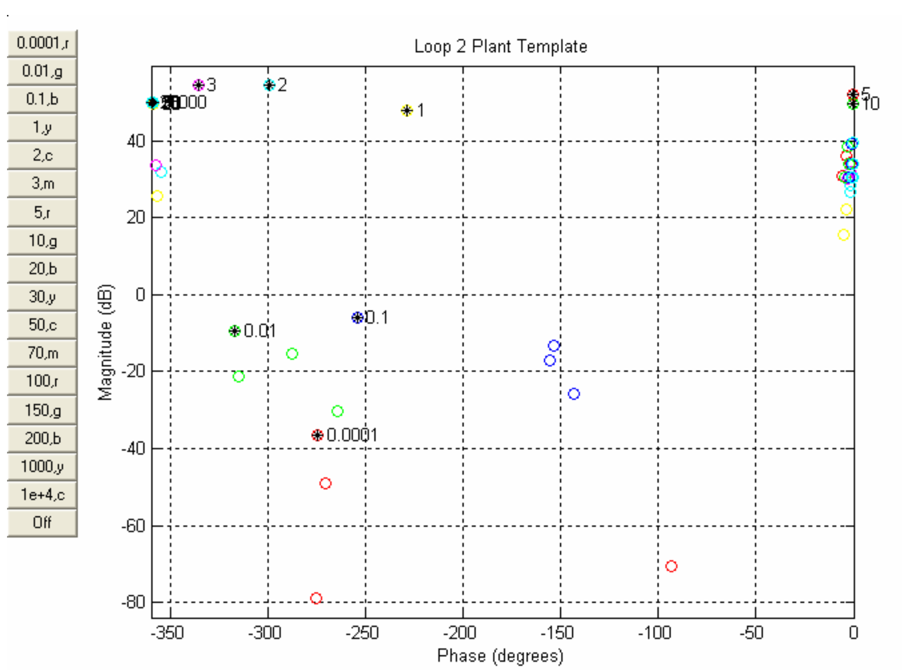

Fig. 5.2 Plant template for q_{n22}

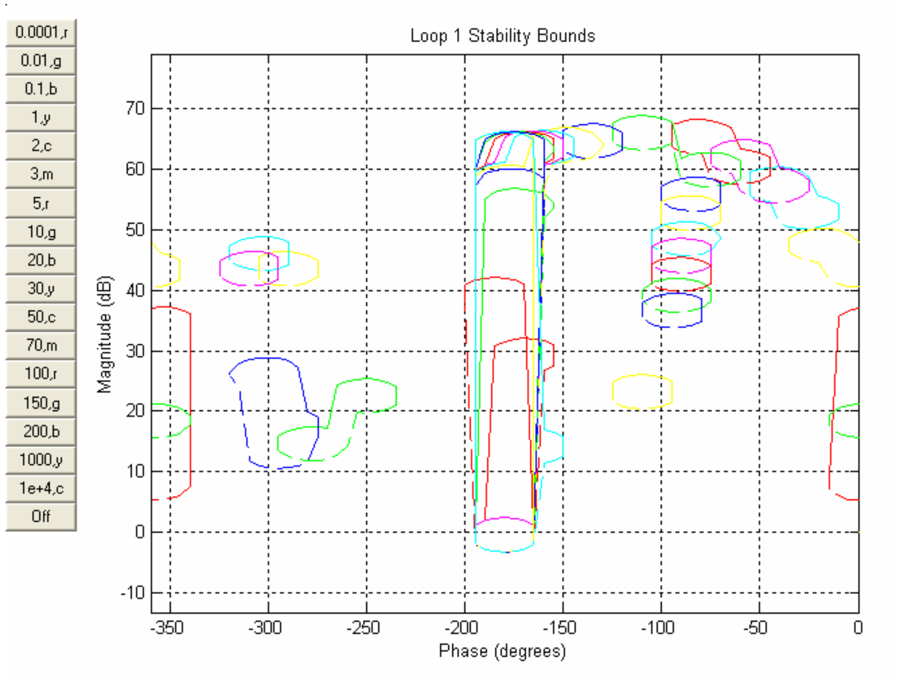

Fig. 5.3 Stability bounds for q_{n11}

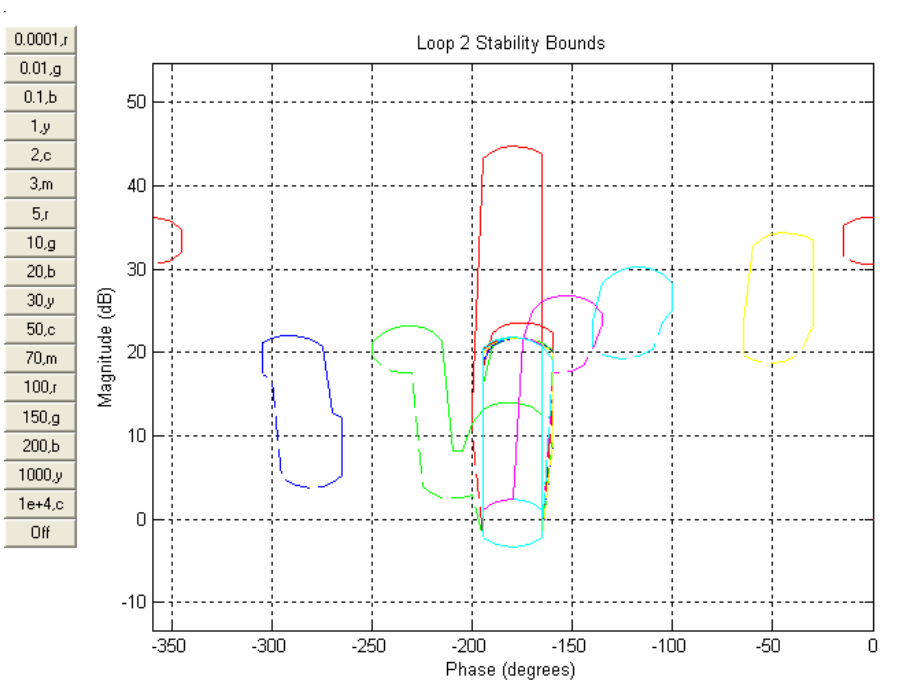

Fig. 5.4 Stability bounds for q_{n22}

For example, if we selected a controller for the transformed problem as follows:

$$G_n = \begin{bmatrix} \dfrac{129142482068.022}{(s+30)(s+15)(s+5)(s+1)} & 0 \\ 0 & \dfrac{1.2829e\text{-}011}{(s+10)(s+2)(s+1.5)(s+1)} \end{bmatrix} \tag{5.25}$$

The 11 entry of the transformed plant b's sensitivity TFM:

$$S_{b11} = \frac{\begin{array}{c}(s+3.872)^5(s+4.999)(s+5.001)(s+1.998)(s+1.497)(s+1)^5(s+10)^3(s+15)^2 \\ (s+30)(s-2.371)(s^2+4.002s+4.005)(s^2+3.003s+2.254)(s^2+0.01758s+0.01056)\end{array}}{\begin{array}{c}(s+2.105e009)(s+15)(s+10.12)(s+4.994)(s+2.202)(s-1.169)(s-0.04698)(s-0.009893) \\ (s+0.005091)(s^2+1.897s+0.9009)(s^2+2.077s+1.086)(s^2+19.88s+98.82) \\ (s^2+8.094s+16.39)(s^2+2.734s+1.898)(s^2 7.402s+13.74) \\ (s^2+4.066s+4.197)(s^2+3.328s+2.842)(s^2+8.323s+31.73)\end{array}} \tag{5.26}$$

which indicates that the closed-loop is not stable.

5.8 Verifying the Closed-loop Stability for the Original Plant

First, we inverse-transform the found diagonal controller G_n back to G from $G = N * G_n$. Then, we verify the closed-loop stability of the original system by:

1. Make sure that there is no RHP pole-zero cancellation in between the controller G and all plants for four flight conditions. If the found controller G has no unstable poles and no unstable zeros, this requirement is satisfied automatically.
2. Make sure that the denominators of the sensitivity TFM's all four entries from all four flight conditions have their roots on the open LHP only.

If the above two requirements are satisfied, the closed-loop system for the original plant family is stable.In our case, the closed-loop system is unstable except for the nominal one because of the RHP poles at the denominators.

Remark: The transformed closed-loop system is unstable from 5.7. Moreover, there are only LHP zeros in N matrix and no unstable poles and unstable zeros in G_n. Thus, we can expect the closed-loop system for the original plant family is unstable as well.

5.9 Discussion

Here, we showed that a proper N matrix can be found for the X-29 flight control problem. For this selected N matrix, the closed-loop nominal plant can be easily stabilized. However, the result does not robustly stabilize the transformed plant and hence the original plant because they violate the diagonal dominance requirement. This is fixed by switching the inputs order. Unfortunately, a proper set of transforming matrices is not found yet.

Several improvements can be made to the X-29 flight control problem and the steps done so far. For example, the plant is described in four flight conditions and could be better put in a continuous parameter uncertain form. This can therefore result in a finer plant template and thus obtain a clearer U-contour. In doing the NS MIMO QFT for the transformed plant, it will be better to argue a RP requirement on the non-diagonal as well. Also, 5.3 is better developed to check the real necessary and sufficient condition rather than just the zero input and output directions.

CHAPTER VI

CONCLUSIONS

6.1 Conclusions

In this thesis, a transformation scheme is proposed to synthesize controllers for RHP dipole systems using NS MIMO QFT. By using the transformation matrices, one can obtain a new set of equivalent plants $\hat{q}_{ii}$ with desired stable and /or MP structure. In effect the scheme helps to eliminate the RHP dipoles that may appear in the equivalent SISO plants and make a design feasible with the NS MIMO QFT.

Compared with the transformed NS MIMO QFT, Singular G Method uses a similar transforming scheme but the transforming matrix used is extreme. Thus, it results in a singular controller.

Different from the standard NS MIMO QFT, the transformation scheme leads to a fully-populated controller. Thus, it is possible to systematically design a robust controller for a MIMO system with RHP dipoles. This is illustrated as a design procedure in a flow chart. The proposed transformations apply in many situations making NS MIMO QFT a viable design method for a much larger class of problems than has been possible thus far.

6.2 Ongoing Work

Work on completing the X-29 design problem is continuing. The diagonal dominance has been fixed by switching the input order. Nonetheless, the search for a useful set of r_i s from an iterating program is not a quick job. Especially, when there are many variables to select.

Although, in this thesis a systematic procedure to find the transforming matrices, M & N, is developed, the existence of a solution for this procedure can not be guaranteed at this point. Even though one may not be able to find a pair of transformation matrices following the developed procedure that does not necessary mean there is no such pair of transforming matrices. In case that the procedure fails, one still can look for a higher order matrix pair. Continuing research will explore other ways of prescribing these transformation matrices. On the other hand, among all the possible M and N pairs there is no clear rule to determine which one is better. Moreover, the transformation of the RS and RP specification from the original system to the transformed system is also not clear, so as are its properties. All these still need to be investigated. Moreover, the roles which the directions play in both Singular G Method and the transformed NS MIMO QFT are also worth investigation.

REFERENCES

[1] Horowitz, I., 1979, "Quantitative synthesis of uncertain multiple input-output feedback system," International Journal of Control, **30**(1), pp. 81-106.

[2] Houpis, C. H., and Rasmussen, S. J., 1999, *Quantitative Feedback Theory: Fundamentals And Applications*, Marcel Dekker, New York.

[3] Yaniv, O., 1999, *Quantitative Feedback Design Of Linear And Nonlinear Control System*, Kluwer Academic Publisher, Norwell, MA.

[4] Kerr, M. L., Jayasuriya, S., and Asokanthan S. F., Accepted for publication in 2004~2005, "On Stability in Non-Sequential MIMO QFT Designs," ASME Journal of Dynamic Systems, Measurement, and Control.

[5] Walke, J. G., 1983, "Design of longitudinal flight control system using the singular "G" method," a M.S. thesis of the Air Force Institute of Technology, Wright-Patterson Air Force Base, OH.

APPENDIX A

Code for the Concocted Example Using Transformed NS MIMO QFT

```
%%% Concocted Example Using Transformed NS MIMO QFT %%%
%% Selecting the N matrix
% Nominal plant
clear;clc;
syms s n11 n12 n21 n22
k=1;a=1;
p11=k/(s+a);
p12=-k/(s+a);
p21=k/(s+a);
p22=k/(s-2);
P=[p11 p12;p21 p22];
pretty(P)
dd=det(P);
dd=factor(dd);
pretty(dd)
% Original Nominal Plant's Q matrix
IP=inv(P);
Q=[1/IP(1,1) 1/IP(1,2);1/IP(2,1) 1/IP(2,2)];
pretty(Q)
% N transfer
N=[n11 n12; n21 n22];
Pn=P*N;
Pn=simplify(Pn);
pretty(Pn)
dn=det(Pn);
IPn=inv(Pn);
IPn=simplify(IPn);
IPn=factor(IPn);
```

```
Qn=[1/IPn(1,1) 1/IPn(1,2);1/IPn(2,1) 1/IPn(2,2)];
Qn=simplify(Qn);
Qn=factor(Qn);
Qn=collect(Qn);
pretty(Qn)
% Result
N=[3*s+2 1;s+3 3];% selected from hand calculation
Pn=P*N;
Pn=simplify(Pn);
pretty(Pn)
IPn=inv(Pn);
Qn=[1/IPn(1,1) 1/IPn(1,2);1/IPn(2,1) 1/IPn(2,2)];
Qn=factor(Qn);
pretty(Qn)
% Verification
Gn=[0.3 0;0 20]; %Solved from nonsiquential QFT; or assigning small/large gain to the
NMP/unstable plant
G=N*Gn;
den=eye(2,2)+P*G;
den=simplify(den);
num=P*G;
num=simplify(num);
S=inv(den);
S=simplify(S);
pretty(S)
T=S*num;
T=simplify(T);
pretty(T)
%% Applying NS MIMO QFT to the transformed problem
clear all;clc;close all;
%%%%%%%%%%
```

```
%  Nominal plant value: a=K=1
%  The Transformed Problem:
%
%          [          k (2 s - 1)                    k       ]
%          [          -----------                 -2 -----     ]
%          [            s + a                       s + a     ]
%     P'=   [                                          ]        1≦K≦2  ;  1≦a≦2
%          [      2                                    ]
%          [k (4 s  + (-1 + a) s + 3 a - 4)    k (4 s - 2 + 3 a)]
%          [------------------------------    -----------------]
%          [       (s + a) (s - 2)            (s + a) (s - 2) ]
%
%  Design the controller G'=diag{g1,g2} such that for all P the system is stable;
%
%*******************Generate Plant ***************************
%   p--plant; P=inv(p);
nom=1;   %nominal plant
c=1;
for k=[1 2]
  for a=[1 2]
       nump11(c,:)=k*[2 -1];               nump12(c,:)=[-2*k];
       denp11(c,:)=[1 a];                  denp12(c,:)=[1 a];
       nump21(c,:)=k*[4 a-1 3*a-4];         nump22(c,:)=k*[4 3*a-2];
       denp21(c,:)=[1 a-2 -2*a];           denp22(c,:)=[1 a-2 -2*a];
       c=c+1;
  end
end
%generate inverse of plant
[nn1,dd1]=mulnd(nump11,denp11,nump22,denp22);
[nn2,dd2]=mulnd(nump12,denp12,nump21,denp21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
```

```
[numP11,denP11]=mulnd(nump22,denp22,dendet,numdet);
[numP22,denP22]=mulnd(nump11,denp11,dendet,numdet);
[numP12,denP12]=mulnd(-nump12,denp12,dendet,numdet);
[numP21,denP21]=mulnd(-nump21,denp21,dendet,numdet);
w  = [1 2 5 10 20 50 70 100 150];  % for stability bounds;
wl  = logspace(-1,10);
m=10; %db
m_lin=10.^(m/20)
% Compute Continuous bode of the plant
p11=freqcp(nump11,denp11,w);
p12=freqcp(nump12,denp12,w);
p21=freqcp(nump21,denp21,w);
p22=freqcp(nump22,denp22,w);

P11=freqcp(numP11,denP11,w);
P12=freqcp(numP12,denP12,w);
P21=freqcp(numP21,denP21,w);
P22=freqcp(numP22,denP22,w);
tit1 = sprintf('Loop 1 Plant Template');
plottmpl(w,w,1./P11,nom);title(tit1);
tit2 = sprintf('Loop 2 Plant Template');
plottmpl(w,w,1./P22,nom);title(tit2);
% Calculating Bounds for Loop #1
bds_st=sisobnds(2,w,w,m_lin,1./P11,[],nom);% Stability
bds=grpbnds(bds_st);
bds=sectbnds(bds); %intersections
tit3 = sprintf('Loop 1 Stability Bounds');
plotbnds(bds);title(tit3);
% Shaping for Loop #1
numg1=1;deng1=1;
lpshape(wl,bds,denP11(nom,:),numP11(nom,:),[],numg1,deng1);
```

```
[numg1,deng1]=getqft('tqeg1.shp');
g1=freqcp(numg1,deng1,w);
G1=zpk(tf(numg1,deng1));
figure;bode(G1,{1E-1,1E5});
% Calculating Bounds for Loop #2
bds_st=sisobnds(2,w,w,m_lin,1./P22,[],nom);% Stability
bds=grpbnds(bds_st);
bds=sectbnds(bds);%intersections
tit4 = sprintf('Loop 2 Stability Bounds');
plotbnds(bds);title(tit4);
% Shaping for Loop #2
numg2=1;deng2=1;
lpshape(wl,bds,denP22(nom,:),numP22(nom,:),[],numg2,deng2);
[numg2,deng2]=getqft('tqeg2.shp');
g2=freqcp(numg2,deng2,w);
G2=zpk(tf(numg2,deng2));
figure;bode(G2,{1E-1,1E5});
%% Result
clc;clear;format long e;
r=[];
for k=2:-1:1
   for a=2:-1:1
% Plant
Na11=k;Na12=-k;
Na21=k;Na22=k;
Da11=[1 a];Da12=[1 a];
Da21=[1 a];Da22=[1 -2];
% N matrix
n12=[1];n22=[3];
n11=[3 2];n21=[1 3];
nd=[1];
```

```
%[nn1,dd1]=mulnd(n11,1,n22,1);
%[nn2,dd2]=mulnd(n12,1,n21,1);
%[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
% Gn
gnn11=[0.12];gnd11=[1 1];
gn12=0;gn21=0;
gnn22=81600;gnd22=[1 817 13600];
% G     %G=N*Gn%
g11=conv(n11,gnn11);
g12=n12*gnn22;
g21=conv(n21,gnn11);
g22=n22*gnn22;
gd11=gnd11;
gd12=gnd22;
gd21=gnd11;
gd22=gnd22;
% P*G
[tn1,td1]=mulnd(Na11,Da11,g11,gd11);
[tn2,td2]=mulnd(Na12,Da12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na11,Da11,g12,gd12);
[tn2,td2]=mulnd(Na12,Da12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g11,gd11);
[tn2,td2]=mulnd(Na22,Da22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g12,gd12);
[tn2,td2]=mulnd(Na22,Da22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
```

```
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PG
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
s11=zpk(tf(numS11,denS11));
s11=minreal(s11)
```

```
[nums11,dens11]=tfdata(s11,'v');
s22=zpk(tf(numS22,denS22));
s22=minreal(s22)
[nums22,dens22]=tfdata(s22,'v');
s12=zpk(tf(numS12,denS12));
s12=minreal(s12)
[nums12,dens12]=tfdata(s12,'v');
s21=zpk(tf(numS21,denS21));
s21=minreal(s21)
[nums21,dens21]=tfdata(s21,'v');
% Closed-loop TFM
[tn1,td1]=mulnd(numS11,denS11,num11,den11);
[tn2,td2]=mulnd(numS12,denS12,num21,den21);
[numT11,denT11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(numS11,denS11,num12,den12);
[tn2,td2]=mulnd(numS12,denS12,num22,den22);
[numT12,denT12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(numS21,denS21,num11,den11);
[tn2,td2]=mulnd(numS22,denS22,num21,den21);
[numT21,denT21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(numS21,denS21,num12,den12);
[tn2,td2]=mulnd(numS22,denS22,num22,den22);
[numT22,denT22]=addnd(tn1,td1,tn2,td2);
t11=zpk(tf(numT11,denT11));
t11=minreal(t11)
t22=zpk(tf(numT22,denT22));
t22=minreal(t22)
t12=zpk(tf(numT12,denT12));
t12=minreal(t12)
t21=zpk(tf(numT21,denT21));
t21=minreal(t21)
```

```
r(:,1)=roots(dens11);
r(:,2)=roots(dens12);
r(:,3)=roots(dens21);
r(:,4)=roots(dens22);
figure(1);
plot(r,'*');
hold on;
sysS=[s11 s12;s21 s22];
figure(2);
sigma(sysS);hold on;
sysT=[t11 t12;t21 t22];
figure(3);
impulse(sysT,2);hold on;
   end
end
hold off;
```

APPENDIX B

Code for the Concocted Example Using Singular G Method

```
%%% Concocted Example Using Singular G Method %%%
%% Searching For a Set of the Variable b & c
clear;clc;
i=1;datar=[];datap=[];
for a=1:0.1:2
    for b=0.1:0.1:0.5
        for c=0.1:0.1:0.5
            r=(2*b+2*b*c-2-a*c)/(b-1+b*c+c);
            datar(i)=r;
            datap(1,i)=a;
            datap(2,i)=b;
            datap(3,i)=c;
            i=i+1;
        end
    end
end
%% Find the Sensitivity TFM for H=15
clear;clc;
syms s k a
% G %
H=15;b=0.5;c=0.4;
N=[b b*c;1 c];
G=H*[b b*c;1 c];
% P %
p11=k/(s+a);
p12=-k/(s+a);
p21=k/(s+a);
p22=k/(s-2);
```

```
P=[p11 p12;p21 p22];
P=simplify(P);
pretty(P)
dP=det(P);
dP=factor(dP);
pretty(dP)
% P*G %
PG=P*G;
PG=simplify(PG);
pretty(PG)
% I+PG %
I=[1 0;0 1];
IP=I+PG;
IP=simplify(IP);
pretty(IP)
% Sencitivity TFM %
S=(IP)^(-1);
S=simplify(S);
pretty(S)
%% Closed-loop Plots
clear;clc;
r=[];
i=1;
for k=1:0.1:2
    for a=1:0.1:2
        e=2;
        f=+3*k-4+2*a;
        g=-4*a+18*k+12*k*a;
        r(:,i)=roots([e f g]);
        i=i+1;
    end
```

```
end
for i=1:2
   for j=1:121
   figure(1);
   plot(r(i,j),'*');
   hold on;
   end
end
hold off;
for k=1:0.5:2
   for a=1:2
      k=1;a=1;
      numS={[2 -4+2*a+18*k -4*a-12*k+12*k*a] [6*k -12*k];[-45*k 30*k-30*k*a] [2 -
4+2*a-15*k 30*k-4*a]};
      denS={[2 3*k-4+2*a -4*a+18*k+12*k*a] [2 3*k-4+2*a -4*a+18*k+12*k*a];[2
3*k-4+2*a -4*a+18*k+12*k*a] [2 3*k-4+2*a -4*a+18*k+12*k*a]};
      sysS=tf(numS,denS);
      figure(2);
      sigma(sysS);hold on;
   end
end
holf off;
for k=1:0.5:2
   for a=1:2
      numT={[-15*k 30*k] [-6*k 12*k];[45*k 30*k*a-30*k] [18*k 12*k*a-12*k]};
      denT={[2 3*k-4+2*a -4*a+18*k+12*k*a] [2 3*k-4+2*a -4*a+18*k+12*k*a];[2
3*k-4+2*a -4*a+18*k+12*k*a] [2 3*k-4+2*a -4*a+18*k+12*k*a]};
      sysT=tf(numT,denT);
      figure(3);
      impulse(sysT);hold on;
   end
```

```
end
hold off;
```

APPENDIX C

Code for X-29 Longitudinal Flight Control

```
%%% X-29 Plant Model %%%
clc;clear;format long e;
% plant a %
A1=[-0.6161e-1 0.26e2 -0.4452 -32.17; -0.1528e-3 -4.277 0.9846 0; 0.2409e-5 0.7687e2
-1.252 0; 0 0 1 0];
B1=[-0.3185 -0.9043e-1; -0.3744e-2 -0.1115e-1; 0.5577 -0.3264; 0 0];
C1=[0.4779e-2 0.1335e3 0.48 0; 0 0 1 0];
D1=[0.1172 0.3481; 0 0];
[n11,d11]=ss2tf(A1,B1,C1,D1,1);
[n12,d12]=ss2tf(A1,B1,C1,D1,2);
Na11=n11(1,:);
Na21=n11(2,:);
Na12=n12(1,:);
Na22=n12(2,:);
Da=d11;
fNa11=roots(Na11)
fNa12=roots(Na12)
fNa21=roots(Na21)
fNa22=roots(Na22)
fDa=roots(Da)
% plant b %
A2=[-0.1621e-1 -0.1454e2 -0.4084 -32.17; -0.3231e-3 -0.1045e1 0.9872 0; 0.1360e-3
0.9765e1 -0.4570 0; 0 0 1 0];
B2=[-0.1012 -0.1812; -0.6389e-3 -0.5101e-2; 0.8576e-1 -0.5019e-1;0 0];
C2=[0.4580e-2 0.1450e2 0.1782 0.2384e-4; 0 0 1 0];
D2=[0.9072e-2 0.7102e-1; 0 0];
[n21,d21]=ss2tf(A2,B2,C2,D2,1);
[n22,d22]=ss2tf(A2,B2,C2,D2,2);
```

```
Nb11=n21(1,:);
Nb21=n21(2,:);
Nb12=n22(1,:);
Nb22=n22(2,:);
Db=d21;
fNb11=roots(Nb11)
fNb12=roots(Nb12)
fNb21=roots(Nb21)
fNb22=roots(Nb22)
fDb=roots(Db)
% plant c %
A3=[-0.1443e-1  -0.8774e1  -0.3365  -32.13;  -0.1392e-3  -0.1311e1  0.9911  0;  0.7e-3
0.2677e2 -0.5191 0; 0 0 1 0];
B3=[-0.1306 -0.1617; -0.9053e-3 -0.6301e-2; 0.1589 -0.118; 0 0];
C3=[0.3224e-2 0.3013e2 0.2043 -0.7153e-4; 0 0 1 0];
D3=[0.21e-1 0.145; 0 0];
[n31,d31]=ss2tf(A3,B3,C3,D3,1);
[n32,d32]=ss2tf(A3,B3,C3,D3,2);
Nc11=n31(1,:);
Nc21=n31(2,:);
Nc12=n32(1,:);
Nc22=n32(2,:);
Dc=d31;
fNc11=roots(Nc11)
fNc12=roots(Nc12)
fNc21=roots(Nc21)
fNc22=roots(Nc22)
fDc=roots(Dc)
% plant d %
A4=[-0.1533e-1  -0.1587e2  -0.1675  -32.02;  -0.8453e-4  -0.4247  0.9976  0.5141e-6;  -
0.2104e-2 0.7490e1 -0.1668 0; 0 0 1 0];
```

```
B4=[-0.6987e-1 -0.1600; -0.1573e-3 -0.1605e-2; 0.5476e-1 -0.3643e-1; 0 0];
C4=[0.2315e-2 0.1158e2 0.6387e-1 -0.3934e-3; 0 0 1 0];
D4=[0.4423e-2 0.4373e-1; 0 0];
[n41,d41]=ss2tf(A4,B4,C4,D4,1);
[n42,d42]=ss2tf(A4,B4,C4,D4,2);
Nd11=n41(1,:);
Nd21=n41(2,:);
Nd12=n42(1,:);
Nd22=n42(2,:);
Dd=d41;
fNd11=roots(Nd11)
fNd12=roots(Nd12)
fNd21=roots(Nd21)
fNd22=roots(Nd22)
fDd=roots(Dd)
% multivariable poles and zeros %
sys1=ss(A1,B1,C1,D1);
sys2=ss(A2,B2,C2,D2);
sys3=ss(A3,B3,C3,D3);
sys4=ss(A4,B4,C4,D4);
z1=zero(sys1);z1=esort(z1)
z2=zero(sys2);z2=esort(z2)
z3=zero(sys3);z3=esort(z3)
z4=zero(sys4);z4=esort(z4)
p1=pole(sys1);p1=esort(p1)
p2=pole(sys2);p2=esort(p2)
p3=pole(sys3);p3=esort(p3)
p4=pole(sys4);p4=esort(p4)
figure(1);pzmap(sys1);sgrid;
figure(2);pzmap(sys2);sgrid;
figure(3);pzmap(sys3);sgrid;
```

```
figure(4);pzmap(sys4);sgrid;

%%% SVD for X-29 plant %%%
% Checking the NMP zero input-output direction
clc;clear;format long e;
% Plant a %
Na11=[1.172000000000000e-001                              4.215693804999958e-001
6.546706291135286e+001   4.132461548651955e+000   -7.737107377179808e-004];
Na12=[3.481000000000001e-001                3.004621760300199e-001              -
7.031347733880426e+001   -4.290260300163865e+000   6.681760043581730e-004];
Na21=[5.577000000000014e-001     2.131840749733499e+000     1.351789739836793e-
001   0];
Na22=[-3.264000000000067e-001   -2.273223021845780e+000   -1.390504955224294e-
001   0];
Da11=[1.000000000000000e+000                5.590610000000000e+000              -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da12=[1.000000000000000e+000                5.590610000000000e+000              -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da21=[1.000000000000000e+000                5.590610000000000e+000              -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da22=[1.000000000000000e+000                5.590610000000000e+000              -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
s=0;
pd=Da11(1)*s^4+Da11(2)*s^3+Da11(3)*s^2+Da11(4)*s+Da11(5);
pn11=Na11(1)*s^4+Na11(2)*s^3+Na11(3)*s^2+Na11(4)*s+Na11(5);
pn12=Na12(1)*s^4+Na12(2)*s^3+Na12(3)*s^2+Na12(4)*s+Na12(5);
pn21=Na21(1)*s^3+Na21(2)*s^2+Na21(3)*s+Na21(4);
pn22=Na22(1)*s^3+Na22(2)*s^2+Na22(3)*s+Na22(4);
plant1=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upa1,Spa1,Vpa1]=svd(plant1)
dot(Upa1(:,1),Vpa1(:,1))
```

```
dot(Upa1(:,2),Vpa1(:,2))
% Plant b %
Nb11=[9.071999999999969e-003 1.932808712000278e-002
1.155041962224416e+000 4.755595748284797e-003 -2.796579577720032e-004];
Nb12=[7.101999999999997e-002 2.408502020000114e-002 -
1.421095010395007e+000 -4.109295365955556e-003 2.950852136856669e-004];
Nb21=[8.576000000000006e-002 8.475674789998777e-002 1.254879739693904e-
003 0];
Nb22=[-5.018999999999885e-002 -1.030980381000131e-001 -8.658115718451487e-
004 0];
Db11=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db12=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db21=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db22=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
s=0;
pd=Db11(1)*s^4+Db11(2)*s^3+Db11(3)*s^2+Db11(4)*s+Db11(5);
pn11=Nb11(1)*s^4+Nb11(2)*s^3+Nb11(3)*s^2+Nb11(4)*s+Nb11(5);
pn12=Nb12(1)*s^4+Nb12(2)*s^3+Nb12(3)*s^2+Nb12(4)*s+Nb12(5);
pn21=Nb21(1)*s^3+Nb21(2)*s^2+Nb21(3)*s+Nb21(4);
pn22=Nb22(1)*s^3+Nb22(2)*s^2+Nb22(3)*s+Nb22(4);
plant1=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upb1,Spb1,Vpb1]=svd(plant1)
dot(Upb1(:,1),Vpb1(:,1))
dot(Upb1(:,2),Vpb1(:,2))
s=1.077771575548320e-002;
pd=Db11(1)*s^4+Db11(2)*s^3+Db11(3)*s^2+Db11(4)*s+Db11(5);
pn11=Nb11(1)*s^4+Nb11(2)*s^3+Nb11(3)*s^2+Nb11(4)*s+Nb11(5);
```

```
pn12=Nb12(1)*s^4+Nb12(2)*s^3+Nb12(3)*s^2+Nb12(4)*s+Nb12(5);
pn21=Nb21(1)*s^3+Nb21(2)*s^2+Nb21(3)*s+Nb21(4);
pn22=Nb22(1)*s^3+Nb22(2)*s^2+Nb22(3)*s+Nb22(4);
plant2=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upb2,Spb2,Vpb2]=svd(plant2)
dot(Upb2(:,1),Vpb2(:,1))
dot(Upb2(:,2),Vpb2(:,2))
% Plant c %
Nc11=[2.100000000000002e-002 4.350065659999158e-002
4.225829411104126e+000 4.912148017935841e-002 -1.646391253174590e-004];
Nc12=[1.450000000000000e-001 5.297899920000204e-002 -
7.436124264278636e+000 -7.703155594005817e-002 2.414747452200994e-004];
Nc21=[1.589000000000143e-001 1.862845259999553e-001 2.834621212944311e-
003 0];
Nc22=[-1.179999999999932e-001 -3.251917000000226e-001 -4.029330462134440e-
003 0];
Dc11=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc12=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc21=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc22=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
s=0;
pd=Dc11(1)*s^4+Dc11(2)*s^3+Dc11(3)*s^2+Dc11(4)*s+Dc11(5);
pn11=Nc11(1)*s^4+Nc11(2)*s^3+Nc11(3)*s^2+Nc11(4)*s+Nc11(5);
pn12=Nc12(1)*s^4+Nc12(2)*s^3+Nc12(3)*s^2+Nc12(4)*s+Nc12(5);
pn21=Nc21(1)*s^3+Nc21(2)*s^2+Nc21(3)*s+Nc21(4);
pn22=Nc22(1)*s^3+Nc22(2)*s^2+Nc22(3)*s+Nc22(4);
plant1=[pn11/pd pn12/pd;pn21/pd pn22/pd];
```

```
[Upc1,Spc1,Vpc1]=svd(plant1)
dot(Upc1(:,1),Vpc1(:,1))
dot(Upc1(:,2),Vpc1(:,2))
s=6.885173072838656e-004;
pd=Dc11(1)*s^4+Dc11(2)*s^3+Dc11(3)*s^2+Dc11(4)*s+Dc11(5);
pn11=Nc11(1)*s^4+Nc11(2)*s^3+Nc11(3)*s^2+Nc11(4)*s+Nc11(5);
pn12=Nc12(1)*s^4+Nc12(2)*s^3+Nc12(3)*s^2+Nc12(4)*s+Nc12(5);
pn21=Nc21(1)*s^3+Nc21(2)*s^2+Nc21(3)*s+Nc21(4);
pn22=Nc22(1)*s^3+Nc22(2)*s^2+Nc22(3)*s+Nc22(4);
plant2=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upc2,Spc2,Vpc2]=svd(plant2)
dot(Upc2(:,1),Vpc2(:,1))
dot(Upc2(:,2),Vpc2(:,2))
% Plant d %
Nd11=[4.422999999999955e-003 4.198247239997510e-003 6.009732055199013e-001 5.601114275346930e-003 -1.388627209596149e-005];
Nd12=[4.373000000000005e-002 5.253591799998514e-003 -7.493085240193560e-001 -5.172401463264253e-003 1.061499190466164e-005];
Nd21=[5.476000000000147e-002 2.306487227999554e-002 3.664198444066069e-004 0];
Nd22=[-3.642999999999996e-002 -2.771510290000112e-002 -1.819212840562523e-004 0];
Dd11=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd12=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd21=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd22=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
s=0;
```

```
pd=Dd11(1)*s^4+Dd11(2)*s^3+Dd11(3)*s^2+Dd11(4)*s+Dd11(5);
pn11=Nd11(1)*s^4+Nd11(2)*s^3+Nd11(3)*s^2+Nd11(4)*s+Nd11(5);
pn12=Nd12(1)*s^4+Nd12(2)*s^3+Nd12(3)*s^2+Nd12(4)*s+Nd12(5);
pn21=Nd21(1)*s^3+Nd21(2)*s^2+Nd21(3)*s+Nd21(4);
pn22=Nd22(1)*s^3+Nd22(2)*s^2+Nd22(3)*s+Nd22(4);
plant1=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upd1,Spd1,Vpd1]=svd(plant1)
dot(Upd1(:,1),Vpd1(:,1))
dot(Upd1(:,2),Vpd1(:,2))
s=1.167101885093847e-003;
pd=Dd11(1)*s^4+Dd11(2)*s^3+Dd11(3)*s^2+Dd11(4)*s+Dd11(5);
pn11=Nd11(1)*s^4+Nd11(2)*s^3+Nd11(3)*s^2+Nd11(4)*s+Nd11(5);
pn12=Nd12(1)*s^4+Nd12(2)*s^3+Nd12(3)*s^2+Nd12(4)*s+Nd12(5);
pn21=Nd21(1)*s^3+Nd21(2)*s^2+Nd21(3)*s+Nd21(4);
pn22=Nd22(1)*s^3+Nd22(2)*s^2+Nd22(3)*s+Nd22(4);
plant2=[pn11/pd pn12/pd;pn21/pd pn22/pd];
[Upd2,Spd2,Vpd2]=svd(plant2)
dot(Upd2(:,1),Vpd2(:,1))
dot(Upd2(:,2),Vpd2(:,2))

%%%  Finding N matrix for X29 Nominal Plant  %%%
clc;clear;format long e;
%%% X29 Nominal plant / Flight Condition a %%%
syms s n11 n12 n21 n22 m11 m12 m21 m22 % Define Symbols
%%% Plant Model in ZKP form %%%
pa11=0.1172*(s-1.866755071718204e-004)*(s+6.333473415037980e-
002)*(s^2+3.533860307397712*s+5.583695423620505e+002)/(s-
6.066135254583974)/(s+1.159471324781247e+001)/(s^2+6.203200677151032e-
002*s+5.367574321054267e-003);
pa12=0.3481*(s-1.553470197710892e-004)*(s+6.115672198662352e-002)*(s-
1.381869977774404e+001)*(s+1.462084742900535e+001)/(s-
```

```
6.066135254583974)/(s+1.159471324781247e+001)/(s^2+6.203200677151032e-
002*s+5.367574321054267e-003);
pa21=0.5577*s*(s+6.449777515801528e-002)*(s+3.758060499422411)/(s-
6.066135254583974)/(s+1.159471324781247e+001)/(s^2+6.203200677151032e-
002*s+5.367574321054267e-003);
pa22=-0.3264*s*(s+6.171576574771799e-002)*(s+6.902815551181606)/(s-
6.066135254583974)/(s+1.159471324781247e+001)/(s^2+6.203200677151032e-
002*s+5.367574321054267e-003);
Pa=[pa11 pa12;pa21 pa22];
Pa=simplify(Pa);pretty(Pa)
dPa=det(Pa);dPa=factor(dPa);
pretty(dPa)
IPa=inv(Pa);IPa=simplify(IPa);
pretty(IPa)
Qa=[1/IPa(1,1) 1/IPa(1,2);1/IPa(2,1) 1/IPa(2,2)];Qa=factor(Qa);
pretty(Qa)
%%% N transfer
N=[n11 n12; n21 n22];
Pan=Pa*N;Pan=simplify(Pan);
pretty(Pan)
IPan=inv(Pan);IPan=simplify(IPan);
pretty(IPan)
Qan=[1/IPan(1,1) 1/IPan(1,2);1/IPan(2,1) 1/IPan(2,2)];
Qan=simplify(Qan);
pretty(Qan)

% Try to make qan11 M.P. but unstable
% Try to make qan22 N.M.P. but stable
%  n11 n22 - n12 n21 = f(s)     where f(s) is stable
%  Poly1= (s-6.066135254583974)*stable
%  Poly2=  stable
```

```
clear;clc;format long e;

% coeficients
% qan11
h1=8.826904659509207e-001;
h2=6.147525393270124;
h3=3.760372141035229e-001;
k1=1.508199978127538;
k2=5.765182306113439;
k3=3.655673760185577e-001;
% qan22
h4=1.23426005835549;
h5=4.439643755774853;
h6=6.894486424004172e+002;
h7=4.351989959207325e+001;
h8=8.148125087765034e-003;
k4=3.665920872982476;
k5=3.164236031744317;
k6=7.404873433734643e+002;
k7=4.518171440648887e+001;
k8=7.036714625689909e-003;
% Define Symbols %
syms s
syms a b c d
syms e f g h k l o r
syms p1 p2
syms p3 p4 p5 p6 p7 p8 p9
% Poly 1 %
n12=a*s+b;
n22=c*s+d;
ps11=(-h1*(n22)+k1*(n12))*s^2+(-h2*(n22)+k2*(n12))*s+(-h3*(n22)+k3*(n12));
```

```
ps11=simplify(ps11);
ps11=collect(ps11,s);
pretty(ps11);
po11=(s-p1)*(s-p2)*(s-6.066135254583974);
po11=expand(po11);
po11=collect(po11,s);
pretty(po11);
S11=solve('-
496910556692507/562949953421312*c+3396164429747685/2251799813685248*a=1','-
3460749133797749/562949953421312*c+6491018221383865/1125899906842624*a-
496910556692507/562949953421312*d+3396164429747685/2251799813685248*b=-
p1-3414930559015427/562949953421312-p2','-
6774084229256261/18014398509481984*c+3292738196831973/9007199254740992*a-
3460749133797749/562949953421312*d+6491018221383865/1125899906842624*b=3
414930559015427/562949953421312*p1+p1*p2+3414930559015427/56294995342131
2*p2','-
6774084229256261/18014398509481984*d+3292738196831973/9007199254740992*b
=-3414930559015427/562949953421312*p1*p2','a','b','c','d');
% Poly 2 %
n11=e*s^3+f*s^2+g*s+h;
n21=k*s^3+l*s^2+o*s+r;
ps22=(h4*(n11)+k4*(n21))*s^4+(h5*(n11)+k5*(n21))*s^3+(h6*(n11)-
k6*(n21))*s^2+(h7*(n11)-k7*(n21))*s+(-h8*(n11)+k8*(n21));
ps22=simplify(ps22);
ps22=collect(ps22,s);
pretty(ps22);
po22=(s-p3)*(s-p4)*(s-p5)*(s-p6)*(s-p7)*(s-p8)*(s-p9);
po22=expand(po22);
po22=collect(po22,s);
pretty(po22);
```

```
S22=solve('1.23426005835549*e+3.665920872982476*k=1','4.439643755774853*e+3.1
64236031744317*k+3.665920872982476*l+1.23426005835549*f=-p4-p3-p9-p8-p7-p6-
p5','1.23426005835549*g+4.439643755774853*f+6.894486424004172e+002*e+3.1642
36031744317*l-
7.404873433734643e+002*k+3.665920872982476*o=p5*p7+p7*p8+p4*p7+p3*p7+p8*
p9+p5*p8+p7*p9+p5*p6+p3*p8+p3*p6+p6*p9+p5*p9+p4*p8+p6*p8+p3*p9+p3*p5+p
3*p4+p6*p7+p4*p5+p4*p6+p4*p9','6.894486424004172e+002*f+4.351989959207325e
+001*e-7.404873433734643e+002*l-
4.518171440648887e+001*k+3.665920872982476*r+1.23426005835549*h+4.4396437
55774853*g+3.164236031744317*o=-p3*p4*p7-p3*p5*p7-p5*p6*p8-p4*p5*p7-
p3*p6*p7-p3*p5*p6-p3*p4*p6-p4*p5*p6-p5*p6*p7-p4*p7*p8-p5*p7*p8-p4*p6*p7-
p3*p4*p5-p3*p7*p8-p4*p6*p8-p6*p7*p8-p3*p5*p8-p3*p4*p8-p4*p5*p8-p3*p6*p8-
p6*p8*p9-p4*p8*p9-p5*p7*p9-p3*p8*p9-p5*p8*p9-p7*p8*p9-p4*p7*p9-p3*p7*p9-
p5*p6*p9-p6*p7*p9-p4*p5*p9-p3*p5*p9-p3*p4*p9-p3*p6*p9-p4*p6*p9','-
4.518171440648887e+001*l-8.148125087765034e-
003*e+4.351989959207325e+001*f+7.036714625689909e-
003*k+4.439643755774853*h+6.894486424004172e+002*g+3.164236031744317*r-
7.404873433734643e+002*o=p3*p4*p5*p6+p4*p5*p6*p8+p3*p5*p6*p8+p3*p4*p6*p8
+p3*p4*p5*p8+p5*p6*p7*p8+p4*p6*p7*p8+p4*p5*p7*p8+p3*p6*p7*p8+p3*p5*p7*p
8+p3*p4*p7*p8+p4*p5*p6*p9+p3*p5*p6*p9+p3*p4*p6*p9+p3*p4*p5*p9+p5*p6*p7*
p9+p4*p6*p7*p9+p4*p5*p7*p9+p3*p6*p7*p9+p3*p5*p7*p9+p3*p4*p7*p9+p5*p6*p8
*p9+p5*p7*p8*p9+p4*p7*p8*p9+p3*p7*p8*p9+p4*p6*p8*p9+p6*p7*p8*p9+p3*p5*p
8*p9+p4*p5*p8*p9+p3*p6*p8*p9+p3*p4*p5*p7+p3*p4*p6*p7+p3*p5*p6*p7+p4*p5*
p6*p7+p3*p4*p8*p9','-8.148125087765034e-
003*f+6.894486424004172e+002*h+7.036714625689909e-
003*l+4.351989959207325e+001*g-7.404873433734643e+002*r-
4.518171440648887e+001*o=-p3*p4*p5*p8*p9-p5*p6*p7*p8*p9-p4*p6*p7*p8*p9-
p3*p6*p7*p8*p9-p4*p5*p6*p7*p9-p3*p5*p6*p7*p9-p4*p5*p7*p8*p9-
p3*p4*p5*p7*p9-p3*p4*p5*p6*p9-p4*p5*p6*p8*p9-p3*p5*p6*p8*p9-
p3*p4*p6*p8*p9-p3*p4*p6*p7*p8-p3*p4*p5*p6*p7-p4*p5*p6*p7*p8-
p3*p4*p6*p7*p9-p3*p4*p7*p8*p9-p3*p4*p5*p7*p8-p3*p4*p5*p6*p8-
```

```
p3*p5*p7*p8*p9-p3*p5*p6*p7*p8','4.351989959207325e+001*h-
8.148125087765034e-003*g-4.518171440648887e+001*r+7.036714625689909e-
003*o=p3*p4*p5*p6*p7*p8+p4*p5*p6*p7*p8*p9+p3*p5*p6*p7*p8*p9+p3*p4*p6*p7
*p8*p9+p3*p4*p5*p7*p8*p9+p3*p4*p5*p6*p8*p9+p3*p4*p5*p6*p7*p9','7.03671462
5689909e-003*r-8.148125087765034e-003*h=-
p3*p4*p5*p6*p7*p8*p9','e','f','g','h','k','l','o','r');

% Search for a Solution %
%    n11 n22 - n12 n21 = f(s)  ,   where f(s) is stable
%    pi's <0
clear;clc;format long e;
i=1;dNr={};datap=[];
for j1=0:1:5
    for j2=0:1:5
        for j3=0:1:5
            for j4=0:1:5
                for j5=0:1:5
                    for j6=0:0.5:3
                        for j7=0:0.5:3
                            for j8=0:0.01:0.1
                                for j9=0:0.001:0.01
                                    p1=-j1;
                                    p2=-j2;
                                    p3=-j3;
                                    p4=-j4;
                                    p5=-j5;
                                    p6=-j6;
                                    p7=-j7;
                                    p8=-j8;
                                    p9=-j9;
```

a=336916532476006945048494603021036565080197937712727092298675 2/13134679
92357882353457075654975722014538330886245324835139459-
757830419951784382188837427890727642456988717736808003413975 04/131346799
2357882353457075654975722014538330886245324835139459*p1*p2-
4151802919874093002318272290760127636700731446555226919999872/1313467992
357882353457075654975722014538330886245324835139459*p1-
4151802919874093002318272290760127636700731446555226919999872/1313467992
357882353457075654975722014538330886245324835139459*p2

b=-
1768720138847782067603843275276779362448584640423249347990708/1313467992
357882353457075654975722014538330886245324835139459-
5236415663477162164366231266058289525473305450455527090383970 56/13134679
92357882353457075654975722014538330886245324835139459*p1*p2-
3228452678194179661393932695511495161858346535021204415932505 6/131346799
2357882353457075654975722014538330886245324835139459*p1-
3228452678194179661393932695511495161858346535021204415932505 6/131346799
2357882353457075654975722014538330886245324835139459*p2

c=426866180399281267036581283094136501306781816011328634545529 6/13134679
92357882353457075654975722014538330886245324835139459-
129485914585511661042996829851361539827947472126011020554542080/13134679
92357882353457075654975722014538330886245324835139459*p1*p2-
7093935319895162940027497478145613938510850633978999271553440/1313467992
357882353457075654975722014538330886245324835139459*p1-
7093935319895162940027497478145613938510850633978999271553440/1313467992
357882353457075654975722014538330886245324835139459*p2

d=-
1719474445132331325066730746192410072551468444214982351414888/1313467992
357882353457075654975722014538330886245324835139459-
487873518920566817075926443256436052975031454342418985200623200/13134679

92357882353457075654975722014538330886245324835139459*p1*p2-
31385642960396415004362909994519880833615427967983683696002816/131346799
23578823534570756549757220145383308862453248351394559*p1-
31385642960396415004362909994519880833615427967983683696002816/131346799
23578823534570756549757220145383308862453248351394559*p2

e=.16771188967190843744892778644333e-
4*p3*p4*p7+.16771188967190843744892778644333e-
4*p3*p5*p7+.16771188967190843744892778644333e-
4*p5*p6*p8+.16771188967190843744892778644333e-
4*p4*p5*p7+.16771188967190843744892778644333e-
4*p3*p6*p7+.16771188967190843744892778644333e-
4*p3*p5*p6+.16771188967190843744892778644333e-
4*p3*p4*p6+.16771188967190843744892778644333e-
4*p4*p5*p6+.16771188967190843744892778644333e-
4*p5*p6*p7+.16771188967190843744892778644333e-
4*p4*p7*p8+.16771188967190843744892778644333e-
4*p5*p7*p8+.16771188967190843744892778644333e-
4*p4*p6*p7+.20585409509137520534651185374607e-
3*p3*p4*p5*p6+.17701040746951615431047030382294e-
2*p4+.17701040746951615431047030382294e-
2*p9+.17701040746951615431047030382294e-
2*p8+.17701040746951615431047030382294e-
2*p6+.10726495948709906150231903754564e-
2*p3*p4+.16771188967190843744892778644333e-
4*p3*p4*p5+.16771188967190843744892778644333e-
4*p3*p7*p8+.16771188967190843744892778644333e-
4*p4*p6*p8+.16771188967190843744892778644333e-
4*p6*p7*p8+.16771188967190843744892778644333e-
4*p3*p5*p8+.16771188967190843744892778644333e-
4*p3*p4*p8+.16771188967190843744892778644333e-

4*p4*p5*p8+.16771188967190843744892778644333e-

4*p3*p6*p8+.16771188967190843744892778644333e-

4*p6*p8*p9+.16771188967190843744892778644333e-

4*p4*p8*p9+.16771188967190843744892778644333e-

4*p5*p7*p9+.16771188967190843744892778644333e-

4*p3*p8*p9+.16771188967190843744892778644333e-

4*p5*p8*p9+.16771188967190843744892778644333e-

4*p7*p8*p9+.16771188967190843744892778644333e-

4*p4*p7*p9+.16771188967190843744892778644333e-

4*p3*p7*p9+.16771188967190843744892778644333e-

4*p5*p6*p9+.16771188967190843744892778644333e-

4*p6*p7*p9+.16771188967190843744892778644333e-

4*p4*p5*p9+.16771188967190843744892778644333e-

4*p3*p5*p9+.16771188967190843744892778644333e-

4*p3*p4*p9+.16771188967190843744892778644333e-

4*p3*p6*p9+.17701040746951615431047030382294e-

2*p7+.17701040746951615431047030382294e-

2*p5+.17701040746951615431047030382294e-

2*p3+.32781281723403729885604413457166e-

2*p3*p4*p5*p6*p7+.62773696376947046872854858558771e-

1*p3*p4*p5*p6*p7*p8+.10726495948709906150231903754564e-

2*p6*p9+.10726495948709906150231903754564e-

2*p5*p9+.10726495948709906150231903754564e-

2*p4*p8+.10726495948709906150231903754564e-

2*p6*p8+.10726495948709906150231903754564e-

2*p3*p9+.10726495948709906150231903754564e-

2*p5*p8+.10726495948709906150231903754564e-

2*p8*p9+.10726495948709906150231903754564e-

2*p7*p8+.10726495948709906150231903754564e-

2*p4*p7+.10726495948709906150231903754564e-

2*p5*p7+.10726495948709906150231903754564e-

2*p3*p8+.10726495948709906150231903754564e-
2*p7*p9+.20585409509137520534651185374607e-
3*p4*p5*p6*p8+.20585409509137520534651185374607e-
3*p3*p5*p6*p8+.20585409509137520534651185374607e-
3*p3*p4*p6*p8+.20585409509137520534651185374607e-
3*p3*p4*p5*p8+.20585409509137520534651185374607e-
3*p5*p6*p7*p8+.20585409509137520534651185374607e-
3*p4*p6*p7*p8+.20585409509137520534651185374607e-
3*p4*p5*p7*p8+.20585409509137520534651185374607e-
3*p3*p6*p7*p8+.20585409509137520534651185374607e-
3*p3*p5*p7*p8+.20585409509137520534651185374607e-
3*p3*p4*p7*p8+.20585409509137520534651185374607e-
3*p4*p5*p6*p9+.20585409509137520534651185374607e-
3*p3*p5*p6*p9+.20585409509137520534651185374607e-
3*p3*p4*p6*p9+.20585409509137520534651185374607e-
3*p3*p4*p5*p9+.20585409509137520534651185374607e-
3*p5*p6*p7*p9+.20585409509137520534651185374607e-
3*p4*p6*p7*p9+.20585409509137520534651185374607e-
3*p4*p5*p7*p9+.20585409509137520534651185374607e-
3*p3*p6*p7*p9+.20585409509137520534651185374607e-
3*p3*p5*p7*p9+.20585409509137520534651185374607e-
3*p3*p4*p7*p9+.20585409509137520534651185374607e-
3*p5*p6*p8*p9+.20585409509137520534651185374607e-
3*p5*p7*p8*p9+.20585409509137520534651185374607e-
3*p4*p7*p8*p9+.20585409509137520534651185374607e-
3*p3*p7*p8*p9+.20585409509137520534651185374607e-
3*p4*p6*p8*p9+.20585409509137520534651185374607e-
3*p6*p7*p8*p9+.20585409509137520534651185374607e-
3*p3*p5*p8*p9+.20585409509137520534651185374607e-
3*p4*p5*p8*p9+.20585409509137520534651185374607e-3*p3*p6*p8*p9-
58.012997967149969559776728894198*p3*p4*p5*p6*p7*p8*p9+.2058540950913752

0534651185374607e-3*p3*p4*p5*p7+.20585409509137520534651185374607e-
3*p3*p4*p6*p7+.20585409509137520534651185374607e-
3*p3*p5*p6*p7+.20585409509137520534651185374607e-
3*p4*p5*p6*p7+.20585409509137520534651185374607e-
3*p3*p4*p8*p9+.10726495948709906150231903754564e-
2*p4*p9+.10726495948709906150231903754564e-
2*p3*p7+.16771188967190843744892778644333e-
4*p4*p6*p9+.10726495948709906150231903754564e-
2*p5*p6+.10726495948709906150231903754564e-
2*p6*p7+.10726495948709906150231903754564e-
2*p4*p5+.10726495948709906150231903754564e-
2*p3*p5+.10726495948709906150231903754564e-
2*p4*p6+.10726495948709906150231903754564e-
2*p3*p6+.32781281723403729885604413457166e-
2*p4*p5*p6*p7*p8+.32781281723403729885604413457166e-
2*p3*p5*p6*p7*p8+.32781281723403729885604413457166e-
2*p3*p4*p6*p7*p8+.32781281723403729885604413457166e-
2*p3*p4*p5*p7*p8+.32781281723403729885604413457166e-
2*p3*p4*p5*p6*p8+.32781281723403729885604413457166e-
2*p4*p5*p6*p7*p9+.32781281723403729885604413457166e-
2*p3*p5*p6*p7*p9+.32781281723403729885604413457166e-
2*p3*p4*p6*p7*p9+.32781281723403729885604413457166e-
2*p3*p4*p5*p7*p9+.32781281723403729885604413457166e-
2*p3*p4*p5*p6*p9+.32781281723403729885604413457166e-
2*p4*p5*p6*p8*p9+.32781281723403729885604413457166e-
2*p3*p5*p6*p8*p9+.32781281723403729885604413457166e-
2*p3*p4*p6*p8*p9+.32781281723403729885604413457166e-
2*p3*p4*p5*p8*p9+.32781281723403729885604413457166e-
2*p5*p6*p7*p8*p9+.32781281723403729885604413457166e-
2*p4*p6*p7*p8*p9+.32781281723403729885604413457166e-
2*p4*p5*p7*p8*p9+.32781281723403729885604413457166e-

2*p3*p6*p7*p8*p9+.32781281723403729885604413457166e-
2*p3*p5*p7*p8*p9+.32781281723403729885604413457166e-
2*p3*p4*p7*p8*p9+.62773696376947046872854858558771e-
1*p3*p4*p5*p6*p7*p9+.62773696376947046872854858558771e-
1*p4*p5*p6*p7*p8*p9+.62773696376947046872854858558771e-
1*p3*p5*p6*p7*p8*p9+.62773696376947046872854858558771e-
1*p3*p4*p6*p7*p8*p9+.62773696376947046872854858558771e-
1*p3*p4*p5*p7*p8*p9+.62773696376947046872854858558771e-
1*p3*p4*p5*p6*p8*p9+.21798759549135961824828703227369

f=-.10581735594452713475320140611046e-2*p3*p4*p7-
.10581735594452713475320140611046e-2*p3*p5*p7-
.10581735594452713475320140611046e-2*p5*p6*p8-
.10581735594452713475320140611046e-2*p4*p5*p7-
.10581735594452713475320140611046e-2*p3*p6*p7-
.10581735594452713475320140611046e-2*p3*p5*p6-
.10581735594452713475320140611046e-2*p3*p4*p6-
.10581735594452713475320140611046e-2*p4*p5*p6-
.10581735594452713475320140611046e-2*p5*p6*p7-
.10581735594452713475320140611046e-2*p4*p7*p8-
.10581735594452713475320140611046e-2*p5*p7*p8-
.10581735594452713475320140611046e-
2*p4*p6*p7+.16091157275600908777474783147905e-3*p3*p4*p5*p6-
.34432696226496828390450391211620-.21645973188296398817367798204861*p4-
.21645973188296398817367798204861*p9-
.21645973188296398817367798204861*p8-
.21645973188296398817367798204861*p6-.84424762139823372976059996947924e-
3*p3*p4-.10581735594452713475320140611046e-2*p3*p4*p5-
.10581735594452713475320140611046e-2*p3*p7*p8-
.10581735594452713475320140611046e-2*p4*p6*p8-
.10581735594452713475320140611046e-2*p6*p7*p8-
.10581735594452713475320140611046e-2*p3*p5*p8-

.10581735594452713475320140611046e-2*p3*p4*p8-

.10581735594452713475320140611046e-2*p4*p5*p8-

.10581735594452713475320140611046e-2*p3*p6*p8-

.10581735594452713475320140611046e-2*p6*p8*p9-

.10581735594452713475320140611046e-2*p4*p8*p9-

.10581735594452713475320140611046e-2*p5*p7*p9-

.10581735594452713475320140611046e-2*p3*p8*p9-

.10581735594452713475320140611046e-2*p5*p8*p9-

.10581735594452713475320140611046e-2*p7*p8*p9-

.10581735594452713475320140611046e-2*p4*p7*p9-

.10581735594452713475320140611046e-2*p3*p7*p9-

.10581735594452713475320140611046e-2*p5*p6*p9-

.10581735594452713475320140611046e-2*p6*p7*p9-

.10581735594452713475320140611046e-2*p4*p5*p9-

.10581735594452713475320140611046e-2*p3*p5*p9-

.10581735594452713475320140611046e-2*p3*p4*p9-

.10581735594452713475320140611046e-2*p3*p6*p9-

.21645973188296398817367798204861*p7-

.21645973188296398817367798204861*p5-

.21645973188296398817367798204861*p3+.26236590447153215229644435045563e-

2*p3*p4*p5*p6*p7+.50904926728163759282275331488011e-1*p3*p4*p5*p6*p7*p8-

.84424762139823372976059996947924e-3*p6*p9-

.84424762139823372976059996947924e-3*p5*p9-

.84424762139823372976059996947924e-3*p4*p8-

.84424762139823372976059996947924e-3*p6*p8-

.84424762139823372976059996947924e-3*p3*p9-

.84424762139823372976059996947924e-3*p5*p8-

.84424762139823372976059996947924e-3*p8*p9-

.84424762139823372976059996947924e-3*p7*p8-

.84424762139823372976059996947924e-3*p4*p7-

.84424762139823372976059996947924e-3*p5*p7-

.84424762139823372976059996947924e-3*p3*p8-
.84424762139823372976059996947924e-
3*p7*p9+.16091157275600908777474783147905e-
3*p4*p5*p6*p8+.16091157275600908777474783147905e-
3*p3*p5*p6*p8+.16091157275600908777474783147905e-
3*p3*p4*p6*p8+.16091157275600908777474783147905e-
3*p3*p4*p5*p8+.16091157275600908777474783147905e-
3*p5*p6*p7*p8+.16091157275600908777474783147905e-
3*p4*p6*p7*p8+.16091157275600908777474783147905e-
3*p4*p5*p7*p8+.16091157275600908777474783147905e-
3*p3*p6*p7*p8+.16091157275600908777474783147905e-
3*p3*p5*p7*p8+.16091157275600908777474783147905e-
3*p3*p4*p7*p8+.16091157275600908777474783147905e-
3*p4*p5*p6*p9+.16091157275600908777474783147905e-
3*p3*p5*p6*p9+.16091157275600908777474783147905e-
3*p3*p4*p6*p9+.16091157275600908777474783147905e-
3*p3*p4*p5*p9+.16091157275600908777474783147905e-
3*p5*p6*p7*p9+.16091157275600908777474783147905e-
3*p4*p6*p7*p9+.16091157275600908777474783147905e-
3*p4*p5*p7*p9+.16091157275600908777474783147905e-
3*p3*p6*p7*p9+.16091157275600908777474783147905e-
3*p3*p5*p7*p9+.16091157275600908777474783147905e-
3*p3*p4*p7*p9+.16091157275600908777474783147905e-
3*p5*p6*p8*p9+.16091157275600908777474783147905e-
3*p5*p7*p8*p9+.16091157275600908777474783147905e-
3*p4*p7*p8*p9+.16091157275600908777474783147905e-
3*p3*p7*p8*p9+.16091157275600908777474783147905e-
3*p4*p6*p8*p9+.16091157275600908777474783147905e-
3*p6*p7*p8*p9+.16091157275600908777474783147905e-
3*p3*p5*p8*p9+.16091157275600908777474783147905e-
3*p4*p5*p8*p9+.16091157275600908777474783147905e-3*p3*p6*p8*p9-

50.136636400298773632227367463787*p3*p4*p5*p6*p7*p8*p9+.1609115727560090
8777474783147905e-3*p3*p4*p5*p7+.16091157275600908777474783147905e-
3*p3*p4*p6*p7+.16091157275600908777474783147905e-
3*p3*p5*p6*p7+.16091157275600908777474783147905e-
3*p4*p5*p6*p7+.16091157275600908777474783147905e-3*p3*p4*p8*p9-
.84424762139823372976059996947924e-3*p4*p9-
.84424762139823372976059996947924e-3*p3*p7-
.10581735594452713475320140611046e-2*p4*p6*p9-
.84424762139823372976059996947924e-3*p5*p6-
.84424762139823372976059996947924e-3*p6*p7-
.84424762139823372976059996947924e-3*p4*p5-
.84424762139823372976059996947924e-3*p3*p5-
.84424762139823372976059996947924e-3*p4*p6-
.84424762139823372976059996947924e-
3*p3*p6+.26236590447153215229644435045563e-
2*p4*p5*p6*p7*p8+.26236590447153215229644435045563e-
2*p3*p5*p6*p7*p8+.26236590447153215229644435045563e-
2*p3*p4*p6*p7*p8+.26236590447153215229644435045563e-
2*p3*p4*p5*p7*p8+.26236590447153215229644435045563e-
2*p3*p4*p5*p6*p8+.26236590447153215229644435045563e-
2*p4*p5*p6*p7*p9+.26236590447153215229644435045563e-
2*p3*p5*p6*p7*p9+.26236590447153215229644435045563e-
2*p3*p4*p6*p7*p9+.26236590447153215229644435045563e-
2*p3*p4*p5*p7*p9+.26236590447153215229644435045563e-
2*p3*p4*p5*p6*p9+.26236590447153215229644435045563e-
2*p4*p5*p6*p8*p9+.26236590447153215229644435045563e-
2*p3*p5*p6*p8*p9+.26236590447153215229644435045563e-
2*p3*p4*p6*p8*p9+.26236590447153215229644435045563e-
2*p3*p4*p5*p8*p9+.26236590447153215229644435045563e-
2*p5*p6*p7*p8*p9+.26236590447153215229644435045563e-
2*p4*p6*p7*p8*p9+.26236590447153215229644435045563e-

2*p4*p5*p7*p8*p9+.26236590447153215229644435045563e-
2*p3*p6*p7*p8*p9+.26236590447153215229644435045563e-
2*p3*p5*p7*p8*p9+.26236590447153215229644435045563e-
2*p3*p4*p7*p8*p9+.50904926728163759282275331488011e-
1*p3*p4*p5*p6*p7*p9+.50904926728163759282275331488011e-
1*p4*p5*p6*p7*p8*p9+.50904926728163759282275331488011e-
1*p3*p5*p6*p7*p8*p9+.50904926728163759282275331488011e-
1*p3*p4*p6*p7*p8*p9+.50904926728163759282275331488011e-
1*p3*p4*p5*p7*p8*p9+.50904926728163759282275331488011e-
1*p3*p4*p5*p6*p8*p9

g=-.25434013742621446625319462135741e-2*p3*p4*p7-
.25434013742621446625319462135741e-2*p3*p5*p7-
.25434013742621446625319462135741e-2*p5*p6*p8-
.25434013742621446625319462135741e-2*p4*p5*p7-
.25434013742621446625319462135741e-2*p3*p6*p7-
.25434013742621446625319462135741e-2*p3*p5*p6-
.25434013742621446625319462135741e-2*p3*p4*p6-
.25434013742621446625319462135741e-2*p4*p5*p6-
.25434013742621446625319462135741e-2*p5*p6*p7-
.25434013742621446625319462135741e-2*p4*p7*p8-
.25434013742621446625319462135741e-2*p5*p7*p8-
.25434013742621446625319462135741e-2*p4*p6*p7-
.40522743562954819130417800974649e-1*p3*p4*p5*p6-
.13220215969456756273917256732944e-1*p4-
.13220215969456756273917256732944e-1*p9-
.13220215969456756273917256732944e-1*p8-
.13220215969456756273917256732944e-1*p6-
.20709656127326634163327587095002e-3*p3*p4-
.25434013742621446625319462135741e-2*p3*p4*p5-
.25434013742621446625319462135741e-2*p3*p7*p8-
.25434013742621446625319462135741e-2*p4*p6*p8-

.25434013742621446625319462135741e-2*p6*p7*p8-
.25434013742621446625319462135741e-2*p3*p5*p8-
.25434013742621446625319462135741e-2*p3*p4*p8-
.25434013742621446625319462135741e-2*p4*p5*p8-
.25434013742621446625319462135741e-2*p3*p6*p8-
.25434013742621446625319462135741e-2*p6*p8*p9-
.25434013742621446625319462135741e-2*p4*p8*p9-
.25434013742621446625319462135741e-2*p5*p7*p9-
.25434013742621446625319462135741e-2*p3*p8*p9-
.25434013742621446625319462135741e-2*p5*p8*p9-
.25434013742621446625319462135741e-2*p7*p8*p9-
.25434013742621446625319462135741e-2*p4*p7*p9-
.25434013742621446625319462135741e-2*p3*p7*p9-
.25434013742621446625319462135741e-2*p5*p6*p9-
.25434013742621446625319462135741e-2*p6*p7*p9-
.25434013742621446625319462135741e-2*p4*p5*p9-
.25434013742621446625319462135741e-2*p3*p5*p9-
.25434013742621446625319462135741e-2*p3*p4*p9-
.25434013742621446625319462135741e-2*p3*p6*p9-
.13220215969456756273917256732944e-1*p7-
.13220215969456756273917256732944e-1*p5-
.13220215969456756273917256732944e-1*p3-
.66231716253383242817045942477287*p3*p4*p5*p6*p7-
12.682419343254558468886475834706*p3*p4*p5*p6*p7*p8-
.20709656127326634163327587095002e-3*p6*p9-
.20709656127326634163327587095002e-3*p5*p9-
.20709656127326634163327587095002e-3*p4*p8-
.20709656127326634163327587095002e-3*p6*p8-
.20709656127326634163327587095002e-3*p3*p9-
.20709656127326634163327587095002e-3*p5*p8-
.20709656127326634163327587095002e-3*p8*p9-

.20709656127326634163327587095002e-3*p7*p8-

.20709656127326634163327587095002e-3*p4*p7-

.20709656127326634163327587095002e-3*p5*p7-

.20709656127326634163327587095002e-3*p3*p8-

.20709656127326634163327587095002e-3*p7*p9-

.40522743562954819130417800974649e-1*p4*p5*p6*p8-

.40522743562954819130417800974649e-1*p3*p5*p6*p8-

.40522743562954819130417800974649e-1*p3*p4*p6*p8-

.40522743562954819130417800974649e-1*p3*p4*p5*p8-

.40522743562954819130417800974649e-1*p5*p6*p7*p8-

.40522743562954819130417800974649e-1*p4*p6*p7*p8-

.40522743562954819130417800974649e-1*p4*p5*p7*p8-

.40522743562954819130417800974649e-1*p3*p6*p7*p8-

.40522743562954819130417800974649e-1*p3*p5*p7*p8-

.40522743562954819130417800974649e-1*p3*p4*p7*p8-

.40522743562954819130417800974649e-1*p4*p5*p6*p9-

.40522743562954819130417800974649e-1*p3*p5*p6*p9-

.40522743562954819130417800974649e-1*p3*p4*p6*p9-

.40522743562954819130417800974649e-1*p3*p4*p5*p9-

.40522743562954819130417800974649e-1*p5*p6*p7*p9-

.40522743562954819130417800974649e-1*p4*p6*p7*p9-

.40522743562954819130417800974649e-1*p4*p5*p7*p9-

.40522743562954819130417800974649e-1*p3*p6*p7*p9-

.40522743562954819130417800974649e-1*p3*p5*p7*p9-

.40522743562954819130417800974649e-1*p3*p4*p7*p9-

.40522743562954819130417800974649e-1*p5*p6*p8*p9-

.40522743562954819130417800974649e-1*p5*p7*p8*p9-

.40522743562954819130417800974649e-1*p4*p7*p8*p9-

.40522743562954819130417800974649e-1*p3*p7*p8*p9-

.40522743562954819130417800974649e-1*p4*p6*p8*p9-

.40522743562954819130417800974649e-1*p6*p7*p8*p9-

.40522743562954819130417800974649e-1*p3*p5*p8*p9-

.40522743562954819130417800974649e-1*p4*p5*p8*p9-

.40522743562954819130417800974649e-

1*p3*p6*p8*p9+11718.120936266314183147397099373*p3*p4*p5*p6*p7*p8*p9-

.40522743562954819130417800974649e-1*p3*p4*p5*p7-

.40522743562954819130417800974649e-1*p3*p4*p6*p7-

.40522743562954819130417800974649e-1*p3*p5*p6*p7-

.40522743562954819130417800974649e-1*p4*p5*p6*p7-

.40522743562954819130417800974649e-1*p3*p4*p8*p9-

.20709656127326634163327587095002e-3*p4*p9-

.20709656127326634163327587095002e-3*p3*p7-

.25434013742621446625319462135741e-2*p4*p6*p9-

.20709656127326634163327587095002e-3*p5*p6-

.20709656127326634163327587095002e-3*p6*p7-

.20709656127326634163327587095002e-3*p4*p5-

.20709656127326634163327587095002e-3*p3*p5-

.20709656127326634163327587095002e-3*p4*p6-

.20709656127326634163327587095002e-3*p3*p6-

.66231716253383242817045942477287*p4*p5*p6*p7*p8-

.66231716253383242817045942477287*p3*p5*p6*p7*p8-

.66231716253383242817045942477287*p3*p4*p6*p7*p8-

.66231716253383242817045942477287*p3*p4*p5*p7*p8-

.66231716253383242817045942477287*p3*p4*p5*p6*p8-

.66231716253383242817045942477287*p4*p5*p6*p7*p9-

.66231716253383242817045942477287*p3*p5*p6*p7*p9-

.66231716253383242817045942477287*p3*p4*p6*p7*p9-

.66231716253383242817045942477287*p3*p4*p5*p7*p9-

.66231716253383242817045942477287*p3*p4*p5*p6*p9-

.66231716253383242817045942477287*p4*p5*p6*p8*p9-

.66231716253383242817045942477287*p3*p5*p6*p8*p9-

.66231716253383242817045942477287*p3*p4*p6*p8*p9-

.66231716253383242817045942477287*p3*p4*p5*p8*p9-

.66231716253383242817045942477287*p5*p6*p7*p8*p9-

.66231716253383242817045942477287*p4*p6*p7*p8*p9-

.66231716253383242817045942477287*p4*p5*p7*p8*p9-

.66231716253383242817045942477287*p3*p6*p7*p8*p9-

.66231716253383242817045942477287*p3*p5*p7*p8*p9-

.66231716253383242817045942477287*p3*p4*p7*p8*p9-

12.682419343254558468886475834706*p3*p4*p5*p6*p7*p9-

12.682419343254558468886475834706*p4*p5*p6*p7*p8*p9-

12.682419343254558468886475834706*p3*p5*p6*p7*p8*p9-

12.682419343254558468886475834706*p3*p4*p6*p7*p8*p9-

12.682419343254558468886475834706*p3*p4*p5*p7*p8*p9-

12.682419343254558468886475834706*p3*p4*p5*p6*p8*p9-

.21818224526122728113624854231396e-1

h=.39513578494375908784377304392288e-

6*p3*p4*p7+.39513578494375908784377304392288e-

6*p3*p5*p7+.39513578494375908784377304392288e-

6*p5*p6*p8+.39513578494375908784377304392288e-

6*p4*p5*p7+.39513578494375908784377304392288e-

6*p3*p6*p7+.39513578494375908784377304392288e-

6*p3*p5*p6+.39513578494375908784377304392288e-

6*p3*p4*p6+.39513578494375908784377304392288e-

6*p4*p5*p6+.39513578494375908784377304392288e-

6*p5*p6*p7+.39513578494375908784377304392288e-

6*p4*p7*p8+.39513578494375908784377304392288e-

6*p5*p7*p8+.33977048777016218373374965099827e-

5+.39513578494375908784377304392288e-

6*p4*p6*p7+.62923487043043714747695863613769e-

5*p3*p4*p5*p6+.20589449019744666509558563641547e-

5*p4+.20589449019744666509558563641547e-

5*p9+.20589449019744666509558563641547e-

5*p8+.20589449019744666509558563641547e-

5*p6+.32192203483003323876919533993521e-

7*p3*p4+.39513578494375908784377304392288e-

6*p3*p4*p5+.39513578494375908784377304392288e-

6*p3*p7*p8+.39513578494375908784377304392288e-

6*p4*p6*p8+.39513578494375908784377304392288e-

6*p6*p7*p8+.39513578494375908784377304392288e-

6*p3*p5*p8+.39513578494375908784377304392288e-

6*p3*p4*p8+.39513578494375908784377304392288e-

6*p4*p5*p8+.39513578494375908784377304392288e-

6*p3*p6*p8+.39513578494375908784377304392288e-

6*p6*p8*p9+.39513578494375908784377304392288e-

6*p4*p8*p9+.39513578494375908784377304392288e-

6*p5*p7*p9+.39513578494375908784377304392288e-

6*p3*p8*p9+.39513578494375908784377304392288e-

6*p5*p8*p9+.39513578494375908784377304392288e-

6*p7*p8*p9+.39513578494375908784377304392288e-

6*p4*p7*p9+.39513578494375908784377304392288e-

6*p3*p7*p9+.39513578494375908784377304392288e-

6*p5*p6*p9+.39513578494375908784377304392288e-

6*p6*p7*p9+.39513578494375908784377304392288e-

6*p4*p5*p9+.39513578494375908784377304392288e-

6*p3*p5*p9+.39513578494375908784377304392288e-

6*p3*p4*p9+.39513578494375908784377304392288e-

6*p3*p6*p9+.20589449019744666509558563641547e-

5*p7+.20589449019744666509558563641547e-

5*p5+.20589449019744666509558563641547e-

5*p3+.12049375933335713007711151686696e-3*p3*p4*p5*p6*p7-

.11135562534481211257792652191658*p3*p4*p5*p6*p7*p8+.3219220348300332387

6919533993521e-7*p6*p9+.32192203483003323876919533993521e-

7*p5*p9+.32192203483003323876919533993521e-

7*p4*p8+.32192203483003323876919533993521e-

7*p6*p8+.32192203483003323876919533993521e-

7*p3*p9+.32192203483003323876919533993521e-

7*p5*p8+.32192203483003323876919533993521e-

7*p8*p9+.32192203483003323876919533993521e-

7*p7*p8+.32192203483003323876919533993521e-

7*p4*p7+.32192203483003323876919533993521e-

7*p5*p7+.32192203483003323876919533993521e-

7*p3*p8+.32192203483003323876919533993521e-

7*p7*p9+.62923487043043714747695863613769e-

5*p4*p5*p6*p8+.62923487043043714747695863613769e-

5*p3*p5*p6*p8+.62923487043043714747695863613769e-

5*p3*p4*p6*p8+.62923487043043714747695863613769e-

5*p3*p4*p5*p8+.62923487043043714747695863613769e-

5*p5*p6*p7*p8+.62923487043043714747695863613769e-

5*p4*p6*p7*p8+.62923487043043714747695863613769e-

5*p4*p5*p7*p8+.62923487043043714747695863613769e-

5*p3*p6*p7*p8+.62923487043043714747695863613769e-

5*p3*p5*p7*p8+.62923487043043714747695863613769e-

5*p3*p4*p7*p8+.62923487043043714747695863613769e-

5*p4*p5*p6*p9+.62923487043043714747695863613769e-

5*p3*p5*p6*p9+.62923487043043714747695863613769e-

5*p3*p4*p6*p9+.62923487043043714747695863613769e-

5*p3*p4*p5*p9+.62923487043043714747695863613769e-

5*p5*p6*p7*p9+.62923487043043714747695863613769e-

5*p4*p6*p7*p9+.62923487043043714747695863613769e-

5*p4*p5*p7*p9+.62923487043043714747695863613769e-

5*p3*p6*p7*p9+.62923487043043714747695863613769e-

5*p3*p5*p7*p9+.62923487043043714747695863613769e-

5*p3*p4*p7*p9+.62923487043043714747695863613769e-

5*p5*p6*p8*p9+.62923487043043714747695863613769e-

5*p5*p7*p8*p9+.62923487043043714747695863613769e-

5*p4*p7*p8*p9+.62923487043043714747695863613769e-

5*p3*p7*p8*p9+.62923487043043714747695863613769e-

5*p4*p6*p8*p9+.62923487043043714747695863613769e-

5*p6*p7*p8*p9+.62923487043043714747695863613769e-

5*p3*p5*p8*p9+.62923487043043714747695863613769e-

5*p4*p5*p8*p9+.62923487043043714747695863613769e-

5*p3*p6*p8*p9+727.68058674328478182257653385667*p3*p4*p5*p6*p7*p8*p9+.62923487043043714747695863613769e-

5*p3*p4*p5*p7+.62923487043043714747695863613769e-

5*p3*p4*p6*p7+.62923487043043714747695863613769e-

5*p3*p5*p6*p7+.62923487043043714747695863613769e-

5*p4*p5*p6*p7+.62923487043043714747695863613769e-

5*p3*p4*p8*p9+.32192203483003323876919533993521e-

7*p4*p9+.32192203483003323876919533993521e-

7*p3*p7+.39513578494375908784377304392288e-

6*p4*p6*p9+.32192203483003323876919533993521e-

7*p5*p6+.32192203483003323876919533993521e-

7*p6*p7+.32192203483003323876919533993521e-

7*p4*p5+.32192203483003323876919533993521e-

7*p3*p5+.32192203483003323876919533993521e-

7*p4*p6+.32192203483003323876919533993521e-

7*p3*p6+.12049375933335713007711151686696e-

3*p4*p5*p6*p7*p8+.12049375933335713007711151686696e-

3*p3*p5*p6*p7*p8+.12049375933335713007711151686696e-

3*p3*p4*p6*p7*p8+.12049375933335713007711151686696e-

3*p3*p4*p5*p7*p8+.12049375933335713007711151686696e-

3*p3*p4*p5*p6*p8+.12049375933335713007711151686696e-

3*p4*p5*p6*p7*p9+.12049375933335713007711151686696e-

3*p3*p5*p6*p7*p9+.12049375933335713007711151686696e-

3*p3*p4*p6*p7*p9+.12049375933335713007711151686696e-

3*p3*p4*p5*p7*p9+.12049375933335713007711151686696e-
3*p3*p4*p5*p6*p9+.12049375933335713007711151686696e-
3*p4*p5*p6*p8*p9+.12049375933335713007711151686696e-
3*p3*p5*p6*p8*p9+.12049375933335713007711151686696e-
3*p3*p4*p6*p8*p9+.12049375933335713007711151686696e-
3*p3*p4*p5*p8*p9+.12049375933335713007711151686696e-
3*p5*p6*p7*p8*p9+.12049375933335713007711151686696e-
3*p4*p6*p7*p8*p9+.12049375933335713007711151686696e-
3*p4*p5*p7*p8*p9+.12049375933335713007711151686696e-
3*p3*p6*p7*p8*p9+.12049375933335713007711151686696e-
3*p3*p5*p7*p8*p9+.12049375933335713007711151686696e-3*p3*p4*p7*p8*p9-
.11135562534481211257792652191658*p3*p4*p5*p6*p7*p9-
.11135562534481211257792652191658*p4*p5*p6*p7*p8*p9-
.11135562534481211257792652191658*p3*p5*p6*p7*p8*p9-
.11135562534481211257792652191658*p3*p4*p6*p7*p8*p9-
.11135562534481211257792652191658*p3*p4*p5*p7*p8*p9-
.11135562534481211257792652191658*p3*p4*p5*p6*p8*p9

k=-.56466054207261314166885936337977e-5*p3*p4*p7-
.56466054207261314166885936337977e-5*p3*p5*p7-
.56466054207261314166885936337977e-5*p5*p6*p8-
.56466054207261314166885936337977e-5*p4*p5*p7-
.56466054207261314166885936337977e-5*p3*p6*p7-
.56466054207261314166885936337977e-5*p3*p5*p6-
.56466054207261314166885936337977e-5*p3*p4*p6-
.56466054207261314166885936337977e-5*p4*p5*p6-
.56466054207261314166885936337977e-
5*p5*p6*p7+.19938963305375075745406627240308-
.56466054207261314166885936337977e-5*p4*p7*p8-
.56466054207261314166885936337977e-5*p5*p7*p8-
.56466054207261314166885936337977e-5*p4*p6*p7-
.69307957324645697872087620932252e-4*p3*p4*p5*p6-

.59596724376407031249222623828091e-3*p4-

.59596724376407031249222623828091e-3*p9-

.59596724376407031249222623828091e-3*p8-

.59596724376407031249222623828091e-3*p6-

.36114487939925327634404169866136e-3*p3*p4-

.56466054207261314166885936337977e-5*p3*p4*p5-

.56466054207261314166885936337977e-5*p3*p7*p8-

.56466054207261314166885936337977e-5*p4*p6*p8-

.56466054207261314166885936337977e-5*p6*p7*p8-

.56466054207261314166885936337977e-5*p3*p5*p8-

.56466054207261314166885936337977e-5*p3*p4*p8-

.56466054207261314166885936337977e-5*p4*p5*p8-

.56466054207261314166885936337977e-5*p3*p6*p8-

.56466054207261314166885936337977e-5*p6*p8*p9-

.56466054207261314166885936337977e-5*p4*p8*p9-

.56466054207261314166885936337977e-5*p5*p7*p9-

.56466054207261314166885936337977e-5*p3*p8*p9-

.56466054207261314166885936337977e-5*p5*p8*p9-

.56466054207261314166885936337977e-5*p7*p8*p9-

.56466054207261314166885936337977e-5*p4*p7*p9-

.56466054207261314166885936337977e-5*p3*p7*p9-

.56466054207261314166885936337977e-5*p5*p6*p9-

.56466054207261314166885936337977e-5*p6*p7*p9-

.56466054207261314166885936337977e-5*p4*p5*p9-

.56466054207261314166885936337977e-5*p3*p5*p9-

.56466054207261314166885936337977e-5*p3*p4*p9-

.56466054207261314166885936337977e-5*p3*p6*p9-

.59596724376407031249222623828091e-3*p7-

.59596724376407031249222623828091e-3*p5-

.59596724376407031249222623828091e-3*p3-

.11036961269700994213239110706339e-2*p3*p4*p5*p6*p7-

.21134953218552691380325212875076e-1*p3*p4*p5*p6*p7*p8-

.36114487939925327634404169866136e-3*p6*p9-

.36114487939925327634404169866136e-3*p5*p9-

.36114487939925327634404169866136e-3*p4*p8-

.36114487939925327634404169866136e-3*p6*p8-

.36114487939925327634404169866136e-3*p3*p9-

.36114487939925327634404169866136e-3*p5*p8-

.36114487939925327634404169866136e-3*p8*p9-

.36114487939925327634404169866136e-3*p7*p8-

.36114487939925327634404169866136e-3*p4*p7-

.36114487939925327634404169866136e-3*p5*p7-

.36114487939925327634404169866136e-3*p3*p8-

.36114487939925327634404169866136e-3*p7*p9-

.69307957324645697872087620932252e-4*p4*p5*p6*p8-

.69307957324645697872087620932252e-4*p3*p5*p6*p8-

.69307957324645697872087620932252e-4*p3*p4*p6*p8-

.69307957324645697872087620932252e-4*p3*p4*p5*p8-

.69307957324645697872087620932252e-4*p5*p6*p7*p8-

.69307957324645697872087620932252e-4*p4*p6*p7*p8-

.69307957324645697872087620932252e-4*p4*p5*p7*p8-

.69307957324645697872087620932252e-4*p3*p6*p7*p8-

.69307957324645697872087620932252e-4*p3*p5*p7*p8-

.69307957324645697872087620932252e-4*p3*p4*p7*p8-

.69307957324645697872087620932252e-4*p4*p5*p6*p9-

.69307957324645697872087620932252e-4*p3*p5*p6*p9-

.69307957324645697872087620932252e-4*p3*p4*p6*p9-

.69307957324645697872087620932252e-4*p3*p4*p5*p9-

.69307957324645697872087620932252e-4*p5*p6*p7*p9-

.69307957324645697872087620932252e-4*p4*p6*p7*p9-

.69307957324645697872087620932252e-4*p4*p5*p7*p9-

.69307957324645697872087620932252e-4*p3*p6*p7*p9-

.69307957324645697872087620932252e-4*p3*p5*p7*p9-

.69307957324645697872087620932252e-4*p3*p4*p7*p9-

.69307957324645697872087620932252e-4*p5*p6*p8*p9-

.69307957324645697872087620932252e-4*p5*p7*p8*p9-

.69307957324645697872087620932252e-4*p4*p7*p8*p9-

.69307957324645697872087620932252e-4*p3*p7*p8*p9-

.69307957324645697872087620932252e-4*p4*p6*p8*p9-

.69307957324645697872087620932252e-4*p6*p7*p8*p9-

.69307957324645697872087620932252e-4*p3*p5*p8*p9-

.69307957324645697872087620932252e-4*p4*p5*p8*p9-

.69307957324645697872087620932252e-

4*p3*p6*p8*p9+19.532098137747700551270543818764*p3*p4*p5*p6*p7*p8*p9-

.69307957324645697872087620932252e-4*p3*p4*p5*p7-

.69307957324645697872087620932252e-4*p3*p4*p6*p7-

.69307957324645697872087620932252e-4*p3*p5*p6*p7-

.69307957324645697872087620932252e-4*p4*p5*p6*p7-

.69307957324645697872087620932252e-4*p3*p4*p8*p9-

.36114487939925327634404169866136e-3*p4*p9-

.36114487939925327634404169866136e-3*p3*p7-

.56466054207261314166885936337977e-5*p4*p6*p9-

.36114487939925327634404169866136e-3*p5*p6-

.36114487939925327634404169866136e-3*p6*p7-

.36114487939925327634404169866136e-3*p4*p5-

.36114487939925327634404169866136e-3*p3*p5-

.36114487939925327634404169866136e-3*p4*p6-

.36114487939925327634404169866136e-3*p3*p6-

.11036961269700994213239110706339e-2*p4*p5*p6*p7*p8-

.11036961269700994213239110706339e-2*p3*p5*p6*p7*p8-

.11036961269700994213239110706339e-2*p3*p4*p6*p7*p8-

.11036961269700994213239110706339e-2*p3*p4*p5*p7*p8-

.11036961269700994213239110706339e-2*p3*p4*p5*p6*p8-

.11036961269700994213239110706339e-2*p4*p5*p6*p7*p9-

.11036961269700994213239110706339e-2*p3*p5*p6*p7*p9-

.11036961269700994213239110706339e-2*p3*p4*p6*p7*p9-

.11036961269700994213239110706339e-2*p3*p4*p5*p7*p9-

.11036961269700994213239110706339e-2*p3*p4*p5*p6*p9-

.11036961269700994213239110706339e-2*p4*p5*p6*p8*p9-

.11036961269700994213239110706339e-2*p3*p5*p6*p8*p9-

.11036961269700994213239110706339e-2*p3*p4*p6*p8*p9-

.11036961269700994213239110706339e-2*p3*p4*p5*p8*p9-

.11036961269700994213239110706339e-2*p5*p6*p7*p8*p9-

.11036961269700994213239110706339e-2*p4*p6*p7*p8*p9-

.11036961269700994213239110706339e-2*p4*p5*p7*p8*p9-

.11036961269700994213239110706339e-2*p3*p6*p7*p8*p9-

.11036961269700994213239110706339e-2*p3*p5*p7*p8*p9-

.11036961269700994213239110706339e-2*p3*p4*p7*p8*p9-

.21134953218552691380325212875076e-1*p3*p4*p5*p6*p7*p9-

.21134953218552691380325212875076e-1*p4*p5*p6*p7*p8*p9-

.21134953218552691380325212875076e-1*p3*p5*p6*p7*p8*p9-

.21134953218552691380325212875076e-1*p3*p4*p6*p7*p8*p9-

.21134953218552691380325212875076e-1*p3*p4*p5*p7*p8*p9-

.21134953218552691380325212875076e-1*p3*p4*p5*p6*p8*p9

l=.34083399246116936659002673601061e-

3*p3*p4*p7+.34083399246116936659002673601061e-

3*p3*p5*p7+.34083399246116936659002673601061e-

3*p5*p6*p8+.34083399246116936659002673601061e-

3*p4*p5*p7+.34083399246116936659002673601061e-

3*p3*p6*p7+.34083399246116936659002673601061e-

3*p3*p5*p6+.34083399246116936659002673601061e-

3*p3*p4*p6+.34083399246116936659002673601061e-

3*p4*p5*p6+.34083399246116936659002673601061e-

3*p5*p6*p7+.34083399246116936659002673601061e-

3*p4*p7*p8+.34083399246116936659002673601061e-
3*p5*p7*p8+.34083399246116936659002673601061e-3*p4*p6*p7-
.24365469690923157903287666446049e-3*p3*p4*p5*p6-
.20153333221545646732803399102433*p4-
.20153333221545646732803399102433*p9-
.20153333221545646732803399102433*p8-
.20153333221545646732803399102433*p6-.70307390878788321908723413903557e-
3*p3*p4+.34083399246116936659002673601061e-
3*p3*p4*p5+.34083399246116936659002673601061e-
3*p3*p7*p8+.34083399246116936659002673601061e-
3*p4*p6*p8+.34083399246116936659002673601061e-
3*p6*p7*p8+.34083399246116936659002673601061e-
3*p3*p5*p8+.34083399246116936659002673601061e-
3*p3*p4*p8+.34083399246116936659002673601061e-
3*p4*p5*p8+.34083399246116936659002673601061e-
3*p3*p6*p8+.34083399246116936659002673601061e-
3*p6*p8*p9+.34083399246116936659002673601061e-
3*p4*p8*p9+.34083399246116936659002673601061e-
3*p5*p7*p9+.34083399246116936659002673601061e-
3*p3*p8*p9+.34083399246116936659002673601061e-
3*p5*p8*p9+.34083399246116936659002673601061e-
3*p7*p8*p9+.34083399246116936659002673601061e-
3*p4*p7*p9+.34083399246116936659002673601061e-
3*p3*p7*p9+.34083399246116936659002673601061e-
3*p5*p6*p9+.34083399246116936659002673601061e-
3*p6*p7*p9+.34083399246116936659002673601061e-
3*p4*p5*p9+.34083399246116936659002673601061e-
3*p3*p5*p9+.34083399246116936659002673601061e-
3*p3*p4*p9+.34083399246116936659002673601061e-3*p3*p6*p9-
.20153333221545646732803399102433*p7-
.20153333221545646732803399102433*p5-

.2015333322154564673280339910243 3*p3-.39006962449537646019555245485773e-2*p3*p4*p5*p6*p7-.74918907416047423226120514825272e-1*p3*p4*p5*p6*p7*p8-
.70307390878788321908723413903557e-3*p6*p9-
.70307390878788321908723413903557e-3*p5*p9-
.70307390878788321908723413903557e-3*p4*p8-
.70307390878788321908723413903557e-3*p6*p8-
.70307390878788321908723413903557e-3*p3*p9-
.70307390878788321908723413903557e-3*p5*p8-
.70307390878788321908723413903557e-3*p8*p9-
.70307390878788321908723413903557e-3*p7*p8-
.70307390878788321908723413903557e-3*p4*p7-
.70307390878788321908723413903557e-3*p5*p7-
.70307390878788321908723413903557e-3*p3*p8-
.70307390878788321908723413903557e-3*p7*p9-
.32016897051329981383463237433554-.24365469690923157903287666446049e-3*p4*p5*p6*p8-.24365469690923157903287666446049e-3*p3*p5*p6*p8-
.24365469690923157903287666446049e-3*p3*p4*p6*p8-
.24365469690923157903287666446049e-3*p3*p4*p5*p8-
.24365469690923157903287666446049e-3*p5*p6*p7*p8-
.24365469690923157903287666446049e-3*p4*p6*p7*p8-
.24365469690923157903287666446049e-3*p4*p5*p7*p8-
.24365469690923157903287666446049e-3*p3*p6*p7*p8-
.24365469690923157903287666446049e-3*p3*p5*p7*p8-
.24365469690923157903287666446049e-3*p3*p4*p7*p8-
.24365469690923157903287666446049e-3*p4*p5*p6*p9-
.24365469690923157903287666446049e-3*p3*p5*p6*p9-
.24365469690923157903287666446049e-3*p3*p4*p6*p9-
.24365469690923157903287666446049e-3*p3*p4*p5*p9-
.24365469690923157903287666446049e-3*p5*p6*p7*p9-
.24365469690923157903287666446049e-3*p4*p6*p7*p9-
.24365469690923157903287666446049e-3*p4*p5*p7*p9-

.24365469690923157903287666446049e-3*p3*p6*p7*p9-

.24365469690923157903287666446049e-3*p3*p5*p7*p9-

.24365469690923157903287666446049e-3*p3*p4*p7*p9-

.24365469690923157903287666446049e-3*p5*p6*p8*p9-

.24365469690923157903287666446049e-3*p5*p7*p8*p9-

.24365469690923157903287666446049e-3*p4*p7*p8*p9-

.24365469690923157903287666446049e-3*p3*p7*p8*p9-

.24365469690923157903287666446049e-3*p4*p6*p8*p9-

.24365469690923157903287666446049e-3*p6*p7*p8*p9-

.24365469690923157903287666446049e-3*p3*p5*p8*p9-

.24365469690923157903287666446049e-3*p4*p5*p8*p9-

.24365469690923157903287666446049e-

3*p3*p6*p8*p9+70.278255361029326733793592815631*p3*p4*p5*p6*p7*p8*p9-

.24365469690923157903287666446049e-3*p3*p4*p5*p7-

.24365469690923157903287666446049e-3*p3*p4*p6*p7-

.24365469690923157903287666446049e-3*p3*p5*p6*p7-

.24365469690923157903287666446049e-3*p4*p5*p6*p7-

.24365469690923157903287666446049e-3*p3*p4*p8*p9-

.70307390878788321908723413903557e-3*p4*p9-

.70307390878788321908723413903557e-

3*p3*p7+.34083399246116936659002673601061e-3*p4*p6*p9-

.70307390878788321908723413903557e-3*p5*p6-

.70307390878788321908723413903557e-3*p6*p7-

.70307390878788321908723413903557e-3*p4*p5-

.70307390878788321908723413903557e-3*p3*p5-

.70307390878788321908723413903557e-3*p4*p6-

.70307390878788321908723413903557e-3*p3*p6-

.39006962449537646019555245485773e-2*p4*p5*p6*p7*p8-

.39006962449537646019555245485773e-2*p3*p5*p6*p7*p8-

.39006962449537646019555245485773e-2*p3*p4*p6*p7*p8-

.39006962449537646019555245485773e-2*p3*p4*p5*p7*p8-

.390069624495376460195552454857773e-2*p3*p4*p5*p6*p8-

.390069624495376460195552454857773e-2*p4*p5*p6*p7*p9-

.390069624495376460195552454857773e-2*p3*p5*p6*p7*p9-

.390069624495376460195552454857773e-2*p3*p4*p6*p7*p9-

.390069624495376460195552454857773e-2*p3*p4*p5*p7*p9-

.390069624495376460195552454857773e-2*p3*p4*p5*p6*p9-

.390069624495376460195552454857773e-2*p4*p5*p6*p8*p9-

.390069624495376460195552454857773e-2*p3*p5*p6*p8*p9-

.390069624495376460195552454857773e-2*p3*p4*p6*p8*p9-

.390069624495376460195552454857773e-2*p3*p4*p5*p8*p9-

.390069624495376460195552454857773e-2*p5*p6*p7*p8*p9-

.390069624495376460195552454857773e-2*p4*p6*p7*p8*p9-

.390069624495376460195552454857773e-2*p4*p5*p7*p8*p9-

.390069624495376460195552454857773e-2*p3*p6*p7*p8*p9-

.390069624495376460195552454857773e-2*p3*p5*p7*p8*p9-

.390069624495376460195552454857773e-2*p3*p4*p7*p8*p9-

.749189074160474232261205148252722e-1*p3*p4*p5*p6*p7*p9-

.749189074160474232261205148252722e-1*p4*p5*p6*p7*p8*p9-

.749189074160474232261205148252722e-1*p3*p5*p6*p7*p8*p9-

.749189074160474232261205148252722e-1*p3*p4*p6*p7*p8*p9-

.749189074160474232261205148252722e-1*p3*p4*p5*p7*p8*p9-

.749189074160474232261205148252722e-1*p3*p4*p5*p6*p8*p9

o=-.210161360134997639851342806557776e-1-

.245107858536112392724661892968861e-2*p3*p4*p7-

.245107858536112392724661892968861e-2*p3*p5*p7-

.245107858536112392724661892968861e-2*p5*p6*p8-

.245107858536112392724661892968861e-2*p4*p5*p7-

.245107858536112392724661892968861e-2*p3*p6*p7-

.245107858536112392724661892968861e-2*p3*p5*p6-

.245107858536112392724661892968861e-2*p3*p4*p6-

.245107858536112392724661892968861e-2*p4*p5*p6-

.245107858536112392724661892968610e-2*p5*p6*p7-

.245107858536112392724661892968610e-2*p4*p7*p8-

.245107858536112392724661892968610e-2*p5*p7*p8-

.245107858536112392724661892968610e-2*p4*p6*p7-

.390557521019316303316094158040840e-1*p3*p4*p5*p6-

.127339735191912001506045036038370e-1*p4-

.127339735191912001506045036038370e-1*p9-

.127339735191912001506045036038370e-1*p8-

.127339735191912001506045036038370e-1*p6-

.199556348910851797422157101346660e-3*p3*p4-

.245107858536112392724661892968610e-2*p3*p4*p5-

.245107858536112392724661892968610e-2*p3*p7*p8-

.245107858536112392724661892968610e-2*p4*p6*p8-

.245107858536112392724661892968610e-2*p6*p7*p8-

.245107858536112392724661892968610e-2*p3*p5*p8-

.245107858536112392724661892968610e-2*p3*p4*p8-

.245107858536112392724661892968610e-2*p4*p5*p8-

.245107858536112392724661892968610e-2*p3*p6*p8-

.245107858536112392724661892968610e-2*p6*p8*p9-

.245107858536112392724661892968610e-2*p4*p8*p9-

.245107858536112392724661892968610e-2*p5*p7*p9-

.245107858536112392724661892968610e-2*p3*p8*p9-

.245107858536112392724661892968610e-2*p5*p8*p9-

.245107858536112392724661892968610e-2*p7*p8*p9-

.245107858536112392724661892968610e-2*p4*p7*p9-

.245107858536112392724661892968610e-2*p3*p7*p9-

.245107858536112392724661892968610e-2*p5*p6*p9-

.245107858536112392724661892968610e-2*p6*p7*p9-

.245107858536112392724661892968610e-2*p4*p5*p9-

.245107858536112392724661892968610e-2*p3*p5*p9-

.245107858536112392724661892968610e-2*p3*p4*p9-

.24510785853611239272466189296861e-2*p3*p6*p9-

.12733973519191200150604503603837e-1*p7-

.12733973519191200150604503603837e-1*p5-

.12733973519191200150604503603837e-1*p3-

.61627293045128167957385613368354*p3*p4*p5*p6*p7-

11.801929610815526719371746930634*p3*p4*p5*p6*p7*p8-

.19955634891085179742215710134666e-3*p6*p9-

.19955634891085179742215710134666e-3*p5*p9-

.19955634891085179742215710134666e-3*p4*p8-

.19955634891085179742215710134666e-3*p6*p8-

.19955634891085179742215710134666e-3*p3*p9-

.19955634891085179742215710134666e-3*p5*p8-

.19955634891085179742215710134666e-3*p8*p9-

.19955634891085179742215710134666e-3*p7*p8-

.19955634891085179742215710134666e-3*p4*p7-

.19955634891085179742215710134666e-3*p5*p7-

.19955634891085179742215710134666e-3*p3*p8-

.19955634891085179742215710134666e-3*p7*p9-

.39055752101931630331609415804084e-1*p4*p5*p6*p8-

.39055752101931630331609415804084e-1*p3*p5*p6*p8-

.39055752101931630331609415804084e-1*p3*p4*p6*p8-

.39055752101931630331609415804084e-1*p3*p4*p5*p8-

.39055752101931630331609415804084e-1*p5*p6*p7*p8-

.39055752101931630331609415804084e-1*p4*p6*p7*p8-

.39055752101931630331609415804084e-1*p4*p5*p7*p8-

.39055752101931630331609415804084e-1*p3*p6*p7*p8-

.39055752101931630331609415804084e-1*p3*p5*p7*p8-

.39055752101931630331609415804084e-1*p3*p4*p7*p8-

.39055752101931630331609415804084e-1*p4*p5*p6*p9-

.39055752101931630331609415804084e-1*p3*p5*p6*p9-

.39055752101931630331609415804084e-1*p3*p4*p6*p9-

.39055752101931630331609415804084e-1*p3*p4*p5*p9-

.39055752101931630331609415804084e-1*p5*p6*p7*p9-

.39055752101931630331609415804084e-1*p4*p6*p7*p9-

.39055752101931630331609415804084e-1*p4*p5*p7*p9-

.39055752101931630331609415804084e-1*p3*p6*p7*p9-

.39055752101931630331609415804084e-1*p3*p5*p7*p9-

.39055752101931630331609415804084e-1*p3*p4*p7*p9-

.39055752101931630331609415804084e-1*p5*p6*p8*p9-

.39055752101931630331609415804084e-1*p5*p7*p8*p9-

.39055752101931630331609415804084e-1*p4*p7*p8*p9-

.39055752101931630331609415804084e-1*p3*p7*p8*p9-

.39055752101931630331609415804084e-1*p4*p6*p8*p9-

.39055752101931630331609415804084e-1*p6*p7*p8*p9-

.39055752101931630331609415804084e-1*p3*p5*p8*p9-

.39055752101931630331609415804084e-1*p4*p5*p8*p9-

.39055752101931630331609415804084e-

1*p3*p6*p8*p9+10910.562099578460692280012623466*p3*p4*p5*p6*p7*p8*p9-

.39055752101931630331609415804084e-1*p3*p4*p5*p7-

.39055752101931630331609415804084e-1*p3*p4*p6*p7-

.39055752101931630331609415804084e-1*p3*p5*p6*p7-

.39055752101931630331609415804084e-1*p4*p5*p6*p7-

.39055752101931630331609415804084e-1*p3*p4*p8*p9-

.19955634891085179742215710134666e-3*p4*p9-

.19955634891085179742215710134666e-3*p3*p7-

.24510785853611239272466189296861e-2*p4*p6*p9-

.19955634891085179742215710134666e-3*p5*p6-

.19955634891085179742215710134666e-3*p6*p7-

.19955634891085179742215710134666e-3*p4*p5-

.19955634891085179742215710134666e-3*p3*p5-

.19955634891085179742215710134666e-3*p4*p6-

.19955634891085179742215710134666e-3*p3*p6-

.61627293045128167957385613368354*p4*p5*p6*p7*p8-
.61627293045128167957385613368354*p3*p5*p6*p7*p8-
.61627293045128167957385613368354*p3*p4*p6*p7*p8-
.61627293045128167957385613368354*p3*p4*p5*p7*p8-
.61627293045128167957385613368354*p3*p4*p5*p6*p8-
.61627293045128167957385613368354*p4*p5*p6*p7*p9-
.61627293045128167957385613368354*p3*p5*p6*p7*p9-
.61627293045128167957385613368354*p3*p4*p6*p7*p9-
.61627293045128167957385613368354*p3*p4*p5*p7*p9-
.61627293045128167957385613368354*p3*p4*p5*p6*p9-
.61627293045128167957385613368354*p4*p5*p6*p8*p9-
.61627293045128167957385613368354*p3*p5*p6*p8*p9-
.61627293045128167957385613368354*p3*p4*p6*p8*p9-
.61627293045128167957385613368354*p3*p4*p5*p8*p9-
.61627293045128167957385613368354*p5*p6*p7*p8*p9-
.61627293045128167957385613368354*p4*p6*p7*p8*p9-
.61627293045128167957385613368354*p4*p5*p7*p8*p9-
.61627293045128167957385613368354*p3*p6*p7*p8*p9-
.61627293045128167957385613368354*p3*p5*p7*p8*p9-
.61627293045128167957385613368354*p3*p4*p7*p8*p9-
11.801929610815526719371746930634*p3*p4*p5*p6*p7*p9-
11.801929610815526719371746930634*p4*p5*p6*p7*p8*p9-
11.801929610815526719371746930634*p3*p5*p6*p7*p8*p9-
11.801929610815526719371746930634*p3*p4*p6*p7*p8*p9-
11.801929610815526719371746930634*p3*p4*p5*p7*p8*p9-
11.801929610815526719371746930634*p3*p4*p5*p6*p8*p9

r=.45754531391960605306795231495774e-
6*p3*p4*p7+.45754531391960605306795231495774e-
6*p3*p5*p7+.45754531391960605306795231495774e-
6*p5*p6*p8+.45754531391960605306795231495774e-
6*p4*p5*p7+.45754531391960605306795231495774e-

6*p3*p6*p7+.45754531391960605306795231495774e-
6*p3*p5*p6+.45754531391960605306795231495774e-
6*p3*p4*p6+.45754531391960605306795231495774e-
6*p4*p5*p6+.45754531391960605306795231495774e-
6*p5*p6*p7+.45754531391960605306795231495774e-
6*p4*p7*p8+.45754531391960605306795231495774e-
6*p5*p7*p8+.45754531391960605306795231495774e-
6*p4*p6*p7+.72861906537074389723851276956649e-
5*p3*p4*p5*p6+.39343537186727770002621719581241e-
5+.23841439510500581138901061156631e-
5*p4+.23841439510500581138901061156631e-
5*p9+.23841439510500581138901061156631e-
5*p8+.23841439510500581138901061156631e-
5*p6+.37276785372631009867859126474791e-
7*p3*p4+.45754531391960605306795231495774e-
6*p3*p4*p5+.45754531391960605306795231495774e-
6*p3*p7*p8+.45754531391960605306795231495774e-
6*p4*p6*p8+.45754531391960605306795231495774e-
6*p6*p7*p8+.45754531391960605306795231495774e-
6*p3*p5*p8+.45754531391960605306795231495774e-
6*p3*p4*p8+.45754531391960605306795231495774e-
6*p4*p5*p8+.45754531391960605306795231495774e-
6*p3*p6*p8+.45754531391960605306795231495774e-
6*p6*p8*p9+.45754531391960605306795231495774e-
6*p4*p8*p9+.45754531391960605306795231495774e-
6*p5*p7*p9+.45754531391960605306795231495774e-
6*p3*p8*p9+.45754531391960605306795231495774e-
6*p5*p8*p9+.45754531391960605306795231495774e-
6*p7*p8*p9+.45754531391960605306795231495774e-
6*p4*p7*p9+.45754531391960605306795231495774e-
6*p3*p7*p9+.45754531391960605306795231495774e-

6*p5*p6*p9+.45754531391960605306795231495774e-
6*p6*p7*p9+.45754531391960605306795231495774e-
6*p4*p5*p9+.45754531391960605306795231495774e-
6*p3*p5*p9+.45754531391960605306795231495774e-
6*p3*p4*p9+.45754531391960605306795231495774e-
6*p3*p6*p9+.23841439510500581138901061156631e-
5*p7+.23841439510500581138901061156631e-
5*p5+.23841439510500581138901061156631e-
5*p3+.13952508742629161782275413932553e-3*p3*p4*p5*p6*p7-
.12894363531857284389712095980372*p3*p4*p5*p6*p7*p8+.3727678537263100986
7859126474791e-7*p6*p9+.37276785372631009867859126474791e-
7*p5*p9+.37276785372631009867859126474791e-
7*p4*p8+.37276785372631009867859126474791e-
7*p6*p8+.37276785372631009867859126474791e-
7*p3*p9+.37276785372631009867859126474791e-
7*p5*p8+.37276785372631009867859126474791e-
7*p8*p9+.37276785372631009867859126474791e-
7*p7*p8+.37276785372631009867859126474791e-
7*p4*p7+.37276785372631009867859126474791e-
7*p5*p7+.37276785372631009867859126474791e-
7*p3*p8+.37276785372631009867859126474791e-
7*p7*p9+.72861906537074389723851276956649e-
5*p4*p5*p6*p8+.72861906537074389723851276956649e-
5*p3*p5*p6*p8+.72861906537074389723851276956649e-
5*p3*p4*p6*p8+.72861906537074389723851276956649e-
5*p3*p4*p5*p8+.72861906537074389723851276956649e-
5*p5*p6*p7*p8+.72861906537074389723851276956649e-
5*p4*p6*p7*p8+.72861906537074389723851276956649e-
5*p4*p5*p7*p8+.72861906537074389723851276956649e-
5*p3*p6*p7*p8+.72861906537074389723851276956649e-
5*p3*p5*p7*p8+.72861906537074389723851276956649e-

5*p3*p4*p7*p8+.72861906537074389723851276956649e-

5*p4*p5*p6*p9+.72861906537074389723851276956649e-

5*p3*p5*p6*p9+.72861906537074389723851276956649e-

5*p3*p4*p6*p9+.72861906537074389723851276956649e-

5*p3*p4*p5*p9+.72861906537074389723851276956649e-

5*p5*p6*p7*p9+.72861906537074389723851276956649e-

5*p4*p6*p7*p9+.72861906537074389723851276956649e-

5*p4*p5*p7*p9+.72861906537074389723851276956649e-

5*p3*p6*p7*p9+.72861906537074389723851276956649e-

5*p3*p5*p7*p9+.72861906537074389723851276956649e-

5*p3*p4*p7*p9+.72861906537074389723851276956649e-

5*p5*p6*p8*p9+.72861906537074389723851276956649e-

5*p5*p7*p8*p9+.72861906537074389723851276956649e-

5*p4*p7*p8*p9+.72861906537074389723851276956649e-

5*p3*p7*p8*p9+.72861906537074389723851276956649e-

5*p4*p6*p8*p9+.72861906537074389723851276956649e-

5*p6*p7*p8*p9+.72861906537074389723851276956649e-

5*p3*p5*p8*p9+.72861906537074389723851276956649e-

5*p4*p5*p8*p9+.72861906537074389723851276956649e-

5*p3*p6*p8*p9+700.50196816660816868021887319693*p3*p4*p5*p6*p7*p8*p9+.72
861906537074389723851276956649e-

5*p3*p4*p5*p7+.72861906537074389723851276956649e-

5*p3*p4*p6*p7+.72861906537074389723851276956649e-

5*p3*p5*p6*p7+.72861906537074389723851276956649e-

5*p4*p5*p6*p7+.72861906537074389723851276956649e-

5*p3*p4*p8*p9+.37276785372631009867859126474791e-

7*p4*p9+.37276785372631009867859126474791e-

7*p3*p7+.45754531391960605306795231495774e-

6*p4*p6*p9+.37276785372631009867859126474791e-

7*p5*p6+.37276785372631009867859126474791e-

7*p6*p7+.37276785372631009867859126474791e-

```
7*p4*p5+.37276785372631009867859126474791e-
7*p3*p5+.37276785372631009867859126474791e-
7*p4*p6+.37276785372631009867859126474791e-
7*p3*p6+.13952508742629161782275413932553e-
3*p4*p5*p6*p7*p8+.13952508742629161782275413932553e-
3*p3*p5*p6*p7*p8+.13952508742629161782275413932553e-
3*p3*p4*p6*p7*p8+.13952508742629161782275413932553e-
3*p3*p4*p5*p7*p8+.13952508742629161782275413932553e-
3*p3*p4*p5*p6*p8+.13952508742629161782275413932553e-
3*p4*p5*p6*p7*p9+.13952508742629161782275413932553e-
3*p3*p5*p6*p7*p9+.13952508742629161782275413932553e-
3*p3*p4*p6*p7*p9+.13952508742629161782275413932553e-
3*p3*p4*p5*p7*p9+.13952508742629161782275413932553e-
3*p3*p4*p5*p6*p9+.13952508742629161782275413932553e-
3*p4*p5*p6*p8*p9+.13952508742629161782275413932553e-
3*p3*p5*p6*p8*p9+.13952508742629161782275413932553e-
3*p3*p4*p6*p8*p9+.13952508742629161782275413932553e-
3*p3*p4*p5*p8*p9+.13952508742629161782275413932553e-
3*p5*p6*p7*p8*p9+.13952508742629161782275413932553e-
3*p4*p6*p7*p8*p9+.13952508742629161782275413932553e-
3*p4*p5*p7*p8*p9+.13952508742629161782275413932553e-
3*p3*p6*p7*p8*p9+.13952508742629161782275413932553e-
3*p3*p5*p7*p8*p9+.13952508742629161782275413932553e-3*p3*p4*p7*p8*p9-
.12894363531857284389712095980372*p3*p4*p5*p6*p7*p9-
.12894363531857284389712095980372*p4*p5*p6*p7*p8*p9-
.12894363531857284389712095980372*p3*p5*p6*p7*p8*p9-
.12894363531857284389712095980372*p3*p4*p6*p7*p8*p9-
.12894363531857284389712095980372*p3*p4*p5*p7*p8*p9-
.12894363531857284389712095980372*p3*p4*p5*p6*p8*p9
                        n12=[a b];n22=[c d];
                        n11=[e f g h];n21=[k l o r];
```

```
                    dN1=conv(n11,n22);dN2=conv(n12,n21);dN=dN1-dN2;
                    dNr(i)={roots(dN)};

datap(1,i)=p1;datap(2,i)=p2;datap(3,i)=p3;datap(4,i)=p4;datap(5,i)=p5;datap(6,i)=p6;data
p(7,i)=p7;datap(8,i)=p8;datap(9,i)=p9;
                    diary on;
                    display(i)
                    display(dNr{i})
                    diary off;
                    i=i+1;
                  end
                end
              end
            end
          end
        end
      end
   end
end

%%% Transfomated X-29 Nominal Result and Transformated Nominal Closed-Loop
Stability Vrification %%%
clc;clear;format long e;
% Plant a %
Na11=[1.172000000000000e-001                    4.215693804999958e-001
6.546706291135286e+001   4.132461548651955e+000   -7.737107377179808e-004];
Na12=[3.481000000000001e-001              3.004621760300199e-001              -
7.031347733880426e+001   -4.290260300163865e+000   6.681760043581730e-004];
Na21=[5.577000000000014e-001    2.131840749733499e+000    1.351789739836793e-
001   -2.609024107869118e-015];
```

```
Na22=[-3.264000000000067e-001   -2.273223021845780e+000   -1.390504955224294e-
001   -4.996003610813204e-016];
Da11=[1.000000000000000e+000                      5.590610000000000e+000                      -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da12=[1.000000000000000e+000                      5.590610000000000e+000                      -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da21=[1.000000000000000e+000                      5.590610000000000e+000                      -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
Da22=[1.000000000000000e+000                      5.590610000000000e+000                      -
6.998678243751319e+001   -4.333352271727545e+000   -3.775288701841896e-001];
% N matrix %
n12=[-8.376009523223778e+002 -5.784774387580804e+003];
n22=[-1.432291144790173e+003 -5.381735341628663e+003];
n11=[2.068406613167494e-001  1.840636113498792e+000  7.205764157500426e+000  -
1.937881730487596e-001];
n21=[2.031426370332519e-001  1.685275398750232e+000  6.623270023616939e+000  -
1.924208023993863e-001];
% Tried Gn %
gn11=1.8e+004;
gn12=0;gn21=0;
gn22=1;
% G %  %G=N*Gn%
g11=n11*gn11;
g12=n12*gn22;
g21=n21*gn11;
g22=n22*gn22;
gd=[1];
% Pa*G %
[tn1,td1]=mulnd(Na11,Da11,g11,gd);
[tn2,td2]=mulnd(Na12,Da12,g21,gd);
[num11,den11]=addnd(tn1,td1,tn2,td2);
```

```
[tn1,td1]=mulnd(Na11,Da11,g12,gd);
[tn2,td2]=mulnd(Na12,Da12,g22,gd);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g11,gd);
[tn2,td2]=mulnd(Na22,Da22,g21,gd);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g12,gd);
[tn2,td2]=mulnd(Na22,Da22,g22,gd);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PaG %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
```

```
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
s11=zpk(tf(numS11,denS11));
s11=minreal(s11)
s22=zpk(tf(numS22,denS22));
s22=minreal(s22)
s12=zpk(tf(numS12,denS12));
s12=minreal(s12)
s21=zpk(tf(numS21,denS21));
s21=minreal(s21)

%%%  2X2 X-29 transformed NS MIMO QFT EXAMPLE
clear all;clc;close all;format long e;
%   All four plants
% Plant a %
Na11=[1.172000000000000e-001                                  4.215693804999958e-001
6.546706291135286e+001    4.132461548651955e+000   -7.737107377179808e-004];
Na12=[3.481000000000001e-001                       3.004621760300199e-001                    -
7.031347733880426e+001   -4.290260300163865e+000    6.681760043581730e-004];
Na21=[5.577000000000014e-001     2.131840749733499e+000     1.351789739836793e-
001   -2.609024107869118e-015];
Na22=[-3.264000000000067e-001   -2.273223021845780e+000   -1.390504955224294e-
001   -4.996003610813204e-016];
```

Da11=[1.000000000000000e+000 5.590610000000000e+000 -6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da12=[1.000000000000000e+000 5.590610000000000e+000 -6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da21=[1.000000000000000e+000 5.590610000000000e+000 -6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da22=[1.000000000000000e+000 5.590610000000000e+000 -6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
% Plant b %
Nb11=[9.071999999999969e-003 1.932808712000278e-002 1.155041962224416e+000 4.755595748284797e-003 -2.796579577720032e-004];
Nb12=[7.101999999999997e-002 2.408502020000114e-002 -1.421095010395007e+000 -4.109295365955556e-003 2.950852136856669e-004];
Nb21=[8.576000000000006e-002 8.475674789998777e-002 1.254879739693904e-003 -3.747002708109903e-016];
Nb22=[-5.018999999999885e-002 -1.030980381000131e-001 -8.658115718451487e-004 -2.914335439641036e-016];
Db11=[1.000000000000000e+000 1.518209999999999e+000 -9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db12=[1.000000000000000e+000 1.518209999999999e+000 -9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db21=[1.000000000000000e+000 1.518209999999999e+000 -9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db22=[1.000000000000000e+000 1.518209999999999e+000 -9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
% Plant c %
Nc11=[2.100000000000002e-002 4.350065659999158e-002 4.225829411104126e+000 4.912148017935841e-002 -1.646391253174590e-004];
Nc12=[1.450000000000000e-001 5.297899920000204e-002 -7.436124264278636e+000 -7.703155594005817e-002 2.414747452200994e-004];

Nc21=[1.589000000000143e-001 1.862845259999553e-001 2.834621212944311e-003 -1.526556658859590e-016];

Nc22=[-1.179999999999932e-001 -3.251917000000226e-001 -4.029330462134440e-003 -1.110223024625157e-016];

Dc11=[1.000000000000000e+000 1.844530000000003e+000 -2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];

Dc12=[1.000000000000000e+000 1.844530000000003e+000 -2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];

Dc21=[1.000000000000000e+000 1.844530000000003e+000 -2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];

Dc22=[1.000000000000000e+000 1.844530000000003e+000 -2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];

% Plant d %

Nd11=[4.422999999999955e-003 4.198247239997510e-003 6.009732055199013e-001 5.601114275346930e-003 -1.388627209596149e-005];

Nd12=[4.373000000000005e-002 5.253591799998514e-003 -7.493085240193560e-001 -5.172401463264253e-003 1.061499190466164e-005];

Nd21=[5.476000000000147e-002 2.306487227999554e-002 3.664198444066069e-004 9.020562075079397e-017];

Nd22=[-3.642999999999996e-002 -2.771510290000112e-002 -1.819212840562523e-004 1.387778780781446e-016];

Dd11=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];

Dd12=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];

Dd21=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];

Dd22=[1.000000000000000e+000 6.068300000000018e-001 -7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];

% N matrix %

n12=[-8.376009523223778e+002 -5.784774387580804e+003];

```
n22=[-1.432291144790173e+003 -5.381735341628663e+003];
n11=[2.068406613167494e-001 1.840636113498792e+000 7.205764157500426e+000 -
1.937881730487596e-001];
n21=[2.031426370332519e-001 1.685275398750232e+000 6.623270023616939e+000 -
1.924208023993863e-001];
nd=[1];
%   Find Transformated Plant P'=P*N
% Pa' %
[tn1,td1]=mulnd(Na11,Da11,n11,nd);
[tn2,td2]=mulnd(Na12,Da12,n21,nd);
[Nan11,Dan11]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(Nan11,Dan11));
s1=minreal(s1);
s1=tf(s1);
[Nan11,Dan11]=tfdata(s1,'v');
[tn1,td1]=mulnd(Na11,Da11,n12,nd);
[tn2,td2]=mulnd(Na12,Da12,n22,nd);
[Nan12,Dan12]=addnd(tn1,td1,tn2,td2);
s2=zpk(tf(Nan12,Dan12));
s2=minreal(s2);
s2=tf(s2);
[Nan12,Dan12]=tfdata(s2,'v');
[tn1,td1]=mulnd(Na21,Da21,n11,nd);
[tn2,td2]=mulnd(Na22,Da22,n21,nd);
[Nan21,Dan21]=addnd(tn1,td1,tn2,td2);
s3=zpk(tf(Nan21,Dan21));
s3=minreal(s3);
s3=tf(s3);
[Nan21,Dan21]=tfdata(s3,'v');
[tn1,td1]=mulnd(Na21,Da21,n12,nd);
[tn2,td2]=mulnd(Na22,Da22,n22,nd);
```

```
[Nan22,Dan22]=addnd(tn1,td1,tn2,td2);
s4=zpk(tf(Nan22,Dan22));
s4=minreal(s4);
s4=tf(s4);
[Nan22,Dan22]=tfdata(s4,'v');
% Pb' %
[tn1,td1]=mulnd(Nb11,Db11,n11,nd);
[tn2,td2]=mulnd(Nb12,Db12,n21,nd);
[Nbn11,Dbn11]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(Nbn11,Dbn11));
s1=minreal(s1);
s1=tf(s1);
[Nbn11,Dbn11]=tfdata(s1,'v');
[tn1,td1]=mulnd(Nb11,Db11,n12,nd);
[tn2,td2]=mulnd(Nb12,Db12,n22,nd);
[Nbn12,Dbn12]=addnd(tn1,td1,tn2,td2);
s2=zpk(tf(Nbn12,Dbn12));
s2=minreal(s2);
s2=tf(s2);
[Nbn12,Dbn12]=tfdata(s2,'v');
[tn1,td1]=mulnd(Nb21,Db21,n11,nd);
[tn2,td2]=mulnd(Nb22,Db22,n21,nd);
[Nbn21,Dbn21]=addnd(tn1,td1,tn2,td2);
s3=zpk(tf(Nbn21,Dbn21));
s3=minreal(s3);
s3=tf(s3);
[Nbn21,Dbn21]=tfdata(s3,'v');
[tn1,td1]=mulnd(Nb21,Db21,n12,nd);
[tn2,td2]=mulnd(Nb22,Db22,n22,nd);
[Nbn22,Dbn22]=addnd(tn1,td1,tn2,td2);
s4=zpk(tf(Nbn22,Dbn22));
```

```
s4=minreal(s4);
s4=tf(s4);
[Nbn22,Dbn22]=tfdata(s4,'v');
% Pc' %
[tn1,td1]=mulnd(Nc11,Dc11,n11,nd);
[tn2,td2]=mulnd(Nc12,Dc12,n21,nd);
[Ncn11,Dcn11]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(Ncn11,Dcn11));
s1=minreal(s1);
s1=tf(s1);
[Ncn11,Dcn11]=tfdata(s1,'v');
[tn1,td1]=mulnd(Nc11,Dc11,n12,nd);
[tn2,td2]=mulnd(Nc12,Dc12,n22,nd);
[Ncn12,Dcn12]=addnd(tn1,td1,tn2,td2);
s2=zpk(tf(Ncn12,Dcn12));
s2=minreal(s2);
s2=tf(s2);
[Ncn12,Dcn12]=tfdata(s2,'v');
[tn1,td1]=mulnd(Nc21,Dc21,n11,nd);
[tn2,td2]=mulnd(Nc22,Dc22,n21,nd);
[Ncn21,Dcn21]=addnd(tn1,td1,tn2,td2);
s3=zpk(tf(Ncn21,Dcn21));
s3=minreal(s3);
s3=tf(s3);
[Ncn21,Dcn21]=tfdata(s3,'v');
[tn1,td1]=mulnd(Nc21,Dc21,n12,nd);
[tn2,td2]=mulnd(Nc22,Dc22,n22,nd);
[Ncn22,Dcn22]=addnd(tn1,td1,tn2,td2);
s4=zpk(tf(Ncn22,Dcn22));
s4=minreal(s4);
s4=tf(s4);
```

```
[Ncn22,Dcn22]=tfdata(s4,'v');
% Pd' %
[tn1,td1]=mulnd(Nd11,Dd11,n11,nd);
[tn2,td2]=mulnd(Nd12,Dd12,n21,nd);
[Ndn11,Ddn11]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(Ndn11,Ddn11));
s1=minreal(s1);
s1=tf(s1);
[Ndn11,Ddn11]=tfdata(s1,'v');
[tn1,td1]=mulnd(Nd11,Dd11,n12,nd);
[tn2,td2]=mulnd(Nd12,Dd12,n22,nd);
[Ndn12,Ddn12]=addnd(tn1,td1,tn2,td2);
s2=zpk(tf(Ndn12,Ddn12));
s2=minreal(s2);
s2=tf(s2);
[Ndn12,Ddn12]=tfdata(s2,'v');
[tn1,td1]=mulnd(Nd21,Dd21,n11,nd);
[tn2,td2]=mulnd(Nd22,Dd22,n21,nd);
[Ndn21,Ddn21]=addnd(tn1,td1,tn2,td2);
s3=zpk(tf(Ndn21,Ddn21));
s3=minreal(s3);
s3=tf(s3);
[Ndn21,Ddn21]=tfdata(s3,'v');
[tn1,td1]=mulnd(Nd21,Dd21,n12,nd);
[tn2,td2]=mulnd(Nd22,Dd22,n22,nd);
[Ndn22,Ddn22]=addnd(tn1,td1,tn2,td2);
s4=zpk(tf(Ndn22,Ddn22));
s4=minreal(s4);
s4=tf(s4);
[Ndn22,Ddn22]=tfdata(s4,'v');
ta=zeros(1,8);
```

```
while length(Nan11)<8
   l=length(Nan11);
   for i=1:l
      ta(9-i)=Nan11(l+1-i);
   end
   Nan11=ta;
   ta=zeros(1,8);
end
while length(Nan12)<8
   l=length(Nan12);
   for i=1:l
      ta(9-i)=Nan12(l+1-i);
   end
   Nan12=ta;
   ta=zeros(1,8);
end
while length(Nan21)<8
   l=length(Nan21);
   for i=1:l
      ta(9-i)=Nan21(l+1-i);
   end
   Nan21=ta;
   ta=zeros(1,8);
end
while length(Nan22)<8
   l=length(Nan22);
   for i=1:l
      ta(9-i)=Nan22(l+1-i);
   end
   Nan22=ta;
   ta=zeros(1,8);
```

```
end
while length(Dan11)<8
   l=length(Dan11);
   for i=1:l
      ta(9-i)=Dan11(l+1-i);
   end
   Dan11=ta;
   ta=zeros(1,8);
end
while length(Dan12)<8
   l=length(Dan12);
   for i=1:l
      ta(9-i)=Dan12(l+1-i);
   end
   Dan12=ta;
   ta=zeros(1,8);
end
while length(Dan21)<8
   l=length(Dan21);
   for i=1:l
      ta(9-i)=Dan21(l+1-i);
   end
   Dan21=ta;
   ta=zeros(1,8);
end
while length(Dan22)<8
   l=length(Dan22);
   for i=1:l
      ta(9-i)=Dan22(l+1-i);
   end
   Dan22=ta;
```

```
    ta=zeros(1,8);
end
while length(Nbn11)<8
    l=length(Nbn11);
    for i=1:l
        ta(9-i)=Nbn11(l+1-i);
    end
    Nbn11=ta;
    ta=zeros(1,8);
end
while length(Nbn12)<8
    l=length(Nbn12);
    for i=1:l
        ta(9-i)=Nbn12(l+1-i);
    end
    Nbn12=ta;
    ta=zeros(1,8);
end
while length(Nbn21)<8
    l=length(Nbn21);
    for i=1:l
        ta(9-i)=Nbn21(l+1-i);
    end
    Nbn21=ta;
    ta=zeros(1,8);
end
while length(Nbn22)<8
    l=length(Nbn22);
    for i=1:l
        ta(9-i)=Nbn22(l+1-i);
    end
```

```
    Nbn22=ta;
    ta=zeros(1,8);
end
while length(Dbn11)<8
    l=length(Dbn11);
    for i=1:l
        ta(9-i)=Dbn11(l+1-i);
    end
    Dbn11=ta;
    ta=zeros(1,8);
end
while length(Dbn12)<8
    l=length(Dbn12);
    for i=1:l
        ta(9-i)=Dbn12(l+1-i);
    end
    Dbn12=ta;
    ta=zeros(1,8);
end
while length(Dbn21)<8
    l=length(Dbn21);
    for i=1:l
        ta(9-i)=Dbn21(l+1-i);
    end
    Dbn21=ta;
    ta=zeros(1,8);
end
while length(Dbn22)<8
    l=length(Dbn22);
    for i=1:l
        ta(9-i)=Dbn22(l+1-i);
```

```
    end
    Dbn22=ta;
    ta=zeros(1,8);
end
while length(Ncn11)<8
    l=length(Ncn11);
    for i=1:l
        ta(9-i)=Ncn11(l+1-i);
    end
    Ncn11=ta;
    ta=zeros(1,8);
end
while length(Ncn12)<8
    l=length(Ncn12);
    for i=1:l
        ta(9-i)=Ncn12(l+1-i);
    end
    Ncn12=ta;
    ta=zeros(1,8);
end
while length(Ncn21)<8
    l=length(Ncn21);
    for i=1:l
        ta(9-i)=Ncn21(l+1-i);
    end
    Ncn21=ta;
    ta=zeros(1,8);
end
while length(Ncn22)<8
    l=length(Ncn22);
    for i=1:l
```

```
        ta(9-i)=Ncn22(l+1-i);
    end
    Ncn22=ta;
    ta=zeros(1,8);
end
while length(Dcn11)<8
    l=length(Dcn11);
    for i=1:l
        ta(9-i)=Dcn11(l+1-i);
    end
    Dcn11=ta;
    ta=zeros(1,8);
end
while length(Dcn12)<8
    l=length(Dcn12);
    for i=1:l
        ta(9-i)=Dcn12(l+1-i);
    end
    Dcn12=ta;
    ta=zeros(1,8);
end
while length(Dcn21)<8
    l=length(Dcn21);
    for i=1:l
        ta(9-i)=Dcn21(l+1-i);
    end
    Dcn21=ta;
    ta=zeros(1,8);
end
while length(Dcn22)<8
    l=length(Dcn22);
```

```
    for i=1:l
        ta(9-i)=Dcn22(l+1-i);
    end
    Dcn22=ta;
    ta=zeros(1,8);
end
while length(Ndn11)<8
    l=length(Ndn11);
    for i=1:l
        ta(9-i)=Ndn11(l+1-i);
    end
    Ndn11=ta;
    ta=zeros(1,8);
end
while length(Ndn12)<8
    l=length(Ndn12);
    for i=1:l
        ta(9-i)=Ndn12(l+1-i);
    end
    Ndn12=ta;
    ta=zeros(1,8);
end
while length(Ndn21)<8
    l=length(Ndn21);
    for i=1:l
        ta(9-i)=Ndn21(l+1-i);
    end
    Ndn21=ta;
    ta=zeros(1,8);
end
while length(Ndn22)<8
```

```
    l=length(Ndn22);
    for i=1:l
        ta(9-i)=Ndn22(l+1-i);
    end
    Ndn22=ta;
    ta=zeros(1,8);
end
while length(Ddn11)<8
    l=length(Ddn11);
    for i=1:l
        ta(9-i)=Ddn11(l+1-i);
    end
    Ddn11=ta;
    ta=zeros(1,8);
end
while length(Ddn12)<8
    l=length(Ddn12);
    for i=1:l
        ta(9-i)=Ddn12(l+1-i);
    end
    Ddn12=ta;
    ta=zeros(1,8);
end
while length(Ddn21)<8
    l=length(Ddn21);
    for i=1:l
        ta(9-i)=Ddn21(l+1-i);
    end
    Ddn21=ta;
    ta=zeros(1,8);
end
```

```
while length(Ddn22)<8
   l=length(Ddn22);
   for i=1:l
      ta(9-i)=Ddn22(l+1-i);
   end
   Ddn22=ta;
   ta=zeros(1,8);
end
%%% SVD for Transformated X-29
clc;clear;format long e;
% Plant a %
s=0;
pd11=Dan11(1)*s^7+Dan11(2)*s^6+Dan11(3)*s^5+Dan11(4)*s^4+Dan11(5)*s^3+Dan1
1(6)*s^2+Dan11(7)*s+Dan11(8);
pd12=Dan12(1)*s^7+Dan12(2)*s^6+Dan12(3)*s^5+Dan12(4)*s^4+Dan12(5)*s^3+Dan1
2(6)*s^2+Dan12(7)*s+Dan12(8);
pd21=Dan21(1)*s^7+Dan21(2)*s^6+Dan21(3)*s^5+Dan21(4)*s^4+Dan21(5)*s^3+Dan2
1(6)*s^2+Dan21(7)*s+Dan21(8);
pd22=Dan22(1)*s^7+Dan22(2)*s^6+Dan22(3)*s^5+Dan22(4)*s^4+Dan22(5)*s^3+Dan2
2(6)*s^2+Dan22(7)*s+Dan22(8);
pn11=Nan11(1)*s^7+Nan11(2)*s^6+Nan11(3)*s^5+Nan11(4)*s^4+Nan11(5)*s^3+Nan1
1(6)*s^2+Nan11(7)*s+Nan11(8);
pn12=Nan12(1)*s^7+Nan12(2)*s^6+Nan12(3)*s^5+Nan12(4)*s^4+Nan12(5)*s^3+Nan1
2(6)*s^2+Nan12(7)*s+Nan12(8);
pn21=Nan21(1)*s^7+Nan21(2)*s^6+Nan21(3)*s^5+Nan21(4)*s^4+Nan21(5)*s^3+Nan2
1(6)*s^2+Nan21(7)*s+Nan21(8);
pn22=Nan22(1)*s^7+Nan22(2)*s^6+Nan22(3)*s^5+Nan22(4)*s^4+Nan22(5)*s^3+Nan2
2(6)*s^2+Nan22(7)*s+Nan22(8);
plant1=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upa1,Spa1,Vpa1]=svd(plant1)
dot(Upa1(:,1),Vpa1(:,1))
```

```
dot(Upa1(:,2),Vpa1(:,2))
% Plant b %
s=0;
pd11=Dbn11(1)*s^7+Dbn11(2)*s^6+Dbn11(3)*s^5+Dbn11(4)*s^4+Dbn11(5)*s^3+Dbn
11(6)*s^2+Dbn11(7)*s+Dbn11(8);
pd12=Dbn12(1)*s^7+Dbn12(2)*s^6+Dbn12(3)*s^5+Dbn12(4)*s^4+Dbn12(5)*s^3+Dbn
12(6)*s^2+Dbn12(7)*s+Dbn12(8);
pd21=Dbn21(1)*s^7+Dbn21(2)*s^6+Dbn21(3)*s^5+Dbn21(4)*s^4+Dbn21(5)*s^3+Dbn
21(6)*s^2+Dbn21(7)*s+Dbn21(8);
pd22=Dbn22(1)*s^7+Dbn22(2)*s^6+Dbn22(3)*s^5+Dbn22(4)*s^4+Dbn22(5)*s^3+Dbn
22(6)*s^2+Dbn22(7)*s+Dbn22(8);
pn11=Nbn11(1)*s^7+Nbn11(2)*s^6+Nbn11(3)*s^5+Nbn11(4)*s^4+Nbn11(5)*s^3+Nbn
11(6)*s^2+Nbn11(7)*s+Nbn11(8);
pn12=Nbn12(1)*s^7+Nbn12(2)*s^6+Nbn12(3)*s^5+Nbn12(4)*s^4+Nbn12(5)*s^3+Nbn
12(6)*s^2+Nbn12(7)*s+Nbn12(8);
pn21=Nbn21(1)*s^7+Nbn21(2)*s^6+Nbn21(3)*s^5+Nbn21(4)*s^4+Nbn21(5)*s^3+Nbn
21(6)*s^2+Nbn21(7)*s+Nbn21(8);
pn22=Nbn22(1)*s^7+Nbn22(2)*s^6+Nbn22(3)*s^5+Nbn22(4)*s^4+Nbn22(5)*s^3+Nbn
22(6)*s^2+Nbn22(7)*s+Nbn22(8);
plant1=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upb1,Spb1,Vpb1]=svd(plant1)
dot(Upb1(:,1),Vpb1(:,1))
dot(Upb1(:,2),Vpb1(:,2))
s=1.077771575548320e-002;
pd11=Dbn11(1)*s^7+Dbn11(2)*s^6+Dbn11(3)*s^5+Dbn11(4)*s^4+Dbn11(5)*s^3+Dbn
11(6)*s^2+Dbn11(7)*s+Dbn11(8);
pd12=Dbn12(1)*s^7+Dbn12(2)*s^6+Dbn12(3)*s^5+Dbn12(4)*s^4+Dbn12(5)*s^3+Dbn
12(6)*s^2+Dbn12(7)*s+Dbn12(8);
pd21=Dbn21(1)*s^7+Dbn21(2)*s^6+Dbn21(3)*s^5+Dbn21(4)*s^4+Dbn21(5)*s^3+Dbn
21(6)*s^2+Dbn21(7)*s+Dbn21(8);
```

```
pd22=Dbn22(1)*s^7+Dbn22(2)*s^6+Dbn22(3)*s^5+Dbn22(4)*s^4+Dbn22(5)*s^3+Dbn
22(6)*s^2+Dbn22(7)*s+Dbn22(8);
pn11=Nbn11(1)*s^7+Nbn11(2)*s^6+Nbn11(3)*s^5+Nbn11(4)*s^4+Nbn11(5)*s^3+Nbn
11(6)*s^2+Nbn11(7)*s+Nbn11(8);
pn12=Nbn12(1)*s^7+Nbn12(2)*s^6+Nbn12(3)*s^5+Nbn12(4)*s^4+Nbn12(5)*s^3+Nbn
12(6)*s^2+Nbn12(7)*s+Nbn12(8);
pn21=Nbn21(1)*s^7+Nbn21(2)*s^6+Nbn21(3)*s^5+Nbn21(4)*s^4+Nbn21(5)*s^3+Nbn
21(6)*s^2+Nbn21(7)*s+Nbn21(8);
pn22=Nbn22(1)*s^7+Nbn22(2)*s^6+Nbn22(3)*s^5+Nbn22(4)*s^4+Nbn22(5)*s^3+Nbn
22(6)*s^2+Nbn22(7)*s+Nbn22(8);
plant2=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upb2,Spb2,Vpb2]=svd(plant2)
dot(Upb2(:,1),Vpb2(:,1))
dot(Upb2(:,2),Vpb2(:,2))
% Plant c %
s=0;
pd11=Dcn11(1)*s^7+Dcn11(2)*s^6+Dcn11(3)*s^5+Dcn11(4)*s^4+Dcn11(5)*s^3+Dcn1
1(6)*s^2+Dcn11(7)*s+Dcn11(8);
pd12=Dcn12(1)*s^7+Dcn12(2)*s^6+Dcn12(3)*s^5+Dcn12(4)*s^4+Dcn12(5)*s^3+Dcn1
2(6)*s^2+Dcn12(7)*s+Dcn12(8);
pd21=Dcn21(1)*s^7+Dcn21(2)*s^6+Dcn21(3)*s^5+Dcn21(4)*s^4+Dcn21(5)*s^3+Dcn2
1(6)*s^2+Dcn21(7)*s+Dcn21(8);
pd22=Dcn22(1)*s^7+Dcn22(2)*s^6+Dcn22(3)*s^5+Dcn22(4)*s^4+Dcn22(5)*s^3+Dcn2
2(6)*s^2+Dcn22(7)*s+Dcn22(8);
pn11=Ncn11(1)*s^7+Ncn11(2)*s^6+Ncn11(3)*s^5+Ncn11(4)*s^4+Ncn11(5)*s^3+Ncn1
1(6)*s^2+Ncn11(7)*s+Ncn11(8);
pn12=Ncn12(1)*s^7+Ncn12(2)*s^6+Ncn12(3)*s^5+Ncn12(4)*s^4+Ncn12(5)*s^3+Ncn1
2(6)*s^2+Ncn12(7)*s+Ncn12(8);
pn21=Ncn21(1)*s^7+Ncn21(2)*s^6+Ncn21(3)*s^5+Ncn21(4)*s^4+Ncn21(5)*s^3+Ncn2
1(6)*s^2+Ncn21(7)*s+Ncn21(8);
```

```
pn22=Ncn22(1)*s^7+Ncn22(2)*s^6+Ncn22(3)*s^5+Ncn22(4)*s^4+Ncn22(5)*s^3+Ncn2
2(6)*s^2+Ncn22(7)*s+Ncn22(8);
plant1=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upc1,Spc1,Vpc1]=svd(plant1)
dot(Upc1(:,1),Vpc1(:,1))
dot(Upc1(:,2),Vpc1(:,2))
s=6.885173072838656e-004;
pd11=Dcn11(1)*s^7+Dcn11(2)*s^6+Dcn11(3)*s^5+Dcn11(4)*s^4+Dcn11(5)*s^3+Dcn1
1(6)*s^2+Dcn11(7)*s+Dcn11(8);
pd12=Dcn12(1)*s^7+Dcn12(2)*s^6+Dcn12(3)*s^5+Dcn12(4)*s^4+Dcn12(5)*s^3+Dcn1
2(6)*s^2+Dcn12(7)*s+Dcn12(8);
pd21=Dcn21(1)*s^7+Dcn21(2)*s^6+Dcn21(3)*s^5+Dcn21(4)*s^4+Dcn21(5)*s^3+Dcn2
1(6)*s^2+Dcn21(7)*s+Dcn21(8);
pd22=Dcn22(1)*s^7+Dcn22(2)*s^6+Dcn22(3)*s^5+Dcn22(4)*s^4+Dcn22(5)*s^3+Dcn2
2(6)*s^2+Dcn22(7)*s+Dcn22(8);
pn11=Ncn11(1)*s^7+Ncn11(2)*s^6+Ncn11(3)*s^5+Ncn11(4)*s^4+Ncn11(5)*s^3+Ncn1
1(6)*s^2+Ncn11(7)*s+Ncn11(8);
pn12=Ncn12(1)*s^7+Ncn12(2)*s^6+Ncn12(3)*s^5+Ncn12(4)*s^4+Ncn12(5)*s^3+Ncn1
2(6)*s^2+Ncn12(7)*s+Ncn12(8);
pn21=Ncn21(1)*s^7+Ncn21(2)*s^6+Ncn21(3)*s^5+Ncn21(4)*s^4+Ncn21(5)*s^3+Ncn2
1(6)*s^2+Ncn21(7)*s+Ncn21(8);
pn22=Ncn22(1)*s^7+Ncn22(2)*s^6+Ncn22(3)*s^5+Ncn22(4)*s^4+Ncn22(5)*s^3+Ncn2
2(6)*s^2+Ncn22(7)*s+Ncn22(8);
plant2=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upc2,Spc2,Vpc2]=svd(plant2)
dot(Upc2(:,1),Vpc2(:,1))
dot(Upc2(:,2),Vpc2(:,2))
% Plant d %
s=0;
pd11=Ddn11(1)*s^7+Ddn11(2)*s^6+Ddn11(3)*s^5+Ddn11(4)*s^4+Ddn11(5)*s^3+Ddn
11(6)*s^2+Ddn11(7)*s+Ddn11(8);
```

```
pd12=Ddn12(1)*s^7+Ddn12(2)*s^6+Ddn12(3)*s^5+Ddn12(4)*s^4+Ddn12(5)*s^3+Ddn
12(6)*s^2+Ddn12(7)*s+Ddn12(8);
pd21=Ddn21(1)*s^7+Ddn21(2)*s^6+Ddn21(3)*s^5+Ddn21(4)*s^4+Ddn21(5)*s^3+Ddn
21(6)*s^2+Ddn21(7)*s+Ddn21(8);
pd22=Ddn22(1)*s^7+Ddn22(2)*s^6+Ddn22(3)*s^5+Ddn22(4)*s^4+Ddn22(5)*s^3+Ddn
22(6)*s^2+Ddn22(7)*s+Ddn22(8);
pn11=Ndn11(1)*s^7+Ndn11(2)*s^6+Ndn11(3)*s^5+Ndn11(4)*s^4+Ndn11(5)*s^3+Ndn
11(6)*s^2+Ndn11(7)*s+Ndn11(8);
pn12=Ndn12(1)*s^7+Ndn12(2)*s^6+Ndn12(3)*s^5+Ndn12(4)*s^4+Ndn12(5)*s^3+Ndn
12(6)*s^2+Ndn12(7)*s+Ndn12(8);
pn21=Ndn21(1)*s^7+Ndn21(2)*s^6+Ndn21(3)*s^5+Ndn21(4)*s^4+Ndn21(5)*s^3+Ndn
21(6)*s^2+Ndn21(7)*s+Ndn21(8);
pn22=Ndn22(1)*s^7+Ndn22(2)*s^6+Ndn22(3)*s^5+Ndn22(4)*s^4+Ndn22(5)*s^3+Ndn
22(6)*s^2+Ndn22(7)*s+Ndn22(8);
plant1=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upd1,Spd1,Vpd1]=svd(plant1)
dot(Upd1(:,1),Vpd1(:,1))
dot(Upd1(:,2),Vpd1(:,2))
s=1.167101885093847e-003;
pd11=Ddn11(1)*s^7+Ddn11(2)*s^6+Ddn11(3)*s^5+Ddn11(4)*s^4+Ddn11(5)*s^3+Ddn
11(6)*s^2+Ddn11(7)*s+Ddn11(8);
pd12=Ddn12(1)*s^7+Ddn12(2)*s^6+Ddn12(3)*s^5+Ddn12(4)*s^4+Ddn12(5)*s^3+Ddn
12(6)*s^2+Ddn12(7)*s+Ddn12(8);
pd21=Ddn21(1)*s^7+Ddn21(2)*s^6+Ddn21(3)*s^5+Ddn21(4)*s^4+Ddn21(5)*s^3+Ddn
21(6)*s^2+Ddn21(7)*s+Ddn21(8);
pd22=Ddn22(1)*s^7+Ddn22(2)*s^6+Ddn22(3)*s^5+Ddn22(4)*s^4+Ddn22(5)*s^3+Ddn
22(6)*s^2+Ddn22(7)*s+Ddn22(8);
pn11=Ndn11(1)*s^7+Ndn11(2)*s^6+Ndn11(3)*s^5+Ndn11(4)*s^4+Ndn11(5)*s^3+Ndn
11(6)*s^2+Ndn11(7)*s+Ndn11(8);
pn12=Ndn12(1)*s^7+Ndn12(2)*s^6+Ndn12(3)*s^5+Ndn12(4)*s^4+Ndn12(5)*s^3+Ndn
12(6)*s^2+Ndn12(7)*s+Ndn12(8);
```

```
pn21=Ndn21(1)*s^7+Ndn21(2)*s^6+Ndn21(3)*s^5+Ndn21(4)*s^4+Ndn21(5)*s^3+Ndn
21(6)*s^2+Ndn21(7)*s+Ndn21(8);
pn22=Ndn22(1)*s^7+Ndn22(2)*s^6+Ndn22(3)*s^5+Ndn22(4)*s^4+Ndn22(5)*s^3+Ndn
22(6)*s^2+Ndn22(7)*s+Ndn22(8);
plant2=[pn11/pd11 pn12/pd12;pn21/pd21 pn22/pd22];
[Upd2,Spd2,Vpd2]=svd(plant2)
dot(Upd2(:,1),Vpd2(:,1))
dot(Upd2(:,2),Vpd2(:,2))

nom=1;   %nominal plant
nump11(1,:)=Nan11;          nump12(1,:)=Nan12;
denp11(1,:)=Dan11;          denp12(1,:)=Dan12;
nump21(1,:)=Nan21;          nump22(1,:)=Nan22;
denp21(1,:)=Dan21;          denp22(1,:)=Dan22;

nump11(2,:)=Nbn11;          nump12(2,:)=Nbn12;
denp11(2,:)=Dbn11;          denp12(2,:)=Dbn12;
nump21(2,:)=Nbn21;          nump22(2,:)=Nbn22;
denp21(2,:)=Dbn21;          denp22(2,:)=Dbn22;

nump11(3,:)=Ncn11;          nump12(3,:)=Ncn12;
denp11(3,:)=Dcn11;          denp12(3,:)=Dcn12;
nump21(3,:)=Ncn21;          nump22(3,:)=Ncn22;
denp21(3,:)=Dcn21;          denp22(3,:)=Dcn22;

nump11(4,:)=Ndn11;          nump12(4,:)=Ndn12;
denp11(4,:)=Ddn11;          denp12(4,:)=Ddn12;
nump21(4,:)=Ndn21;          nump22(4,:)=Ndn22;
denp21(4,:)=Ddn21;          denp22(4,:)=Ddn22;

%generate inverse of plant
```

```
[nn1,dd1]=mulnd(nump11,denp11,nump22,denp22);
[nn2,dd2]=mulnd(nump12,denp12,nump21,denp21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numP11,denP11]=mulnd(-nump22,-denp22,dendet,numdet);
[numP22,denP22]=mulnd(-nump11,-denp11,dendet,numdet);
[numP12,denP12]=mulnd(nump12,-denp12,dendet,numdet);
[numP21,denP21]=mulnd(nump21,-denp21,dendet,numdet);
w  = [0.0001 0.01 0.1 1 2 3 5 10 20 30 50 70 100 150 200 1000 10000];  % for stability
bounds;
wl  = logspace(-3,10000,400);
m=10; %db
m_lin=10.^(m/20);
%************Compute Continuous bode of the
plant*****************************
p11=freqcp(nump11,denp11,w);
p12=freqcp(nump12,denp12,w);
p21=freqcp(nump21,denp21,w);
p22=freqcp(nump22,denp22,w);
P11=freqcp(numP11,denP11,w);
P12=freqcp(numP12,denP12,w);
P21=freqcp(numP21,denP21,w);
P22=freqcp(numP22,denP22,w);
tit1 = sprintf('Loop 1 Plant Template');
plottmpl(w,w,1./P11,nom);title(tit1);
tit2 = sprintf('Loop 2 Plant Template');
plottmpl(w,w,1./P22,nom);title(tit2);
%***********Calculating Bounds for Loop
#1**********************************
% Stability
bds_st=sisobnds(2,w,w,m_lin,1./P11,[],nom);
bds=grpbnds(bds_st);
```

```
bds=sectbnds(bds);                    %intersections
tit3 = sprintf('Loop 1 Stability Bounds');
plotbnds(bds);title(tit3);
%*****************Shaping                    for                    Loop
#1*****************************************
numg1=1;
deng1=1;
lpshape(wl,bds,denP11(nom,:),numP11(nom,:),[],numg1,deng1);

[numg1,deng1]=getqft('x29g1.shp');
g1=freqcp(numg1,deng1,w);
G1=zpk(tf(numg1,deng1));
figure;bode(G1);
%***********Calculating          Bounds          for          Loop
#2***********************************
% Stability
bds_st=sisobnds(2,w,w,m_lin,1./P22,[],nom);
bds=grpbnds(bds_st);
bds=sectbnds(bds);                    %intersections
tit4 = sprintf('Loop 2 Stability Bounds');
plotbnds(bds);title(tit4);
%*****************Shaping                    for                    Loop
#2*****************************************
numg2=1;
deng2=1;
lpshape(wl,bds,denP22(nom,:),numP22(nom,:),[],numg2,deng2);

[numg2,deng2]=getqft('x29g2.shp');
g2=freqcp(numg2,deng2,w);
G2=zpk(tf(numg2,deng2));
figure;bode(G2);
```

```
% Checking Transformated closed-loop Stability for
% Gn %
g11=numg1;
gd11=deng1;
g12=0;gd12=1;
g21=0;gd21=1;
g22=numg2;
gd22=deng2;
% Pan*Gn %
[tn1,td1]=mulnd(Nan11,Dan11,g11,gd11);
[tn2,td2]=mulnd(Nan12,Dan12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nan11,Dan11,g12,gd12);
[tn2,td2]=mulnd(Nan12,Dan12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nan21,Dan21,g11,gd11);
[tn2,td2]=mulnd(Nan22,Dan22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nan21,Dan21,g12,gd12);
[tn2,td2]=mulnd(Nan22,Dan22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
```

```
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PanGn %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
roots(numS11)
roots(denS11)
roots(numS22)
roots(denS22)
roots(numS12)
roots(denS12)
roots(numS21)
roots(denS21)
% Pbn*Gn %
[tn1,td1]=mulnd(Nbn11,Dbn11,g11,gd11);
```

```
[tn2,td2]=mulnd(Nbn12,Dbn12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nbn11,Dbn11,g12,gd12);
[tn2,td2]=mulnd(Nbn12,Dbn12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nbn21,Dbn21,g11,gd11);
[tn2,td2]=mulnd(Nbn22,Dbn22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nbn21,Dbn21,g12,gd12);
[tn2,td2]=mulnd(Nbn22,Dbn22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PbnGn %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
```

```
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
roots(numS11)
roots(denS11)
roots(numS22)
roots(denS22)
roots(numS12)
roots(denS12)
roots(numS21)
roots(denS21)
% Pcn*Gn %
[tn1,td1]=mulnd(Ncn11,Dcn11,g11,gd11);
[tn2,td2]=mulnd(Ncn12,Dcn12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ncn11,Dcn11,g12,gd12);
[tn2,td2]=mulnd(Ncn12,Dcn12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ncn21,Dcn21,g11,gd11);
[tn2,td2]=mulnd(Ncn22,Dcn22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ncn21,Dcn21,g12,gd12);
[tn2,td2]=mulnd(Ncn22,Dcn22,g22,gd22);
```

```
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PcnGn %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
```

```
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
roots(numS11)
roots(denS11)
roots(numS22)
roots(denS22)
roots(numS12)
roots(denS12)
roots(numS21)
roots(denS21)
% Pdn*Gn %
[tn1,td1]=mulnd(Ndn11,Ddn11,g11,gd11);
[tn2,td2]=mulnd(Ndn12,Ddn12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ndn11,Ddn11,g12,gd12);
[tn2,td2]=mulnd(Ndn12,Ddn12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ndn21,Ddn21,g11,gd11);
[tn2,td2]=mulnd(Ndn22,Ddn22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Ndn21,Ddn21,g12,gd12);
[tn2,td2]=mulnd(Ndn22,Ddn22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
```

```
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PdnGn %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
roots(numS11)
roots(denS11)
roots(numS22)
roots(denS22)
roots(numS12)
roots(denS12)
roots(numS21)
roots(denS21)
```

```
%%% Closed-loop stability verification for the original plant %%%
clc;clear;format long e;
% Plant a %
Na11=[1.172000000000000e-001 4.215693804999958e-001
6.546706291135286e+001 4.132461548651955e+000 -7.737107377179808e-004];
Na12=[3.481000000000001e-001 3.004621760300199e-001 -
7.031347733880426e+001 -4.290260300163865e+000 6.681760043581730e-004];
Na21=[5.577000000000014e-001 2.131840749733499e+000 1.351789739836793e-
001 -2.609024107869118e-015];
Na22=[-3.264000000000067e-001 -2.273223021845780e+000 -1.390504955224294e-
001 -4.996003610813204e-016];
Da11=[1.000000000000000e+000 5.590610000000000e+000 -
6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da12=[1.000000000000000e+000 5.590610000000000e+000 -
6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da21=[1.000000000000000e+000 5.590610000000000e+000 -
6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
Da22=[1.000000000000000e+000 5.590610000000000e+000 -
6.998678243751319e+001 -4.333352271727545e+000 -3.775288701841896e-001];
% N matrix %
n12=[-8.376009523223778e+002 -5.784774387580804e+003];
n22=[-1.432291144790173e+003 -5.381735341628663e+003];
n11=[2.068406613167494e-001 1.840636113498792e+000 7.205764157500426e+000 -
1.937881730487596e-001];
n21=[2.031426370332519e-001 1.685275398750232e+000 6.623270023616939e+000 -
1.924208023993863e-001];
nd=[1];
% Gn %
gnn11=[1.462782619748287e+023 1.131138796635021e+028
3.693058160206754e+030];
```

```
gnd11=[1.000000000000000e+000                                   1.200000011325681e+013
1.359072000029040e+018      3.473052600014921e+020      1.912730220185428e+022
1.042093560010359e+023    8.542799999639254e+022];
gn12=0;gn21=0;
gnn22=7.000000000000033e-005;
gnd22=[1.000000000000000e+000                                   1.500000000000006e+001
7.300000000000030e+001    1.290000000000007e+002    7.000000000000033e+001];
% G %  %G=N*Gn%
g11=conv(n11,gnn11);
g12=n12*gnn22;
g21=conv(n21,gnn11);
g22=n22*gnn22;
gd11=gnd11;
gd12=gnd22;
gd21=gnd11;
gd22=gnd22;
% Pa*G %
[tn1,td1]=mulnd(Na11,Da11,g11,gd11);
[tn2,td2]=mulnd(Na12,Da12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na11,Da11,g12,gd12);
[tn2,td2]=mulnd(Na12,Da12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g11,gd11);
[tn2,td2]=mulnd(Na22,Da22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Na21,Da21,g12,gd12);
[tn2,td2]=mulnd(Na22,Da22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
```

```
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PaG %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
Sa11=zpk(tf(numS11,denS11));
Sa11=minreal(Sa11)
```

```
Sa22=zpk(tf(numS22,denS22));
Sa22=minreal(Sa222)
% Plant b %
Nb11=[9.071999999999969e-003 1.932808712000278e-002
1.155041962224416e+000 4.755595748284797e-003 -2.796579577720032e-004];
Nb12=[7.101999999999997e-002 2.408502020000114e-002 -
1.421095010395007e+000 -4.109295365955556e-003 2.950852136856669e-004];
Nb21=[8.576000000000006e-002 8.475674789998777e-002 1.254879739693904e-
003 -3.747002708109903e-016];
Nb22=[-5.018999999999885e-002 -1.030980381000131e-001 -8.658115718451487e-
004 -2.914335439641036e-016];
Db11=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db12=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db21=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
Db22=[1.000000000000000e+000 1.518209999999999e+000 -
9.142737911599991e+000 -1.455733700726020e-001 -9.692664975499991e-002];
% Pb*G %
[tn1,td1]=mulnd(Nb11,Db11,g11,gd11);
[tn2,td2]=mulnd(Nb12,Db12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nb11,Db11,g12,gd12);
[tn2,td2]=mulnd(Nb12,Db12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nb21,Db21,g11,gd11);
[tn2,td2]=mulnd(Nb22,Db22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nb21,Db21,g12,gd12);
[tn2,td2]=mulnd(Nb22,Db22,g22,gd22);
```

```
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PbG %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
```

```
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
Sb11=zpk(tf(numS11,denS11));
Sb11=minreal(Sb11)
% Plant c %
Nc11=[2.100000000000002e-002 4.350065659999158e-002
4.225829411104126e+000 4.912148017935841e-002 -1.646391253174590e-004];
Nc12=[1.450000000000000e-001 5.297899920000204e-002 -
7.436124264278636e+000 -7.703155594005817e-002 2.414747452200994e-004];
Nc21=[1.589000000000143e-001 1.862845259999553e-001 2.834621212944311e-
003 0];
Nc22=[-1.179999999999932e-001 -3.251917000000226e-001 -4.029330462134440e-
003 0];
Dc11=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc12=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc21=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
Dc22=[1.000000000000000e+000 1.844530000000003e+000 -
2.582578434780001e+001 -3.460338977622594e-001 -9.024301691999996e-002];
% Pc*G %
[tn1,td1]=mulnd(Nc11,Dc11,g11,gd11);
[tn2,td2]=mulnd(Nc12,Dc12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nc11,Dc11,g12,gd12);
[tn2,td2]=mulnd(Nc12,Dc12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nc21,Dc21,g11,gd11);
[tn2,td2]=mulnd(Nc22,Dc22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nc21,Dc21,g12,gd12);
```

```
[tn2,td2]=mulnd(Nc22,Dc22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PcG %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
```

```
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
Sc11=zpk(tf(numS11,denS11));
Sc11=minreal(Sc11);
% Plant d %
Nd11=[4.422999999999955e-003 4.198247239997510e-003 6.009732055199013e-
001 5.601114275346930e-003 -1.388627209596149e-005];
Nd12=[4.373000000000005e-002 5.253591799998514e-003 -7.493085240193560e-
001 -5.172401463264253e-003 1.061499190466164e-005];
Nd21=[5.476000000000147e-002 2.306487227999554e-002 3.664198444066069e-
004 0];
Nd22=[-3.642999999999996e-002 -2.771510290000112e-002 -1.819212840562523e-
004 0];
Dd11=[1.000000000000000e+000 6.068300000000018e-001 -
7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd12=[1.000000000000000e+000 6.068300000000018e-001 -
7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd21=[1.000000000000000e+000 6.068300000000018e-001 -
7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
Dd22=[1.000000000000000e+000 6.068300000000018e-001 -
7.393810256099999e+000 -2.146239075044308e-001 -4.888496216588181e-002];
% Pd*G %
[tn1,td1]=mulnd(Nd11,Dd11,g11,gd11);
[tn2,td2]=mulnd(Nd12,Dd12,g21,gd21);
[num11,den11]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nd11,Dd11,g12,gd12);
[tn2,td2]=mulnd(Nd12,Dd12,g22,gd22);
[num12,den12]=addnd(tn1,td1,tn2,td2);
[tn1,td1]=mulnd(Nd21,Dd21,g11,gd11);
[tn2,td2]=mulnd(Nd22,Dd22,g21,gd21);
[num21,den21]=addnd(tn1,td1,tn2,td2);
```

```
[tn1,td1]=mulnd(Nd21,Dd21,g12,gd12);
[tn2,td2]=mulnd(Nd22,Dd22,g22,gd22);
[num22,den22]=addnd(tn1,td1,tn2,td2);
s1=zpk(tf(num11,den11));
s1=minreal(s1);
s1=tf(s1);
[num11,den11]=tfdata(s1,'v');
s2=zpk(tf(num12,den12));
s2=minreal(s2);
s2=tf(s2);
[num12,den12]=tfdata(s2,'v');
s3=zpk(tf(num21,den21));
s3=minreal(s3);
s3=tf(s3);
[num21,den21]=tfdata(s3,'v');
s4=zpk(tf(num22,den22));
s4=minreal(s4);
s4=tf(s4);
[num22,den22]=tfdata(s4,'v');
% I+PdG %
[inum11,iden11]=addnd(1,1,num11,den11);
inum12=num12;
iden12=den12;
inum21=num21;
iden21=den21;
[inum22,iden22]=addnd(1,1,num22,den22);
% Sencitivity TFM %
[nn1,dd1]=mulnd(inum11,iden11,inum22,iden22);
[nn2,dd2]=mulnd(inum12,iden12,inum21,iden21);
[numdet,dendet]=addnd(nn1,dd1,-nn2,dd2);
[numS11,denS11]=mulnd(inum22,iden22,dendet,numdet);
```

```
[numS22,denS22]=mulnd(inum11,iden11,dendet,numdet);
[numS12,denS12]=mulnd(-inum12,iden12,dendet,numdet);
[numS21,denS21]=mulnd(-inum21,iden21,dendet,numdet);
Sd11=zpk(tf(numS11,denS11));
Sd11=minreal(Sd11);
```

VITA

The author, Chen-yang Lan, was born in 1976 in Tainan, Taiwan. He graduated from National Tsing Hua University in Hsinchu, Taiwan with a Bachelor of Science degree in Mechanical Engineering in June 1998. He was ranked in the top 10 of his graduating class. Upon graduation, he served in the Army of Taiwan as a second lieutenant for two years. After fulfilling his duty in the army, he pursued a career as an engineer with the Taiwan Semiconductor Manufacturing Company in the lithographic department. In August 2001, he started his Master's degree at Texas A&M University in Mechanical Engineering. He is also a member of the American Society of Mechanical Engineers.

His permanent address is:

#12, Lane 530, Sect. 1, Chin-Hwa Rd.,

Tainan 70213, Taiwan.

His e-mail address is:

jimcylan@hotmail.com