İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ANALYSIS AND DESIGN OF FIXED ORDER STABILIZING CONTROLLERS FOR SISO AND TITO SYSTEMS: A COMPUTER ALGEBRA POINT OF VIEW

Ph.D. Thesis by
İlker ÜSTOĞLU

Department : Electrical Engineering

Programme : Control and Automation Engineering

JUNE 2009

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ANALYSIS AND DESIGN OF FIXED ORDER STABILIZING CONTROLLERS FOR SISO AND TITO SYSTEMS: A COMPUTER ALGEBRA POINT OF VIEW

Ph.D. Thesis by
İlker ÜSTOĞLU
(504032101)

Date of submission : 04 May 2009
Date of defence examination: 18 June 2009

Supervisor (Chairman) : Assoc. Prof. Dr. M. T. SÖYLEMEZ (ITU)
Members of the Examining Committee : Prof. Dr. Leyla GÖREN SÜMER (ITU)
Prof. Dr. M. K. Külmiz ÇEVİK (ITU)
Prof. Dr. Galip CANSEVER (YTU)
Assoc. Prof. Dr. Haluk GÖRGÜN (YTU)

JUNE 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SISO VE TITO SİSTEMLER İÇİN SABİT MERTEBELİ KONTROLÖRLERİN ANALİZ VE TASARIMI: BİR BİLGİSAYAR CEBRİ YAKLAŞIMI

DOKTORA TEZİ
İlker ÜSTOĞLU
(504032101)

Tezin Enstitüye Verildiği Tarih: 04 Mayıs 2009
Tezin Savunulduğu Tarih : 18 Haziran 2009

Tez Danışmanı : Doç. Dr. M. Turan SÖYLEMEZ (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Leyla GÖREN SÜMER(İTÜ)
Prof. Dr. M. K. Külmiz ÇEVİK (İTÜ)
Prof. Dr. Galip CANSEVER (YTÜ)
Doç. Dr. Haluk GÖRGÜN (YTÜ)

HAZİRAN 2009

FOREWORD

I would like to gratefully acknowledge the enthusiastic supervision of Assoc. Prof. Dr. M. Turan Söylemez during this work for his detailed and constructive comments, and for his important support throughout this work. His wide knowledge and his logical way of thinking have been of great value for me. Especially, throughout my thesis-writing period, he provided encouragement, good company, lots of good ideas and interesting problems to be solved in near future.

I would like to express my deep and sincere gratitude to Prof. Dr. Leyla Gören Sümer for her constructive criticism and excellent advice during the preparation of this thesis. She has been actively interested in my work and has always been available to advise me. Her ideals and concepts have had a remarkable influence on my entire career.

I also would like to thank the other member of my thesis committee; Prof. Dr. M. K. Külmiz Çevik not only for his excellent comments throughout this work, but also for his magic in the classroom first made linear system theory, nonlinear systems, and geometric control theory fascinating for me.

My sincere thanks are due to the official referees, for their detailed review, constructive criticism.

I am grateful to all the people from Department of Control Engineering, Istanbul Technical University, for being a family during the many years.

I want to thank my parents, my sisters and their families for their understanding, endless patience and encouragement when it was most required. The understanding and support from my beloved wife Kübra is a powerful source of inspiration and energy. Without her help and encouragement, this study would not have been completed.

Last but not least I want to express my deeply-felt thanks to Mehmet Haydaroğlu for his warm encouragement and guidance.

May 2009

İlker Üstoğlu

Electrical Engineer, M. Sc.

TABLE OF CONTENTS

Page

ABBREVIATIONS

BIBO **:** Bounded Input Bounded Output
MIMO **:** Multi Input Multi Output
SISO **:** Single Input Single Output
PID **:** Proportional, Integral, Derivative
TITO **:** Two Input Two Output

LIST OF TABLES

Page

LIST OF FIGURES

Page

ANALYSIS AND DESIGN OF FIXED ORDER STABILIZING CONTROLLERS FOR SISO AND TITO SYSTEMS: A COMPUTER ALGEBRA POINT OF VIEW

SUMMARY

This thesis presents a toolbox developed in the symbolic algebra environment of Mathematica for calculating the transfer function or the state-space description of a given system composed by several subsystems. The user can interactively describe the system interconnections either using a graphical user interface developed as a part of the toolbox, or by the help of a so-called interconnection matrix that defines the signal flow graph of the system. The toolbox can handle multi-input multi-output (MIMO) systems as well as single-input single-output (SISO) systems, and is capable of executing all calculations symbolically. The gain formula of Mason is used in all underlying calculations for SISO systems.

The potential of symbolic algebra for the design of control systems is illustrated through several examples. It has been shown in particular, that (a) the exact calculations provided by symbolic algebra can be used to utilize some of the direct (but numerically error prone) methods efficiently in control system design, and (b) symbolic manipulations by a computer can help control engineers at several stages of the design. The block diagram reduction, calculation of all stabilizing controllers, dominant pole assignment and robust pole assignment are taken as case studies.

This thesis also presents a method for PID controller design, which can achieve dominant pole assignment using two of the controller parameters. The non-dominant poles are restricted on the left of the line $s = \hat{\sigma}$, where $\hat{\sigma}$ is the minimum feasible value, called as the feasibility border. It is obvious that a dominant pole assignment is not practical if $\hat{\sigma}$ is close to the real parts of the required dominant poles. Hence, finding $\hat{\sigma}$ for a given system is very important. The method, which parameterizes all such controllers in order to allow further design criteria, can be applied to other kinds of low-order compensators.

In this thesis the characteristic values and characteristic value plots are examined, reducibility and irreducibility of characteristic equations are discussed, the real-axis crossings of the characteristic value plots and their relation to the stabilizing gain intervals is introduced, and the number of unstable closed-loop poles for gain intervals is considered. Furthermore, constant diagonal controllers of type $diag(k,k)$ are used to stabilize TITO systems, the problem is discussed for irreducible and reducible cases separately, for each case a fast and efficient algorithm is presented. Some tutorial examples are given to introduce how the proposed algorithms work. The problem in parameter space where the number of constant gains is two is presented, i.e., constant diagonal controllers of type $diag(k_1,k_2)$ are considered

SISO VE TITO SİSTEMLER İÇİN SABİT MERTEBELİ KONTROLÖRLERİN ANALİZ VE TASARIMI: BİR BİLGİSAYAR CEBRİ YAKLAŞIMI

ÖZET

Bu tezde sembolik cebir ortamlarından biri olan Mathematica'da geliştirilmiş bir araç kutusu sunulmaktadır. Bu araç ile çeşitli alt sistemlerden oluşan bir sistemin gerek transfer fonksiyonu gerekse durum uzayı temsili hesaplanabilmektedir. Kullanıcı sistem bağlantılarını, bu araç kutusu ile birlikte gelen grafiksel kullanıcı arayüzü ile oluşturabileceği gibi işaret akış diyagramından yola çıkılarak elde edilen bağlantı matrisi ile de verebilir. Geliştirilen bu araç kutusu tek-girişli tek-çıkışlı sistemlerle olduğu kadar çok-girişli çok-çıkışlı sistemlerle de uyumlu ve tüm hesaplamalarını aynı zamanda sembolik olarak yapacak şekilde tasarlanmıştır. Burada Mason kazanç formülü arka plandaki hesaplamalarda temel görevi üstlenmektedir.

Bu tezde sembolik cebirin kontrol sistem tasarımında kullanılabilirliği çeşitli örneklerle verilmiştir. Özellikle (a) sembolik cebirin sağladığı tam hesaplardan, nümerik hataya açık yöntemleri etkin kullanımda yararlanılabileceği (b) sembolik işlemlerin kontrol mühendislerine tasarımın çeşitli aşamalarında nasıl yardımcı olabileceği gösterilmiştir. Blok diyagramı indirgeme, tüm kararlı kontrolörlerin bulunması, baskın kutup atama ve dayanıklı kutup atama problemleri tezin bu kısmında ele alınmıştır.

Tez ayrıca PID kontrolör tasarımında baskın kutup ataması kontrolörün iki parametresi kullanılarak nasıl yapılabilir problemini ele alırken aynı zamanda geriye kalan üçüncü parametre ile baskın olmayan kutuplar en fazla ne kadar uzağa atanabilir sorusuna cevap vermektedir. Bu sınır $s = \hat{\sigma}$ doğrusu ile verilmek üzere $\hat{\sigma}$ baskın kutuplara yakın ise baskın kutup atama tekniğinin pratik olmayacağı çok açıktır. Bu nedenle $\hat{\sigma}$'nın bulunması özellikle önemlidir. Tez kapsamında verilen yöntem diğer düşük mertebeden kontrolörler içinde uygulanabilir.

Bu tez kapsamında ayrıca karakteristik değerler ve eğrileri, bunların indirgenebilir olup olmamaları, reel ekseni kesim noktaları ve bu noktalar civarındaki davranışları, ve bunların kararlı kılan kontrolör parametre aralıklarının hesaplanmasında kullanılması ele alınmıştır. Özellikle TITO sistemleri kararlı kılan $diag(k,k)$ tipinde kontrolörler düşünülmüş, indirgenebilir ve indirgenemez durumlar için tüm kararlı kılan kazançların bulunması yönünde hızlı algoritmalar geliştirilmiştir. Çeşitli açıklayıcı örnekler bu algoritmaların nasıl çalıştığını göstermek amacıyla verilmiştir. Parametre uzayında inceleme yaparak sistemi kararlı kılan tüm $diag(k_1,k_2)$ tipinde kontrolörler üzerinde durulmuştur.

1. INTRODUCTION

A control system is an interconnection of components to perform certain tasks and to generate desired output signal, when it is driven by the input signal. In contrast to an open-loop system, a closed-loop control system uses sensors to measure the actual output to adjust the input in order to achieve desired output. Most industrial control systems are no longer single-input and single-output (SISO) but multi-input and multi-output (MIMO) systems with a high coupling between the channels. In order to analyze and design a control system, it is advantageous if a mathematical representation of system dynamics is available. The system dynamics is usually governed by a set of differential equations. In the case of linear, time-invariant systems, these differential equations are linear ordinary differential equations, which is the case this thesis considers. Let $x(t) \in \mathbb{R}^n$ be the state vector, $u(t) \in \mathbb{R}^m$ the control (input) vector, and $y(t) \in \mathbb{R}^p$ the measurement (control) vector, a linear, time-invariant, continuous-time control system can be described by the following model,

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned} \qquad \textbf{(1.1)}$$

If we assume that the initial conditions of the state variables are all zero and use the Laplace transform, a transfer function matrix corresponding to the system in (1.1) can be derived as

$$G(s) = C(s\,I_n - A)^{-1}B + D \qquad \textbf{(1.2)}$$

and can be further denoted in a short form by

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]. \qquad \textbf{(1.3)}$$

A fundamental issue in control-systems design is the stability. When a dynamic system is described by its input-output relationship such as a transfer function (matrix), the system is stable if it generates bounded outputs for any bounded inputs.

This is called as the bounded-input bounded-output (BIBO) stability. For a linear, time-invariant system modeled by a transfer function matrix $G(s)$, the BIBO stability is guaranteed if and only if all the poles of $G(s)$ are in the open-left-half complex plane. When a system is described by a state-space model such as in (1.1), asymptotic stability can be defined. A system is asymptotically stable if, for an identically zero input, the system state will converge to zero (equilibrium point) from any initial states. A linear, time-invariant system described by a model of (1.1) is asymptotically stable, if, and only if all the eigenvalues of the state matrix A are in the open-left-half complex plane. The asymptotic stability of a system implies that the system is also BIBO stable, but not vice versa. However, for a system in (1.1), if (A, B, C, D) is a minimal realization, the BIBO stability of the system implies that the system is asymptotically stable. For an interconnected, feedback system, a stability concept called the internal stability arises. An interconnected system is internally stable if the subsystems of all input-output pairs are asymptotically stable. For the system given in Figure 1.1, there are two inputs r and d (the disturbance at the output), and two outputs y and u (the output of the controller K).

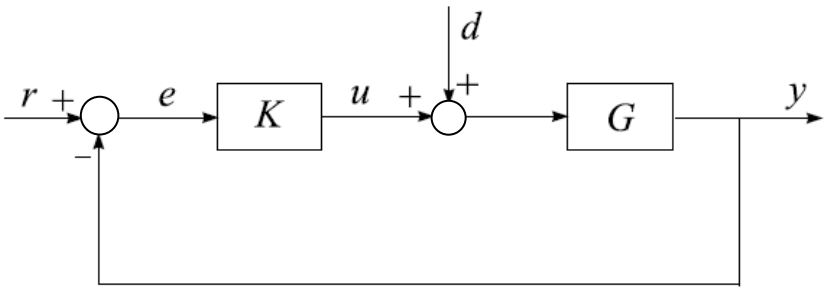

Figure 1.1 : A feedback control system.

The transfer functions from the inputs to the outputs, respectively, are

$$T_{yr} = GK(I + GK)^{-1} \quad \textbf{(1.4)}$$

$$T_{yd} = G(I + KG)^{-1} \quad \textbf{(1.5)}$$

$$T_{ur} = K(I + GK)^{-1} \quad \textbf{(1.6)}$$

$$T_{ud} = -KG(I + KG)^{-1} \quad \textbf{(1.7)}$$

Hence, the system is internally stable if and only if all the transfer functions in (1.4)–(1.7) are BIBO stable, or the transfer function matrix M from $\begin{pmatrix} r \\ d \end{pmatrix}$ to $\begin{pmatrix} y \\ u \end{pmatrix}$ is BIBO stable, where

$$M = \begin{bmatrix} GK(I+GK)^{-1} & G(I+KG)^{-1} \\ K(I+GK)^{-1} & -KG(I+KG)^{-1} \end{bmatrix}. \tag{1.8}$$

The stability of (1.8) is equivalent to the stability of

$$\tilde{M} = \begin{bmatrix} I-GK(I+GK)^{-1} & G(I+KG)^{-1} \\ K(I+GK)^{-1} & I-KG(I+KG)^{-1} \end{bmatrix}. \tag{1.9}$$

Simple matrix manipulations yield in

$$\tilde{M} = \begin{bmatrix} (I+GK)^{-1} & G(I+KG)^{-1} \\ K(I+GK)^{-1} & (I+KG)^{-1} \end{bmatrix} = \begin{bmatrix} I & -G \\ -K & I \end{bmatrix}^{-1} \tag{1.10}$$

Hence, the feedback system in Figure 1.1 is internally stable if, and only if, (1.10) is stable. If there is no unstable pole/zero cancellation between G and K, then any one of the four transfer functions being BIBO stable would be enough to guarantee that the whole system is internally stable [1].

Consider a system given in the form of (1.2) with (A,B,C,D) assumed to be minimal. Recall that H_∞ denotes the space of functions with no poles in the closed right-half complex plane. Matrices $(\hat{M}(s),\hat{N}(s)) \in H_\infty$, $((M(s),N(s)) \in H_\infty)$ constitute a left (right) coprime factorisation of $G(s)$ if, and only if,

(i) $\hat{M}(s)$ ($M(s)$) is square, and $\det(\hat{M}(s)) \neq 0$ ($\det(M(s)) \neq 0$),

(ii) the plant model is given by $G = \hat{M}^{-1} \cdot \hat{N}$ $(G = N \cdot M^{-1})$

(iii) There exists $(\hat{V},\hat{U}) \in H_\infty$ $((V,U) \in H_\infty)$ such that

$$\hat{M} \cdot \hat{V} + \hat{N} \cdot \hat{U} = I \tag{1.11}$$

$$(U \cdot N + V \cdot M = I) \tag{1.12}$$

The two equations in (iii), (1.11) and (1.12), are called Bezout identities [2] and are necessary and sufficient conditions for $(\hat{M}(s), \hat{N}(s))$ $((M(s), N(s)))$ being left coprime (right coprime), respectively. Transfer function matrices are coprime if they share no common zeros in the right-half complex plane, including at the infinity. The left and right coprime factorizations of $G(s)$ can be grouped together to form a Bezout double identity as the following

$$\begin{bmatrix} V & U \\ -\hat{N} & \hat{M} \end{bmatrix} \cdot \begin{bmatrix} M & -\hat{U} \\ N & \hat{V} \end{bmatrix} = I \tag{1.13}$$

For $G(s)$ of minimal realization, the formulae for the coprime factors can be derived as in the following theorem [3].

Theorem 1.1: Let constant matrices F and H be such that $A+BF$ and $A+HC$ are stable. Then the transfer function matrices $\hat{M}$ and $\hat{N}$ (M and N) defined below constitute a left (right) coprime factorization of $G(s)$,

$$\begin{bmatrix} \hat{N}(s) & \hat{M}(s) \end{bmatrix} = \left[\begin{array}{c|cc} A+HC & B+HD & -H \\ \hline C & D & I \end{array}\right] \tag{1.14}$$

$$\begin{bmatrix} N(s) \\ M(s) \end{bmatrix} = \left[\begin{array}{c|c} A+BF & B \\ \hline C+DF & D \\ F & I \end{array}\right] \tag{1.15}$$

Furthermore, $\hat{U}(s)$, $\hat{V}(s)$, $U(s)$, $V(s)$ satisfy the Bezout double identity (1.13),

$$\begin{bmatrix} \hat{U}(s) & \hat{V}(s) \end{bmatrix} = \left[\begin{array}{c|cc} A+HC & H & B+HD \\ \hline F & 0 & I \end{array}\right] \tag{1.16}$$

$$\begin{bmatrix} U(s) \\ V(s) \end{bmatrix} = \left[\begin{array}{c|c} A+BF & H \\ \hline F & 0 \\ C+DF & I \end{array}\right] \tag{1.17}$$

Recall that the pairs $(\hat{U},\hat{V})$ and (U,V) are stable and coprime. Using (1.10), it is straightforward to show the following lemma.

Lemma 1.2. $K = \hat{V}^{-1} \cdot \hat{U} = U \cdot V^{-1}$ is a stabilizing controller, i.e. the closed-loop system in Figure 1.1 is internally stable.

The set of all stabilizing controllers for G can be obtained in the following Youla Parameterization Theorem [4,5]:

Theorem 1.3: The set of all stabilizing controllers for G is

$$\{(\hat{V} + Q\hat{N})^{-1}(\hat{U} + Q\hat{M}):\ Q \in H_\infty\}. \tag{1.18}$$

The set can also be expressed as

$$\{(U + MQ)(UV + NQ)^{-1}:\ Q \in H_\infty\}. \tag{1.19}$$

Let us consider an *n*-dimensional square MIMO system (see Figure 1.2), i.e. having the same number of inputs and outputs. $G(s) = \{g_{ij}(s)\}$ denotes the square transfer function matrix of the open loop system of size $n \times n$ with entries $g_{ij}(s)$ (i, j = 1, 2, ..., *n*), which are scalar proper rational functions in complex variable s. The elements $g_{ii}(s)$ on the principal diagonal are the transfer functions of the separate channels, and the nondiagonal elements $g_{ij}(s)$ $(i \neq j)$ are the transfer functions of cross-connections from the j^{th} channel to the i^{th}.

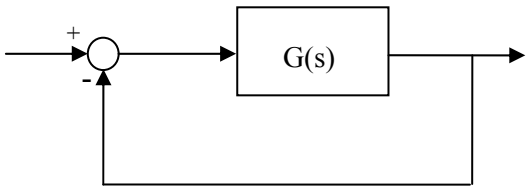

Figure 1.2 : Block diagram of a linear MIMO unit feedback system.

The output of the closed-loop system is related to the input by the following closed-loop system transfer function matrices

$$T(s) = (I + G(s))^{-1} G(s) = G(s)(I + G(s))^{-1} \tag{1.20}$$

where I denotes the identity matrix and $S(s) \triangleq (I + G(s))^{-1}$. The transfer function matrices $S(s)$ and $T(s)$ are usually called the sensitivity function matrix and

complementary sensitivity function matrix. Note that it is easy to check that $S(s)+T(s)=I$.

The notions of poles and zeros of linear MIMO systems were discussed in [6–8] form the necessary basis on which the frequency-domain multivariable control theory is built. The transfer function matrices $G(s)$, $T(s)$ and $S(s)$ may be regarded as some linear operators mapping an *n*-dimensional complex space $\mathbb{C}^n$ of the input vectors into the corresponding spaces of the output or error vectors. This suggests using mathematical tools of the theory of linear algebraic operators and functional analysis for the study of linear MIMO systems [9–12]. The roots of the equation

$$\det(\lambda I - G(s)) = 0 \qquad \textbf{(1.21)}$$

are functions of variable s. These complex functions $\lambda_i(s)$ are called characteristic transfer functions of the open-loop MIMO system [13].

If we assume that all $\lambda_i(s)$ (i = 1, 2, ... , *n*) are distinct, then the corresponding normalized eigenvectors $x_i(s)$ of G(s) are linearly independent, and constitute the basis of the *n*-dimensional complex space $\mathbb{C}^n$. We call this basis as the canonical basis of the open-loop MIMO system. Using the modal matrix $M(s)=\left[x_1(s) \quad x_2(s) \quad \cdots \quad x_n(s)\right]$, we can represent the matrix G(s) by the similarity transformation in the following form:

$$G(s) = M(s) \cdot diag(\lambda_i(s)) M^{-1}(s) \qquad \textbf{(1.22)}$$

where $diag(\lambda_i(s))$ denotes the diagonal matrix with the elements $\lambda_i(s)$ on the principal diagonal.

For the stability of the linear MIMO system in Figure 1.1, it is necessary and sufficient that the roots of the equation (1.21) lie in the open left half-plane of the complex plane [14].

Further, for simplicity, we shall call Equation (1.21) the characteristic equation of the closed-loop MIMO system. Using the canonical representation of the transfer function matrix $G(s)$ via similarity transformation (1.22), (1.21) may be reduced to the following form:

$$\det(M(s)\cdot diag(1+\lambda_i(s))M^{-1}(s))=0 \qquad \textbf{(1.23)}$$

which immediately results in

$$\prod_{i=1}^{n}(1+\lambda_i(s))=0\,. \qquad \textbf{(1.24)}$$

(1.23) shows that the characteristic equation of the *n*-dimensional closed-loop MIMO system splits into *n* corresponding equations of the one-dimensional characteristic systems. This means that the complex plane of the closed-loop MIMO system roots can be considered as superpositions of *n* complex planes of the closed-loop characteristic systems roots. For the stability of a linear MIMO system, it is necessary and sufficient that all closed-loop characteristic systems be stable. We can state that the described approach enables replacing the stability analysis of an *n*-channel linear MIMO system by the stability analysis of *n* SISO characteristic systems. Note that this approach reduces an *n*-dimensional task to *n* one-dimensional tasks. For the stability analysis of characteristic systems, any of the well known stability criteria used for common SISO systems can be applied. However, for practical applications, the most convenient is the Nyquist criterion, of which generalizations to the multivariable case are given, for example, in [14–16].

Define as the characteristic gain loci the Nyquist plots of the open-loop characteristic transfer functions $\lambda_i(s)$ (i = 1, 2, ..., *n*), that is the curves in the complex plane which correspond to $\lambda_i(jw)$ as angular frequency w changes from -∞ to +∞. Then, if the open-loop MIMO characteristic equation has *l* poles in the right half-plane, for the stability of the closed-loop system, it is necessary and sufficient that the total sum of anticlockwise encirclements of the critical point (−1, j0) by the characteristic gain loci $\lambda_i(jw)$ be equal to *l*.

As stated before stability is the most important property in the design of all dynamical systems. A reasonable approach to controller design is to find the set of all stabilizing compensators and then using a member of this set to satisfy further design criteria. As stated before a complete parameterization of all stabilizing controllers for a given system was suggested by Youla [4,5]. An important disadvantage of this parameterization is that the order of the controller cannot be fixed. As a result, the

order of the controller tends to be quite high most of the time. Therefore, in the last few years computation of all stabilizing controllers of a given order is examined by several researchers. In order to describe the borders of absolute and relative stability regions in the parameter space, Shafiei and Shenton have developed a graphical method [17,18]. Ho, Datta and Bhattacharrya have proposed a method based on the generalized Hermite-Biehler theorem, which is also applicable to characterize stabilizing PID controllers [19,20]. Furthermore, Söylemez, Munro and Baki have given a numerical frequency domain method in order to find the set of the so-called D-stabilizing low order controllers [21,22]. This method is based on the calculation of the real-axis intersections of the Nyquist plot. Ackermann and Kaesbauer have generalized these results to a larger class of problems [23]. Gryazina and Polyak have investigated the geometry of D-decomposition for polynomials and extended it for multi-input multi-output (MIMO) systems [24]. Furthermore, they have focused on the estimation of the number of all root-invariant regions, regions where the number of stable and unstable roots of the closed-loop characteristic polynomial remains unchanged. It is a common fact that it is more difficult to design controllers for MIMO systems because there are usually interactions between different control loops. To overcome this difficulty decentralized controllers are considered which have fewer tuning parameters compared to general multivariable controllers; for example for an $n-$input $n-$output system decentralized PID controllers have $3n$ tuning parameters where as the full matrix PID controller has $3n^2$ parameters [25]. Furthermore, decentralized PID controllers are widely used in process control due their simplicity and facility in working in case of actuator and/or sensor failure because it is relatively easy to tune manually as only one loop is directly affected by the failure [26]. Söylemez and Üstoğlu have provided a tutorial example on constant output feedback controller design for TITO systems and showed how symbolic algebra is used as an efficient tool to solve such complicated problems [27]. If a MIMO system described by a $n\times n$ transfer-function matrix $G(s)$ is diagonal dominant over the bandwidth of interest, or there exists an input compensator matrix $C(s)$ to achieve diagonal dominance, then the stability and time domain behavior of the system can be inferred from the diagonal elements of $G(s)C(s)$. This idea can be traced back to Rosenbrock where the single-input–single-output (SISO) frequency domain design techniques are applied to multi-loop systems satisfying diagonal

dominance properties [28,29]. The relative gain array, the (inverse–) Nyquist array approach, the block Nyquist array method, the Perron-Frobenius scaling procedure and the characteristic locus method are among the analysis and design methods to reduce the interaction in a multivariable system. However, these approaches do not provide the set of all stabilizing controllers. Generalizing the Nyquist stability criterion for MIMO case is particularly important because plotting the characteristic values of the open-loop transfer function enables us to check the stability of the closed-loop system for a gain parameter. Using the concepts of the algebraic function theory Barman and Katzenelson have developed a test to discuss the stability of the system [30]. MacFarlene and Postlethwaite, on the other hand, used Riemann surfaces to investigate the same problem [31]. Desoer and Wang used basic facts of analytic function theory to develop a stability test based on the eigenloci [15].

In this thesis the characteristic values and characteristic value plots are examined, reducibility and irreducibility of characteristic equations are discussed, the real-axis crossings of the characteristic value plots and their relation to the stabilizing gain intervals is introduced, and the number of unstable closed-loop poles for gain intervals is considered. Furthermore, constant diagonal controllers of type $diag(k,k)$ are used to stabilize TITO systems, the problem is discussed for irreducible and reducible cases separately, and algorithms are presented to solve this problem. Some tutorial examples are given to introduce how the proposed algorithms work. The problem in parameter space where the number of constant gains is two is presented, i.e., constant diagonal controllers of type $diag(k_1,k_2)$ are considered.

PID controllers by far are the most common controllers in use today [32]. Their success in industry is due to their simplicity, practicality and satisfactory performance. Naturally, there are numerous methods for designing PID controllers. Most of the methods in the literature are actually about tuning three parameters of the PID controller after doing a few experiments on the system to be controlled. Ziegler-Nichols [33] and Cohen-Coon tuning rules can be counted among the most famous such methods [32,34]. When the system to be controlled is not complex (first and second order) these rules usually offer a “good” set of parameters as a starting point for tuning the PID controller parameters. For high order systems, however, simple tuning rules do not always give satisfactory results and therefore more complicated design techniques are required to obtain acceptable results from PID controllers.

A large group of PID design methods given in the literature are based on calculation of a good set of parameters by optimizing for certain design criteria. Many of the robust PID control techniques proposed in the literature can be counted in this group [35–39]. Although some of the internal model control [40] based methods allow a design parameter to be tuned for better performance, they are only applicable to first or second order plants, and model order reduction is required for plants with higher dynamics [41,42].

A common problem with most of the design methods that offer a good set of parameters is that, when the real system does not perform as required there is usually no design freedom to help the control engineer. Hence, the practicing control engineer is usually left on his/her own to play with the three parameters (around the good set) using ad hoc methods to find a satisfactory controller. Therefore, a reasonable approach to controller design is to find the set of all stabilizing compensators and then using a member of this set to satisfy further design criteria. The so-called Youla parameterization provides an algebraic formulation of all stabilizing controllers for a given system and many of the modern control synthesis techniques are based on this method [43]. An important disadvantage of this parameterization is that the order of the controller cannot be fixed, and the order of the controllers found tends to be quite high. To this extent, in the last few years, researchers on PID controller design have focused on finding all stabilizing low-order (PID) controllers rather than finding a good set of parameters using parameter space approach [17,23,44], by calculating the real axis intersections of the Nyquist plot [22], and by generalizing the Hermite-Biehler theorem [45,46]. Such approaches are usually very practical and provide an insight into the PID control since they provide a large design freedom to use in implementation [47]. A serious drawback of these approaches is that the closed-loop system time domain performance characteristics are not taken into account. Therefore the control engineer may face with a large range of possible (stabilizing) controllers to select a suitable set of parameters.

For many systems, it is possible to change the time domain performance as required by placing the closed-loop system poles at desired locations [48]. Using PID controllers, analytical pole placement is possible when the order of the system to be controlled is one or two. For higher order systems up to three poles (the dominant

poles) can be located at desired positions to give the required closed loop performance [49,50]. However, if some of the remaining (unassigned) closed-loop system poles are at undesired locations a new set of dominant poles is chosen and a redesign is carried out.

It is almost always possible to assign two of the closed-loop system poles using two of the PID controller parameters (k_p, k_i). The remaining parameter (k_d) is left as a free parameter and is used to place the rest of the closed-loop poles (non-dominant) on the left of a given point σ in the left half s-plane.

It is a common fact that for the real parts of the roots of a polynomial $p(s)$ to be smaller than σ, the polynomial $p(s+\sigma)$ must be Hurwitz. Obviously, for a given plant transfer function and a pair of desired dominant poles, it may not be always possible to place all non-dominant poles on the left of a given $s=\sigma$ line. Such σ values are called unfeasible, and for these values there will be no stabilizing intervals of the free parameter (k_d). For a given system and a pair of desired poles finding the minimum feasible σ value $\hat{\sigma}$, called as the feasibility border [51], is very useful because if $\hat{\sigma}$ is near to or greater than the real parts of the required dominant poles the dominant pole assignment is not practical.

In this thesis, the main idea introduced in [51] is applied to dominant pole assignment problem using PID controllers.

It is very well known that using the right tool in the right place can save a lot of time and effort. This is certainly true when it comes to using symbolic algebra (also known as computer algebra) to solve some of the problems faced frequently in control systems community. It is an unfortunate fact that a great number of control system theoreticians and practitioners are unaware of these efficient tools, which can make their lives much easier. The use of symbolic algebra tools brings a new way of thinking in the solutions of many problems faced frequently in the control systems engineering [52,53]. A book by Munro [54] collected many interesting application areas of computer algebra to control system theory, and formed an important step towards taking attention of control engineers to this new way of thinking. As the awareness of control engineers on the power of computer algebra languages increased several symbolic toolboxes specifically for control system design such as the Control System Professional Suite [55], parametric uncertain systems toolbox

[56], and block diagram reduction toolbox [57] started to appear. Broadly speaking, it is possible to state that using computer algebra brings two main advantages among many others: (a) using exact calculations, and (b) manipulation of symbols.

This thesis also provides a tutorial overview on some of the application areas of computer algebra in control system design giving special focus on these two main advantages. For this purpose, the advantage of doing exact calculations by the help of a symbolic algebra language is presented. The particular example of state-feedback pole assignment problem for single input systems is considered as a case study. Then, the discussion is extended to illustrate the use of manipulation of symbols in several control engineering problems.

Manipulation of symbols can be used in almost any part of control engineering. An interesting example is finding closed-loop system transfer functions and characteristic polynomials for a given system topology. The advantage of using symbolic block diagram reduction algorithms [57] is also discussed in this thesis. It is shown in particular that it is possible to enter a system's topology using a graphical user interface (GUI) tool, and the transfer function and/or state space representation of overall system can be found in terms of subsystems.

A considerable research has recently focused on finding stabilizing gain intervals for single-input single-output systems [19,21,58]. Symbolic algebra can provide an easy to use alternative in such calculations. It has been shown through examples that finding all stabilizing gain intervals becomes a considerably easy task for both continuous and discrete time systems by the help of computer algebra.

Another example where symbolic manipulations prove to be useful can be given on finding practical low-order controllers that meet several design criteria for a given system. It is, for instance, possible to find out all dominant pole assignment PID controllers very easily by the help of symbolic algebra. Furthermore, the design freedom can be used such that the non-dominant poles and zeros of the closed-loop system are in places not affecting the transient response of the system. Two tutorial examples explaining this process are provided in within the thesis.

In some cases, symbolic algebra is used as an intermediary tool to obtain a simplified symbolic expression for a given formula, before starting an iterative numerical process to manipulate the formula again and again. Although the resulting symbolic

expression can be rather complicated (in some cases filling many pages) using the simplified symbolic expression instead of the original formula can save a lot of time in such cases. To this extend, the pole assignment problem for systems with parameter uncertainties is discussed as a case study [48,59,60].

2. EXACT AND SYMBOLIC MANIPULATIONS OF FORMULAE

2.1 Objectives

In this chapter, the potential of symbolic algebra for the design of control systems is illustrated through several examples. It has been shown in particular, that (i) the exact calculations provided by symbolic algebra can be used to utilize some of the direct (but numerically error prone) methods efficiently in control system design, and (ii) symbolic manipulations by a computer can help control engineers at several stages of the design. The block diagram reduction, calculation of all stabilizing controllers, dominant pole assignment and robust pole assignment are taken as case studies. This chapter has been published in Taylor and Francis, International Journal of Control [27]. Parts of the chapter have been presented at the 12th IEEE Mediterranean Conference on Control and Automation in Kuşadası, June 2004 [57].

2.2 Introduction

Symbolic algebra is the field of computer science and mathematics that is concerned with the development, and application of algorithms that analyze and manipulate mathematical expressions. Theoreticians and practitioners use computers as a indispensable experimental tool to obtain numerical and graphical solutions to problems that are too difficult or even impossible to solve by hand. There are now computer programs that find exact solutions to differential equations, integrate complicated functions, simplify algebraic expressions, and perform many other operations encountered in science, and engineering. In the last two decades, a large number of computer algebra systems have been developed such as Axiom, Derive, Macsyma, Maple, Mathematica, MuPAD, and Reduce. All of these packages include tools for exact symbolic computations. A sampling of these tools includes algebraic manipulation, solution of equations, trigonometry, arithmetic, polynomial operations, calculus, linear algebra, differential equations, and advanced algebra, such as group theory, and Galois groups. Even the so-called code generation is also possible; they can translate formulas to conventional programming languages, and to word

processing languages. It is an unfortunate fact that a great number of control system theoreticians and practitioners are still unaware of these efficient tools, which they could have used to solve some of the problems faced frequently in control systems community. The use of symbolic algebra tools brings a new way of thinking in the solutions of many problems faced frequently in the control systems engineering. As the awareness of control engineers on the power of computer algebra languages increased several symbolic toolboxes specifically for control system design such as the Control System Professional Suite [55], parametric uncertain systems toolbox, and block diagram reduction toolbox started to appear. We believe that, it is possible to state that using computer algebra brings two main advantages among many others: (i) using exact calculations, and (ii) manipulation of symbols. The aim of this chapter is to provide a tutorial overview on some of the application areas of computer algebra in control system design giving special focus on these two main advantages. For this purpose, the advantage of doing exact calculations by the help of a symbolic algebra language is shown in 2.3. The particular example of state-feedback pole assignment problem for single input systems is considered as a case study. Then, the discussion is extended to illustrate the use of manipulation of symbols in several control engineering problems in 2.4. An interesting example is finding closed-loop system transfer functions and characteristic polynomials for a given system topology. The advantage of using symbolic block diagram reduction algorithms is discussed in 2.4.1. It is shown in particular that it is possible to enter a system's topology using a graphical user interface (GUI) tool, and the transfer function and/or state space representation of overall system can be found in terms of subsystems.

Finding stabilizing gain intervals for single-input single-output systems is a popular topic on which a considerable research has recently been focused. It has been shown in 2.4.2 through examples that finding all stabilizing gain intervals becomes a considerably easy task for both continuous and discrete time systems by the help of computer algebra.

Another example is on finding out all dominant pole assignment PID controllers by the help of symbolic algebra. The design freedom can be used such that the non-dominant poles and zeros of the closed-loop system are in places not affecting the transient response of the system. Two tutorial examples explaining this process are provided in 2.4.3.

The pole assignment problem for systems with parameter uncertainties is discussed as a case study in 2.4.4.

2.3 Exact Calculations

Many control problems require numerical awareness in their solution as discussed in detail in the February 2004 issue of IEEE Control Systems Magazine. Solving an ill-posed problem using a finite precision machine usually requires great skill and experience. Complicated numerical algorithms are usually deployed in the solution of such problems. Symbolic algebra can be an easy to use alternative for some of these problems.

Many symbolic algebra tools allow the representation of rational numbers exactly both to eliminate the errors caused by floating point number truncations and to allow exact manipulation of data. When all the numbers are integers or rational numbers in an expression then usually all the internal algorithms keep the numbers in exact format to provide an exact solution [52]. In the following, two examples are provided to demonstrate this fact: One of these examples is on simple matrix manipulations, and the other is on the solution of the pole assignment problem using Ackermann's formula.

2.3.1 Example

Consider an ill-conditioned matrix A as follows,

$$A = \begin{bmatrix} -9.9999 & -10 \\ 10 & 10.0001 \end{bmatrix} \qquad \textbf{(2.25)}$$

Let us assume that we would like to calculate $I = A^{-2}A^2$ for which we know the result should be the identity matrix. Using machine precision numbers, a numerical calculation of I will always involve numerical errors. The order of such errors would depend on the algorithms used in the calculation of the inverse of the A matrix. (2.2) and (2.3) demonstrate the errors that occur in calculations using two well-known programming environments, namely Mathematica™ and Matlab™, respectively.

As can be seen from the results, even such simple calculations may result in considerably big numerical errors unless precautions are taken.

$$A^{-2}A^{2} = \begin{bmatrix} 0.761545 & -0.238457 \\ 0.238519 & 1.23852 \end{bmatrix} \tag{2.26}$$

$$A^{-2}A^{2} = \begin{bmatrix} 0.9669 & -0.0331 \\ 0.0331 & 1.0331 \end{bmatrix} \tag{2.27}$$

The advantage of using symbolic algebra becomes apparent, when exact (or rational) numbers are used instead of machine precision (or floating point) numbers. By using Mathematica's *Rationalize* command the elements of the *A* matrix are converted to rational numbers before doing the required calculation. This time, the inverse matrix is calculated symbolically (exactly), and therefore the result of the calculation is obtained without any error given as follows

$$A = \begin{bmatrix} -\frac{99999}{10000} & -10 \\ 10 & \frac{100001}{10000} \end{bmatrix} \tag{2.28}$$

$$A^{-2}A^{2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.29}$$

Actually ill-conditioned matrices appear frequently in control system analysis and design. A well-known example for ill-conditioned matrices is the controllability matrix [48]. For a system given as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.30}$$

where *u*, *y* and *x* represents the inputs, outputs and states, respectively, the controllability matrix is defined as

$$\Phi = \begin{bmatrix} B & AB & A^{2}B & \cdots & A^{n-1}B \end{bmatrix} \tag{2.31}$$

where *n* is the order of the system. As *n* gets bigger, it is known that the controllability matrix becomes ill-conditioned generically (for almost all *A* and *B*

matrices) and therefore taking the numerical inverse of this matrix involves errors [48,61]. Therefore methods that employ the inverse of the controllability matrix are susceptible to numerical errors. The well-known formula of Ackermann for pole assignment is one such method. Ackermann's formula states that for a controllable single-input system as defined in (2.6), and a target characteristic polynomial $p_c(s)$ such that

$$p_c(s) = \prod_{i=1}^{n} (s - p_i) \tag{2.32}$$

where p_i are desired closed-loop system poles, the unique constant state feedback controller that places the closed-loop system poles at the desired locations is given as

$$K = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \Phi^{-1} p_c(A) \tag{2.33}$$

where $p_c(A)$ is a matrix polynomial having same coefficients with $p_c(s)$. It can be shown that when the order of the system is more than 10 the numerical errors involved in the calculation of the state-feedback matrix are getting beyond acceptable values [61]. Actually there exist numerous methods that address the numerical problems arising in pole assignment problem [52,62]. It should be remarked that when all the calculations are to be executed exactly (symbolically) the required state-feedback controller (K) can be found without any numerical errors using any pole assignment method, including the Ackermann's formula. The following example explains this fact.

2.3.2 Example

$$A=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 8 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}, \quad B=\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{bmatrix} \tag{2.34}$$

Consider the system given in (2.10), and assume that the closed-loop system poles are required to be at P = {-1, -2, -3, -4, -5, -6, -7, -8, -9, -10}. Using Matlab's *acker* command to calculate the required state-feedback matrix results in a warning message stating that the poles are more than 10% in error. Actually, this can be observed by calculating the closed-loop system poles; $-13.4442 \pm j3.7684$, $-7.1594 \pm j5.5798$, $-3.6025 \pm j3.3678$, $-2.1016 \pm j1.4672$, $-1.1860 \pm j0.0913$. Similar results can be found using different numerical programming environments. Therefore using numerical calculations to apply Ackermann's formula to this example is not feasible. Nevertheless, if exact (symbolic) calculations are carried out instead of numerical calculations, the required state-feedback compensator can be found as

$$K=\left(39916800,-39449025,\frac{1547257250}{81},-\frac{20451132625}{3456},\right.$$
$$\left.\frac{229799994347}{180000},-\frac{3514468111}{18000},-\frac{710110159}{34300},-\frac{5616941}{3920},\frac{7381}{126},0\right) \tag{2.35}$$

Note that since this is an exact result the closed-loop system poles are precisely at the required locations when this state-feedback matrix is used. This single example, we believe, changes the way we look at many control problems. Using symbolic algebra platforms properly it is possible to use some well-known (numerically error prone) methods directly to obtain exact solutions to some of the control problems.

It should be remarked that pole assignment is not the only control system design method where numerical problems may arise. Actually, in many control related problems symbolic algebra can be used to relive numerical problems. For instance, it has been reported that numerical problems in calculations of controllability and observability grammians by [63]. A symbolic algebra approach to this problem can be very efficient as reported by [64].

2.4 Manipulation of Symbols

A symbolic algebra environment allows the user to use symbols in calculations, and therefore, derive generic solutions to given problems. Such generic solutions can be very useful in an area like control systems theory where complicated mathematical formulas are used extensively. Using simplification tools provided by computer algebra languages, the control theoretician can find simple rules and formulas for a complex looking problem. It is our personal experience that leaving a few variables as symbols in equations, and then continuing calculations can illuminate the insight of the problem at hand (possible linearities, simplifications, root of difficulties, etc.) and sparkle new ideas in one's mind.

Actually, there are numerous application areas in control engineering where symbolic calculations make the life easier for the control engineer. In the following we shall illustrate a few of these areas.

2.4.1 Block Diagram Reduction

An important step in the analysis and design of control systems is the derivation of a mathematical model that represents the real system in the form of a transfer function or state-space description. Frequently, the system is composed of subsystems interconnected in a rather complicated way. Usually, the sub-systems and their interconnections are shown by the help of a block diagram (or a signal flow graph), and it is required to calculate a transfer function between two points of the block diagram so as to find the required system transfer function. For relatively small block diagrams, the system transfer function can be calculated directly solving a set of linear equations. For more complicated block diagrams, well-known techniques of block diagram reduction or Mason’s gain formula [65,66], which is a standard part of the curriculum of most of the control and systems related undergraduate programs

around the world, can be used. Performing such calculations by hand, however, is error prone, especially if symbolic calculations are to be carried out, and may cause significant time losses, especially after considering the fact that block diagram reduction is usually the first step in design and therefore an error in this step would nullify the rest of the design. Symbolic algebra can provide a safe and fast alternative to hand calculations in such computations and allows the engineer to focus on the real problem i.e., analysis and/or design. Using the block diagram reduction toolbox, [57,67], for example, it is possible to draw a block diagram of a given system using a graphical user interface developed in .NET framework [68] (see Figure 2.1) and then automatically calculate transfer function between any two points of the system.

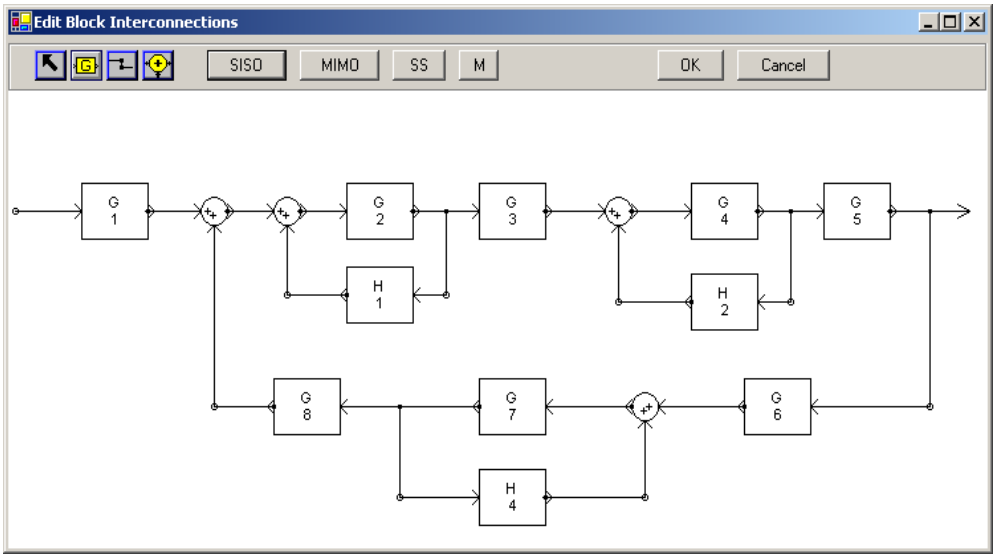

Figure 2.1 : The graphical user interface to draw block diagrams. The example is produced from the signal flow graph given Example 5.7 in [69].

Methods for symbolic transfer function evaluation before 1970 were surveyed in [70]. There are several computer-aided procedures capable of numerical and/or symbolic analysis of interconnected systems such as parameters extraction method [71], tree enumeration method [72], summing and branching matrices method [73], fast Fourier transform method [74], numerical interpolation method [75], signal flow graph method [76], algebraic formulation method [77].

Parameter extraction and numerical interpolation methods are suitable if many of the branches in the signal flow graph are characterized by numerical values, i.e. a small number of graph parameters are used as symbols. Hence these two methods cannot be considered as completely symbolic. The Fast Fourier transform method proposed

by Lee [74] allows only the numerical calculation of the coefficients of the transfer function for a given system. Algebraic formulation method of Mielke [76] is computationally competitive to the parameter extraction method and also provides a numerical solution to the problem. Summing and branching matrices method [73], where two matrices are used to describe the topology of a signal flow graph, is an alternative to Mason's formula. Signal flow graph and tree enumeration methods are topological methods that allow derivation of transfer functions from a given signal flow graph. These two methods are suitable when all the parameters are given symbolically. Although these methods are purely symbolic the computing time increases exponentially with the increase in the number of nodes and branches.

As a result the importance of an efficient algorithm becomes significant. In this section, a toolbox written in Mathematica [78] to handle block diagram reduction operations is presented. Particularly, an algorithm that uses Mason's formula directly, and is applicable to both numerical and symbolic problems is given. It is shown that it is possible to determine all *feed-forward paths*, *loops,* and *nontouching loops* in a system for a given list of input and output signals (nodes). For the description of the topological structure of a signal flow graph only one matrix, called the interconnection matrix is used. An important property of the toolbox is that it allows manipulation of multi-input multi-output (MIMO) systems. An algorithm to calculate the transfer function in a MIMO signal flow graph is also given in this chapter. Related with this algorithm, it has also been shown that the state space description of the overall system can be calculated using the state-space descriptions of the subsystems. Note that it is possible to provide the mathematical description of each subsystem explicitly, or implicitly (as a symbol representing the subsystem). In the examples provided throughout the section, however, only implicit representation of subsystems is assumed, as the solution for the explicit representations can be found easily using a combination of Simplify[] and ReplaceAll[] commands in Mathematica.

The organization of the rest of the section is as follows: after providing some basic definitions and terminology in 2.4.1.1, a graphical user interface (GUI) developed using the *.NET/Link* toolkit that comes with *Mathematica* is presented in 2.4.1.2. It is possible to capture the topological structure of the overall system using this GUI, which provides a *SIMULINK* like environment to enter the block diagram

representation of the system. 2.4.1.3 deals with the derivation of overall system transfer functions with the help of the gain formula of Mason. An algorithm to find transfer functions in MIMO case is given in 2.4.1.4, and 2.4.1.5 presents a discussion on how to calculate state-space descriptions of overall systems.

2.4.1.1 Definitions and Terminology

A signal flow graph consists of *nodes* and oriented *branches*. The value of any node variable (n_i) is the sum of all incoming signals, which is then transmitted on all outgoing branches. A signal travels along branches only in the direction specified by the arrow of the branch and the multiplication factor of a branch connected between two nodes is the transfer function of the corresponding plant. An *input node* is a node, which has one and only one outgoing branch. An *output node* is one, which has one and only one incoming branch. Note that any node can be converted to an input or output node by adding pseudo nodes to the graph. A *path* consists of connected branches in the direction of branch arrows. A *forward path* is a path from an input node to an output node that does not cross any nodes more than once. If no node is crossed more than once, the path is said to be open. The path is closed, if it ends at the same node where it began and does not cross any other node more than once. A *loop* is a closed path, which starts at a node and ends at the same node. Loops are nontouching if they do not have common nodes or branches. The *loop gain* is the product of all branch gains in the loop.

The system *interconnection matrix* (M) is defined as a matrix where $(i,j)^{th}$ element (M_{ij}) presents the gain (transfer function) between the i^{th} and j^{th} nodes of a signal flow graph that represents the system interconnections. Note that the following equation holds:

$$n = Mn \tag{2.36}$$

where n is a vector formed by node variables (n_i).

2.4.1.2 Graphical User Interface

The toolbox provides a graphical user interface (GUI) to allow the user to describe the block diagram of the system efficiently (see Figure 2.1 or Figure 2.2). It is possible to draw blocks, summation points and connectors with the help of this GUI to describe the system interconnections. Each block can be labelled separately to

describe a different subsystem. Subsystems with the same label are considered to be identical systems in the following calculations. A subsystem is considered to be a single-input single-output dynamic (order>0) system unless otherwise stated by the user (using interactive menus and input boxes). Note that in calculation of state-space descriptions of overall systems knowing which components are dynamic and which components are gains is important.

Optionally, the detailed description of each subsystem can also be provided to allow system specific calculations.

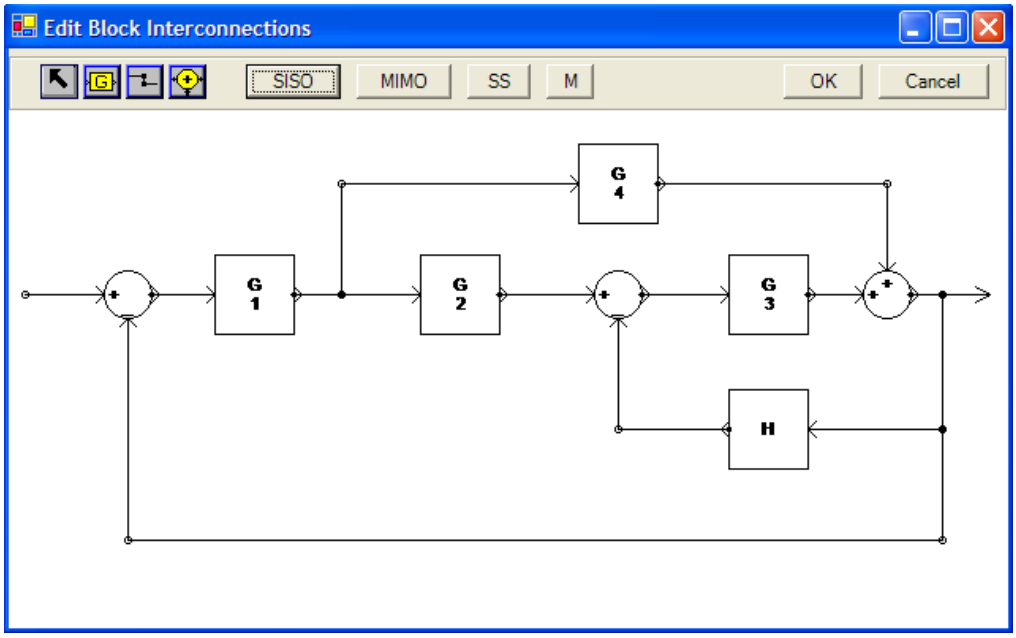

Figure 2.2 : Graphical User Interface.

The GUI can be used to derive the system interconnection matrix, to determine the inputs and outputs of the given system, and to calculate the transfer function or the state space description of the overall system. The command EditBlocks[] opens the graphical user interface, where the *select button* is used in order to select any object on the GUI, that can be deleted, copied, rotated of fliped after right clicking the mouse just above the object. Furthermore, it provides access to the properties menu of the selected object. The *blocks button* is used in order to add a block into the system, adding a node or summing object is done by clicking the *sum button*. The connectors, i.e. the branches in the signal flow graph, or the paths joining two objects in the GUI are added by clicking the *connector button*. Right clicking the connector object provides to define whether the node is an input or an output node (see Figure 2.3) .

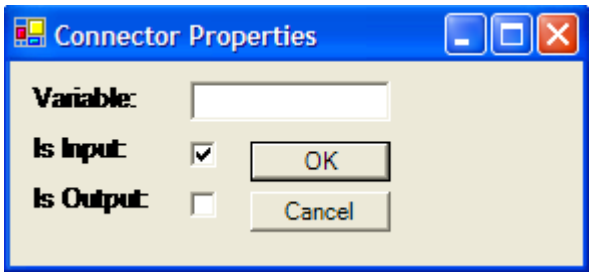

Figure 2.3 : Connector Properties Interface.

Once the design of the block diagram representation of the systems is finished the user can press the *M button* to check the interconnection matrix, input-output variables and their indices. If the *SISO button* is pressed the interconnection matrix, forward path gains, determinant of the graph, cofactors and the transfer function between the input node and the output node are displayed. The system interconnection matrix for the system given in Figure 2.2, for example, is calculated by the GUI as given in (2.13).

$$M = \begin{bmatrix} 0 & G_1 & 0 & 0 \\ 0 & 0 & G_2 & G_4 \\ 0 & 0 & 0 & G_3 \\ -1 & 0 & -H & 0 \end{bmatrix} \qquad \textbf{(2.37)}$$

Note that the signal flow graph that corresponds to this interconnection matrix is given in Figure 2.4.

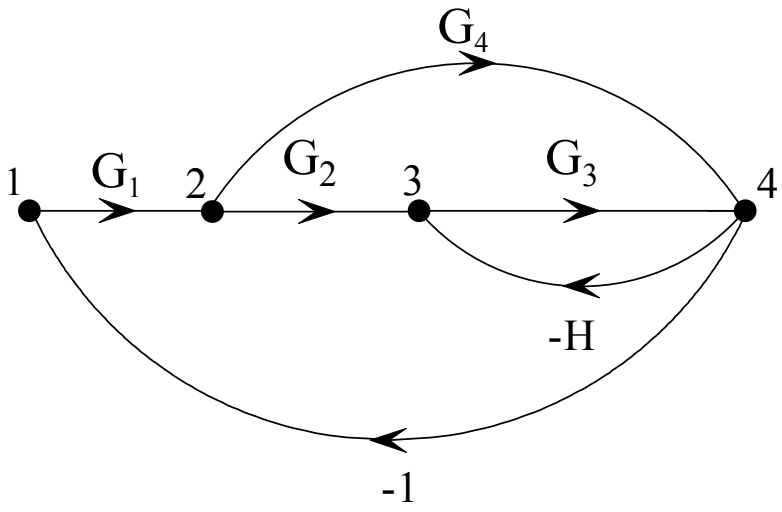

Figure 2.4 : The signal flow graph corresponding to the system interconnection matrix given in (2.13).

2.4.1.3 Calculating transfer functions

The relationship between an input variable and an output variable of a signal flow graph can be determined by Mason's formula, which is given by

$$P = \Delta^{-1} \sum_k P_k \Delta_k \qquad \textbf{(2.38)}$$

where P_k is the path gain of k^{th} forward path, Δ is the determinant of the graph defined as; Δ = 1 - (sum of loop gains of all individual loops) + (sum of gain products of all possible combinations of two nontouching loops) - (sum of gain products of all possible combinations of three nontouching loops) + …

$$\Delta = 1 - \sum_m P_{m1} + \sum_m P_{m2} - \sum_m P_{m3} + \ldots \qquad \textbf{(2.39)}$$

in which P_{mr} is the gain product of m^{th} possible combination of r nontouching loops, and Δ_k is the cofactor of k^{th} forward path given by the determinant of the graph with the loops touching the k^{th} forward path removed.

The heart of the toolbox consists of a collection of modules to compute the forward paths, loops, nontouching loops, determinant of the graph, and the gain (transfer function) from an input node to an output node for a given topological structure defined by an interconnection matrix. A diagram that shows the names and interconnections of these modules can be seen in Figure 2.5.

Here, the P_k module finds all possible forward paths using a recursive search algorithm (see Figure 2.6). In finding the forward loops, the algorithm starts from a given input node and checks whether branches from this node to other nodes exist by the help of the interconnection matrix.

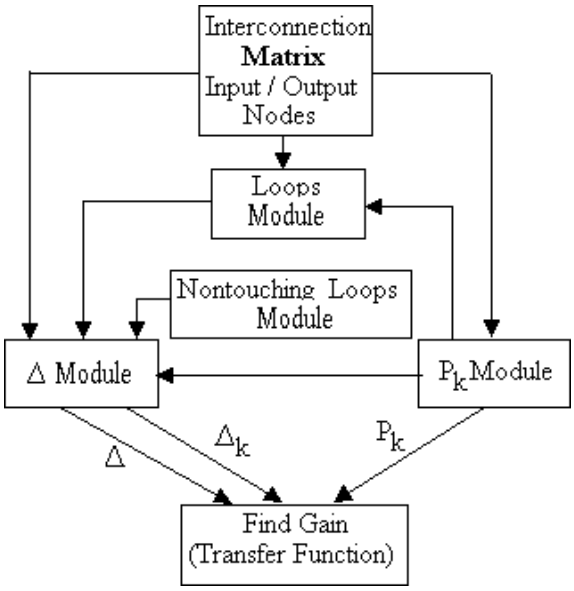

Figure 2.5 : The core modules and their interconnections.

When a branch is detected, a recursive call is made to find the forward paths from the adjacent node to the output node. A list of visited nodes is kept to guarantee that each node is passed only once. After finding the forward paths the corresponding path gains are also calculated using the gains defined in the interconnection matrix. For example, the P_k module returns the following list of paths for the interconnection matrix in (1), if the input and output nodes are selected as 1 and 4, respectively:

$$P = \{\{1,2,3,4\},\{1,2,4\}\} \tag{2.40}$$

Note that a loop can be seen as a forward path from a node to itself. Therefore, *loops* module uses P_k to find all possible loops in a given subgraph. Here a subgraph is obtained by removing some of the nodes from the system. The list of the nodes to be removed from the system is given as an argument to this module. Given two sets of loop lists, *nontouching loops* module determines all possible combinations r nontouching loops. The powerful set operators of Mathematica (Intersection and Union) are used in these calculations. The Δ module finds the determinant of a given subgraph using (2.15). The closed-loop system transfer function for the system shown in Figure 2.15 is calculated as follows by the FindGain module of the toolbox:

$$G_C = \frac{G_1G_2G_3 + G_1G_4}{1 + HG_3 + G_1G_2G_3 + G_1G_4} \tag{2.41}$$

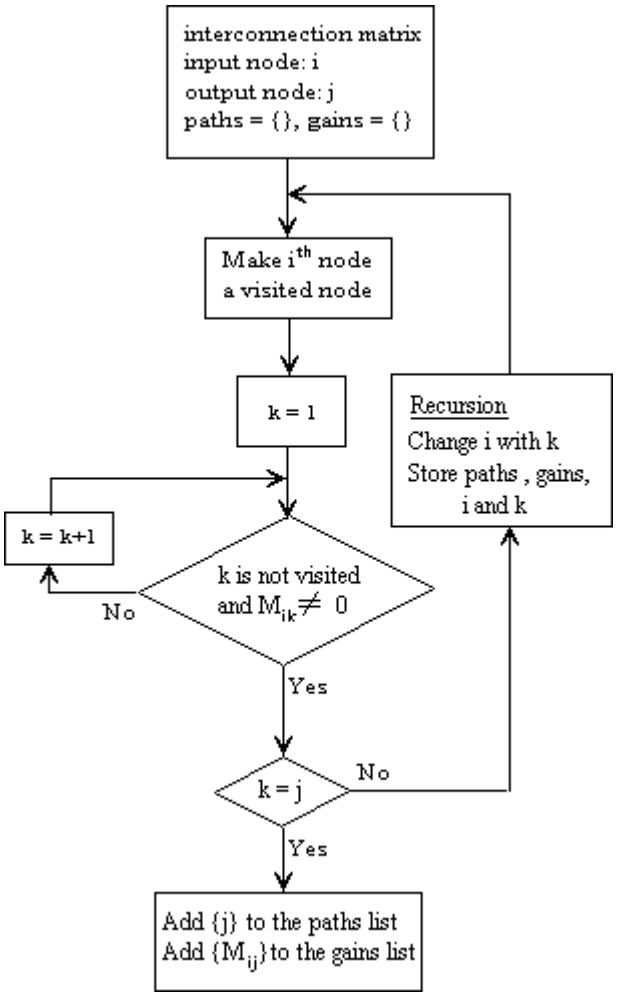

Figure 2.6 : The algorithm to find the forward paths (P_k).

The interconnection matrix for the system given in Figure 2.1 is as follows,

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_1 \\ 0 & 0 & 0 & 0 & G_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_2 & 0 \\ 0 & 0 & H_4 & G_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_2 & 0 \\ 1 & 0 & 0 & 0 & 0 & G_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & H_2 & 0 & G_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & H_1 & 0 & 0 & G_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_2 & 0 \end{bmatrix} \quad \textbf{(2.42)}$$

All forward paths and loops between the input and output are calculated using the interconnection matrix. The closed-loop system transfer function is then computed by the help of Mason's rule (for SISO systems);

$$G_c = \frac{G_1 G_2 G_3 G_4 G_5 (G_7 H_4 - 1)}{-(G_4 H_2 - 1)(G_7 H_4 - 1) + G_2 (G_3 G_4 G_5 G_6 G_7 G_8 + H_1 (G_4 H_2 - 1)(G_7 H_4 - 1))} \quad \textbf{(2.43)}$$

It should be noted that the resulting closed-loop system transfer function is found in a completely symbolic form in terms of the transfer functions of the subsystems, which is very convenient for finding transfer functions of given system topologies. When the transfer functions of the subsystems are given, it is always possible to compute the closed-loop system transfer function explicitly using standard commands of symbolic algebra. Obviously, it is also possible to obtain semi-explicit forms leaving some (or parts) of the transfer functions in symbolic form. This is particularly helpful in model matching and pole assignment problems (more on pole assignment is in the following sections).

2.4.1.4 MIMO Systems

Mason's gain formula cannot be applied to multi-input multi-output systems due to the fact that matrix algebra is a non-commutative algebra. To find the transfer function between two nodes of a MIMO signal flow graph equation (2.12) can be used. This equation can be interpreted as follows: The j^{th} line of the interconnection matrix M expresses a matrix equation that relates all node variables to the jth node variable. Hence, using these linear matrix equations the transfer function between a given input node (i) and an output node (j) can always be found, provided that there exists a forward path from node i to node j.

After checking that there exists a forward path, FindMIMOGain module defined in the toolbox, therefore, uses the recursive algorithm shown in Figure 2.7 to find the required transfer function. Here, J is used to denote the identity matrix (I is not used due to the fact that I is a predefined symbol in Mathematica). Note that since Mathematica's support for matrix expressions is limited, the equations mentioned in the algorithm are expressed as lists and handled specially.

Note also that the algorithm presented in Figure 2.7 works only if the input node (i) has no incoming branches associated with it. This means that the i^{th} column of the

interconnection matrix has to be zero. If this is not the case, the toolbox automatically expands the signal flow graph (interconnection matrix) to satisfy this assumption.

For example, the expanded (MIMO) system interconnection matrix for the system given in Figure 2.2 can be expressed as follows:

$$M = \begin{bmatrix} 0 & G_1 & 0 & 0 & 0 \\ 0 & 0 & G_2 & G_4 & 0 \\ 0 & 0 & 0 & G_3 & 0 \\ -J & 0 & -H & 0 & 0 \\ J & 0 & 0 & 0 & 0 \end{bmatrix} \quad \textbf{(2.44)}$$

The corresponding closed-loop system transfer function (from node 5 to node 4) is then obtained by the previously mentioned algorithm as given in (2.21)

$$G_C = (J + G_3H + G_4G_1 + G_3G_2G_1)^{-1}(G_3G_2G_1 + G_4G_1) \quad \textbf{(2.45)}$$

We note that the algorithm provided in Figure 2.7 does not necessarily produce the simplest algebraic form of the required transfer function. In order to allow matrix expression simplifications, the toolbox gives support to NCAlgebra Suite [79] that can be found on the Internet. If the setup process for NCAlgebra Suite is skipped MIMO and SS functions may not work correctly. In this respect the $NCDir$ variable in SetNCDir.m has to be changed suitably. For example, it is possible to show that (2.21) is equivalent to (2.22) with the help of NCSimplfyRational[] command in this suite.

$$G_C = J - (J + G_3H + G_4G_1 + G_3G_2G_1)^{-1}(G_3H + J) \quad \textbf{(2.46)}$$

2.4.1.5 Using State-Space Descriptions

A unique aspect of the toolbox is that it allows manipulation of state-space descriptions. In particular, it is possible to find the state-space description and the transfer function of the closed-loop system in terms of the state-space descriptions of subsystems. The state space matrices (A, B, C, and D) for each subsystem can be given as symbols or explicit expressions. In these calculations, some of the subsystems can be designated as pure gain matrices (A=0, B=0 and C=0) or as systems with no direct coupling (D=0).

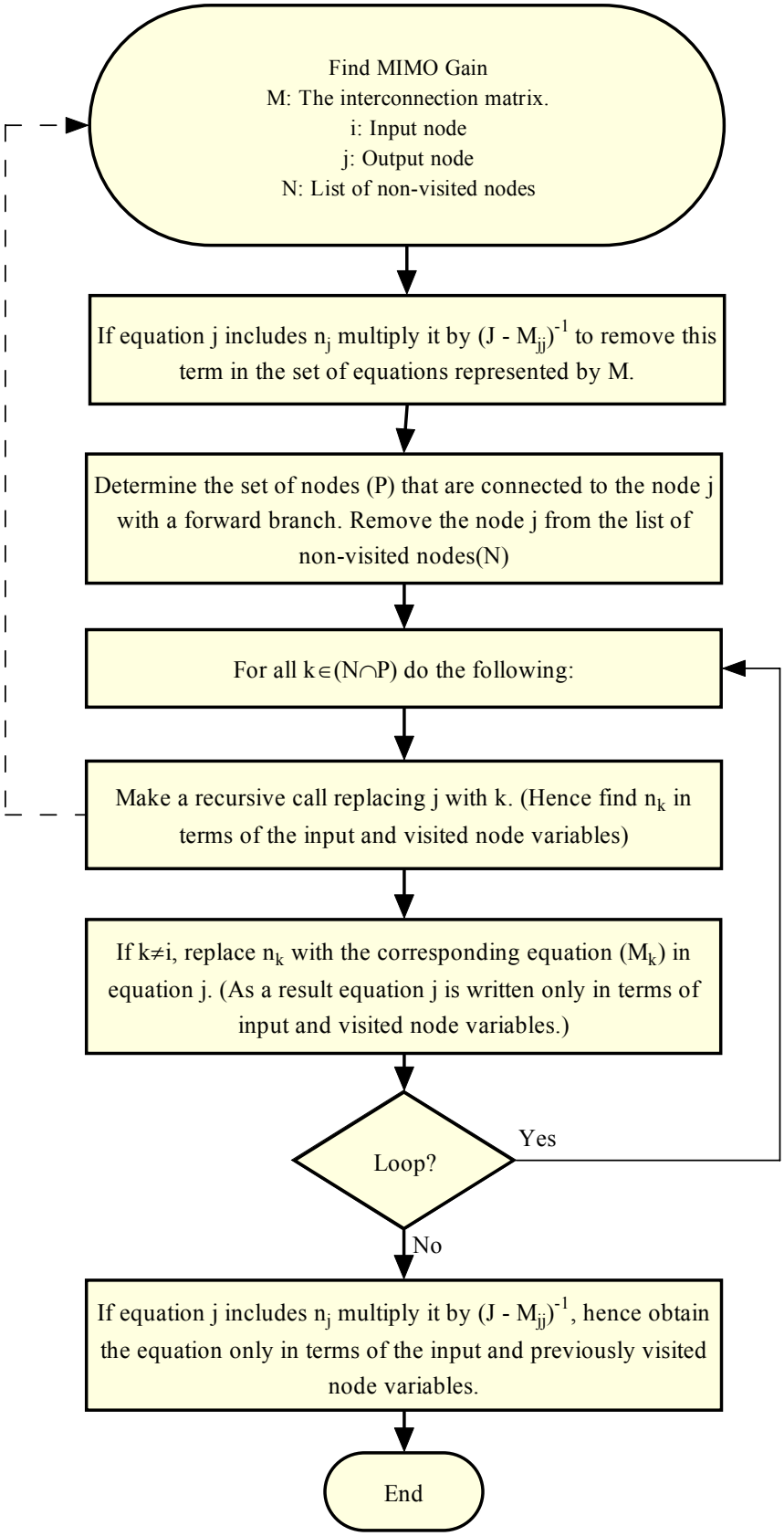

Figure 2.7 : Algorithm used by FindMIMOGain.

In order to use state-space descriptions, the interconnection matrix is expanded such that two nodes (say nk and nm) are added for each dynamic subsystem described in state-space (A, B, C, and D) connecting the nodes ni and nj. These two additional nodes are used to represent the state variables (with node nm) and their derivatives (with node nk) for the subsystem. The following four branches are then added to the signal flow graph (interconnection matrix):

$$\begin{aligned} M_{mk} &= A \\ M_{ik} &= B \\ M_{mj} &= C \\ M_{ij} &= D \end{aligned} \qquad \textbf{(2.47)}$$

A fifth branch $M_{km} = s^{-1}J$ should also be added to the newly expanded interconnection matrix when the closed-loop system transfer function is to be calculated. Once the expanded system interconnection matrix is obtained the solution for the closed-loop system transfer function or the state-space description (A_c, B_c, C_c and D_c) can be obtained using the algorithm defined in the previous section. In calculation of the state-space description of the closed-loop system, the fact that the closed-loop system A matrix (A_c) is the gain between all the state-variable nodes (n_m) and all the nodes that represent their derivatives (n_k) is used. Similarly it is useful to remind the fact that B_c is the gain between the input node and all n_k, C_c is the gain between all n_m and the output node, and D_c is the gain between the input node and the output node.

The expanded signal flow graph for the system described in Figure 2.2 can be seen in Figure 2.8. Here, only the subsystems G_2 and G_3 are assumed to be dynamic systems described by state space matrices (A_2, B_2, C_2, and D_2) and (A_3, B_3, C_3, and D_3=0), respectively. Note that all these assumptions can be entered using the graphical user interface described in 2.4.1.2. The toolbox keeps such information using an internal representation of the overall system. The node numbers of state variable vectors (n_m), and their derivatives (n_k) are also stored in this internal representation to automate the calculation of the closed-loop system state-space descriptions.

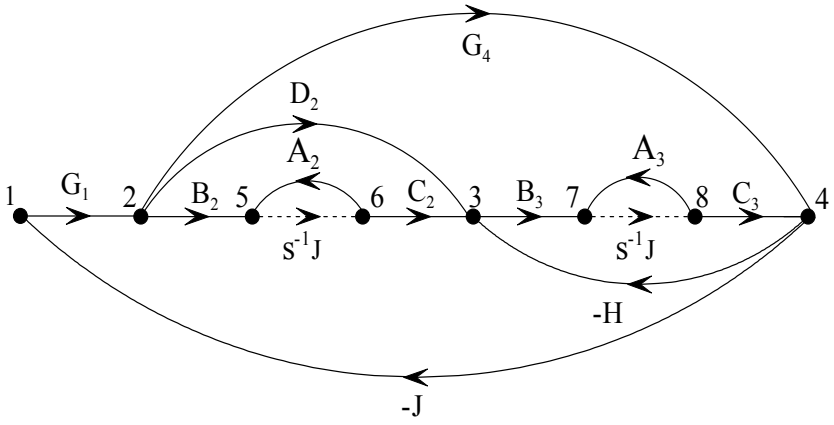

Figure 2.8 : Expanded signal flow graph for the system described in (2.13).

The corresponding system interconnection matrix is given in (2.24).

$$M = \begin{bmatrix} 0 & G_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_2 & G_4 & B_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & B_3 & 0 \\ -J & 0 & -H & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_2 & 0 & A_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_3 & 0 & 0 & A_3 & 0 \end{bmatrix} \quad \textbf{(2.48)}$$

The closed-loop system state-space matrices (A_c, B_c, C_c, and D_c) for the system described in Figure 2.2 are found by the toolbox as follows:

$$A_C = \begin{bmatrix} A_2 & -B_2.G_1.(J+G_4G_1)^{-1}.C_3 \\ B_3.C_2 & -B_3.H.(J+G_4G_1)^{-1}.C_3 - B_3D_2G_1.(J+G_4G_1)^{-1}.C_3 + A_3 \end{bmatrix} \quad \textbf{(2.49)}$$

$$B_C = \begin{bmatrix} B_2.G_1.(J+G_4.G_1)^{-1} \\ -B_3.H + B_3.H.(J+G_4.G_1)^{-1} + B_3.D_2.G_1.(J+G_4.G_1)^{-1} \end{bmatrix} \quad \textbf{(2.50)}$$

$$C_C = \begin{bmatrix} 0 & (J+G_4.G_1)^{-1}.C_3 \end{bmatrix} \quad \textbf{(2.51)}$$

$$D_C = J - (J+G_4.G_1)^{-1} \quad \textbf{(2.52)}$$

An interesting application area of the block diagram reduction toolbox is the calculation of the closed-loop system characteristic polynomial in terms of the numerators and denominators of the transfer functions of the subsystems. It should

be remarked that the characteristic polynomial found in this way depends only on the topology of the closed-loop system (and not the specific transfer functions of the subsystems). A. D. Lewis reported that finding the characteristic polynomial of a given signal flow graph in the most general form (in terms of numerators and denominators of subsystems) is an open problem in systems and control engineering [80,81]. We believe that block diagram reduction toolbox can help us better understand and possibly answer this interesting problem.

When some or all of the signals in a block diagram are vectors, then care must be taken to calculate the resulting transfer function in matrix format. Mason's gain formula does not simply work for such multi-input multi-output (MIMO) systems and therefore a symbolic solution of matrix equations has to be found. This is handled automatically by the block diagram reduction toolbox [57]. It should also be noted that since Mathematica does not support non-commutative (matrix) algebra directly, manipulations of the resulting matrix formulations is not easy [82]. Nevertheless, it is possible (although not necessary) to use the NCAlgebra Suite [79], [83], if simplifications are to be made on the resulting formulations. A simple example is given in Figure 2.9 and Figure 2.10 to illustrate the use of block diagram reduction toolbox with MIMO subsystems.

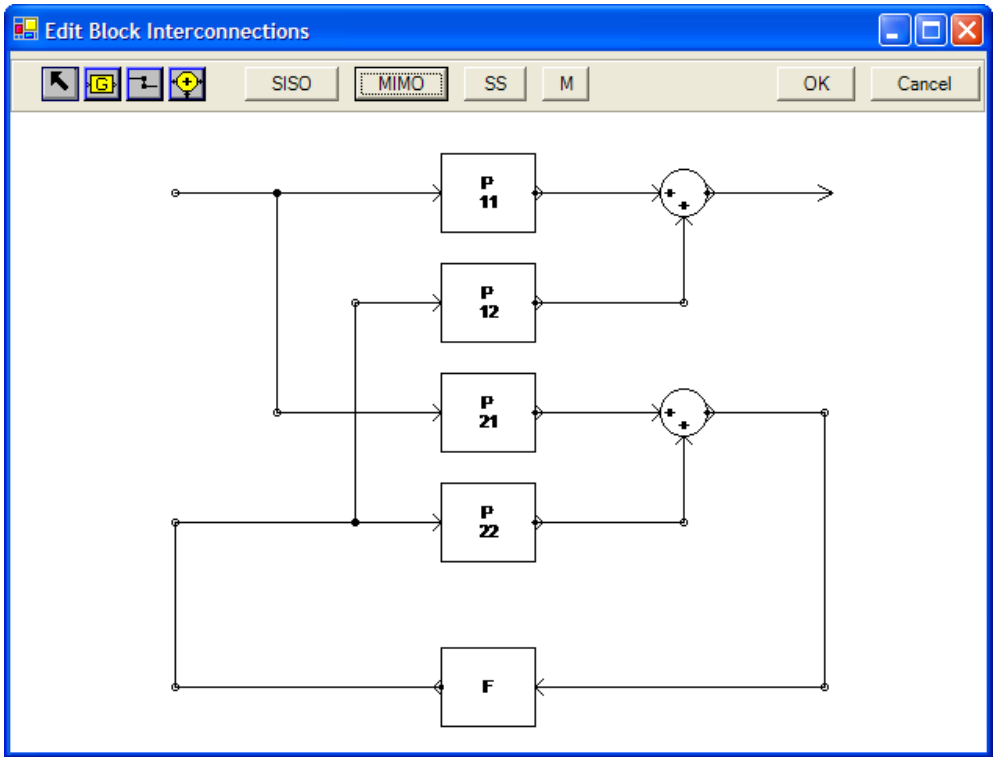

Figure 2.9 : Lower linear fractional transformation (LFT).

Finding the state-space equations of a system described by a block diagram is not a trivial task and requires great skill and attention. Even a very simple looking block diagram may require considerable effort and time, if state space equations are to be found (see Figure 2.11). It is not uncommon that graduate students come to their supervisor with a brilliant looking theorem only to find out that the state space form of the closed-loop system was calculated incorrectly at the very beginning of their calculations. Symbolic algebra can also help researchers to reduce the chance of errors in such calculations. The block diagram reduction toolbox allows users to enter the state space matrices (A, B, C, and D) in symbolic form or explicitly for each subsystem (see Figure 2.12). Note that it is possible to designate some of the subsystems as pure gain matrices (A=0, B=0, and C=0) or as systems with no direct coupling (D=0).

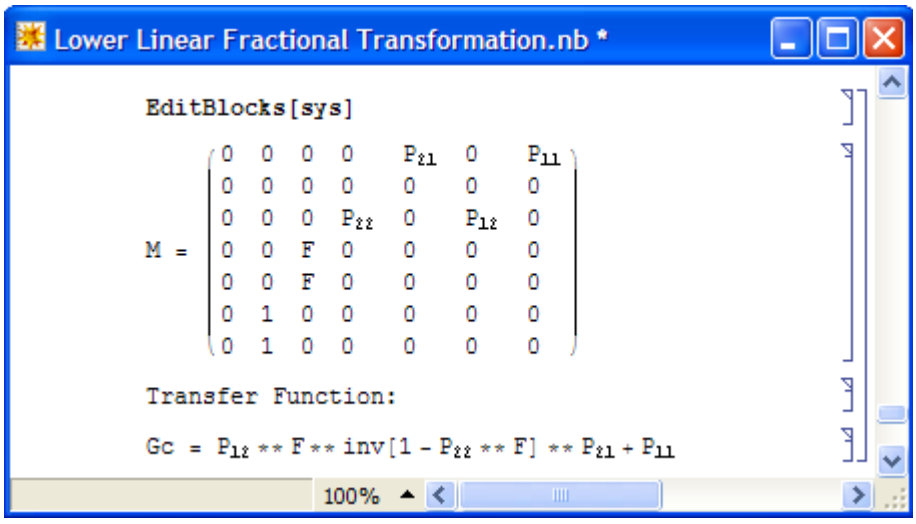

Figure 2.10 : Finding transfer function of a system with MIMO blocks.

Note also that NCAlgebra Suite uses ** operator to denote matrix multiplication.

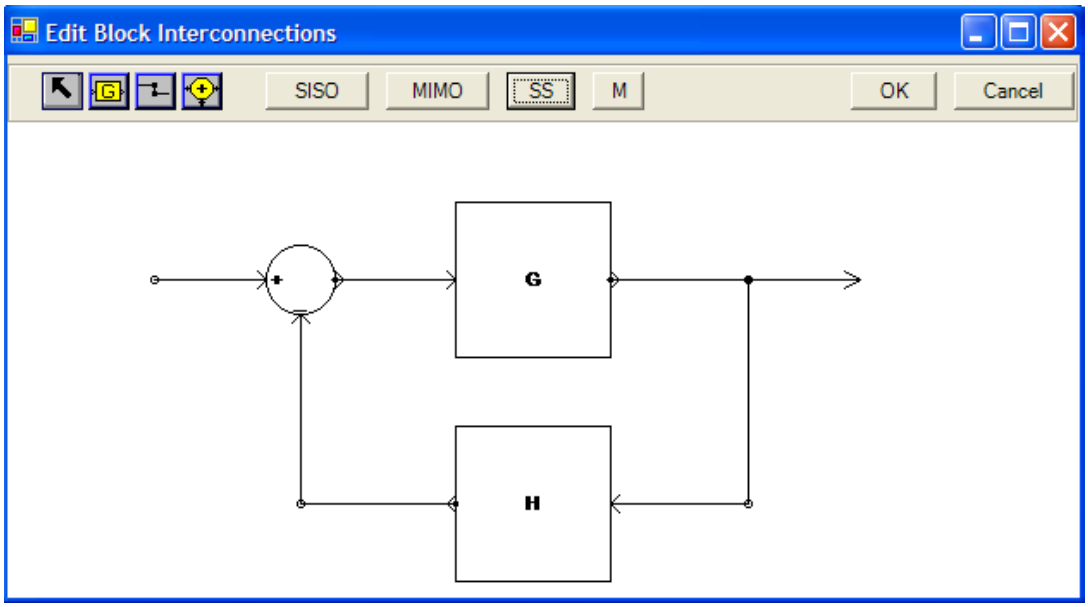

Figure 2.11 : Simple feedback control structure.

The toolbox first finds an extended interconnection matrix as given in (2.21), and then employs the block diagram reduction algorithms for MIMO systems so as to calculate the state space matrices of the closed-loop system as shown in (2.22) [57].

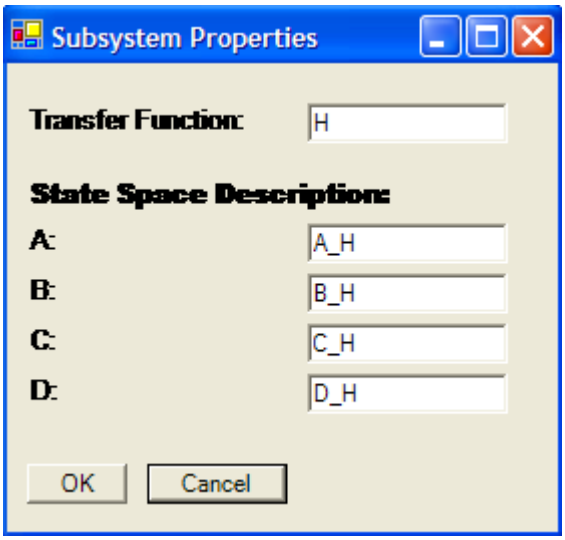

Figure 2.12 : Entering the state-space descriptions of subsystems.

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_G & 0 & D_G \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_F & 0 & C_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -B_G & 0 & -D_G \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -B_G & 0 & -D_G \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_G & 0 & C_G & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & B_F & 0 & D_F & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & B_F & 0 & D_F & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \textbf{(2.53)}$$

$$A_c = \begin{bmatrix} A_F - B_F D_G (I + D_F D_G)^{-1} C_F & B_F (I + D_G D_F)^{-1} C_G \\ -B_G (I + D_F D_G)^{-1} C_F & A_G - B_G D_F (I + D_G D_F)^{-1} C_G \end{bmatrix} \quad \textbf{(2.54)}$$

$$B_c = \begin{bmatrix} B_F D_G (I + D_F D_G)^{-1} \\ B_G (I + D_F D_G)^{-1} \end{bmatrix} \quad \textbf{(2.55)}$$

$$C_c = ((I + D_F D_G)^{-1} C_F - D_G \quad (I + D_G D_F)^{-1} C_G) \quad \textbf{(2.56)}$$

$$D_c = D_G (I + D_F D_G)^{-1} \quad \textbf{(2.57)}$$

2.4.2 Calculation of Stabilizing Gains

Consider a single input single output system with a transfer function

$$G(s) = \frac{N(s)}{D(s)} \quad \textbf{(2.58)}$$

where *N(s)* and *D(s)* are numerator and denominator polynomials, respectively. Recently, there has been an extensive research on finding all stabilizing constant gain intervals for such systems [19–22]. In order to calculate all stabilizing gain intervals efficiently, a generalised version of the Hermite-Biehler theorem [19,58] or a generalised version of the Nyquist theorem [21] can be used. These approaches,

however, require calculation of several sub-expressions and might not be convenient for small to mid-sized problems where a quick answer to the problem is sought. With the help of a symbolic algebra environment, a direct solution to the problem can be found. In the following example, symbolic calculation properties of Mathematica is used to find the Routh table of the closed-loop system in symbolic form and then all stabilizing gain intervals are found through the Routh Criterion.

Here, we consider a system with the transfer function

$$G(s) = \frac{s^3 + 3s^2 + 2s - 2}{s^4 + 5s^3 + 10s^2 + 4s + 6} \qquad \textbf{(2.59)}$$

taken from Example 3.2 of Silva et al [84]. A Routh table for the closed-loop system characteristic polynomial ($p_c(s) = D(s) + kN(s)$) can be derived in symbolic format:

Table 2.1: Routh table for the system taken from Example 3.2 of [84]**.**

s^4	1	$10+3k$	$6-2k$
s^3	$5+k$	$4+2k$	0
s^2	$\frac{46+23k+3k^2}{5+k}$	$6-2k$	
s^1	$\frac{34+174k+72k^2+8k^3}{46+23k+3k^2}$		
s^0	$6-2k$		

Simplification of the condition that all the elements in the first column (the result of the RouthTabulation command, which is a simple script for constructing the Routh table [85] should be positive gives us the result that all stabilizing gains for this system are between -0.213882 and 3. It should be remarked that this example is being used in the computer laboratory sessions of the Control System Design course (KON 314) given in the third year of Control Engineering Program in Istanbul Technical University. Our experience tells us that third year undergraduate students can use these tools very efficiently. It is worth to remark, however, that for high order systems (order more than 10) the Routh table can become very complex and finding a solution to the stabilizability problem directly is very difficult if not

impossible. Therefore the application of algebraic methods suggested in the literature [21] is suggested for systems with high order.

It should also be noted that a similar strategy can be used in finding stabilizing gain intervals for discrete time systems. Here, bilinear transformation can be applied to convert the system to a continuous time system as far as the stability is concerned. The system with transfer function

$$G(z) = \frac{100z^3 + 2z^2 + 3z + 11}{100z^5 + 2z^4 + 5z^3 - 41z^2 + 52z + 70} \quad \textbf{(2.60)}$$

given as an example in Hao et al [86], is considered, and it has been found out that all stabilizing gains for this system are between -0.417762 and -0.126272.

2.4.3 Dominant Pole Assignment

Another application area of symbolic algebra where an insight to a control engineering problem can be obtained is the dominant pole placement problem. Here, the closed-loop system characteristic polynomial can be obtained in terms of the free parameters of a fixed structure controller. Then, the coefficients of this polynomial is equated to the coefficients of the polynomial obtained by multiplying the desired polynomial ($p_d(s)$: polynomial formed by the required "dominant" closed-loop system poles) and the residue polynomial ($p_e(s)$: the polynomial formed by other poles of the closed-loop system). The solution can then be refined to restrict non-dominant poles and/or closed-loop system zeros in a left half plane, or to improve robustness. The following example illustrates how symbolic algebra can be used in such a design methodology.

2.4.3.1 Example

Consider a linear time invariant system with the transfer function

$$G(s) = \frac{2}{s^4 + 22s^3 + 160s^2 + 416s + 256} \quad \textbf{(2.61)}$$

and assume that the system is to be controlled by a PID controller;

$$F(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad \textbf{(2.62)}$$

The closed-loop system characteristic polynomial can be calculated as given below

$$p(s,K_p,K_i,K_d) = s^5 + 22s^4 + 160s^3 + (416 + 2K_d)s^2 + (256 + 2K_p)s + 2K_i \quad \textbf{(2.63)}$$

If two of the closed-loop system poles are required to be at $-\frac{1}{2} \pm \frac{1}{2} j$, then it is possible to write the closed-loop system characteristic polynomial as

$$p_c(s) = p_d(s)\, p_e(s) \quad \textbf{(2.64)}$$

where $p_d(s)$ is the desired polynomial given as

$$p_d(s) = (s + \frac{1}{2} + \frac{1}{2} j)(s + \frac{1}{2} - \frac{1}{2} j) \quad \textbf{(2.65)}$$

and $p_e(s)$ is the residue polynomial given as

$$p_e(s) = e_3 s^3 + e_2 s^2 + e_1 s + e_0 \quad \textbf{(2.66)}$$

Here, the coefficients of the residue polynomial (e_i) are not known. It is (almost) always possible to assign two of the closed-loop system poles arbitrarily using two of the controller parameters (say K_i and K_p). The remaining controller parameter (say K_d) is left as a free parameter to satisfy further design criteria.

Therefore by equating the coefficients of the closed-loop system characteristic polynomials obtained above, it is possible to find a symbolic solution to the general pole assignment problem. In the solution provided, the unknowns (e_0, e_1, e_2, e_3, K_i and K_p) are found in terms of the free parameter (K_d). By the help of symbolic algebra we found that all PID controllers that assign two of the closed-loop system poles to $-\frac{1}{2} \pm \frac{1}{2} j$ as

$$K_p = K_d + \frac{321}{8} \quad \textbf{(2.67)}$$

$$K_i = \frac{2K_d + 267}{4} \quad \textbf{(2.68)}$$

where K_d is a free parameter chosen from the real field. We remark that not all values of K_d result in the dominance of the assigned poles. In order to ensure the dominance of the assigned poles, we may require the roots of the residue polynomial to be restricted on a left half plane (say on the left of $s=-5/2$ line). This is equivalent to requiring Hurwitz stability of $p_e(s+\sigma)$, where σ is taken as -5/2 for our example. Symbolic algebra can help us at this point by calculating $p_e(s+\sigma)$ easily as given below

$$p_e(s+\frac{5}{2}) = s^3 + \frac{27}{2}s^2 + \frac{209}{4}s + \frac{291}{8} + 2K_d \qquad \textbf{(2.69)}$$

and then finding the conditions on K_d to make $p_e(s+\sigma)$ Hurwitz stable.

It is now obvious that all dominant pole assignment controllers for the problem considered are described by (2.43) and (2.44) where the free parameter is restricted as $-18.1875 < K_d < 334.5$. It is even possible to take a further step in the design by restricting the zeros of the controller to be on the left of $s=-5/2$ line, and therefore guaranteeing that the time response of the closed-loop system are mainly shaped by the assigned poles. For this purpose, the polynomial

$$NF(s-5/2) = K_d(s-5/2)^2 + K_p(s-5/2) + K_i \qquad \textbf{(2.70)}$$

is required to be Hurwitz stable. After the application of this condition we can state that all dominant pole assignment PID controllers that assign two of the closed-loop system poles at $-\frac{1}{2} \pm \frac{1}{2}j$ and all the remaining closed-loop system poles and zeros to the left of $s=-5/2$ line are given by (2.43) and (2.44), where $7.89706 < K_d < 10.0313$. It is possible to show that all these controllers have a very similar closed-loop time-response as dictated by the dominant poles (see Figure 2.13). It is worth to mention that all the steps in such a design method are easily tractable and open for further investigations. It should be noted that formal methods such as the employment of polynomial Diophantine equations can also be used in the solution of dominant pole assignment problems [48]. The aim here is to demonstrate the power of symbolic algebra as for solving rather complicated problems directly. Symbolic algebra also helps us to understand the nature of problems and the design

methodologies at hand in detail. This has great value not only in research but also in teaching. It has been observed that third year undergraduate students can work out even more complicated design problems by the help of symbolic algebra.

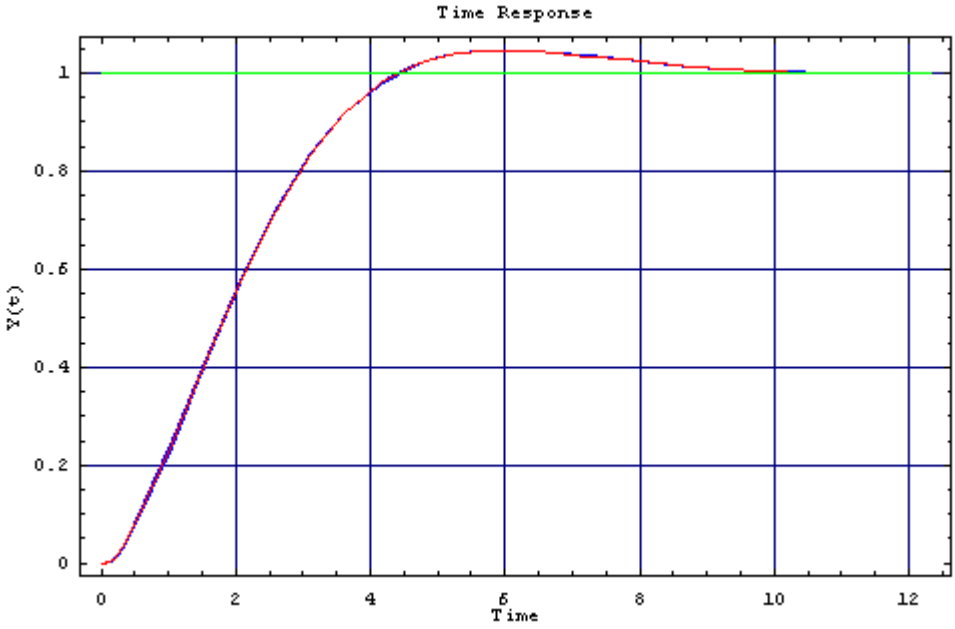

Figure 2.13 : Step responses of the closed-loop systems for $7.897 < K_d < 10.03$.

In (2.37), the coefficients of the controller enter into the closed-loop system characteristic polynomial linearly. This is very convenient as far as the dominant pole assignment problem is concerned, since the solution to the problem can be obtained by solving linear equalities. Fortunately, when dynamic compensators are used to control single-input single-output systems we always end up with a similar closed-loop system characteristic polynomial, and therefore can apply a similar design technique to assign a few of the closed-loop system poles to required locations and restrict the remaining poles in a left half plane. For multi-input multi-output (MIMO) systems, however, things can be more complicated since the controller coefficients enter into the closed-loop system characteristic polynomial non-linearly. Nonetheless, symbolic algebra languages can still be used to solve such complicated problems as illustrated by the following example.

2.4.3.2 Example

Consider a linear time-invariant MIMO system described in state space as

$$\dot{x} = Ax + Bu$$
$$y = Cx \tag{2.71}$$

where u, y, and x are input, output and state vectors, respectively, and

$$A = \begin{bmatrix} -4.2 & 4 & 0.6 & 3.8 & 1.6 \\ 1.6 & 6 & 1.2 & 2.6 & 5.2 \\ -0.6 & 6 & -0.2 & 3.4 & 3.8 \\ 1.2 & -17 & -5.6 & -9.8 & -12.6 \\ -4.4 & 1 & 3.2 & 2.6 & 0.2 \end{bmatrix} \tag{2.72}$$

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 1 & 1 \\ 5 & 2 \\ 7 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 0 & 2 & 1 \\ 2 & 4 & 2 & 1 & 1 \end{bmatrix} \tag{2.73}$$

Assume that it is required to assign three of the closed-loop system poles at $s_1 = -1/2$, $s_{2,3} = -\frac{1}{2} \pm \frac{1}{2} j$ and restrict the rest of the closed-loop system poles to the left of $s = -5/2$ line using a constant output feedback controller K such that

$$u = r - Ky \tag{2.74}$$

Note that due to Söylemez et al [87] it is possible to assign 3 poles of the closed-loop system as required. However, since the method suggested in [87] involves linear equations, it does not provide enough design freedom to restrict the remaining closed-loop poles to the left of $s = -5/2$ line. Actually, in order to solve this problem, nonlinear equations need to be solved as will be shown in the following.

Considering the fact that the closed-loop system A matrix (A_c) is given by $A_c = A - BKC$, the closed-loop system characteristic polynomial can be calculated. It should be noted that the elements of the A matrix are converted to rational numbers to avoid possible numerical problems as discussed in section 2.3. Using a strategy similar to the previous example all constant output-feedback compensators that assign three of the closed-loop system poles at desired locations are given as follows

$$K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{2.75}$$

such that

$$\begin{aligned} k_{11} &= \frac{122047843993880021 - 547588313659627733k_{22} - 4309919793p\sqrt{\Delta}}{107806832675576053860} \\ k_{12} &= \frac{-1590518241 - 94771357786k_{22} + p\sqrt{\Delta}}{180628080480} \\ k_{21} &= \frac{-1274729635 + 60104488802k_{22} + 5p\sqrt{\Delta}}{57297048772} \end{aligned} \tag{2.76}$$

where

$$\Delta = 4645701746401 - 600153564736\ k_{22} + 878672403k_{22}^2 \tag{2.77}$$

and $k_{22} \in \mathbb{R}$. In (2.52), *p* can be +1 or -1 (i.e. two sets of solutions exist). It should be noted that the results are exact; not numerical approximations. In order to restrict non-dominant poles to the left of $s = -5/2$ line, it is necessary and sufficient to have all coefficients of $p_e(s-5/2)$ positive, since this polynomial is of second order. After applying these conditions for each solution set, Mathematica returns the following conditions on k_{22} for dominant pole assignment (numerical values are given).

$$-0.295308 < k_{22} \le 0.00890075 \quad \text{or} \quad k_{22} \ge 0.0594016 \tag{2.78}$$

for *p=+1*, and

$$-0.0304795 < k_{22} \le 0.00890075 \quad \text{or} \quad 0.0594016 \le k_{22} \le 2.59539 \tag{2.79}$$

for *p=-1*.

As a result, symbolic algebra helped us to solve a rather complicated problem very easily. It would be very difficult, if not impossible, to solve such problems by hand calculations and/or using numerical software, especially after considering the fact that the examples provided here are rather simple for illustration purposes.

2.4.4 Control of Systems with Parameter Uncertainty

Many practical systems include parametric uncertainties in their mathematical model [87]. Leaving the uncertain parameters in symbolic format and doing calculations symbolically as further as possible is usually desired for such systems, as such an approach can provide a better understanding of the problem and simplifications to reduce computation time, which is typically an important issue when a certain property of the system (such as controllability, observability or stability) is to be tested for all possible uncertainties. A particular application area of computer algebra with this respect is the robust pole assignment problem [60,88]. In the following, we provide an example to illustrate the efficiency of symbolic algebra in drawing pole spread and designing robust pole assignment controllers.

2.4.4.1 Example

Consider the system given in the example in 2.4.3.2, and assume that (1,1) and (1,5) elements of the system *A* matrix as well as (1,1) element of the system *B* matrix are subject to perturbation such that $a_{11} = -4.2 + q_1$, $a_{15} = 1.6 + q_2$ and $b_{11} = 1 + q_3$, where q_i represent the uncertain variables and $|q_i| \le 0.5$ for *i=1,2,3*. Then, for any of the dominant pole assignment controllers found in the previous example, the closed-loop system poles perturb from nominal values as q_1 and q_2 deviates from zero. Finding the pole spread of the closed-loop system for all possible perturbations of q_1 and q_2 for a given value of k_{22} is desired usually to test the stability and time domain performance of the closed-loop system. For instance, if $k_{22}=1$ (with $p=+1$) the controller can be calculated as

$$K = \begin{bmatrix} -0.858 & -0.483 \\ 1.817 & 1 \end{bmatrix} \qquad \textbf{(2.80)}$$

and the pole spread of the closed-loop system for all possible perturbations of the uncertain parameters is shown in Figure 2.14. In calculation of the pole spread, symbolic algebra can be useful at several stages. First of all, by calculating the closed-loop system characteristic polynomial in symbolic form the structure of the uncertainty of this polynomial can be determined automatically. For example, the closed-loop system characteristic polynomial for the system we consider is of affine-linear type (i.e. uncertain parameters q_i get into the coefficients of the polynomial

affine-linearly). This means using only the edges of the parameter box is enough in determination of the pole spread [89]. It should be noted that although for this particular system determining the structure of the closed-loop system characteristic polynomial is relatively easy (one can see the affine linear structure after observing the fact that uncertain parameters are only affecting the first row of the *A* and *B* matrices), this is not always the case.

In computations related with parametric uncertain systems, it is very usual that the same calculations are done over and over again (thousands if not millions of times) for different values of uncertain parameters. Therefore, finding a simple formula before starting numerical calculations is of great importance. This is one of the places where symbolic algebra can provide a tremendous help. It is, for example, possible to find the closed-loop system characteristic polynomial ($p_c(s,q,k)$) in the most general form in terms of uncertain parameters q_i and the free parameter k_{22}. We remark that even a further optimization on this formula is possible to increase computational speed by determining common factors and calculating them separately [90].

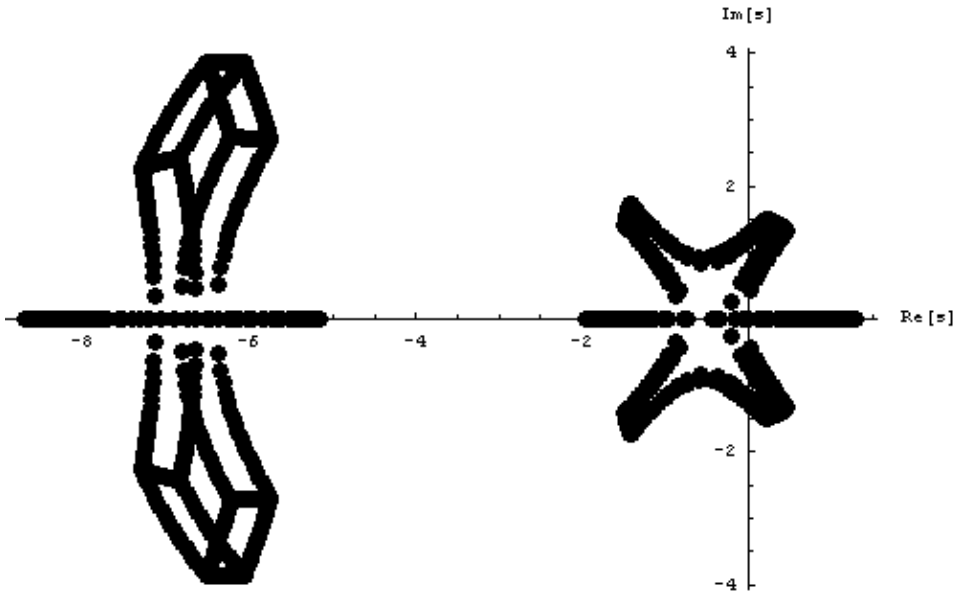

Figure 2.14 : Pole spread of the closed-loop system for $k_{22}=1$ and $p=+1$.

After calculating the general form of the closed-loop system characteristic polynomial, it is then possible to try different values of k_{22} both for $p=+1$ and $p=-1$ in order to find a robust constant feedback controller. Since there is only one free parameter (k_{22}), it is possible to grid k_{22} in the allowable intervals for $p=+1$ and

$p=-1$, and use a pole-colouring approach [59] to evaluate a suitable cost function with respect to the corresponding pole spread. After minimizing such a cost function it has been observed that the closed-loop system polynomial with $k_{22}=0.9$ (and $p=-1$) corresponding to

$$K = \begin{bmatrix} -0.121 & -0.526 \\ 0.214 & 0.9 \end{bmatrix} \qquad \textbf{(2.81)}$$

has an acceptable pole spread (see Figure 2.15).

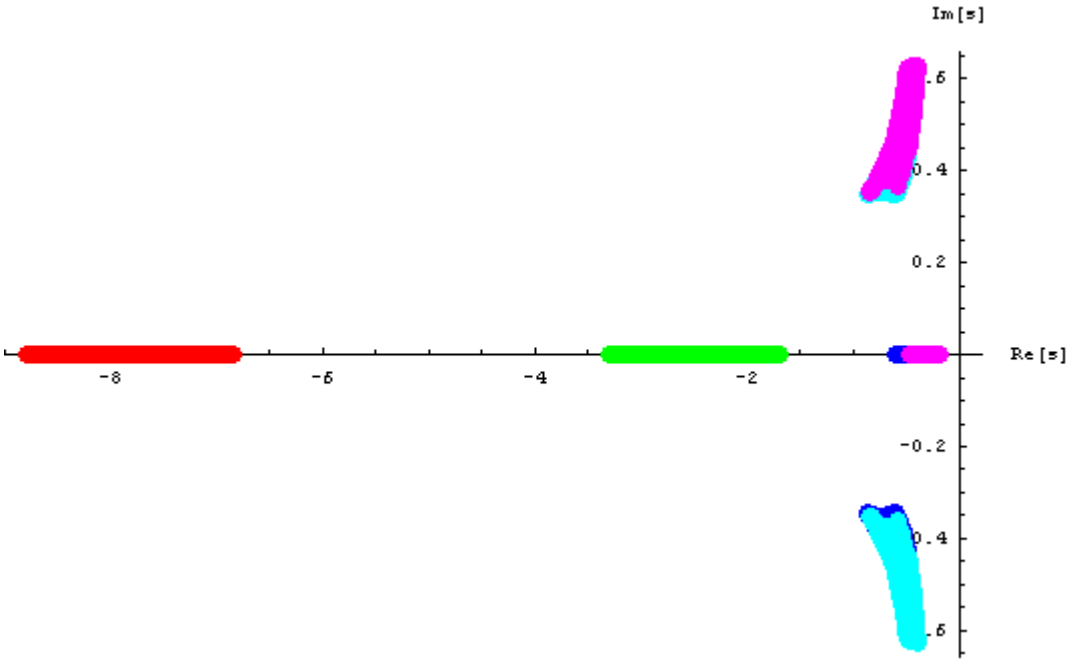

Figure 2.15 : Pole spread of the closed-loop system for $k_{22}=0.9$ and $p=-1$.

3. FEASIBILITY CONDITIONS ON PID CONTROLLER SYNTHESIS USING DOMINANT POLE ASSIGNMENT

3.1 Objectives

The focus of this chapter is to develop a method for PID controller design which can achieve dominant pole assignment using two of the controller parameters. The non-dominant poles are restricted on the left of the line $s = \hat{\sigma}$, where $\hat{\sigma}$ is the minimum feasible value, called as the feasibility border. It is obvious that a dominant pole assignment is not practical if $\hat{\sigma}$ is close to the real parts of the required dominant poles. Hence, finding $\hat{\sigma}$ for a given system is very important. The method, which parameterizes all such controllers in order to allow further design criteria, can be applied to other kinds of low-order compensators. Parts of this chapter have been presented at the 9th European Control Conference in Kos, July 2007 [91] .

3.2 Introduction

PID controllers by far are the most common controllers in use today. Their success in industry is due to their simplicity, practicality and satisfactory performance. Naturally, there are numerous methods for designing PID controllers. Most of the methods in the literature are actually about tuning three parameters of the PID controller after doing a few experiments on the system to be controlled [92–96]. When the system to be controlled is not complex (first and second order) the so-called tuning rules usually offer a "good" set of parameters as a starting point for tuning the PID controller parameters. For high order systems, however, simple tuning rules do not always give satisfactory results and therefore more complicated design techniques are required to obtain acceptable results from PID controllers.

A common problem with most of the design methods that offer a good set of parameters is that, when the real system does not perform as required there is usually no design freedom to help the control engineer. Hence, the practicing control engineer is usually left on his/her own to play with the three parameters (around the

good set) using ad hoc methods to find a satisfactory controller. Therefore, a reasonable approach to a controller design is to find the set of all stabilizing compensators and then using a member of this set to satisfy further design criteria. The so-called Youla parameterization provides an algebraic formulation of all stabilizing controllers for a given system and many of the modern control synthesis techniques are based on this method. An important disadvantage of this parameterization is that the order of the controller cannot be fixed, so the order of the controllers found tends to be quite high. In the last few years, researchers on PID controller design have focused on finding all stabilizing PID controllers rather than finding a good set of parameters. From control engineering point of view there is a serious drawback in selecting a suitable set of parameters within a large range of possible stabilizing controllers. Using PID controllers, analytical pole placement is possible when the order of the system to be controlled is one or two. For higher order systems up to three dominant poles can be located at desired positions to give the required closed loop performance. However, if some of the remaining (unassigned) closed-loop system poles are at undesired locations a new set of dominant poles is chosen and a redesign is carried out.

It is almost always possible to assign two of the closed-loop system poles using two of the PID controller parameters (k_p, k_i). The remaining parameter (k_d) is left as a free parameter and will be used to place the rest of the closed-loop poles (non-dominant) on the left of a given point σ in the left half s-plane.

It is a common fact that for the real parts of the roots of a polynomial $p(s)$ to be smaller than σ, the polynomial $p(s+\sigma)$ must be Hurwitz. Obviously, for a given plant transfer function and a pair of desired dominant poles, it may not be always possible to place all non-dominant poles left of a given $s=\sigma$ line. Such σ values are called unfeasible, and for these values there will be no stabilizing intervals of the free parameter (k_d). For a given system and a pair of desired poles finding the minimum feasible σ value $\hat{\sigma}$, called as the feasibility border, is very useful because if $\hat{\sigma}$ is near to or greater than the real parts of the required dominant poles the dominant pole assignment is not practical. In this chapter, the main idea introduced in [51] is applied to dominant pole assignment problem using PID controllers.

Section 3.3 contains the preliminaries for dominant pole assignment using PID controllers and some numerical examples.

3.3 Dominant Pole Placement

Consider a linear time-invariant system defined by a strictly proper rational transfer function

$$G(s)=\frac{N(s)}{D(s)}=\frac{b_m s^m+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0} \tag{3.1}$$

The transfer function of a PID controller can be given as

$$C(s)=\frac{k_d s^2+k_p s+k_i}{s} \tag{3.2}$$

where, k_p, k_i, and k_d are the proportional, integral and derivative gain terms, respectively. The characteristic polynomial of the (unit feedback) closed-loop system is as follows

$$\begin{aligned} p_c(s,\,q) &= N(s)(k_d s^2+k_p s+k_i)+s\,D(s) \\ &= \begin{bmatrix} 1 & s & \cdots & s^{n+1} \end{bmatrix}\left(\delta_0+N_1\begin{bmatrix} k_i \\ k_p \end{bmatrix}+N_2 k_d\right) \end{aligned} \tag{3.3}$$

in which

$$\delta_0=\begin{bmatrix} 0 \\ a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ 1 \end{bmatrix},\; N_1=\begin{bmatrix} b_0 & 0 \\ b_1 & b_0 \\ \vdots & \vdots \\ b_m & b_{m-1} \\ 0 & b_m \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{(n+2)\times 2} \qquad N_2=\begin{bmatrix} 0 \\ 0 \\ b_0 \\ b_1 \\ \vdots \\ b_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.4}$$

Additionally, closed-loop system zeros are the roots of the following polynomial

$$p_z(s,q) = N(s)(k_d s^2 + k_p s + k_i) \tag{3.5}$$

The polynomial that corresponds to dominant poles is called as the desired polynomial and can be given as

$$p_d(s) = (s - p_1)(s - p_2) \triangleq s^2 + d_1 s + d_0 \tag{3.6}$$

where p_1 and p_2 are the desired poles, which specify certain damping ratio and natural frequency requirements. Furthermore, the coefficients of the so called residue polynomial $p_e(s)$, the polynomial that corresponds to non-dominant poles of the closed loop system, are not known. The residue polynomial is of the form

$$p_e(s) = e_{n-1} s^{n-1} + \cdots + e_1 s + e_0 \tag{3.7}$$

It is then possible to partition the closed-loop system characteristic polynomial as follows:

$$p_c(s,\, q) = p_d(s)\, p_e(s) \tag{3.8}$$

which represented in matrix form can be given as

$$p_d(s)\, p_e(s) = (1 \;\; s \;\; \cdots \;\; s^{n+1})\, De \tag{3.9}$$

where the following definitions can be made

$$D = \begin{bmatrix} d_0 & 0 & \cdots & 0 \\ d_1 & d_0 & & \vdots \\ 1 & d_1 & \cdots & 0 \\ 0 & 1 & \cdots & d_0 \\ \vdots & & \ddots & d_1 \\ 0 & & & 1 \end{bmatrix}_{(n+2)\times n} \qquad e = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-2} \\ e_{n-1} \end{bmatrix} \tag{3.10}$$

From (3.3) and (3.9) it is possible to show that

$$\delta_0 + N_1 \begin{bmatrix} k_i \\ k_p \end{bmatrix} + N_2 k_d = De \tag{3.11}$$

Rearranging this equation results in

$$\begin{bmatrix} e \\ k_i \\ k_p \end{bmatrix} = \begin{bmatrix} D & -N_1 \end{bmatrix}^{-1} (\delta_0 + N_2 k_d) \qquad \textbf{(3.12)}$$

By solving (3.12) it is possible to find a symbolic solution to the pole assignment problem, in which the unknowns are obtained in terms of the free parameter. Substituting these symbolic results back in the residue polynomial will end up with a polynomial $p_e(s+\sigma, k_d)$ depending on the free parameter k_d. In order to guarantee the dominance of the placed poles, we may require the real parts of the roots of the residue polynomial to be restricted on the left of the line $s = \sigma$ where σ is a negative real number. The problem can be put as given the plant transfer function, what is smallest value σ such that $p_e(s+\sigma, k_d)$ is stabilizable by a constant k_d? From (3.12) the coefficients of the residue polynomial are found as follows:

$$e = \begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} D & -N_1 \end{bmatrix}^{-1} (\delta_0 + N_2 k_d) \qquad \textbf{(3.13)}$$

where I_n denotes $n \times n$ identity matrix. It is now possible to express the residue polynomial as

$$p_e(s) = A(s) + k_d B(s) \qquad \textbf{(3.14)}$$

where $A(s)$ and $B(s)$ are defined as

$$A(s) = S_n \begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} D & -N_1 \end{bmatrix}^{-1} \delta_0 \qquad \textbf{(3.15)}$$

$$B(s) = S_n \begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} D & -N_1 \end{bmatrix}^{-1} N_2 \qquad \textbf{(3.16)}$$

in which $S_n = (1 \;\; s \;\; s^2 \;\; \cdots \;\; s^{n-1})$. The crucial observation about (3.14) is that the free parameter k_d enters into the residual polynomial linearly. Therefore the problem of restricting non-dominant poles in the left-half plane is equivalent to finding the stabilizing k_d gain intervals. Algebraic solutions to this problem have been developed by several researchers in recent years [19–21,58,97]. Furthermore, for all-pole SISO systems, where $N(s) = b_0$ in (3.1), N_1 and N_2 turn out to be as follows:

$$N_1 = \begin{bmatrix} b_0 & 0 \\ 0 & b_0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{(n+2)\times 2} \qquad N_2 = \begin{bmatrix} 0 \\ 0 \\ b_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.17}$$

Using (3.17) together with (3.16) it is possible to show that $B(s) = b_0$.

A recent study [51] provides direct formulations of the lower and upper bounds for all-pole SISO systems closed under constant gain feedback (k), to calculate the smallest value of σ (called $\hat{\sigma}$), where the closed-loop system characteristic polynomial is given as follows

$$p_c(s,k) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 + b_0 k \tag{3.18}$$

This value satisfies the following inequality

$$\lambda_{r\max} \le \hat{\sigma} \le \lambda_{h\max} \tag{3.19}$$

where $\lambda_{r\max}$ is the maximum real root of all derivatives of the closed-loop system characteristic polynomial $p_c(s,k)$. On the other hand, $\lambda_{h\max}$ is the maximum of the real roots of the polynomials $p_{hm}(s) \triangleq |H_m|$. Here, H_m is the m th Hurwitz matrix for the closed-loop system characteristic polynomial with its coefficients replaced by polynomials

$$p_m(s) = \frac{p_c^{(m)}(s,k)}{m!} \tag{3.20}$$

for $m = 1, 2, ..., n-1$. Note that for all-pole systems the residue polynomial is of the form given as follows

$$p_e(s) = A(s) + k_d b_0 \tag{3.21}$$

which makes it possible to apply the result given in (3.19) to dominant pole assignment problem using PID controllers.

3.3.1 Example

Consider a linear time invariant system with the transfer function

$$G(s) = \frac{42}{s^5 + 31s^4 + 348s^3 + 1694s^2 + 3116s + 480} \tag{3.22}$$

The polynomials $p_m(s)$ are given as follows

$$p_1(s) = 5s^4 + 124s^3 + 1044s^2 + 3388s + 3116 \tag{3.23}$$

$$p_2(s) = 10s^3 + 186s^2 + 1044s + 1694 \tag{3.24}$$

$$p_3(s) = 10s^2 + 124s + 348 \tag{3.25}$$

$$p_4(s) = 5s + 31 \tag{3.26}$$

with maximum real root $\lambda_{r\max} = -1.49$. Using these polynomials, the second set of polynomials $p_{hm}(s)$ can be determined from the following determinants

$$p_{h1}(s) = p_4(s) \tag{3.27}$$

$$p_{h2}(s) = \begin{vmatrix} p_4(s) & p_2(s) \\ 1 & p_3(s) \end{vmatrix} \tag{3.28}$$

$$p_{h3}(s) = \begin{vmatrix} p_4(s) & p_2(s) & 0 \\ 1 & p_3(s) & p_1(s) \\ 0 & p_4(s) & p_2(s) \end{vmatrix} \tag{3.29}$$

$$p_{h4}(s)=\begin{vmatrix} p_4(s) & p_2(s) & 0 & 0 \\ 1 & p_3(s) & p_1(s) & 0 \\ 0 & p_4(s) & p_2(s) & 0 \\ 0 & 1 & p_3(s) & p_1(s) \end{vmatrix} \tag{3.30}$$

where the maximum real root is found to be $\lambda_{h\max}=-1.49$. Hence, we can immediately conclude that $\hat{\sigma}=-1.49$. As mentioned before, for the real parts of the roots of $p_c(s,k)$ to be smaller than σ, the following polynomial has to be Hurwitz.

$$\begin{aligned} p(s+\sigma,\,k) = {} & s^5+s^4(31+5\sigma)+s^3(10\sigma^2+124\sigma+348) \\ & +s^2(10\sigma^3+186\sigma^2+1044\sigma+1694) \\ & +s(5\sigma^4+124\sigma^3+1044\sigma^2+3388\sigma+3116) \\ & +\sigma^5+31\sigma^4+348\sigma^3+1694\sigma^2+3116\sigma+480+42k \end{aligned} \tag{3.31}$$

By the help of a simple script the Hurwitz determinants for (3.31) can be calculated.

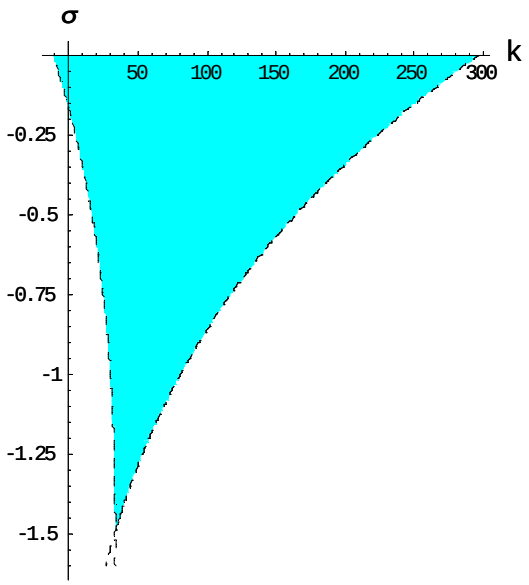

Figure 3.1 : Feasible region for the Example in 3.3.1 .

In order the polynomial to be stable each Hurwitz determinant has to be greater than zero, which results in a set of semi-algebraic curves in the parameter space, i.e., (k,σ)-plane. By the help of *IneqGraphics* package [98] these inequalities and the stability region can be visualized as shown in Figure 3.1.

3.3.2 Example

Let us now consider the same system given in (3.22) and assume that the system is to be controlled by a PID controller. The closed-loop system characteristic polynomial can be calculated as given below

$$p(s, k_p, k_i, k_d) = s^6 + 31s^5 + 348s^4 + 1694s^3 + (3116 + 42k_d)s^2 + (480 + 42k_p)s + 42k_i \quad \textbf{(3.32)}$$

If two of the closed-loop system poles are required to be at $-\frac{1}{2} \pm i\frac{1}{2}$, then it is possible to write the closed-loop system characteristic polynomial as the product of the desired polynomial which is given as

$$p_d(s) = s^2 + s + \frac{1}{2} \quad \textbf{(3.33)}$$

and the residue polynomial given as

$$p_e(s) = e_4 s^4 + e_3 s^3 + e_2 s^2 + e_1 s + e_0 \quad \textbf{(3.34)}$$

All PID controllers that assign two of the closed-loop system poles to $-\frac{1}{2} \pm i\frac{1}{2}$ are found as

$$k_p = k_d + \frac{3593}{84} \quad \textbf{(3.35)}$$

$$k_i = \frac{k_d}{2} + \frac{6383}{336} \quad \textbf{(3.36)}$$

where, k_d is the free parameter. The residue polynomial is obtained as follows

$$p_e(s) = s^4 + 30s^3 + \frac{635}{2}s^2 + \frac{2723}{2}s + \frac{6383}{4} + 42k_d \quad \textbf{(3.37)}$$

By the help of a simple script for constructing the Routh table [27,85] $p_e(s)$ is found to be Hurwitz if $-37.994 < k_d < 256.043$. Using the result given in (3.19) the lower and upper bounds are obtained as $\lambda_{r\max} = \lambda_{h\max} = -4.0525$. Therefore it is possible to

state that there exists a constant derivative term gain that makes the real parts of all the poles to the left of $s=-4.0525$ line. Using the graphical approach it is also possible to visualize this result. For that purpose let us consider the shifted residue polynomial given as follows

$$\begin{aligned} p_e(s+\sigma,\, k_d) &= s^4 + s^3(30+4\sigma) + s^2(6\sigma^2+90\sigma+\frac{635}{2}) \\ &\quad + s(4\sigma^3+90\sigma^2+635\sigma+\frac{2723}{2}) \\ &\quad + (\sigma^4+30\sigma^3+\frac{635}{2}\sigma^2+\frac{2723}{2}\sigma+\frac{6383}{4}+42k_d). \end{aligned} \tag{3.38}$$

The feasible region is given in Figure 3.2 and it is observed that $\sigma > -4.0525$. This value is obtained if $k_d = 10.3427$, and the PID controller is given as follows

$$C(s) = 10.3427s + 53.1165 + \frac{24.1684}{s} \tag{3.39}$$

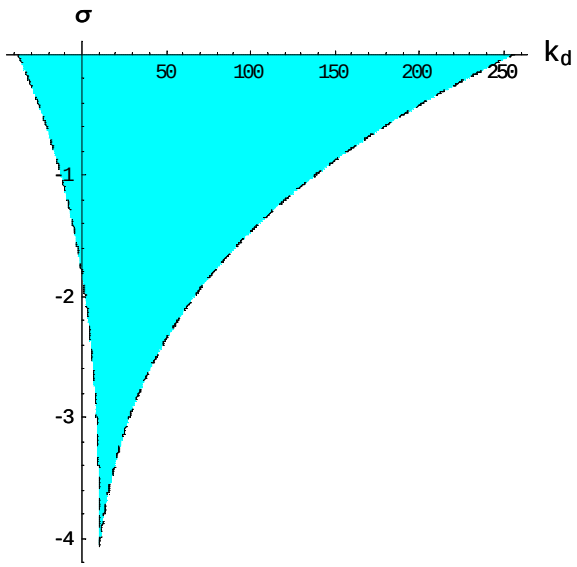

Figure 3.2 : Feasible region for the example in 3.3.2 .

Using the controller the closed-loop poles are obtained at $\{-10.9475 \pm i1.9419, -4.0525, -4.0525, -0.5 \pm i0.5\}$, and the zeros are at $\{-4.6311, -0.5046\}$. We can immediately conclude that the dominant pole assignment is achieved, and the non-dominant poles are kept on the left of the $s=-4.0525$ line. One of the transmission zeros however appears close to the

dominant poles. The closed-loop system time response for $\{k_p = 53.1165\ ,\ k_i = 24.1684\ ,\ k_d = 10.3427\}$ is given in Table 3.1. As a second step it is even possible to restrict the zeros of the system to be on the left of a $s = \hat{\sigma}$ line in order to guarantee the time response of the closed loop system to be shaped by the dominant poles. For this purpose, the polynomial $p_z(s+\sigma, q) = k_d(s+\sigma)^2 + k_p(s+\sigma) + k_i$ is required to be Hurwitz stable. Using parameterized expressions of k_p and k_i yields in the following polynomial

$$p_z(s+\sigma, k_d) = 42k_d\, s^2 + (-84\sigma k_d + 42k_d + \frac{3593}{2})s + (42\sigma^2 k_d - 42\sigma k_d - \frac{3593}{2}\sigma + 21k_d + \frac{6383}{8}). \quad \textbf{(3.40)}$$

for which the feasible set is given by a set of inequalities given in Table 3.1.

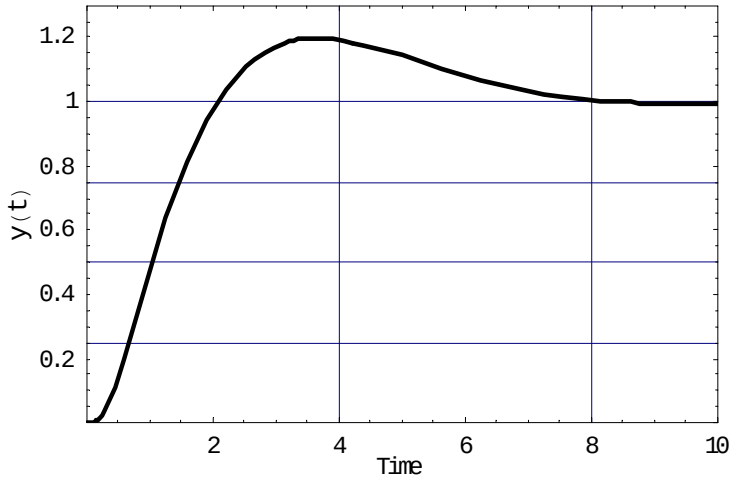

Figure 3.3 : Step response of the compensated system

Table 3.1 : Feasible Sets.

$\sigma \geq 0.059$	$k_d > -\dfrac{3593}{168\sigma + 84}$
$-0.5 \leq \sigma < 0.059$	$k_d > -\dfrac{14372\sigma + 6383}{336\sigma^2 + 336\sigma + 168}$
$-0.947 < \sigma < -0.5$	$\dfrac{-(14372\sigma + 6383)}{336\sigma^2 + 336\sigma + 168} < k_d < \dfrac{-3593}{168\sigma + 84}$

These feasibility constraints are depicted in Figure 3.4. From this figure it is possible to determine that the minimum feasible value for σ is -0.947, and this value is attained for $k_d = 47.8198$. From (3.35) and (3.36) the integral and proportional gain terms can be obtained as $k_i = 42.9069$ and $k_p = 90.5936$. Using this PID controller the closed-loop system poles are found to be at $\{-12.5977 \pm i\,4.1547\}$, $\{-0.5 \pm i\,0.5\}$, $\{-2.4023 \pm i\,3.8356\}$ and the zeros are at $\{-0.947, -0.947\}$. We observe that, all the zeros are kept on the $s = -0.947$ line.

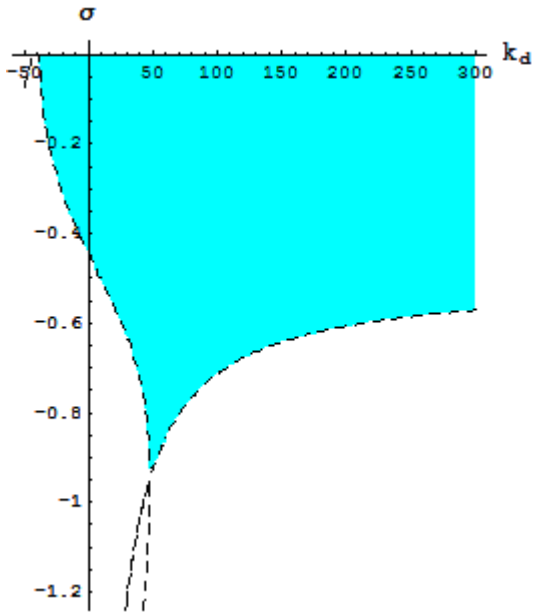

Figure 3.4 : Feasibility conditions.

The closed-loop system time response for this compensator (solid line $(k_p = 90.5936$, $k_i = 42.9069$, $k_d = 47.8198))$ and the previous compensator (dashed line $(k_p = 53.1165$, $k_i = 24.1684$, $k_d = 10.3427))$ are given in Figure 3.5. As can be observed from this figure the adverse effects of the compensator zeros have been alleviated using this last approach.

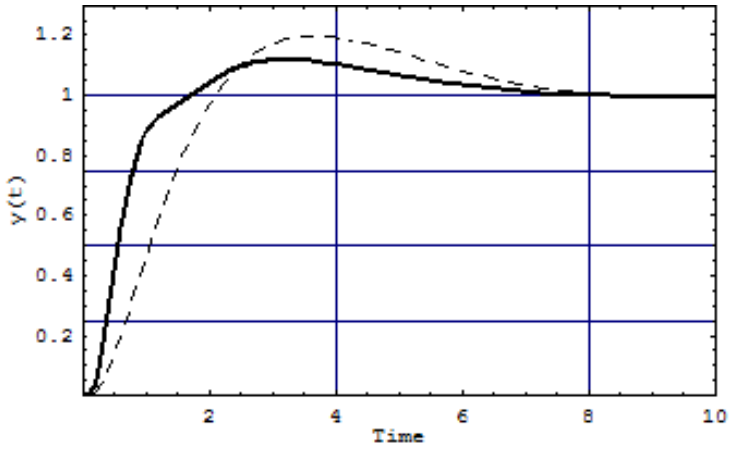

Figure 3.5 : Step response of the compensated system.

4. STABILIZING CONSTANT DIAGONAL CONTROLLERS FOR TITO SYSTEMS

4.1 Objectives

This chapter presents two algorithms for determining the stabilizing gain intervals for two-input two-output systems where the controller is a constant diagonal controller. The algorithms are based on the characteristic values of the transfer function matrix representation of the system, and differ from each other in reducibility of the corresponding characteristic equations.

4.2 Introduction

Stability is the most important property in the design of all dynamical systems. A reasonable approach to controller design is to find the set of all stabilizing compensators and then using a member of this set to satisfy further design criteria. A complete parameterization of all stabilizing controllers for a given system was suggested by Youla. An important disadvantage of this parameterization is that the order of the controller cannot be fixed. As a result, the order of the controller tends to be quite high most of the time. Therefore, in the last few years computation of all stabilizing controllers of a given order is examined by several researchers. It is a common fact that it is more difficult to design controllers for MIMO systems because there are usually interactions between different control loops. To overcome this difficulty decentralized controllers are considered which have fewer tuning parameters compared to general multivariable controllers. For example, decentralized PID controllers are widely used in process control due their simplicity and facility in working in case of actuator and/or sensor failure because it is relatively easy to tune manually as only one loop is directly affected by the failure.. If a MIMO system described by a $n \times n$ transfer-function matrix $G(s)$ is diagonal dominant over the bandwidth of interest, or there exists an input compensator matrix

$C(s)$ to achieve diagonal dominance, then the stability and time domain behavior of the system can be inferred from the diagonal elements of $G(s)C(s)$. The relative gain array, the (inverse–) Nyquist array approach, the block Nyquist array method, the Perron-Frobenius scaling procedure and the characteristic locus method are among the analysis and design methods to reduce the interaction in a multivariable system. However, these approaches do not provide the set of all stabilizing controllers. Generalizing the Nyquist stability criterion for MIMO case is particularly important because plotting the characteristic values of the open-loop transfer function enables us to check the stability of the closed-loop system for a gain parameter. This chapter is organized as follows: we start with some mathematical preliminaries in 4.3, the characteristic values and characteristic value plots are examined, reducibility and irreducibility of characteristic equations are discussed, the real-axis crossings of the characteristic value plots and their relation to the stabilizing gain intervals is introduced, and the number of unstable closed-loop poles for gain intervals is considered. In 4.4 constant diagonal controllers of type $diag(k,k)$ are used to stabilize TITO systems, the problem is discussed for irreducible and reducible cases separately, algorithms are presented to solve this problem. Some tutorial examples are given to introduce how the proposed algorithms work. In 4.5 the problem in parameter space where the number of constant gains is two is presented.

4.3 Mathematical Preliminaries

Consider a system described by 2×2 transfer function matrix $G(s)$, the characteristic equation is defined as

$$p(\lambda, s) = \left|\lambda(s)I - G(s)\right| = 0 \qquad \textbf{(4.1)}$$

where $\lambda(s)$ is a complex variable, which is denoted as characteristic value and I is the identity matrix of dimension 2. In general, for a $n\times n$ transfer function matrix the function $p(\lambda, s)$ can be factored into a product of polynomials $p_i(\lambda, s)$ in λ, $(\mathrm{i} = 1, 2, \cdots, \mathrm{r})$, which are irreducible over the field of rational functions, i.e., $p(\lambda, s) = p_1(\lambda, s)\, p_2(\lambda, s) \cdots p_r(\lambda, s)$. Calculating the characteristic values and letting s traverse the Nyquist contour in the complex plane constitutes a set of

curves, which are known as generalized Nyquist diagrams or characteristic value plots. Note that these diagrams can be plotted quickly by the *CharacteristicValuePlot* command of the Polynomial Control Systems Toolbox [99,100]. Using these diagrams, it is possible to generalize the Nyquist stability criterion for SISO systems to the multivariable case. The generalized Nyquist theorem is stated as follows,

Theorem 1: The feedback system (see Figure 4.1) is asymptotically stable if and only if

(i) $\left|I + k\,G(s)\right| \neq 0$ for all $s \in D_N$ and

(ii) $N[\left|I + k\,G(s)\right|, 0] + u_0 = 0$,

where D_N is the usual Nyquist contour and u_0 is the number of unstable poles of the open-loop system. Here, $N[h(s), s_0]$ denotes the winding number of a complex function $h(s)$ about the point s_0 [101].

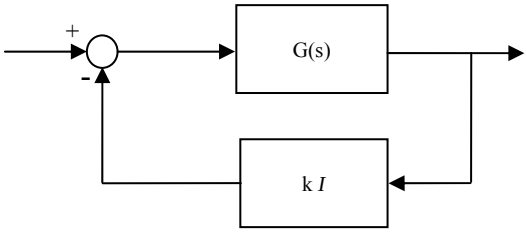

Figure 4.1 : Static output feedback with constant gain k.

Let us consider the system given by the transfer function matrix

$$G(s) = \frac{1}{(s^2+3s+2)(s+3)}\begin{bmatrix} 2s^2+12s-2 & 15s-15 \\ -2s+2 & 2s^2+s+9 \end{bmatrix} \qquad \textbf{(4.2)}$$

The characteristic values of the system given by the transfer function matrix (4.2) are as follows:

$$\lambda_1(s) = \frac{2}{s+3} \qquad \textbf{(4.3)}$$

$$\lambda_2(s) = \frac{2s+1}{s^2+3s+2} \qquad \textbf{(4.4)}$$

For this example, the characteristic value plots, i.e., the Nyquist plots for $\lambda_1(s)$ and $\lambda_2(s)$ are shown in Figure 4.2 and Figure 4.3, respectively.

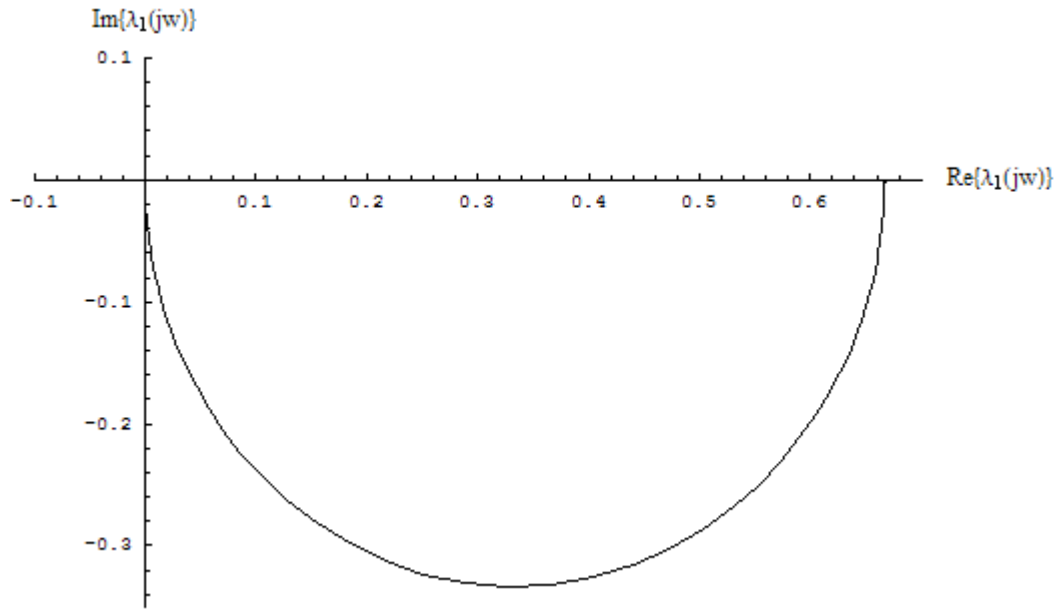

Figure 4.2 : Characteristic value plots for the system in Example in (4.2).

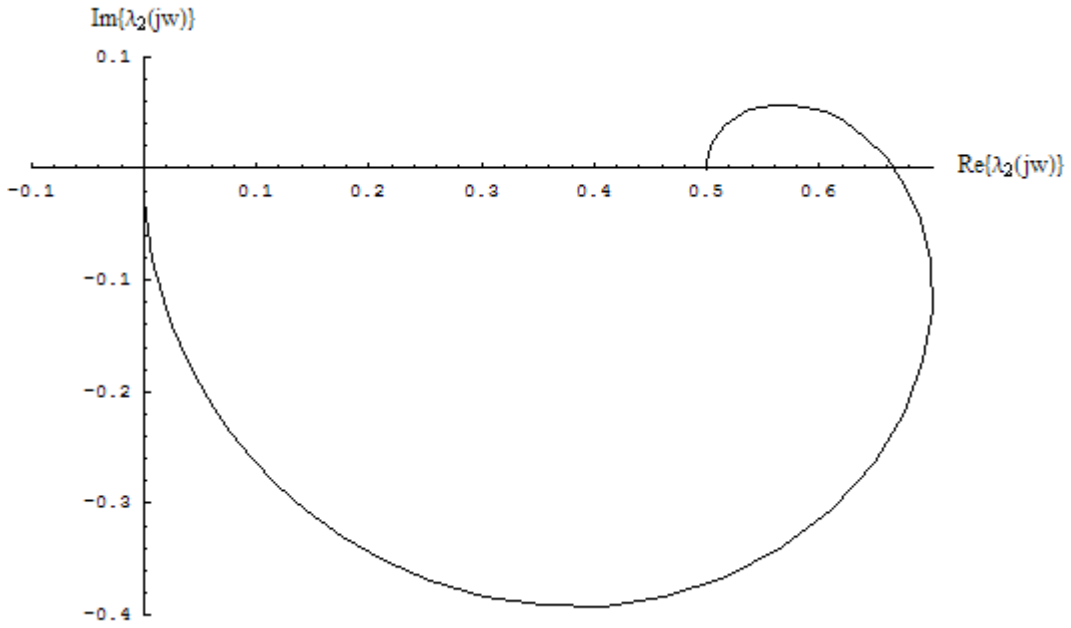

Figure 4.3 : Characteristic value plots for the system in Example in (4.2).

Let us now consider the transfer function matrix

$$G(s)=\begin{bmatrix} \dfrac{1}{s+1} & \dfrac{1}{s+2} \\ \dfrac{2s+1}{(s+1)(s+2)} & \dfrac{2}{s+1} \end{bmatrix} \tag{4.5}$$

where the characteristic function given in (4.5)

$$p(\lambda,\, s)=\lambda^2+\lambda\frac{-3}{s+1}+\frac{5s+7}{(s+1)^2(s+2)^2} \tag{4.6}$$

is irreducible over the field of rational functions. However, it is still possible to plot the characteristic loci by calculating the characteristic values at each frequency numerically as given in Figure 4.4 and Figure 4.5.

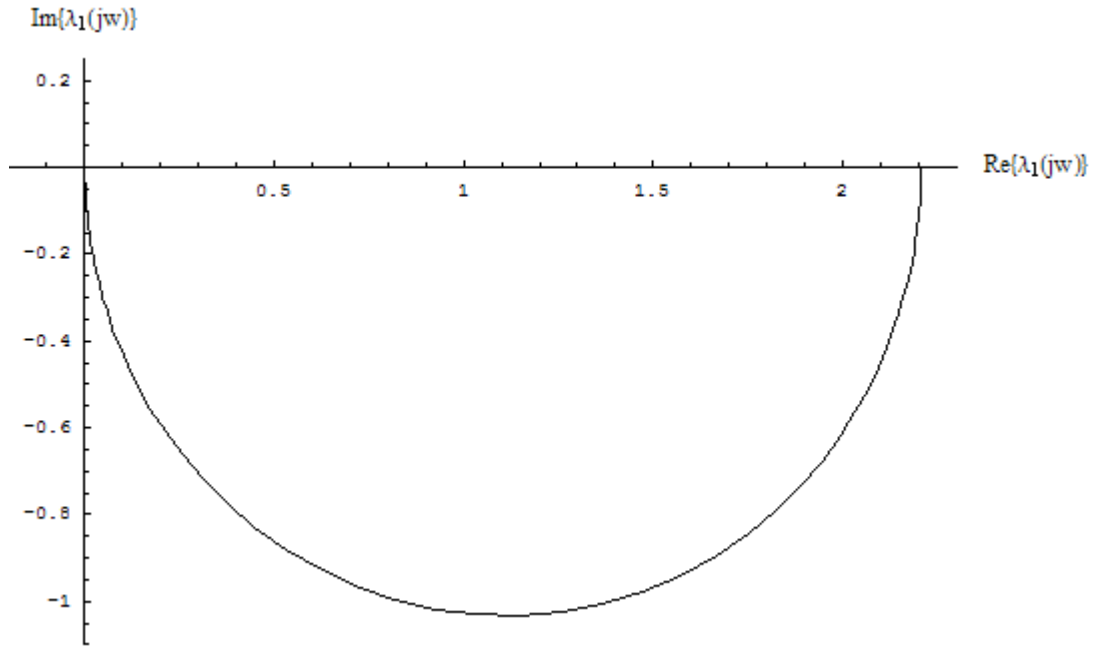

Figure 4.4 : Characteristic value plots for the system in Example in (4.5).

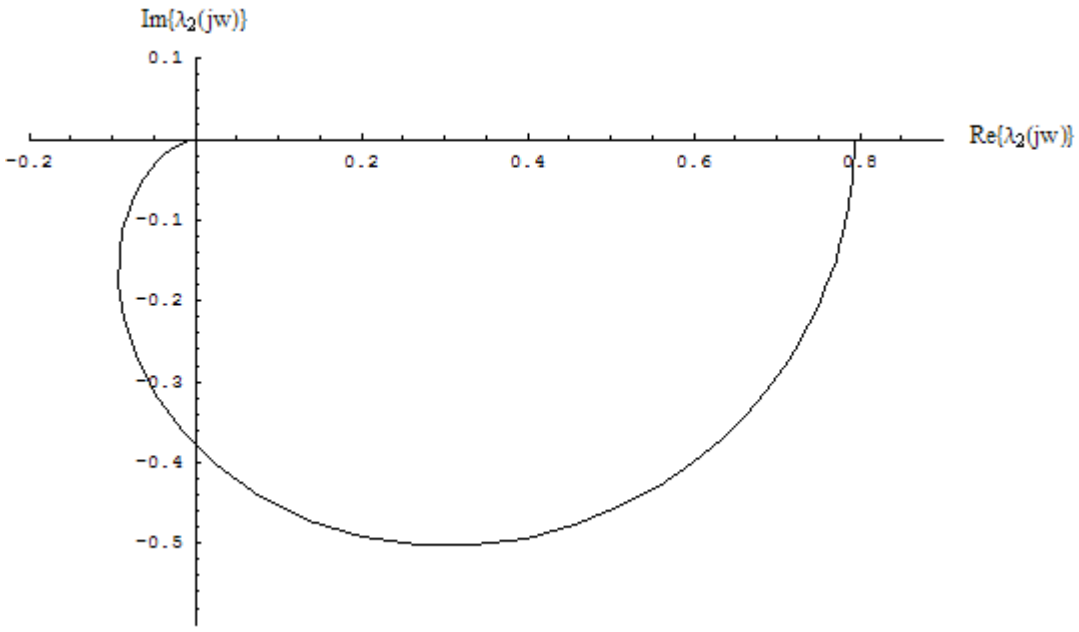

Figure 4.5 : Characteristic value plots for the system in Example in (4.5).

Note that, for a TITO system given as

$$G(s)=\begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \tag{4.7}$$

(4.1) is of the form

$$\lambda^2(s)-\tau(s)\lambda(s)+\delta(s)=0 \tag{4.8}$$

where

$$\tau(s)=g_{11}(s)+g_{22}(s) \tag{4.9}$$

and

$$\delta(s)=g_{11}(s)g_{22}(s)-g_{12}(s)g_{21}(s) \tag{4.10}$$

The characteristic values $\lambda_1(s)$ and $\lambda_2(s)$ are the solutions to (4.8). Recall also that the generalized Nyquist theorem requires that the net sum of encirclements of the point -1 by the characteristic values equal to the number of open-loop unstable poles of the system. Hence, it is of special importance to determine where the characteristic locus intersects with the real axis, i.e. where the imaginary part of $\lambda_i(jw)$, $i=1,2$ is

zero. The direction of these crossings is also important, because the net count of crossings at an intersection point will indicate whether there are closed-loop poles to cross the imaginary axis. Associated with the first and the second characteristic values, the set of frequencies for which the characteristic value plots cross the real axis at a point x_i is denoted by ${}_1\Omega_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,\mu}\}$ and ${}_2\Omega_i = \{w_{i,\mu+1}, w_{i,2}, \cdots, w_{i,\mu+\eta}\}$, respectively. Note that the intersection of these sets is not necessarily empty. If one of these sets is empty then it means that the corresponding characteristic value has no real axis crossing at the point x_i. For each of these frequencies the direction of the crossing is said to be positive and denoted by ${}_m d_{i,j} = 1$, $j = 1, 2, \cdots, \mu+\eta$, if the corresponding characteristic value plot crosses from the negative imaginary part of the complex plane to the positive imaginary half, and is said to be negative and denoted as ${}_m d_{i,j} = -1$, $j = 1, 2, \cdots, \mu+\eta$, $m = \{1, 2\}$, otherwise. The net count of crossings for the first characteristic value, ${}_1 c_i$, at a crossing point x_i is defined as

$$ {}_1 c_i = \sum_{j=1}^{\mu} {}_1 d_{i,j} \qquad \textbf{(4.11)} $$

The net count of crossings for the second characteristic value, ${}_2 c_i$, on the other hand is defined as

$$ {}_2 c_i = \sum_{j=\mu+1}^{\mu+\eta} {}_2 d_{i,j} \qquad \textbf{(4.12)} $$

and the final value of the crossings at this point is calculated as

$$ c_i = {}_1 c_i + {}_2 c_i \qquad \textbf{(4.13)} $$

Theorem 2: For a given TITO system $G(s)$ and a closed contour $\mathbb{C} = f(w)$, there exits a frequency w^* for which the imaginary part of one or both of the characteristic values is zero if, and only if, there exists a real gain $k^* = -\dfrac{1}{x_i}$ for which at least a pole of the closed loop system is on the contour $\mathbb{C}$. Furthermore, as the gain increases

around k^* the net number of poles moving from one side of the D-Stability region to the other side of this region is given by c_i.

Proof: Consider that the closed-loop characteristic equation is given by

$$\left|I + k^* G(s)\right| = 0 \qquad \textbf{(4.14)}$$

the closed-loop system has a pole on the border of the D-Stability region, $\mathbb{C}$, if and only if the characteristic equation is satisfied for a particular frequency, w^*, i.e.

$$\left|I + k^* G(f(w^*))\right| = 0 \qquad \textbf{(4.15)}$$

Hence, if the gain is given by

$$k^* = -\frac{1}{\lambda_i(f(w^*))} \qquad \textbf{(4.16)}$$

where $\lambda_i(.)$ is the characteristic value, we get

$$\left|\lambda_i(f(w^*))I - G(f(w^*))\right| = 0 \qquad \textbf{(4.17)}$$

Note that k^* is real if, and only if, the imaginary part of $\lambda_1(.)$ and $\lambda_2(.)$ is zero. In order to prove the second part of the theorem, the generalized Nyquist theorem is used.

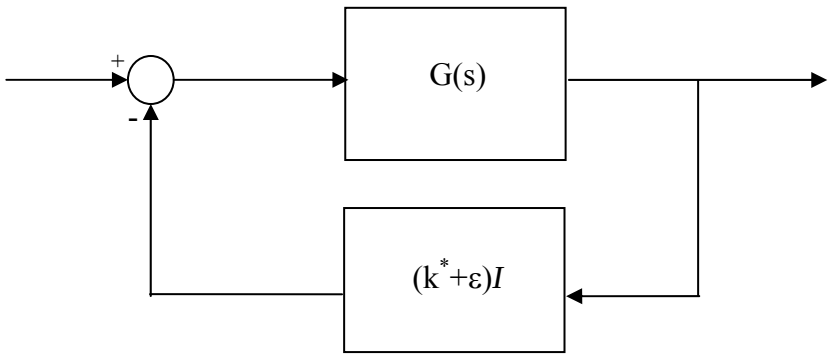

Figure 4.6 : Static output feedback with constant gain $k^*+\varepsilon$

Let N_1 denote the clockwise encirclements by the Nyquist plot of $G(s)$ around the point $-\frac{1}{k^* + \varepsilon}$. The number of $\mathbb{C}^+$ roots of $\left|I + (k^* + \varepsilon)G(s)\right|$, p_{c1}, is equal to u_0, the number of unstable roots of $G(s)$, plus N_1. Here, $\mathbb{C}^+$ denotes the closed right half plane $\{s : \mathrm{Re}\{s\} \geq 0\}$. That is

$$p_{c1} = u_0 + N_1 \qquad \textbf{(4.18)}$$

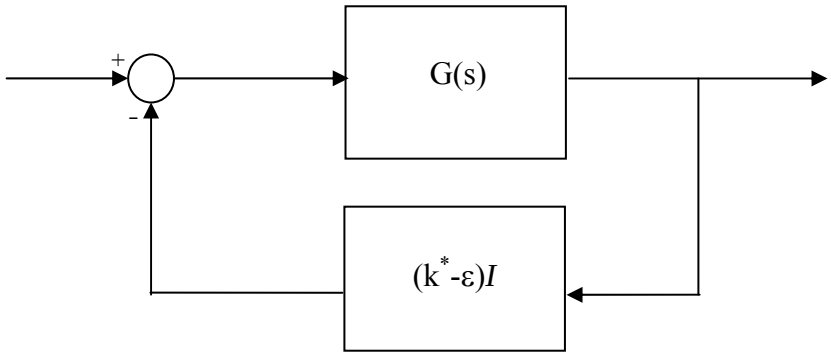

Figure 4.7 : Static output feedback with constant gain k^*-ε

Let N_2 denote the clockwise encirclements by the Nyquist plot of $G(s)$ around the point $-\frac{1}{k^* - \varepsilon}$. The number of $\mathbb{C}^+$ roots of $\left|I + (k^* - \varepsilon)G(s)\right|$, p_{c2}, is equal to the number of unstable roots of $G(s)$ plus N_2. That is

$$p_{c2} = u_0 + N_2 \qquad \textbf{(4.19)}$$

From $p_{c1} = u_0 + N_1$ and $p_{c2} = u_0 + N_2$,

$$p_{c1} - p_{c2} = N_1 - N_2 = c_i \qquad \textbf{(4.20)}$$

As k^* decreases from $k^* + \varepsilon$ (see Figure 4.6) to $k^* - \varepsilon$ (see Figure 4.7) the net change in the number of unstable closed-loop system poles is given by c_i. Note that the number of unstable closed-loop system poles remain unchanged as the gain varies within the interval $\left(-\frac{1}{x_{i-1}}, -\frac{1}{x_i}\right)$. Let us assume that the characteristic

values intersect with the real axis at q distinct points given with the list of ascending order $\{x_1, x_2, \cdots, x_q\}$. Defining $x_0 = -\infty$ and $x_{q+1} = \infty$ the real axis can be partitioned into $q+1$ intervals $\{I_1, I_2, \cdots, I_{q+1}\}$ with $I_i = [x_{i-1}, x_i)$. Hence, defining the intervals $K_i = \{k : \frac{-1}{x_{i-1}} < k < \frac{-1}{x_i}\}$ and denoting the number of unstable closed-loop poles for each interval K_i as u_i, it is possible to state the following lemma.

Lemma 1: For a given system $G(s)$ as defined above the number of unstable closed-loop system poles for gains in intervals K_i is given by

$$u_i = u_0 + \sum_{n=0}^{i-1} c_n \qquad \textbf{(4.21)}$$

where u_0 is the number of unstable open-loop system poles.

Proof: Due to previous theorem the number of unstable closed-loop poles in the j[th] interval is given by

$$u_i = u_{i-1} + c_{i-1} \qquad \textbf{(4.22)}$$

It is possible to observe that the number of unstable closed-loop system poles in the first interval is equal the number of unstable open-loop poles, u_0, plus the net count of positive crossings of the Nyquist plot at $-\infty$, i.e. c_0. This yields in

$$u_1 = u_0 + c_0 \qquad \textbf{(4.23)}$$

Hence, the recursive formula of $u_i = u_{i-1} + c_{i-1}$ is equivalent to $u_i = u_0 + \sum_{n=0}^{i-1} c_n$.

4.4 Constant Diagonal Controllers of Type diag(k, k) for TITO Systems

Consider a TITO process and a controller $K = diag(k, k)$ with negative feedback configuration, the closed-loop characteristic equation is of the form

$$p_c(s, k) = 1 + k\,\tau(s) + k^2\delta(s) = 0 \qquad \textbf{(4.24)}$$

where $\tau(s)$ and $\delta(s)$ are defined as in (4.9) and (4.10) respectively. The real and the imaginary parts of $p_c(jw, k) = 0$ are of the form

$$1 + k\mathrm{T}_R(w) + k^2\Delta_R(w) = 0 \quad \textbf{(4.25)}$$

$$k\mathrm{T}_I(w) + k^2\Delta_I(w) = 0 \quad \textbf{(4.26)}$$

with

$$\Delta_R(w) = \mathrm{Re}[\delta(jw)] \quad \textbf{(4.27)}$$

$$\Delta_I(w) = \mathrm{Im}[\delta(jw)] \quad \textbf{(4.28)}$$

$$T_R(w) = \mathrm{Re}\{\tau(jw)\} \quad \textbf{(4.29)}$$

$$T_I(w) = \mathrm{Im}\{\tau(jw)\} \quad \textbf{(4.30)}$$

From (4.26), we immediately observe that $k = 0$ or

$$k = -\frac{\mathrm{T}_I(w)}{\Delta_I(w)} \quad \textbf{(4.31)}$$

If we substitute (4.31) into (4.25) we get

$$1 - \frac{\mathrm{T}_I(w)}{\Delta_I(w)}\mathrm{T}_R(w) + \frac{\mathrm{T}_I(w)^2}{\Delta_I(w)^2}\Delta_R(w) = 0 \quad \textbf{(4.32)}$$

which yields in

$$q(w) = \Delta_I(w)^2 - \Delta_I(w)\mathrm{T}_I(w)\mathrm{T}_R(w) + \mathrm{T}_I(w)^2\Delta_R(w) = 0 \quad \textbf{(4.33)}$$

Solving (4.33) in real w's results in the critical frequencies where the closed-loop system has pole or poles on the imaginary axis, it is possible to name these frequencies as crossing frequencies. The corresponding gains can be determined by substituting these frequencies into (4.31). For $w = 0$ it is sufficient to solve

$$1 + k\mathrm{T}_R(0) + k^2\Delta_R(0) = 0 \quad \textbf{(4.34)}$$

for real k. As an example let us consider the process by Niederlinski, [102]

$$G(s)=\begin{bmatrix} \dfrac{0.5}{(0.1s+1)^2(0.2s+1)^2} & \dfrac{-1}{(0.1s+1)(0.2s+1)^2} \\ \dfrac{1}{(0.1s+1)(0.2s+1)^2} & \dfrac{2.4}{(0.1s+1)(0.5s+1)(0.2s+1)^2} \end{bmatrix} \tag{4.35}$$

within the negative feedback configuration, and find the critical frequencies by solving (4.33). These frequencies are determined as $\{0.83085, 4.30431\}$ and the corresponding gains are obtained as $\{-1.05249, 1.1352\}$. For $w=0$ we have a complex k, hence we omit it. If we consider the process, which is slightly different than (4.35), where the numerator of $g_{22}(s)$ is replaced by 2.7

$$G(s)=\begin{bmatrix} \dfrac{0.5}{(0.1s+1)^2(0.2s+1)^2} & \dfrac{-1}{(0.1s+1)(0.2s+1)^2} \\ \dfrac{1}{(0.1s+1)(0.2s+1)^2} & \dfrac{2.7}{(0.1s+1)(0.5s+1)(0.2s+1)^2} \end{bmatrix} \tag{4.36}$$

The critical frequencies are calculated as $\{0.503197, 4.32873\}$ and corresponding gains are $\{-0.996988, 1.08832\}$. For $w=0$ we have $k=-0.485848$ and $k=-0.875854$. The case of reducible characteristic equation

4.4.1 The case of reducible characteristic equation

Let the characteristic values of the TITO system, $\lambda_1(s)$ and $\lambda_2(s)$, be of the form $\dfrac{N_1(s)}{D_1(s)}$ and $\dfrac{N_2(s)}{D_2(s)}$, respectively. The polynomials $N_i(s)$ and $D_i(s)$ are defined as follows

$$N_i(s)=a_{i,m}s^m+a_{i,m-1}s^{m-1}+\cdots+a_{i,1}s+a_{i,0} \tag{4.37}$$

$$D_i(s)=b_{i,n}s^n+b_{i,n-1}s^{n-1}+\cdots+b_{i,1}s+b_{i,0} \tag{4.38}$$

with $m<n$. This kind of formulation of the characteristic values means that we have a reducible characteristic equation. Furthermore, assume that the open-loop system has no imaginary axis poles. Let us consider $\lambda_1(jw)$ and $\lambda_2(jw)$;

$$\lambda_1(jw) = \frac{N_1(jw)}{D_1(jw)} = \frac{N_{1,re} + j\,N_{1,im}}{D_{1,re} + j\,D_{1,im}} \tag{4.39}$$

$$\lambda_2(jw) = \frac{N_2(jw)}{D_2(jw)} = \frac{N_{2,re} + j\,N_{2,im}}{D_{2,re} + j\,D_{2,im}} \tag{4.40}$$

where $D_{1,re} = \mathrm{Re}\{D_1(jw)\}$, $D_{1,im} = \mathrm{Im}\{D_1(jw)\}$ and $N_{1,re}$, $N_{1,im}$, $N_{2,re}$, $N_{2,im}$, $D_{2,re}$, $D_{2,im}$ are defined similarly. Note that

$$D_{i,re} = D_{i,e}(-w^2) \tag{4.41}$$

$$D_{i,im} = D_{i,o}(-w^2)\,w \tag{4.42}$$

$$N_{i,re} = N_{i,e}(-w^2) \tag{4.43}$$

$$N_{i,im} = N_{i,o}(-w^2)\,w \tag{4.44}$$

with $D_i(jw) = D_{i,e}(-w^2) + jwD_{i,o}(-w^2)$ and $N_i(jw) = N_{i,e}(-w^2) + jw\,N_{i,o}(-w^2)$, $i = \{1, 2\}$. It is possible to write

$$\lambda_1(jw) = \frac{N_1(jw)}{D_1(jw)} = \frac{N_{1,re} + j\,N_{1,im}}{D_{1,re} + j\,D_{1,im}} = \frac{X_1(w^2)}{Z_1(w^2)} + jw\frac{Y_1(w^2)}{Z_1(w^2)} \tag{4.45}$$

$$\lambda_2(jw) = \frac{N_2(jw)}{D_2(jw)} = \frac{N_{2,re} + j\,N_{2,im}}{D_{2,re} + j\,D_{2,im}} = \frac{X_2(w^2)}{Z_2(w^2)} + jw\frac{Y_2(w^2)}{Z_2(w^2)} \tag{4.46}$$

with

$$X_i(w^2) = D_{i,e}N_{i,e} + w^2\,D_{i,o}N_{i,o} \tag{4.47}$$

$$Y_i(w^2) = D_{i,e}N_{i,o} - D_{i,o}N_{i,e} \tag{4.48}$$

$$Z_i(w^2) = D_{i,e}^2 + w^2\,D_{i,o}^2 \tag{4.49}$$

where $D_{i,e}$, $D_{i,o}$, $N_{i,e}$ and $N_{i,o}$ are used as abbreviations for $D_{i,e}(-w^2)$, $D_{i,o}(-w^2)$, $N_{i,e}(-w^2)$, $N_{i,o}(-w^2)$, respectively.

Let us denote $v = w^2$, and the positive real roots of $Y_1(v)$ as $v_1, v_2, \cdots, v_\mu$ and the positive real roots of $Y_2(v)$ as $v_{\mu+1}, v_{\mu+2}, \cdots, v_{\mu+\eta}$. Note that the polar plot of each characteristic value crosses the real axis only if $w = 0$, $w = \infty$, or $w = \pm\sqrt{v_m}$ for $m = 1, 2, \cdots, \mu + \eta$. The real axis crossing points are found as

$$ {}_1x_i = \frac{X_1(v_m)}{Z_1(v_m)} \qquad \textbf{(4.50)} $$

for $m = 1, 2, \cdots, \mu$ and

$$ {}_2x_i = \frac{X_2(v_m)}{Z_2(v_m)} \qquad \textbf{(4.51)} $$

for $m = \mu + 1, \mu + 2, \cdots, \mu + \eta$ with $i = 1, 2, \cdots, q$.

Theorem 3: Consider a linear time invariant TITO system by a proper rational transfer function matrix $G(s)$ and characteristic values $\lambda_1(s)$ and $\lambda_2(s)$, and assume that the system has no roots on the imaginary axis. Let $X_i(w^2)$, $Y_i(w^2)$, and $Z_i(w^2)$ ($i = \{1, 2\}$) be polynomials as defined in (4.47), (4.48) and (4.49). Let us denote the first coefficient of $Y_1(v)$ and $Y_2(v)$ as $y_{1,1}$ and $y_{2,1}$, the last nonzero coefficient of $Y_1(v)$ and $Y_2(v)$ as $y_{1,0}$ and $y_{2,0}$, respectively. For a given gain $k \in K_i = \left(\frac{-1}{x_{i-1}}, \frac{-1}{x_i} \right)$ the number of unstable poles of the closed-loop system is given by (4.21), and the direction of the crossings is calculated as

$$ {}_md_{i,j} = \begin{cases} (1-(-1)^l)\,\mathrm{sgn}(Y_m^{(l)}(v_j)) & \text{if } \ 0 < v_j < \infty \\ \mathrm{sgn}(y_{m,0}) & \text{if } \ v_j = 0 \\ -\mathrm{sgn}(y_{m,1}) & \text{if } \ v_j = \infty \end{cases} \qquad \textbf{(4.52)} $$

in which $Y_m^{(l)}(v)$ is the first nonzero derivative of $Y_m(v)$ at the point v_j with $m = \{1, 2\}$.

Proof: Let us consider , and define the functions $H_i(v)$ as

$$H_i(v) = \sqrt{v}\,\frac{Y_i(v)}{Z_i(v)} \tag{4.53}$$

Assume that for a given frequency $w_{i,j} = \sqrt{v_{i,j}}$, $H_1(v) = 0$ and/or $H_2(v) = 0$. For a given small positive real number, ε, where $0 < \varepsilon << 1$ determine $H_i(v_{i,j} + \varepsilon)$ and $H_i(v_{i,j} - \varepsilon)$. If $\text{sgn}(H_i(v_{i,j} + \varepsilon)) \neq \text{sgn}(H_i(v_{i,j} - \varepsilon))$ there exists a net crossing at the frequency $w_{i,j}$ and the sign of the crossing is given by $\text{sgn}(H_i(v_{i,j} + \varepsilon))$. Note that there exists a similar crossing at $-w_{i,j}$, hence the number of net crossing is $d_{i,j} = 2\,\text{sgn}(H_i(v_{i,j} + \varepsilon))$. If $\text{sgn}(H_i(v_{i,j} + \varepsilon)) = \text{sgn}(H_i(v_{i,j} - \varepsilon))$ there does not exist a net crossing. It is possible to write

$$d_{i,j} = \text{sgn}(H_i(v_{i,j} + \varepsilon)) - \text{sgn}(H_i(v_{i,j} - \varepsilon)) \tag{4.54}$$

For a given polynomial $Y_i(v)$,

$$Y_i(v + \varepsilon) = Y_i(v) + \sum_{j=1}^{\infty} \frac{Y_i^{(j)}(v)\,\varepsilon^j}{j!} \tag{4.55}$$

where $Y_i^{(l)}(v)$ denotes the l^{th} derivative of the $Y_i(v)$ with respect to v. It is obvious that $Z_i(v_{i,j}) > 0$, therefore $H_i(v_{i,j}) = 0$ if, and only if $Y_i(v_{i,j}) = 0$. Assume that the first $(l-1)$ derivatives of $Y_i(v)$ are zero at $v_{i,j}$, $Y_i(v_{i,j} + \varepsilon)$ and $Z_i(v_{i,j} + \varepsilon)$ can be approximated by

$$Y_i(v_{i,j} + \varepsilon) \cong \frac{1}{l!} Y_i^{(l)}(v_{i,j})\,\varepsilon^l \tag{4.56}$$

$$Z_i(v_{i,j} + \varepsilon) \cong Z_i(v_{i,j}) \tag{4.57}$$

This yields in

$$H_i(v_{i,j} + \varepsilon) \cong \sqrt{v_{i,j}}\,\frac{\frac{1}{l!} Y_i^{(l)}(v_{i,j})\,\varepsilon^l}{Z_i(v_{i,j})} \tag{4.58}$$

Therefore,

$$\text{sgn}(H_i(\nu_{i,j}+\varepsilon)) = \text{sgn}(Y_i^{(l)}(\nu_{i,j})) \tag{4.59}$$

$$\text{sgn}(H_i(\nu_{i,j}-\varepsilon)) = (-1)^l \,\text{sgn}(Y_i^{(l)}(\nu_{i,j})) \tag{4.60}$$

We can immediately conclude that

$$d_{i,j} = (1-(-1)^l)\,\text{sgn}(Y_i^{(l)}(\nu_{i,j})) \tag{4.61}$$

Let us consider the case $\nu_{i,j}=0$. Since the characteristic value plot is symmetric about the real axis, there always exists a net crossing for $\nu_{i,j}=0$. The direction of this crossing is equal to $\text{sgn}(\lambda_i(j\varepsilon))$, where ε is a small positive number. Then

$$\text{Im}(\lambda_i(j\varepsilon)) = \varepsilon\frac{Y(\varepsilon^2)}{Z(\varepsilon^2)} \tag{4.62}$$

Note that $Z_i(\varepsilon^2) \cong Z_i(0) = b_{i,0}^2 > 0$ and $\text{sgn}(Y_i(\varepsilon^2)) = \text{sgn}(y_{i,0})$. Therefore,

$$d_{i,j} = \text{sgn}(\text{Im}(\lambda_i(j\varepsilon))) = \text{sgn}(y_{i,0}) \tag{4.63}$$

Finally, let us consider $\nu_{i,j}=\infty$. The semi-circular part of the Nyquist contour can be parameterized as $s = R\,e^{j\theta}$, where R goes to infinity and θ changes from $\frac{\pi}{2}$ to $-\frac{\pi}{2}$. Noting that for $\theta=\frac{\pi}{2}$

$$\text{sgn}(\text{Im}(\lambda_i(R\,e^{j\frac{\pi}{2}}))) = \text{sgn}(\text{Im}(\lambda_i(j\,R))) = \text{sgn}(Y_i(R^2)) \tag{4.64}$$

and for $\theta=-\frac{\pi}{2}$

$$\text{sgn}(\text{Im}(\lambda_i(R\,e^{j\frac{\pi}{2}}))) = \text{sgn}(\text{Im}(\lambda_i(j\,R))) = \text{sgn}(Y_i(R^2)) \tag{4.65}$$

the direction of the crossing is given as $d_{i,j} = -\text{sgn}(Y_i(R^2))$. Note that, when R goes to infinity the sign of $Y_i(R^2)$ is the same as that of the first coefficient $y_{i,1}$ of the polynomial $Y_i(R^2)$. Hence, $d_{i,j} = -\text{sgn}(y_{i,1})$.

4.4.1.1 Example

$$\begin{bmatrix} \frac{7s^5 - 9s^4 - 10s^3 - 348s^2 - 808s - 1132}{(s^2+2s+2)^2(s^2+4s+5)} & \frac{-3s^5 + 13s^4 - 10s^3 + 260s^2 + 608s + 832}{(s^2+2s+2)^2(s^2+4s+5)} \\ \frac{6s^5 - 26s^4 + 20s^3 - 520s^2 - 1216s - 1664}{(s^2+2s+2)^2(s^2+4s+5)} & \frac{-2s^5 + 30s^4 - 40s^3 + 432s^2 + 1016s + 1364}{(s^2+2s+2)^2(s^2+4s+5)} \end{bmatrix} \quad \textbf{(4.66)}$$

The characteristic value plots (Nyquist plots for each characteristic value) of the system is as shown in Figure 4.8 and Figure 4.9, where the characteristic values are

$$\lambda_1(s) = \frac{4(s^3 - 3s^2 + 2s - 15)}{(s^2 + 2s + 2)^2} \quad \textbf{(4.67)}$$

$$\lambda_2(s) = \frac{s^3 + 15s^2 - 62s + 266}{s^4 + 6s^3 + 15s^2 + 18s + 10} \quad \textbf{(4.68)}$$

with,

$$N_1(s) = 4s^3 - 12s^2 + 8s - 60 \quad \textbf{(4.69)}$$

$$D_1(s) = s^4 + 4s^3 + 8s^2 + 8s + 4 \quad \textbf{(4.70)}$$

and,

$$N_2(s) = s^3 + 15s^2 - 62s + 266 \quad \textbf{(4.71)}$$

$$D_2(s) = s^4 + 6s^3 + 15s^2 + 18s + 10 \quad \textbf{(4.72)}$$

If the even-odd decompositions are applied to the numerator and denominator polynomials, we have,

$$D_{1,e}(-w^2) = w^4 - 8w^2 + 4 \quad \textbf{(4.73)}$$

$$D_{1,o}(-w^2) = -4w^2 + 8 \quad \textbf{(4.74)}$$

$$N_{1,e}(-w^2) = 12w^2 - 60 \quad \textbf{(4.75)}$$

$$N_{1,o}(-w^2) = -4w^2 + 8 \quad \textbf{(4.76)}$$

and,

$$D_{2,e}(-w^2) = w^4 - 15w^2 + 10 \quad \textbf{(4.77)}$$

$$D_{2,o}(-w^2) = -6w^2 + 18 \quad \textbf{(4.78)}$$

$$N_{2,e}(-w^2) = -15w^2 + 266 \quad \textbf{(4.79)}$$

$$N_{1,o}(-w^2) = -w^2 - 62 \quad \textbf{(4.80)}$$

From these equation we get

$$X_1(w^2) = 28w^6 - 220w^4 + 592w^2 - 240 \quad \textbf{(4.81)}$$

$$Y_1(w^2) = -4w^6 + 88w^4 - 416w^2 + 512 \quad \textbf{(4.82)}$$

$$Z_1(w^2) = w^8 + 8w^4 + 16 \quad \textbf{(4.83)}$$

and,

$$X_2(w^2) = -9w^6 + 845w^4 - 5256w^2 + 2660 \quad \textbf{(4.84)}$$

$$Y_2(w^2) = -w^6 - 137w^4 + 2786w^2 - 5408 \quad \textbf{(4.85)}$$

$$Z_2(w^2) = w^8 + 6w^6 + 29w^4 + 24w^2 + 100 \quad \textbf{(4.86)}$$

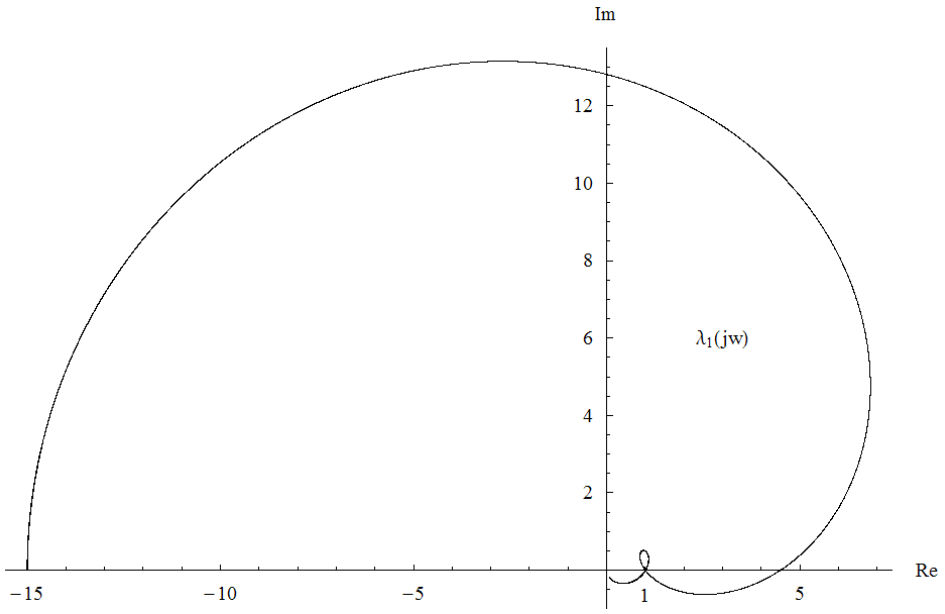

Figure 4.8 : First characteristic value plot for the system in Example in 4.4.1.1.

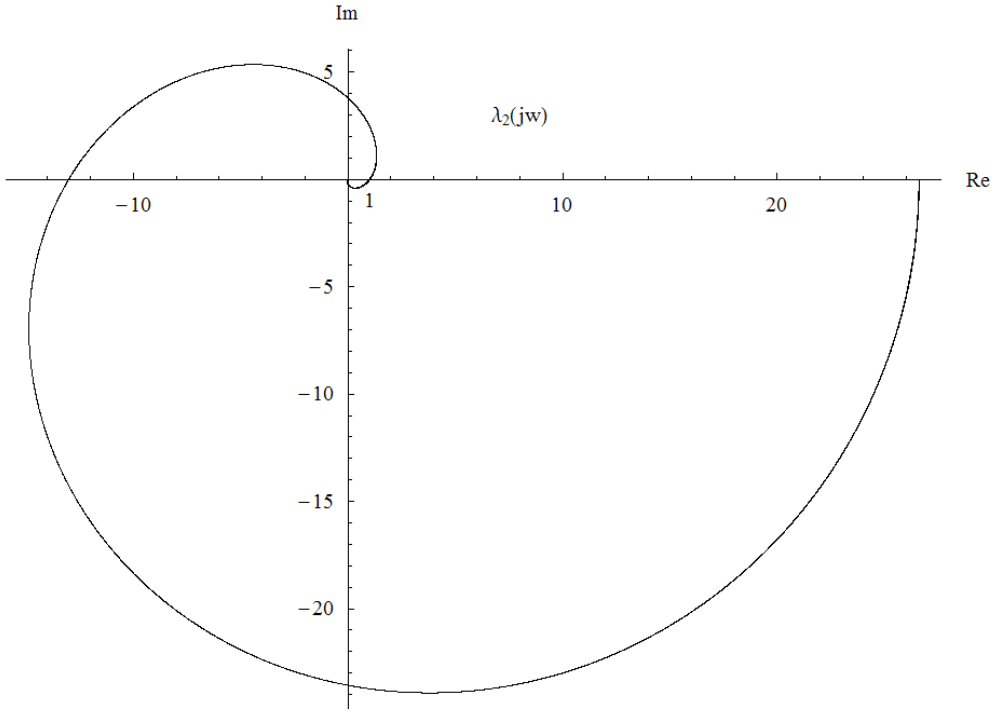

Figure 4.9 : Second characteristic value plot for the system in Example in 4.4.1.1.

Remembering $w = \sqrt{v}$, recall that

$$X_1(v) = 28v^3 - 220v^2 + 592v - 240 \qquad \textbf{(4.87)}$$

$$Y_1(v) = -4v^3 + 88v^2 - 416v + 512 \qquad \textbf{(4.88)}$$

$$Z_1(v) = v^4 + 8v^2 + 16 \quad \textbf{(4.89)}$$

$$X_2(v) = -9v^3 + 845v^2 - 5256v + 2660 \quad \textbf{(4.90)}$$

$$Y_2(v) = -v^3 - 137v^2 + 2786v - 5408 \quad \textbf{(4.91)}$$

$$Z_2(v) = v^4 + 6v^3 + 29v^2 + 24v + 100 \quad \textbf{(4.92)}$$

The positive real roots of $Y_1(v)$ are $v_1 = 2$, $v_2 = 4$ and $v_3 = 16$. The crossing points for these frequencies are given by $x_1 = 4.5$, $x_2 = 1$ and $x_3 = 1$. The last and the first coefficients of $Y_1(v)$ are $y_{1,0} = 512$ and $y_{1,1} = -4$, respectively. Note that $Y_1'(2) = -112$, $Y_1'(4) = 96$ and $Y_1'(16) = -672$. The positive real roots of $Y_2(v)$ are $v_4 = 2.1781$ and $v_5 = 16$. The crossing points to these frequencies are given by $x_4 = -13.01$ and $x_5 = 1$. The last and the first coefficients of $Y_2(v)$ are $y_{2,0} = -5408$ and $y_{2,1} = -1$, respectively. Note that $Y_2'(2.1781) = 2174.96$ and $Y_2'(16) = -2366$. The net crossing counts are calculated as given in Table 4.1.

Table 4.1 : Critical frequencies, locations, directions, net crossing counts

i	v_i	${}_1x_i = \frac{X_1(v_i)}{Z_1(v_i)}$	${}_2x_i = \frac{X_2(v_i)}{Z_2(v_i)}$	${}_1d_{i,j}$	${}_2d_{i,j}$	c_i
1	0	-15	26.6	1	-1	0
2	2	4.5	-	-2	-	-2
3	2.1781	-	-13.01	-	2	2
4	4	1	-	2	-	2
5	16	1	1	-2	-2	-4
6	∞	0	0	1	1	2

Forming I_i and K_i the number of root-invariant closed-loop poles are calculated.

Table 4.2 : Decision table for stability; root-invariant intervals and critical gains.

	I_i		K_i			
i	x_{i-1}	x_i	$\frac{-1}{x_{i-1}}$	$\frac{-1}{x_i}$	u_i	Stable
1	$-\infty$	-15	0^+	0.0666	0	True
2	-15	-13.01	0.0666	0.0768	1	False
3	-13.01	0^-	0.0768	∞	3	False
4	0^+	1	$-\infty$	-1	5	False
5	1	4.5	-1	-0.2222	3	False
6	4.5	26.6	-0.2222	-0.0376	1	False
7	26.6	∞	-0.0376	0^-	0	True

An examination of Table 4.2 reveals that the closed-loop system is only stable for gains $k \in (-0.0376, 0.0666)$.

4.4.2 The case of irreducible characteristic equation

The characteristic values λ_1 and λ_2 can be calculated as

$$\lambda_{1,2}(s) = \frac{\tau(s) \pm \sqrt{\tau^2(s) - 4\delta(s)}}{2} = \frac{\tau(s)}{2} \pm \sqrt{\frac{\tau^2(s)}{4} - \delta(s)} \qquad \textbf{(4.93)}$$

provided that $\sqrt{\frac{\tau^2(s)}{4} - \delta(s)}$ is a rational polynomial in s. Otherwise the characteristic equation is irreducible in λ. Recall that, given an $n-$ square transfer function matrix it is possible to reduce it to its irreducible rational canonical form using the method proposed in [14]. However, if we replace s with jw we have

$$\lambda_1(jw) = \frac{\tau(jw)}{2} + \sqrt{\frac{\tau^2(jw)}{4} - \delta(jw)} = A_1(w) + j\,A_2(w) \qquad \textbf{(4.94)}$$

$$\lambda_2(jw) = \frac{\tau(jw)}{2} - \sqrt{\frac{\tau^2(jw)}{4} - \delta(jw)} = B_1(w) + j\,B_2(w) \qquad \textbf{(4.95)}$$

where,

$$A_1(w) = \frac{1}{2}\left[T_R(w) + D_R(w)\right] \quad \textbf{(4.96)}$$

$$B_1(w) = \frac{1}{2}\left[T_R(w) - D_R(w)\right] \quad \textbf{(4.97)}$$

$$A_2(w) = \frac{1}{2}\left[T_I(w) + D_I(w)\right] \quad \textbf{(4.98)}$$

$$B_2(w) = \frac{1}{2}\left[T_I(w) - D_I(w)\right] \quad \textbf{(4.99)}$$

with $T_R(w)$ and $T_I(w)$ defined in (4.29) and (4.30), respectively. Furthermore,

$$D_R(w) = \mathrm{Re}\left\{\sqrt{\tau^2(jw) - 4\delta(jw)}\right\} \quad \textbf{(4.100)}$$

$$D_I(w) = \mathrm{Im}\left\{\sqrt{\tau^2(jw) - 4\delta(jw)}\right\} \quad \textbf{(4.101)}$$

The critical frequencies w^* for the first and the second characteristic values are the frequencies which are the solutions to

$$A_2(w) = 0, \quad \textbf{(4.102)}$$

and,

$$B_2(w) = 0 \quad \textbf{(4.103)}$$

respectively. The real axis crossing points are

$$_1x_i = A_1(w^*) \quad \textbf{(4.104)}$$

and,

$$_2x_i = B_1(w^*) \quad \textbf{(4.105)}$$

The direction of crossings are calculated as follows

$$ {}_1d_{i,j}(w^*) = \frac{\partial}{\partial w}(A_2(w))\Big|_{w=w^*} \quad \textbf{(4.106)} $$

$$ {}_2d_{i,j}(w^*) = \frac{\partial}{\partial w}(B_2(w))\Big|_{w=w^*} \quad \textbf{(4.107)} $$

4.4.2.1 Example

Let us consider the process by Niederlinski [102] where the characteristic equation

$$ \Delta(\lambda, s) = \lambda^2 + \lambda \frac{-50(49s+290)}{(s^2+15s+50)^2(s+2)} + \frac{62500(s^2+12s+44)}{(s+10)^3(s+5)^4(s+2)} \quad \textbf{(4.108)} $$

which is irreducible in λ over the field of rational polynomials. Recall that the critical frequencies are $\{0.83085, 4.30431\}$ as mentioned previously. Table 4.3 helps us to better understand how the proposed algorithm works:

Table 4.3 : Critical frequencies, locations, directions, net crossing counts

w^*	$A_2(jw^*)$	$A_1(jw^*)$	$B_2(jw^*)$	$B_1(jw^*)$	${}_1d_{i,j}$	${}_2d_{i,j}$	c_i
0	$\neq 0$	-	$\neq 0$	-	-	-	-
0.83085	$\neq 0$	-	0	0.950126	-	-1	-2
4.30431	$\neq 0$	-	0	-0.880906	-	+1	2
∞	0	0	0	0	-1	+1	0

Table 4.4 : Decision table for stability; root-invariant intervals and critical gains.

	I_i		K_i			
i	x_{i-1}	x_i	$\frac{-1}{x_{i-1}}$	$\frac{-1}{x_i}$	u_i	Stable
1	$-\infty$	-0.880906	0^+	1.1352	0	True
2	-0.880906	0^-	1.1352	$+\infty$	2	False
3	0^+	0.950126	$-\infty$	-1.05249	2	False
4	0.950126	$+\infty$	-1.05249	0^-	0	True

Table 4.4 indicates that the system is stable for $k \in (-1.05249, 1.1352)$.

4.4.3 Algorithms

The Algorithm 1 is the algorithm to determine the stabilizing intervals for constant feedback controllers for TITO systems where the characteristic polynomial is reducible, it is direct application of Theorem 3.

Algorithm 1

Step 1: Find the frequencies w^*, satisfying (4.33).

Step 2: Calculate the points x_i using (4.50) and (4.51).

Step 3: Relabel x_i such that $x_{i-1} < x_i$.

Step 4: Relabel the frequencies that yield the same x_i as $\{w_{i,1}, w_{i,2}, \cdots, w_{i,\mu}\}$ corresponding to the first characteristic value and as $\{w_{i,\mu+1}, w_{i,2}, \cdots, w_{i,\mu+\eta}\}$ corresponding to the second characteristic value.

Step 5: Find the directions of the crossings ${}_1d_{i,j}$ at x_i using (4.52) and calculate ${}_1c_i$ given in (4.11).

Step 6: Find the directions of the crossings ${}_2d_{i,j}$ at x_i using (4.52) and calculate ${}_2c_i$ given in (4.12).

Step 7: Find the total number of crossings as $c_i = {}_1c_i + {}_2c_i$.

Step 8: Form the intervals I_i and K_i, for each of these calculate the number of unstable poles u_i by using (4.21).

Step 9: Return the intervals, if there are any, K_i for which $u_i = 0$ as the stabilizing intervals.

Algorithm 2 is suitable for irreducible case.

Algorithm 2

Step 1: Find the frequencies w^*, satisfying (4.102) and (4.103).

Step 2: Calculate the points x_i using (4.104) and (4.105).

Step 3: Relabel x_i such that $x_{i-1} < x_i$.

Step 4: Relabel the frequencies that yield the same x_i as $\{w_{i,1}, w_{i,2}, \cdots, w_{i,\mu}\}$ corresponding to the first characteristic value and as $\{w_{i,\mu+1}, w_{i,2}, \cdots, w_{i,\mu+\eta}\}$ corresponding to the second characteristic value.

Step 5: Find the directions of the crossings ${}_1d_{i,j}$ at x_i using (4.106) and calculate ${}_1c_i$ given in (4.11).

Step 6: Find the directions of the crossings ${}_2d_{i,j}$ at x_i using (4.107) and calculate ${}_2c_i$ given in (4.12).

Step 7: Find the total number of crossings as $c_i = {}_1c_i + {}_2c_i$.

Step 8: Form the intervals I_i and K_i, for each of these calculate the number of unstable poles u_i by using (4.21).

Step 9: Return the intervals, if there are any, K_i for which $u_i = 0$ as the stabilizing intervals.

For each characteristic value the direction of the crossings can be calculated more easily. Depending on the multiplicity of the critical frequencies a table for the direction of crossings can be constructed. Assume that the imaginary part of one of the characteristic values is zero at a couple of frequencies as given in Figure 4.10. Note that only the positive frequencies are considered. If the frequencies are sorted such that $w_i < w_{i+1}$ and if the direction of the crossing is known for a particular frequency, we can immediately determine the direction of the crossings for preceding and/or subsequent frequency without any calculation. As can be observed from Figure 4.10 the direction of the crossing for w_1 is positive, the multiplicities of w_2 and w_3 are odd and even, respectively. As a result the direction of the crossing at the point corresponding to w_2 has to be negative, and for w_3 there is no crossing.

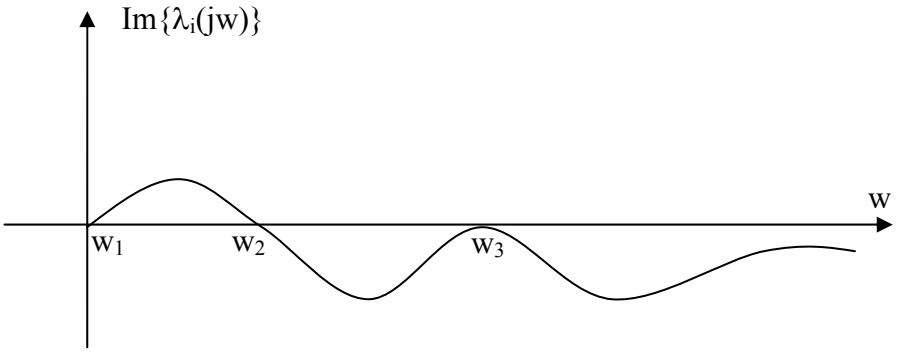

Figure 4.10 : Direction of crossings at particular frequencies

4.5 Stabilizing Constant Diagonal Controllers in Parameter Space

Consider a two input two output (TITO) process given by the transfer matrix $G(s)$. The process is to be controlled in a negative feedback configuration by the static controller:

$$K = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \tag{4.109}$$

with $k_1, k_2 \in \mathbb{R}$. The characteristic equation $\det(I + G(s)K)$ is of the form

$$p_c(s) = 1 + g_{11}(s)k_1 + g_{22}(s)k_2 + \det(G(s))k_1k_2 = 0 \tag{4.110}$$

The boundary of the stability domain is given by the equation

$$p_c(jw, k_1, k_2) = 0 \tag{4.111}$$

Note that the real and the imaginary part of (4.111) are two equations in two variables (k_1 and k_2) and the parameter w, that are obtained as follows

$$\mathrm{Re}\{p_c(jw)\} = 1 + \alpha_1(w)k_1 + \alpha_2(w)k_2 + \alpha_{12}(w)k_1k_2 = 0 \tag{4.112}$$

$$\mathrm{Im}\{p_c(jw)\} = \beta_1(w)k_1 + \beta_2(w)k_2 + \beta_{12}(w)k_1k_2 = 0 \tag{4.113}$$

where,

$$\alpha_1(w) = \mathrm{Re}\{g_{11}(jw)\} \tag{4.114}$$

$$\alpha_2(w) = \mathrm{Re}\{g_{22}(jw)\} \tag{4.115}$$

$$\alpha_{12}(w) = \mathrm{Re}\{\det(G(jw))\} \tag{4.116}$$

$$\beta_1(w) = \mathrm{Im}\{g_{11}(jw)\} \tag{4.117}$$

$$\beta_2(w) = \mathrm{Im}\{g_{22}(jw)\} \tag{4.118}$$

$$\beta_{12}(w) = \mathrm{Im}\{\det(G(jw))\} \tag{4.119}$$

Note that for $w = 0$, $G(jw)$ is real and therefore the following equation

$$1 + \alpha_1(0)\, k_1 + \alpha_2(0)\; k_2 + \alpha_{12}(0)\; k_1 k_2 = 0 \tag{4.120}$$

which defines a hyperbola provided that $\det(G(0)) \neq 0$ (otherwise it is a straight line) is obtained. For real $w \neq 0$ the set of equations (4.112) and (4.113) are solved and hence a parametric curve for $w \in (-\infty, \infty)$ is defined. This curve divides the (k_1, k_2)–plane into regions in which the number of stable and unstable roots of $p_c(jw, k_1, k_2)$ remains unchanged. It is not difficult to show that these equations have no real solution (k_1, k_2) if

$$\Delta(w) = (\alpha_2\beta_1 - \alpha_1\beta_2 + \beta_{12})^2 - 4\beta_1(\alpha_2\beta_{12} - \alpha_{12}\beta_2) \tag{4.121}$$

is negative. Assume that w^* is real and satisfies $\Delta(w) < 0$. It is immediately possible to conclude that there exist no pair of (k_1, k_2) such that a crossing at that frequency occurs.

4.5.1.1 Example

Let us consider the process given in (4.35) within the negative feedback configuration. We shall determine the set of all stabilizing controllers in form (4.109). Straightforward calculations result in a region in the (k_1, k_2)–plane (Figure

4.11), which ensures that for any gain pairs chosen within this region the closed-loop system will be stable.

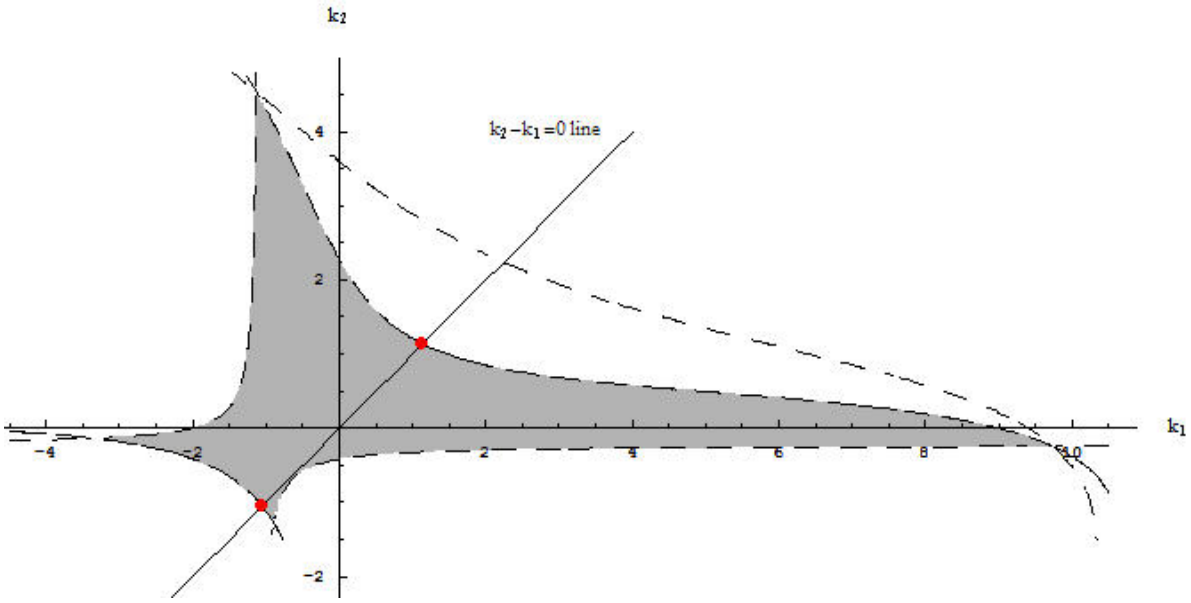

Figure 4.11 : Stabilizing gains in the parameter space for system in (4.35)

Solving $\Delta(w)=0$ in real w results in the following frequencies; $w_{1,2}=0$, $w_{3,4}=\pm 1.29566$, $w_{5,6}=\pm 4.3043$, where $\Delta(w)<0$ holds if $1.29566<w<4.3043$, $-4.3043<w<-1.29566$. Let us determine the intersection points of the $k_2-k_1=0$ line with the boundary of the stability region. For this example the critical values of k are calculated as $k=-1.05249$ and $k=1.1352$ if $-1.05249<k<1.1352$ the closed-loop system is stable. Let us consider the system given in (4.36). In this case, the region of all stabilizing controllers is presented in Figure 4.12. If we want to determine all controllers of the form $K=diag(k,k)$ for the perturbed case the critical points are calculated as $\{-0.996988,-0.875854,-0.485848,1.08832\}$ with stabilizing intervals $-0.996988<k<-0.875854$ and $-0.485848<k<1.08832$.

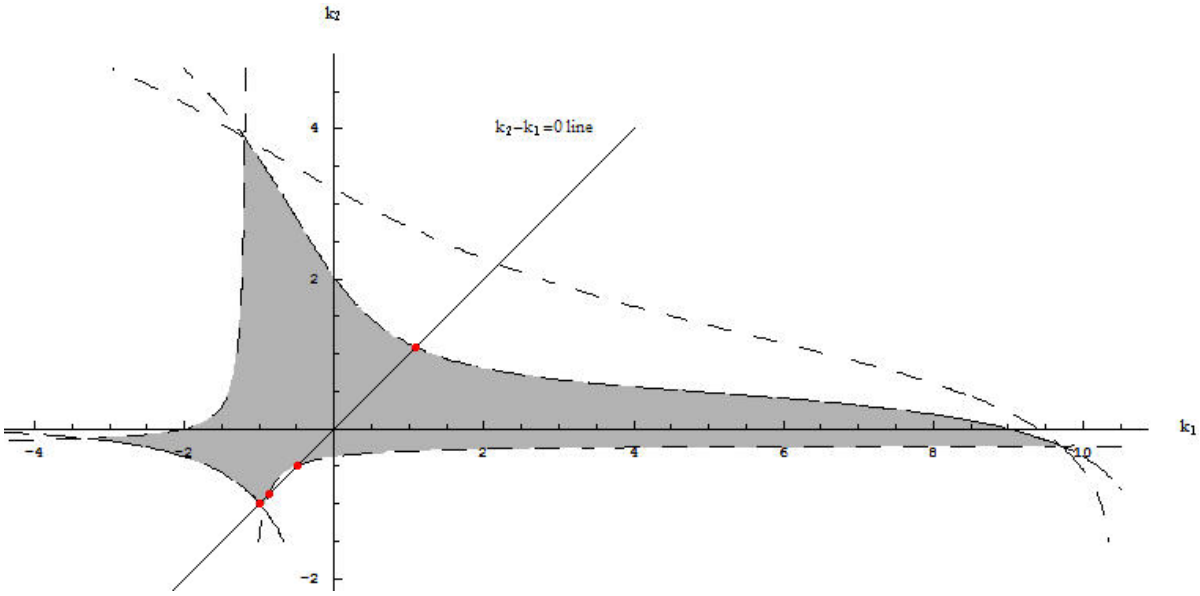

Figure 4.12 : Stabilizing gains in the parameter space for system in (4.36)

5. CONCLUSION AND RECOMMENDATIONS

Symbolic algebra presents us a new vision for the solution of numerically error prone problems such as pole assignment. It has been demonstrated that using the exact calculation properties of symbolic algebra languages it is actually possible to use direct and simple methods for solving some of the numerically sensitive problems in control system design. The advantage of manipulating symbols has also been exposed at several stages in the design of control systems.

The design stages considered in this research were the block diagram reduction in system modeling, finding stabilizing gain intervals, designing controllers to satisfy time domain performance, and computing robust controllers. It has been shown through tutorial examples that from modeling to the design of robust controllers computer algebra can simplify the task of designers by reducing the chance of making errors in algebraic manipulations, highlighting possible linearities and nonlinearities, simplifying formulations, and reducing the overall computation time. The author believes that there are still many parts of control engineering analysis & design where the advantages of symbolic algebra can be exploited. Possible future research areas include manipulation of multi-input multi-output systems automatically, exact simulation of control systems, and algebraic derivation of stabilizability conditions for more complicated controllers.

In this thesis, a new toolbox has also been presented to calculate the transfer function, or the state-space description of a system formed by subsystems. All calculations can be carried out symbolically to find out closed-loop transfer functions (or state-space solutions) in symbolic form. The author believes that the toolbox will not only be used by students that learn control engineering and by practicing control engineers, but also by theoreticians who would like to derive algebraic expressions for the description of control systems with given topologies. A possible usage of the toolbox can be in the area of parametric uncertain systems where the symbolic calculation of the overall transfer function for a given system structure is a common

practice. Possible future work includes integration of the toolbox with the existing symbolic algebra packages such as Parametric Uncertain Systems Toolbox.

In this thesis two fast and efficient algorithms are developed for determining the stabilizing gain intervals for TITO systems with reducible and irreducible characteristic equations, respectively. The key feature of these algorithms is that they avoid the difficulties associated with calculation the roots of a polynomial. They are based on the real-axis intersections and the direction of these intersections of the characteristic value plots. In an upcoming work the robust stability conditions will be considered. Possible future work includes integration of these algorithms with the Block Diagram Reduction Toolbox.

REFERENCES

[1] **Desoer, C. A., and Chan W. S.,** 1975: The feedback interconnection of lumped linear time-invariant systems, *Journal of the Franklin Institute*, **300**, 335–351.

[2] **MacDuffee, C. C.,** 1950: *The Theory of Matrices*. Chelsea Publishing Co., New York.

[3] **Maciejowski, J. M.,** 1989: *Multivariable Feedback Design*. Addison-Wesley, Wokingham, England, ISBN: 0-201-18243-2.

[4] **Youla, D. C., and Bongiorno, J. J., and Lu, C. N.,** 1974: Single-loop feedback stabilization of linear multivariable plants, *Automatica*, **10(2)**, 159–173.

[5] **Youla, D. C., Jabr, H. A., and Bongiorno, J. J.,** 1976: Modern Wiener-Hopf design of optimal controllers – part II: multivariable case, *IEEE Transactions on Automatic Control Theory*, **21(3)**, 319–338.

[6] **Sain, M. K., and Schrader, C. B.,** 1990: The role of zeros in the performance of multi-input, multi-output feedback systems, *IEEE Transactions on Education*, **33**, 244–257.

[7] **Wonham, W. M.,** 1979: *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York.

[8] **Vardulakis, A. I. G.,** 1991: *Linear Multivariable Control: Algebraic Analysis and Synthesis Methods*, JohnWiley & Sons, Chichester, Sussex, UK.

[9] **Strang, G.,** 1976: *Linear Algebra and its Applications*, Academic Press, New York.

[10] **Gantmacher, F. R.,** 1964: *The Theory of Matrices*, Chelsea Publishing Company, New York, NY

[11] **Derusso, P. M., Roy, R. J., and Close, C. M.,** 1965: *State Variables for Engineers*, John Wiley & Sons, New York.

[12] **Porter, W. A.,** 1966: *Modern Foundations of Systems Engineering*, The Macmillan Company, New York.

[13] **MacFarlane, A. G. J., and Belletrutti, J. J.,** 1970: The characteristic locus design method, *Automatica*, **9(5)**, 575–588.

[14] **Postlethwaite, I., MacFarlane, A. G. J.,** 1979: *A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems*, Springer-Verlag.

[15] **Desoer, C. A., and Wang, Y. -T.,** 1980: On the generalized Nyquist stability criterion, *IEEE Transactions on Automatic Control*, **25(2)**, 187–196.

[16] **Stevens, P. K.,** 1981: Generalization of the Nyquist stability criterion, *IEEE Transactions on Automatic Control*, **26(3)**, 664–669.

[17] **Shafiei, Z., and Shenton, A. T.,** 1994: Tuning of PID-type controllers for stable and unstable systems with time delay, *Automatica*, **30**, 1609–1615.

[18] **Shafiei, Z., and Shenton, A. T.,** 1997: Frequency-domain design of PID controllers for stable and unstable systems with time delay, *Automatica*, **33(12)**, 2223–2232.

[19] **Ho, M.-T., Datta, A., and Bhattacharrya S. P.,** 1996. A new approach to feedback stabilization, *The 35th IEEE Conference on Decision and Control.,* Kobe, Japan, December 11-13.

[20] **Ho, M.-T., Datta, A., and Bhattacharrya S. P.,** 1997. A linear programming characterization of all stabilizing PID controllers, *IEEE Proc. American Control Conference,* Albuquerque, NM, USA, June 4-6.

[21] **Munro, N., Söylemez, M. T., and Baki, H.**, 1999. Computation of D-Stabilizing low-order compensators, *Control System Centre Report,* **882**, UMIST, Manchester, UK.

[22] **Söylemez, M. T., Munro, N., and Baki, H.,** 2003: Fast Calculation of Stabilizing PID controllers, *Automatica,* **39**, 121–126.

[23] **Ackermann, J., and Kaesbauer, D.,** 2003: Stable polyhedra in parameter space, *Automatica*, **39**, 937–943.

[24] **Gryazina, E. N., and Polyak, B. T.,** 2006: Stability regions in the parameter space: D-decomposition revisited, *Automatica*, **42(1)**, 13–26.

[25] **Palmor, Z. J., Halevi, Y., and Krasney, N.,** 1995: Automatic tuning of decentralized PID controllers for TITO processes, *Automatica*, **31(7)**, 1001–1010.

[26] **Skogestad, S., and Morari, M.,** 1989: Robust performance of decentralized control systems by independent designs, *Automatica*, **25(1)**, 119–125.

[27] **Söylemez, M. T., and Üstoğlu, İ.**, 2006: Designing control systems using exact and symbolic manipulations of formulae, *International Journal of Control,* **79**(11), 1418–1430.

[28] **Rosenbrock H. H.,** 1969. Design of multivariable control systems using the inverse Nyquist array, *Proc. Inst. Elect. Eng,* **116**, 1929–1936.

[29] **Rosenbrock H. H.,** 1970: *State-Space and Multivariable Theory*, London: Nelson, UK.

[30] **Barman, J. F., and Katzenelson, J.,** 1974: A generalized Nyquist-type stability criterion for multivariable feedback systems, *International Journal of Control,* **20(4)**, 593–622.

[31] **MacFarlane, A. G. J., and Postlethwaite, I.,** 1977: Generalized Nyquist stability criterion and multivariable root loci, *International Journal of Control,* **25(1)**, 81–127.

[32] **Åström, K. J., and Hägglund, T.,** 1995: *PID Controllers: Theory, Design, and Tuning*. 2nd Ed., Instrument Society of America, NC, USA.

[33] **Ziegler, J. G., and Nichols, N. B.,** 1942: Optimum setting for automatic controllers. *Transactions of ASME.* Vol. **64**, pp.759–768.

[34] **Goodwin, G. C., Graebe, S. F., and Salgado, M. E.,** 2001: *Control System Design*. ISBN: 0-13-958653-9, Prentice-Hall, New Jersey, USA

[35] **Toscano, R.,** 2005: A simple robust PI/PID controller design via numerical optimization approach, *Journal of Process Control*, **15**, 81–88.

[36] **Daley, S.,** (Editor) 1999: PID Tuning Methods (special edition). *Computing and Control Engineering Journal*, **10(2)**, 42–69.

[37] **Saeki, M., and Kimura, J.,** 1997. Design method of robust PID controller and CAD systems, *IFAC Symposium on System Identification,* Kitakyushu, Japan, July 8–11.

[38] **Ohta, Y., Li, J., Tagawa, K., and Haneda, H.,** 1997: Robust PID controller design. *Proceedings of NOLTA,* pp.1053–1056.

[39] **Malan, S. A. and Yılmaz, P.,** 1994. Robust tuning for PID controllers with multiple performance specifications, *Conference on Decision Control*, Lake Buena Vista, Florida, USA, December 14–16.

[40] **Morari, M., and Zafiriou, E.,** 1989: *Robust Process Control*. ISBN: 0-137-82153-0, Prentice-Hall, New Jersey, USA.

[41] **Skogestad, S.,** 2003: Simple analytical rules for model reduction and PID controller tuning, *Journal of Process Control*, **15**, 291–309.

[42] **Isaksson, A. J., and Graebe, S. F.,** 1999: Analytical PID parameter expressions for higher order systems, *Automatica*, **35**, 1121–1130.

[43] **Skogestad, S., and Postlethwaite, I.,** 1996: *Multivariable Feedback Control: Analysis and Design*. John Wiley, Chicester, England.

[44] **Ackermann, J., and Kaesbauer, D.,** 2001. Design of robust PID controllers, *ECC'01, The 6th European Control Conference,* Porto, Portugal, September 4–7.

[45] **Datta, A., Ho, M.-T., and Bhattacharrya, S. P.,** 2000: *Structure and Synthesis of PID Controllers*. ISBN: 1-85233-614-5, Springer-Verlag, London, UK.

[46] **Ho, M.-T., Datta, A., and Bhattacharrya, S. P.,** 2001: Robust and non-fragile PID controller design, *International Journal of Robust and Nonlinear Cont*rol, **11**, 681–708.

[47] **Åström, K. J., and Hägglund, T.,** 2002: The future of PID control, *Control Engineering Practice*, **9**, 1163–1175.

[48] **Söylemez, M. T.,** 1999: *Pole Assignment for Uncertain Systems*. RSP Press. UMIST Control System Centre Series, London, ISBN:0-86380-246-X.

[49] **Cominos, P., and Munro, N.,** 2001: PID Controllers: Recent tuning methods and design to specification, *IEE Proceedings – Control Theory Appl.*, **149**, 46–53.

[50] **Åström, K. J., Hägglund, T., Hang, C. C., and Ho, W. K.,** 1993: Automatic tuning and adaptation for PID Controllers – a survey, *Control Engineering Practice*, **1**, 699–714.

[51] **Söylemez, M. T.,** 2006: The smallest stabilizable left half plane for all pole plants, *IEE Proceedings: Control Theory and Applications*, **153(1)**, 124–126.

[52] **Söylemez, M. T. and Munro, N**., 1998. Pole assignment and symbolic algebra: a new way of thinking, *UKACC'98 International Conference on Control,* Swansea, UK, September 1-4.

[53] **Chetty, M.**, 1997. Symbolic algebra systems: A new way of thinking in system analysis and control, *AUPEC'99, The Australian Universities Power Engineering Conference*, Darwin, Australia, September 26-29.

[54] **Munro, N.,** (Editor) 1999: *Symbolic Methods in Control System Analysis and Design*. IEE Control Engineering Series, IEE, Bath, UK.

[55] **Paláncz, B., Benyó, Z., and Kovács, L.,** 2005: Product review: Control System Professional Suite, *IEEE Control Systems Magazine*, **25(2)**, 67-75.

[56] **Söylemez, M. T. and Munro, N.**, 1997. Development of a robust eigenvalue assignment toolbox using a Kharitonov based approach, *IEE Colloquium*, Savoy Palace, London, UK, IEE: Professional Group B1, Digest No:97/380.

[57] **Söylemez, M. T., and Üstoğlu, İ.**, 2004. Block Diagram Reduction Using Symbolic Algebra, *The 12th IEEE Mediterranean Conference on Control and Automation MED'04,* Kuşadası, Türkiye, June 06-09.

[58] **Ho, M.-T., Datta, A., and Bhattacharrya S. P.,** 1997. Control system design using low order controllers: constant gain, PI and PID, *Proc. American Control Conference,* Albuquerque, NM, USA, June 4-6.

[59] **Söylemez M. T., Munro N.,** 1997: Robust pole assignment in uncertain systems. *Proc. IEE: Control Theory and Applications*, Vol. **144**, no. 3, pp. 217-224.

[60] **Söylemez. M. T. and Munro, N.**, 1999. Pole assignment for uncertain systems, in *Symbolic Methods in Control System Analysis and Design,* IEE, pp. 251-272, Ed. Munro, N., Bath, UK, ISBN: 0-85296-943.

[61] **Van Dooren, P.,** 2004: The basics of developing numerical algorithms, *IEEE Control Systems Magazine,* **24(1)**, 18–27.

[62] **Higham, N. J., Konstantinov, M., Mehrmann , V., and Petkov P.,** 2004: The sensitivity of computational control problems, *IEEE Control Systems Magazine,* **24(1)**, 28–43.

[63] **Van Huffel, S., Sima, V., Varga, A., Hammarling S., and Delebecque, F.,** 2004: High performance numerical software for control, *IEEE Control Systems Magazine,* **24(1)**, 60–76.

[64] **Baki, H., and Munro, N**., 1998. Implementation of balanced realisation algorithms in a symbolic environment, *UKACC'98 International Conference on Control,* Swansea, UK, September 1-4.

[65] **Mason, S. J.,** 1953: Feedback theory: some properties of signal flow graphs, *Proc. IRE,* **41(9)**, 1144–1156.

[66] **Mason, S. J.,** 1956: Feedback theory: further properties of signal flow graphs, *Proc. IRE,* **44(7)**, 920–926.

[67] **Url-1** <*http://www.elk.itu.edu.tr/~soylemez/sa/bdrt* >, accessed at 29.12.2008.

[68] **Url-2** <*http://documents.wolfram.com/mathematica/Add-onsLinks/NETLink*>, accessed at 29.12.2008.

[69] **Nise, N. S.,** 2000: *Control Systems Engineering*. John–Wiley, New York, USA.

[70] **Lin, P. M.,** 1973: A survey of applications of symbolic network functions, *IEEE Transactions Circuit Theory,* Vol. **CT-20**, pp.732–737.

[71] **Lin, P. M., and Alderson, G. E.,** 1973: Computer generation of symbolic network functions – A new theory and implementation, *IEEE Transactions Circuit Theory,* Vol. **CT-20**, pp.48–56.

[72] **Shieu, S. D., and Chan, S. P.,** 1974: Topological formulation of symbolic network functions and sensitivity analysis of active networks, *IEEE Transactions Circuit and Systems,* Vol. **CAS-21**, no.2, pp.39–45.

[73] **Lee, A. Y.,** 1974: Signal flow graphs – Computer-aided system analysis and sensitivity calculations, *IEEE Transactions Circuit and Systems,* Vol. **CAS-21**, no.2, pp.209–216.

[74] **Lee, T.,** 1976: A simple method to determine the characteristic function by discrete Fourier series and fast Fourier transforms, *IEEE Transactions Circuit and Systems,* **21**, 242–243.

[75] **Singhal, K., Vlach, J.,** 1977: Symbolic analysis of analog and digital circuits, *IEEE Transactions Circuit and Systems,* Vol. **CAS-24**, no.11, pp.598–609.

[76] **Mielke, R. R.,** 1978: A new signal flow graph formulation of symbolic network functions, *IEEE Transactions Circuit and Systems,* Vol. **CAS-25**, no.6, pp.334–340.

[77] **Sannuti, P., Puri, N. N.,** 1980: Symbolic network analysis – An algebraic formulation, *IEEE Transactions Circuit and Systems,* Vol. **CAS-27**, no.8, pp.679–687.

[78] **Wolfram S.,** 2003: *The Mathematica Book*. 5th Ed., Cambridge University Press, ISBN: 1579550223, UK.

[79] **Url-3** <*http://math.ucsd.edu/~ncalg*>, accessed at 29.12.2008.

[80] **Lewis A. D.,** 2004. What is the characteristic polynomial of a signal flow graph?, in *Unsolved Problems in Mathematical Systems and Control Theory,* Eds. Blondel, V.D., Megretski, A., pp. 326-329, Princeton University Press, NJ, USA, ISBN:0-691-11748-9.

[81] **Url-4** <*http://press.princeton.edu/math/blondel/* >, accessed at 29.12.2008.

[82] **Helton, J. W., Stankus, M., Wavrik, J. J.,** 1998: Computer simplification of formulas in linear systems theory, *IEEE Transactions on Automatic Control Theory*, **43(3)**, 302–314.

[83] **Kronewitter, F. R.,** 2000. Noncommutative computer algebra in linear algebra and control theory, *PhD Thesis,* University of California, San Diego, USA.

[84] **Silva, G. J., Datta, A., and Bhattaharyya, A. P.,** 2005: *PID Controllers for Time-Delay Systems*. ISBN: 0-8176-4266-8. Birkhauser, Boston, USA.

[85] **Url-5** <*http://www.elk.itu.edu.tr/~soylemez/sa/rt.htm*>, accessed at 29.12.2008.

[86] **Hao, X., Datta, A., and Bhattaharyya, S. P.,** 2001: Computation of all stabilizing PID gains for digital control systems, *IEEE Transactions on Automatic Control Theory*, **46(4)**, 647–652.

[87] **Söylemez, M. T., and Munro, N.,** 2001: A parametric solution to the pole assignment problem using dynamic output-feedback, *IEEE Transactions on Automatic Control*, **46(5)**, 711–723.

[88] **Munro, N., and Söylemez, M. T.,** 1996. The use of symbolic algebra for uncertain systems, *UKACC'96 International Conference on Control,* Exeter, UK, September 2-5.

[89] **Bartlett, A. C., Hollot , C. V., and Lin, H.,** 1988: Root locations of an entire polytope of polynomials: it suffices to check edges, *Math. Contr. Signals Syst.*, **1**, 61–71.

[90] **Url-6** <*http://library.wolfram.com/infocenter/MathSource/3947/* >, accessed at 29.12.2008.

[91] **Üstoğlu, İ., and Söylemez, M. T.,** 2007. Feasibility Conditions on PID Controller Synthesis using Dominant Pole Assignment, *ECC'07, The 9th European Control Conference,* Kos, Greece, July 2–5.

[92] **Yu, C. C.,** 1999: *Autotuning of PID Controllers*. Springer-Verlag. London, Great Britain.

[93] **Tan, K. K., Wang, Q.-G., Hang, C. C., and Hägglund, T.,** 1999: Advances in PID control. Springer-Verlag. London, Great Britain.

[94] **O'Dwyer, A.,** 2003: Handbook of PI and PID Controller Tuning Rules. Imperial College Press, London, ISBN 1-8609-4342-X.

[95] **Johnson, M. A., and Moradi, M. H.,** 2005: PID Control: New Identification And Design Methods. Springer, Berlin, ISBN 1-8523-3702-8.

[96] **Visioli, A.,** 2006: *Practical PID Control*. Springer, London, UK ISBN: 978-1-84628-585-1.

[97] **Keel, L. H., and Bhattaharyya, S. P.,** 1990: State space design of low-order stabilizers, *IEEE Transactions on Automatic Control*, **35(2)**, 182–186.

[98] **Url-7** <*http://library.wolfram.com/infocenter/MathSource/5711/* >, accessed at 01.01.2009.

[99] **Url-8** <*http://www.wolfram.com/products/applications/pcs/*>, accessed at 11.01.2009.

[100] **Söylemez, M. T., and Üstoğlu İ.,** 2007: Polynomial control systems, *IEEE Control System Magazine*, **27(4)**, 124–137.

[101] **Bryant G. F., and Yeung, L. F.,** 1996: *Multivariable Control System Design Techniques: Dominance and Direct Methods*, Wiley.

[102] **Niederlinski, A.,** 1971: A heuristic approach to the design of linear multivariable interacting control systems, *Automatica*, **7(6)**, 691–701.

CURRICULUM VITA

Candidate's full name: İlker ÜSTOĞLU

Place and date of birth: Istanbul, 24.11.1974

Permanent Address: I.T.U. Department of Control Engineering, Maslak.

Universities and Colleges attended: I.T.U. Control and Computer Engineering, 1999

I.T.U. Electrical Engineering, 1997

Deutsche Schule Istanbul, 1993

Publications:

- Söylemez, M. T., and **Üstoğlu, İ.**, 2006: Designing control systems using exact and symbolic manipulations of formulae, *International Journal of Control*, **79(11)**, 1418–1430.
- Söylemez, M. T., and **Üstoğlu İ.,** 2007: Polynomial control systems, *IEEE Control System Magazine*, **27(4)**, 124–137.
- **Üstoğlu, İ.,** and Söylemez, M. T., 2007. Feasibility Conditions on PID Controller Synthesis using Dominant Pole Assignment, *ECC'07, The 9th European Control Conference,* Kos, Greece, July 2–5.
- Söylemez, M. T., and **Üstoğlu, İ.,** 2004. Block Diagram Reduction Using Symbolic Algebra, *The 12th IEEE Mediterranean Conference on Control and Automation MED'04,* Kuşadası, Türkiye, June 6–9.
- Çamlıbel, M. K., and **Üstoğlu, İ.,** 2005: Feedback linearization of piecewise linear systems, *16th IFAC World Congress on Automatic Control*, Prague, Czech Republic, July 3–8.
- **Üstoğlu, İ.,** 2006: Stability of Takagi-Sugeno fuzzy systems, *3rd International Conference on Informatics in Control, Automation and Robotics*, Setubal, Portugal,

August 1–5.

▪ **Üstoğlu, İ.,** Yeşil, E., Güzelkaya, M., and Eksin, İ., 2004: Exponential stability of discrete-time Takagi-Sugeno fuzzy systems, *ISSCI 2004 - 5th International Symposium on Soft Computing for Industry*, Seville, Spain, June 28 – July 1.

▪ Kaymakçı, Ö.T., and **Üstoğlu, İ.,** 2006: Ölü-zamanlı ısıl sistemleri kararlı kılan PI kontrolörler, *TOK'06*, November 6–8.

▪ **Üstoğlu, İ.,** and Söylemez, M. T., 2006: Statik çıkış geribeslemesi probleminin parametre uzayında LMI problemine indirgenmesi, *TOK'06*, November 6–8.

▪ **Üstoğlu, İ.,** and Çamlıbel, M. K. 2005: Parça parça lineer sistemlerin geribesleme ile lineerleştirilmesi, TOK'05, June 2–3.

▪ Yeşil, E., **Üstoğlu, İ.,** and Güzelkaya, M. 2000: Sistemlerin bulanık modellenmesi, ELECO Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu, November 8–12.