Mathematica implementation of output-feedback pole assignment for uncertain systems via symbolic algebra

Author final version (International Journal of Control, Vol. 79, No. 11, November 2006, 1431-1446)

Mathematica Implementation of Output-Feedback Pole Assignment for Uncertain Systems via Symbolic Algebra

X. Zheng[*], A.C. Zolotas[*] and H. Wang[†]

Abstract

This paper presents the application of symbolic algebra techniques to the MATHEMATICA implementation of a set of output-feedback pole assignment algorithms, for systems characterised by parametric uncertainty. For multivariable systems there may be more than one feedback matrix solutions leading to the same closed-loop poles based on the same algorithm used. Thus over-parameterised solutions are sought by generalising the existing algorithms with extra degrees of freedom retained in the symbolic variables. The general parametric form of output-feedback compensators is developed in terms of the uncertain parameters and symbols representing the extra degrees of freedom. The implementation of three output-feedback pole assignment techniques is presented, with the theory briey introduced and examples illustrating the effectiveness of the algorithms described.

[*]Control Systems Group, Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, LE11 3TU, UK, {x.zheng, a.c.zolotas}@lboro.ac.uk; *corresponding author*: X. Zheng.

[†]Control Systems Centre, School of Electrical and Electronics Engineering, The University of Manchester, P.O. Box 88, Manchester M60 1QD, UK, hong.wang@manchester.ac.uk.

Notation

n	Number of states
m	Number of inputs
l	Number of outputs
$\boldsymbol{A}$	$n \times n$ system matrix
$\boldsymbol{B}$	$n \times m$ input matrix
$\mathbf{b}_i$	i^{th}column of the input matrix
$\boldsymbol{C}$	$l \times n$ output matrix
$[\boldsymbol{A}, \boldsymbol{B}]$	A system defined by $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices
$[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]$	A system defined by $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ matrices
Φ	Controllability matrix of a system
$\boldsymbol{I}_n$	$n \times n$ identity matrix
i	The imaginary number of$\sqrt{-1}$
q	Order of the output-feedback compensator
$\mathbf{q}$	$r \times 1$ uncertain parameter vector
r	Degree of uncertainties
$\boldsymbol{A}^{\text{t}}$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of matrix $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$	Determinant of matrix $\boldsymbol{A}$
$\mathbf{Adj}(\boldsymbol{A})$	Adjoint matrix of matrix $\boldsymbol{A}$
$\lfloor x \rfloor$	The nearest integer lower than or equal to$x$
$\lceil x \rceil$	The nearest integer greater than or equal to$x$
Γ	The set of desired closed-loop poles
φ_i	Constants defined as $\varphi_i = \left\lfloor \frac{\max(m,l)}{\min(m,l-i+1}} \right\rfloor$
φ	A constant defined as $\varphi = \sum_{i=1}^{\min(m,l)} \varphi_i$
$\boldsymbol{G}(s)$	Transfer function matrix of the open-loop system
$\boldsymbol{F}(s)$	Output-feedback compensator matrix
SISO	Single Input Single Output
SIMO	Single Input Multi Output
MISO	Multi Input Single Output
MIMO	Multi Input Multi Output

1 Introduction

The Pole Assignment problem relates to moving all or a portion of the poles of a given time-invariant linear system, to a specified set of prescribed locations in the complex plane by means of state or output feedback. State-feedback methods are easy to solve and rather straightforward to implement if all states of the system are accessible. This however is hardly what happens in reality with states been difficult to measure or even inaccessible or significantly corrupted by noise, and usually observers are included to provide the necessary estimates in the expense of increasing overall complexity [7, 13].

Output-feedback compensation is necessary for the aforementioned reasons. However, since calculating output-feedback compensators involves solving high nonlinear equations [11], it is important to develop methods which can render the calculations easier. Another objective for the output-feedback pole assignment is, to reduce the least order of the compensator assigning all the closed-loop poles arbitrarily, which is equal to saying design a fixed-order compensator which increases the maximum number of poles that can be assigned arbitrarily.

Pearson [2, 8, 9] found the relationship between the minimum order of the output-feedback compensator and the controllability and observability indices. In addition, Seraji [4] improved the methods by Chen [3] for designing dyadic (rank one) dynamic compensators to cover both complete and partial pole assignment. Moreover, Munro and Novin-Hirbod [6] further improved Seraji's method to full-rank output-feedback compensators, having better disturbance rejecting properties. Recently, Soylemez and Munro [14, 15] introduced a method of partial output-feedback pole assignment, to further improve the maximum number of poles that can be assinged with a fixed order compensator. The necessary order of the synthesized output-feedback compensator is the lowest for arbitrary assignment of all the closed-loop poles using Soylemez and Munro's method.

The implementation of algorithms for output-feedback pole assignment in a numerical environment is usually complicated and also deficient in accuracy. However, in a symbolic environment the computation becomes simpler, more straightforward, and easier to handle in a mathematical programming sense. Many engineering methods

that were considered impractical when implemented numerically, actually turned out being practical when implemented in a symbolic algebra computing environment [5]. There is no doubt that implementation using symbolic computation based on symbols and fractional numerical forms with infinite precision yields improved accuracy (avoiding accumulative computation errors in a numerical environment).

There are quite a few software packages for symbolic computation, such as DERIVE, MAPLE, MATHEMATICA, MuPad. These systems have a similar set of basic commands for algebraic manipulations, e.g. the transformation (simplifying, expanding, factorizing) of expressions and solving equations. They also have commands for selecting different parts of an expression (e.g. `Last` and `First` in MATHEMATICA), very useful for realizing various algorithms [16]. DERIVE offers less capabilities compared to the other programs. MATHEMATICA is considered to be the best system [1], based upon various criteria of power, purpose, availability, flexibility etc. Moreover, MATHEMATICA offers great flexibility for programming, i.e. capable of functional programming, object-oriented programming, and rule based programming in addition to the traditional procedural programming. Thus, MATHEMATICA can be a useful tool for realizing a variety of algorithms.

This paper presents the implementation of three output-feedback pole assignment algorithms: (i) dyadic method by Seraji [4], (ii) full-rank method by Munro and Novin-Hirbod [6] and (iii) constant [15] and dynamic [14] partial-pole placement by Soylemez and Munro; using symbolic algebra computation in `MATHEMATICA` for parametric uncertain systems, i.e. systems whose uncertainties are represented by symbols in their models. Symbolic Algebra computation offers an advantage for realizing, and extending the above algorithms for parametric uncertain systems. The compensators are obtained in a general parametric framework by retaining the extra degrees of freedom as symbols (also referred to as free parameters) in their structure. These free parameters can be then optimised to achieve specific internal stability and/or robustness of the resulting closed-loop system. Another useful merit of utilising symbolic computations is that the repetitive calculation required by numeric computations for the process of optimization of the free parameters can be avoided.

2 Pole assignment for parametric uncertain systems

Parametric uncertain systems are systems characterised by uncertainty in their parameters. The uncertain parameters are commonly represented by a vector as $\mathbf{q} = [q_1\ q_2\ \cdots\ q_r]$, where q_i is the uncertain parameter bounded as $q_i^- \leq q_i \leq q_i^+$, r is the dimension of the uncertain vector. The state space representation of the model of a parametric uncertain linear system can be written as

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}(\mathbf{q}) + \boldsymbol{B}(\mathbf{q})\mathbf{u}(t) \quad (1)$$

$$\mathbf{y}(t) = \boldsymbol{C}(\mathbf{q})\mathbf{x}(t) \quad (2)$$

where $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are matrix functions of $\mathbf{q}$. Moreover, the nominal operating condition for $\mathbf{q}$ is denoted by $\mathbf{q}_n$.

Soylemez and Munro [12] presented a reasonable approach for finding a solution to robust output-feedback pole assignment problem: *Given a pre-specified set of numbers,* $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_p\}$*, closed under complex conjugation, where p is the number of closed-loop poles to be assigned. Firstly, find the general parametric form of the compensator,* $\boldsymbol{F}(\mathbf{k})$ *of order q, where* $\mathbf{k}$ *is the vector of free parameters, such that the poles of the resulting closed-loop system,* $[\boldsymbol{A - BFC, B, C}]$*, are equal to* Γ *under nominal working conditions; then find the set of* $\mathbf{k}$ *vectors that satisfy pre-specified performance criteria for all possible perturbations* $\mathbf{q}$. However, the approach adopted in this paper is, instead of finding $\boldsymbol{F}(\mathbf{k})$ for the nominal working conditions q_n, to find the general compensator $\boldsymbol{F}$ directly in terms of both the uncertain parameters $\mathbf{q}$ and the free parameters $\mathbf{k}$, i.e. $\boldsymbol{F}(s, \mathbf{q}, \mathbf{k})$.

3 Dyadic design method

Seraji's algorithm [4] introduced a simple frequency-domain method for the design of physically realizable dynamic output-feedback compensators to achieve pole assignment in single-input or single-output systems, i.e. SISO, SIMO or MISO. In terms of MIMO systems, pseudo single-input or pseudo single-output systems can be obtained using the dyadic technique before calculating the output-feedback compensator with Seraji's algorithm.

The Dyadic method for output-feedback pole assignment can be stated as follows: Given a multivariable system $[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]$, whose transfer function $\boldsymbol{G}(s) = \boldsymbol{C}\,(sI - \boldsymbol{A})^{-1}\,\boldsymbol{B}$ is an $m \times l$ matrix, find the output-feedback compensator matrix $\boldsymbol{F}$ (of rank one) in the form of outer product of two vectors $\mathbf{f}$ and $\mathbf{m}^t$ [1]

$$\boldsymbol{F} = \mathbf{f}\ \mathbf{m}^t \tag{3}$$

where $\mathbf{m}^t(s)$ is an l-row vector of order q given by:

$$\mathbf{m}^{\mathrm{t}} = \frac{\mathbf{N}(s)}{p_f(s)} = \frac{1}{s^q + d_q\ s^{q-1} + \cdots + d_1}[N_1(s) \cdots N_l(s)] \tag{4}$$

where

$$N_i(s) = b_{qi}\ s^q + b_{q-1,i}\ s^{q-1} + \cdots + b_{0i}, \quad i = 1, 2, \cdots, l \tag{5}$$

$\mathbf{f}$ is a predefined constant m-column vector such that the resulting pseudo single-input system $[\boldsymbol{A},\ \boldsymbol{B}\mathbf{f},\ \boldsymbol{C}]$ is completely controllable, which equally means that the corresponding m-column vector, $\mathbf{g}(s) = G(s)\mathbf{f}$, is completely controllable. $\mathbf{m}^t(s)$ is designed for the pseudo system so that the resulting closed-loop system poles are moved to the desired set of poles, $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_p\}$, where p is the number of closed-loop poles to be assigned. The open-loop transfer function of the pseudo single-input system is given by

$$\boldsymbol{G}(s) = \boldsymbol{C}[s\boldsymbol{I}_n - \boldsymbol{A}]^{-1}(\boldsymbol{B}\mathbf{f}) \tag{6}$$

$$= \frac{\mathbf{w}(s)}{p(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \cdots + a_0} \begin{pmatrix} W_1(s) \\ \vdots \\ W_l(s) \end{pmatrix} \tag{7}$$

where $p(s)$ is the open-loop system characteristic polynomial, $\mathbf{w}(s)$ is the numerator matrix, and $W_i = m_{ni}s^{n-1} + \ldots + m_{1i}$. The closed-loop system characteristic polynomial can be written as

$$p_c = p_f(s)p(s) + \mathbf{N}(s)\mathbf{w}(s) \tag{8}$$

[1] Note that the output-feedback compensator can also be considered as $\boldsymbol{F} = \mathbf{m}^t\ \mathbf{f}$. Here, $\mathbf{f}$ is a constant l-row vector pre-sepcified such that the corresponding pseudo single-output system $[\boldsymbol{A}, \boldsymbol{B},\ \mathbf{f}\boldsymbol{C}]$ is completely observable.

From the given desired set of closed-loop poles Γ, the closed-loop characteristic polynomial can also be expressed as

$$p_c = (s-\gamma_1)(s-\gamma_2)\cdots(s-\gamma_p) \tag{9}$$

Equating coefficients of like powers of s in equation (8) and equation (9) gives

$$\boldsymbol{E}\mathbf{c} = \mathbf{h} \tag{10}$$

where $\boldsymbol{E}$ is an $(n+q)\times(ql+q+l)$ matrix, $\mathbf{h}$ is an $(n+q)$ column vector and $\mathbf{c}$ is the $(ql+q+l)$ column vector formed by the unkown parameters in $m^t(s)$. It can be comprehended that there are $(q+l(q+1))$ elements that can be used to assign the closed-loop poles to desired positions. Hence depending on the order of the compensator, q, some or all of the poles of the $(n+q)^{\text{th}}$ order closed-loop system can be positioned arbitrarily. If $ql+q+l \geq n+q$, it is possible to assign all the $(n+q)$ poles of the closed-loop system and there are $ql+l-n$ free parameters out of the parameters of the compensator; If $ql+q+l < n+q$, only $ql+q+l$ closed-loop poles can be assigned arbitrarily at most. In a more accurate way, the number of closed-loop poles that can be assigned arbitriliy is determined by the rank of matrix $\boldsymbol{E}$.

4 Full-rank design method

Munro and Novin-Hirbod [6] introduced a method that generates full-rank output-feedback compensators for multivariable systems. This method finds the output-feedback compensator as summation of a sequence of dyadic compensators expressed as

$$\begin{aligned} \boldsymbol{F} &= \sum_{\mu}^{i=1} \mathbf{f}^{(i)}\mathbf{m}^{(i)} \\ &= \boldsymbol{F_c}\,\boldsymbol{F_o} \end{aligned} \tag{11}$$

where $\boldsymbol{F_o} = [\mathbf{m}^1 \cdots \mathbf{m}^m]^{\text{t}}$, $\boldsymbol{F_c} = [\mathbf{f^1} \cdots \mathbf{f}^m]$, $\mu = \min(m,l)$, m and l are the number of inputs and outputs of the system respectively. In each step except for the first step, $\mathbf{f}^{(i)} = [f_1^{(i)} \cdots f_i^{(i)}\ 0 \cdots 0]^{\text{t}}$, which picks up the first i inputs, is designed to render as many poles assigned in the previous steps as possible uncontrollable (retained) from the

pseudo single-input system considered in its current step, by satisfying the following equation

$$\mathbf{Adj}(\gamma_j \, \boldsymbol{I}_n - \boldsymbol{A}_{\mathrm{cl}}^{(i-1)}) \, \boldsymbol{B}\mathbf{f}^{(i)} = 0 \tag{12}$$

where $\boldsymbol{A}_{\mathrm{cl}}^{(i-1)}$ is the state matrix of the resulting closed-loop system after $(i-1)$ steps, γ_j is the pole assigned in the past $(i-1)$ steps, $\boldsymbol{I}_n$ is identity matrix of the same dimension as matrix $\boldsymbol{A}$. For the first step, there is no pole to be retained, so $f_1^{(1)}$ is a free parameter. Due to the fact that if an $\mathbf{f}$ vector yields a closed-loop solution for dyadic pole assignment, then ρ $\mathbf{f}$ yields the same solution, where ρ is a nonzero scalar. Let $\mathbf{f}_1^{(1)} = 1$ for simplicity. In each step, the pseudo single-input system $[\boldsymbol{A}_{\mathrm{cl}}^{(i-1)}, \boldsymbol{B}\mathbf{f}^{(i)}, \boldsymbol{C}]$ is considered. $\mathbf{m}^{(i)}$ is designed so that at least one more closed-loop pole is to be moved to the specified location.

To make the algorithm more practical, it is assumed that the set of desired closed-loop poles, $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_p\}$, can be partitioned into m disjoint subsets [2], $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \cdots \Gamma_m$, such that each subset, Γ_i has n_i elements closed under complex conjugation as

Assumption 1.

$$\begin{aligned} \sum_{i=1}^{m} n_i &= p \\ n_i &= 1 \ (for \ i = 1, 2, \cdots, m-1) \\ n_m &= p - m + 1 \end{aligned} \tag{13}$$

Constant dyadic design is used for the first $(m-1)$ steps making sure that one more closed-loop pole is assigned in each step. For the final step, there should be $(m-1)$ closed-loop poles to be rendered uncontrollable. If $l+(m-1) \geq n$, the constant dyadic compensator is sufficient to assign all the remaining closed-loop poles. If $l+(m-1) < n$, a dynamic dyadic compensator of order $q \geq \left(\frac{n-(m+l-1)}{l}\right)$ is needed to assign all the remaining closed-loop poles.

[2] Here,we assume that $m < l$

5 Partial pole assignment method

Soylemez and Munro developed a new technique for partial pole assignment using constant [15] and dynamic output-feedback [14].

The technique, for either constant or dynamic assignment, consists of m steps for multivariable systems of m inputs and l outputs assuming that $m \leq l$ (see [3]), to assign as many poles as possible to desired locations. This technique is also called Pole Assignment by Pole Retention from Inputs (PAPRI) because in each step, $\varphi_i = \left\lfloor \frac{max(m,l)}{min(m,l)-i+1} \right\rfloor$ poles can be assigned and retained simultaneously. It is almost always possible to arbitrarily assign $\min(n, \varphi)$ closed-loop poles by a static linear output-feedback compensator, or to arbitrarily assign all the closed-loop system poles using a compensator of order $\lceil (n-\varphi)/max(m,l) \rceil$, where $\varphi = max(m,l) + \left\lfloor \frac{max(m,l)}{2} \right\rfloor + \cdots + \left\lfloor \frac{max(m,l)}{min(m,l)} \right\rfloor$. To make the algorithm more practical, it is assumed that the set of desired closed-loop poles, Γ, can be partitioned into m subsets given as $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \cdots \Gamma_m$, where each subset, Γ_i, closed under complex conjugation has n_i elements, and n_i satisfies the following

Assumption 2.

$$\begin{aligned} \sum_{i=1}^{m} n_i &= p \\ n_i &\leq \varphi_i \ (for\ i = 1, 2, \cdots, m-1) \\ n_m &\leq \varphi_m + q(l+1) \end{aligned} \tag{14}$$

For each step of the first $(m-1)$ steps, one input of the system is selected such that system has at least n_i controllable modes through this input. Then the constant output-feedback compensator via the chosen input is designed such that n_i closed-loop poles are assigned to the specified locations and simultaneously rendered uncontrollable through the remaining inputs of the system, which means that the following steps do not affect the already assigned poles. For the last step, the remaining n_m closed-loop poles are assigned and no further pole retention is required. If $n_m \leq l$, the constant compensator is sufficient to assign these n_m poles; else if $n_m > l$, a dynamic

[3] If $m > l$, then everything said can be applied to the dual system $[\ \boldsymbol{A}^{\mathrm{t}},\ \ \boldsymbol{B}^{\mathrm{t}},\ \ \boldsymbol{C}^{\mathrm{t}}\ ]$, and the compensator found at last can be transposed to find the compensator for the original system

compensator of order $q = \lceil \frac{n_m - l}{l+1} \rceil$ is necessary. In the following, we will briefly introduce the connection between pole assignment, partial pole assignment and pole retention. For more details the reader is referred to [15, 14].

5.1 Pole assignment

Assume a single-input system $[\boldsymbol{A}, \mathbf{b}, \boldsymbol{C}]$, with the objective being to assign all poles. In this case, the technique for output-feedback pole assignment developed is based on the mapping approach [17]. It was found that the mapping approach is fast and hence sufficient when implementing symbolically due to the simplicity of the equations involved, although has poor performance if implemented in a numerical environment [13]. For constant compensation, the output-feedback constant vector $\mathbf{k}_{\mathrm{y}}$ is given as

$$\begin{aligned} \mathbf{k}_{\mathrm{y}} &= [\boldsymbol{X}\Phi^{\mathrm{t}}\boldsymbol{C}^{\mathrm{t}}]^{-1}\delta & (15) \\ &= \boldsymbol{Z}^{-1}\delta & (16) \end{aligned}$$

where $\boldsymbol{Z} = \boldsymbol{X}\Phi\boldsymbol{C}$, $\boldsymbol{\Phi}$ is the controllability matrix of the system,i.e.

$$\boldsymbol{\Phi} = [\mathbf{b} \;\; \boldsymbol{A}\mathbf{b} \ldots \;\; \boldsymbol{A}^{n-1}\mathbf{b}] \tag{17}$$

δ is the difference vector given by the coefficients of the difference polynomial, $\delta(s) = p_c(s) - p(s)$, i.e.

$$\delta = [\alpha_{n-1} - a_{n-1} \;\; \alpha_{n-2} - a_{n-2} \cdots \;\; \alpha_0 - a_0]^{\mathrm{t}} \tag{18}$$

where a_i are the coefficients of the open-loop system characteristic polynomial,

$$\begin{aligned} p(s) &= |s\boldsymbol{I}_n - \boldsymbol{A}| \\ &= s^n + a_{n-1}s^{n-1} + \cdots + a_0 & (19) \end{aligned}$$

α_i are the coefficients of the desired closed-loop system characteristic polynomial,

$$p_c(s) = \sum_{i=1}^{n}(s - \gamma_i) \tag{20}$$

$\boldsymbol{X}$ is a lower-triangular Toeplitz matrix given as

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ a_{n-1} & 1 & 0 & \dots & 0 \\ a_{n-2} & a_{n-1} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \dots & 1 \end{pmatrix} \tag{21}$$

For dynamic compensation, the following equation is given instead of equation (16)

$$\mathbf{f} = \boldsymbol{Z}^{-1}\delta \tag{22}$$

where $\mathbf{f}$, as defined in equation (4), is a $(ql + q + l)$ elements vector that consists of all the parameters of the dynamic compensator $F(s)$ of order q , i.e.

$$\mathbf{f} = \begin{bmatrix} d_1 & \dots & d_q & b_{01} & b_{11} & \dots & b_{q1} & \cdots & b_{0l} & b_{1l} & \dots & b_{ql} \end{bmatrix} \tag{23}$$

Here, δ is the difference vector given by the coefficients of the polynomial, $p_c(s) - p(s)s^q$. $\boldsymbol{Z}$ is a $(n + q) \times (ql + q + l)$ matrix formed by the coefficients of the open-loop system characteristic polynomial, $p(s)$, and coefficients of the numerator polynomial matrix of the open-loop system transfer function matrix $\mathbf{w}(s)$ as defined in equation (7)

$$\boldsymbol{Z} = \left(\begin{array}{cccc|ccc|ccc} a_0 & 0 & \dots & 0 & m_{11} & 0 & \dots & 0 & m_{l1} & 0 & \dots & 0 \\ a_1 & a_0 & \dots & 0 & m_{12} & m_{11} & \dots & 0 & m_{l2} & m_{l1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & a_0 & \vdots & \vdots & & m_{11} & \vdots & \vdots & & m_{l1} \\ a_{n-1} & a_{n-2} & & \vdots & m_{1n} & \vdots & & \vdots & m_{ln} & \vdots & & \vdots \\ 1 & a_{n-1} & & \vdots & 0 & m_{1n} & & \vdots & 0 & m_{ln} & & \vdots \\ 0 & 1 & & \vdots & \vdots & 0 & & \vdots & \vdots & 0 & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & & a_{n-1} & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & & 1 & 0 & 0 & & m_{1n} & 0 & 0 & & m_{ln} \end{array}\right) \tag{24}$$

5.2 Partial pole placement

For each step, only part of the closed-loop system poles are assigned to the desired positions, Γ_i, through one chosen input using the technique of partial pole placement .

Again consider the previous assumed single-input system $[\boldsymbol{A}, \quad \mathbf{b}, \quad \boldsymbol{C}]$, with an output-feedback compensator of order q is to be designed so that p of the $(n+q)$ closed-loop poles are assigned to desired locations. The closed-loop characteristic polynomial is separated into two parts

$$p_c = p_d(s)p_e(s) \tag{25}$$

where $p_d(s)$ is the achievable part formed by the p poles assigned

$$p_d(s) = s^p + d_{p-1}s^{p-1} + \ldots + d_1 s + d_0 \tag{26}$$

$p_e(s)$ is the residue polynomial formed by the rest of the closed-loop poles

$$p_e(s) = s^t + e_{t-1}s^{t-1} + \ldots + e_1 s + e_0 \tag{27}$$

where $t = n + q - p$
It is known that

$$\delta(s) = p_c(s) - p(s)s^q \tag{28}$$

and it is possible to write

$$p_c(s) = p_d(s)s^t + \sum_{i=1}^{t} p_d(s)e_{t-i}s^{t-i} \tag{29}$$

where $\mathbf{e}$ is a vector formed by the unknown coefficients in polynomial $p_e(s)$. Then we can get

$$\delta = \delta_0 + \boldsymbol{D}_p\mathbf{e} \tag{30}$$

where δ_0 is a vector formed by the coefficients of $p_d(s)s^t - p(s)s^q$, $D_p = [\ d_1 \quad d_2 \quad \ldots \quad d_t\ ]$, where d_i is the coefficient of $p_d(s)e_{t-i}s^{t-i}$.

To assign p closed-loop poles with a compensator of order q, there are $ql + q + l - p$ free variables available out of the compensator parameters. Thus, partition $\mathbf{f}$ into two parts

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \tag{31}$$

where $\mathbf{f}_1$ is a vector that contains $ql + q + l - p$ elements formed by the free parameters of the compensator and $\mathbf{f}_2$ is a vector that contains p element formed by the parameters

of the compensator to be determined. Accordingly, matrix $\boldsymbol{Z}$ is partitioned as

$$\boldsymbol{Z} = [\boldsymbol{Z}_1 \quad \boldsymbol{Z}_2] \tag{32}$$

Then, it is possible to get

$$\widehat{\delta} = \widehat{\boldsymbol{X}}\ \widehat{\mathbf{f}} \tag{33}$$

where $\widehat{\boldsymbol{X}} = [\boldsymbol{Z}_2 \quad -\boldsymbol{D}_p]$ is a square matrix, $\widehat{\delta} = \delta_0 - \boldsymbol{Z}_1\mathbf{f}_1$, $\widehat{\mathbf{f}} = [\mathbf{f}_2 \quad \mathbf{e}]$. In this way, the p elements in $\mathbf{f}_2$ and the $t = n + q - p$ coefficients of the residue polynomial p_e can be found in terms of the $ql + q + l - p$ free parameters in $\mathbf{f}_1$

$$\widehat{\mathbf{f}} = \widehat{\boldsymbol{X}}^{-1}\ \widehat{\delta} \tag{34}$$

It should be noted that the system $[\boldsymbol{A} \quad \mathbf{b}_i \quad \boldsymbol{C}]$ should have at least p controllable and observable modes.

5.3 Pole retention

In each step except the final step of this algorithm, a constant compensator is applied and pole assignment is carried out by pole retention, that is simultaneous pole assignment and retention. The free variables in $\mathbf{k}_{y1}$ that occurs in partial pole assignment are used to make the assigned closed-loop poles uncontrollable (retained) from the rest of the inputs (pole retention) [15]. Combined with equation (34), the output-feedback compensator vector can be written as

$$\mathbf{k}_y = \begin{bmatrix} \mathbf{k}_{y1} \\ \begin{bmatrix} \boldsymbol{I}_p & 0 \end{bmatrix} \widehat{\mathbf{k}} \end{bmatrix} \tag{35}$$

$$= \begin{bmatrix} \mathbf{k}_{y1} \\ \begin{bmatrix} \boldsymbol{I}_p & 0 \end{bmatrix} \widehat{\boldsymbol{X}}^{-1}(\delta_0 - \boldsymbol{Z}_1\mathbf{k}_{y1}) \end{bmatrix} \tag{36}$$

To make the assigned poles uncontrollable (retained) from the rest of the inputs of the system, it needs to satisfy

$$\mathbf{Adj}(\gamma_k \boldsymbol{I}_n - \boldsymbol{A}_{\mathrm{cl}})\mathbf{b}_j = 0 \tag{37}$$

where $i \neq j$, $\boldsymbol{A}_{\mathrm{cl}}$ is the closed-loop system state matrix given by

$$\boldsymbol{A}_{\mathrm{cl}} = \boldsymbol{A} - \mathbf{b}_i\mathbf{k}_y'\boldsymbol{C} \tag{38}$$

It's been proven that $\mathbf{k}_{y1}$ enters into equation (37) linearly [15]. Moreover, the rank of adjoint matrix in equation (37) is 1. Therefore, matrix equation (37) can be satisfied by loosing $m-1$ degrees of freedom, that is only pick up any nonzero row of the equation. For step i, where $i = 1, 2, \cdots, m-1$, to make all the p poles assigned in this step uncontrollable from the rest $(m-i)$ inputs, $\mathbf{k}_{y1}$ has to satisfy $(m-i)p$ linear equations. In order to find a solution for $\mathbf{k}_{y1}$, the length of $\mathbf{k}_{y1}$ should be greater or equal to the number of equations, i.e.

$$l - p \geq (m-i)p \tag{39}$$

Thus there will be $l - p - (m-i)p$ extra degrees of freedom from $\mathbf{k}_{y1}$.

6 Symbolic computation in MATHEMATICA

MATHEMATICA, offering the capabilities of a symbolic computational environment, can handle the problem of output-feedback pole assignment in a rather straightforward way. Most of the inbuilt MATHEMATICA functions use natural names for the operations considered. Note that MATHEMATICA distinguishes between upper-case letters and lower-case letters, for example `MatrixForm` is different from `matrixForm`. To represent a parametric uncertain system given as

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{pmatrix} \qquad \boldsymbol{B} = \begin{pmatrix} q_1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{40}$$

the code in MATHEMATICA is

```
Amx = {{1, 0, 0}, {0, -2, 0}, {0, 0, -3}};
Bmx = {{q1, 0}, {0,1},{1, 0}};
Cmx = {{1, 0, 0}, {0, 1, 1}}
```

To calculate the transfer function of the system by using the relationship of

$$\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D} \tag{41}$$

with the relevant code being

```
g = Cmx.Inverse[s IdentityMatrix[n]-Amx].Bmx+Dmx
```

The package `Control Systems Professional` provides a convenient way for solving common problems in control design, which is applicable for both symbolic and numeric models. An alternative way to get the transfer function matrix model for the state-space system in equation (41), is the following

```
TransferFunction[StateSpace[Amx,Bmx,Cmx]]
```

Both methods involve the inverse operation of $(s\boldsymbol{I} - \boldsymbol{A})^{-1}$, which can be computationally expensive (based on the size of matrix $\boldsymbol{A}$). Caution must be taken when using matrix inverse operations especially for large parametric systems because it significantly slows down the solution process.

In addition, the command `ControllabilityIndices` (available in the `Controllability` package [10]) can be used to find all possible conditional controllability indices for parametric uncertain systems. For example, to find the controllability indices for a system described by equation (40), the code required is

```
ControllabilityIndices[Amx,Bmx]
```

and returns the following result

```
{{{2,1},q1!= 0},{{1,2},False}}
```

meaning that the controllability indices of the system are $\mu_c = \{2, 1\}$ when $q_1 \neq 0$. The command `ObservabilityIndices` was developed based upon `ControllabilityIndices` for finding observability indices

```
ObservabilityIndices[amx_,cmx_]}:=Module[{a,b}, a=amx;
b=Transpose[cmx]; ControllabilityIndices[a,b] ]
```

However, it is essential to write packages when implementing big algorithms in `MATHEMATICA`, either to be used standalone or to complement currently available ones. It is an efficient way to collect all algorithms/routines in a toolbox framework for easy use in problem solving. In this work separate packages were built for implementing the algorithms listed in section 1, and these are described in the following sections.

6.1 Dyadic method package

This package provides a set of commands to find a general output-feedback compensator for parametric uncertain systems (SISO, SIMO, MISO, MIMO) via Seraji's approach to dynamic compensator design.
First the package `OFBDyadic` needs loading

```
<<OFBDyadic‘
```

which returns the available commands

```
 OFBDyadic toolbox is loaded.
 Available commands:
 Seraji,SerajiMIMOToMISO, SerajiMIMOToSIMO,SymDyadicObser,SymDyadicContr
```

Help on any of the command can be obtained by typing `?Command_Name`, for example to get help on the `Seraji` command type

```
?Seraji
```

returning the full explanation on its usage

```
Seraji[amx,bmx,cmx,r,polescls,sQ] implements Seraji's dynamic output
-feedback compensator design method, for single-input or
single-output systems given by '[amx,bmx,cmx]', to assign the
closed-loop poles to the set of desired positions 'polescls',e.g
{-1,-2,-3+2i,-3-2i}; 'r' is the order of the dynamic compensator, it
returns  the generalised compensator and the conditions required for
the equations to have solutions. If [amx,bmx,cmx] is an uncertain
system with symboles in it, set 'sQ' to 1, or set sQ to 0
```

The following example illustrate the usage and the effectiveness of the developed packages.

Dyadic Compensator Example

Consider the system given by equation (40). We seek to design a first-order output-feedback compensator to place the poles of the resulting closed-loop system to the desired set $\Gamma = \{-1 \pm i, -4, -5\}$. Let $\mathbf{f} = [1,\ f_1]^{\mathrm{t}}$,

```
in[1]:= fmx={{1},{f1}}; % fmx is a column vector
        Bmx.fmx
out[1]= {{q1},{f1},{1}}
```

Vector $\mathbf{f}$ has to be pre-specified (if using a numerical environment), albeit this is not necessary in the symbolic computation as the solution of the final compensator is given in a symbolic (generic) form (symbols rather than numbers). Then we must check for the conditional controllability and observability indices of the corresponding pseudo single-input system $[\boldsymbol{A}, \mathbf{b},\ \boldsymbol{C}]$, where $\mathbf{b} = \boldsymbol{B}\mathbf{f}$. Thus write,

```
in[2]:= ControllabilityIndices[Amx,Bmx.fmx]
in[3]:= ObservabilityIndices[Amx,Cmx]
out[2]= {{{3},f1q1!= 0}}
out[3]= {{{2,1},False},{{1,2},True}}
```

The above confirms that the pseudo single-input system is completely controllable when $f_1 q_1 \neq 0$. Next the solution for $\mathbf{m^t}$ is to be found. Firstly, we represent $\mathbf{m^t}$ in symbolic form as defined from equation (4).

```
in[4]:= coeff1 = Array[f, {2, 2}]
out[4]= {{f[1,1],f[1,2]},{f[2,1],f[2,2]}}
in[5]:= coeff2 = Array[g, {1}]
out[5]= {g[1]} in[6]:= pt = Table[s^i, {i, 0, r}]
out[6]= {1,s}
in[7]:= NN = pt.coeff1; % the numerator Matrix
        polycom = coeff2.pt;  % characteristic polynomial of the
                              % compensator
        compensator = NN/polycom;
```

The resulting compensator from the above code is

$$\left\{\frac{\mathtt{f[1,\ 1]} + \mathtt{s\ f[2,\ 1]}}{\mathtt{s} + \mathtt{g[1]}},\ \frac{\mathtt{f[1,\ 2]} + \mathtt{s\ f[2,\ 2]}}{\mathtt{s} + \mathtt{g[1]}}\right\}$$

where the variables `f[1,1],f[2,1],f[1,2],f[2,2]`, i.e. the parameters of the compensator, are to be determined. These can be determined by calculating the closed-loop characteristic polynomial from equation (8), by first obtaining $\mathbf{w}(s)$ and $p(s)$.

```
in[8]:= G = TransFunction[Amx, Bmx.fmx, C];
        p = Det[s IdentityMatrix[3]-a];
        w = Flatten[Simplify[G p]];
```

Hence the open-loop transfer function is given by and the numerator matrix $\mathbf{w}(s)$ of

$$\left\{\left\{\frac{q1}{-1+s}\right\}, \left\{\frac{2+s+f1\ (3+s)}{6+5\,s+s^2}\right\}\right\}$$

the open-loop transfer function is

```
{q1(s+2)(s+3),(-1+s)(2+s+f1(3+ s))}
```

Thus, the closed-loop characteristic polynomial is calculated by using

```
in[9]:= pc = Simplify[Expand[p polycom + w.NN]]
```

and the solution obtained is

```
s^4 + 6 q1 f[1, 1] - 2 f[1, 2] - 3 f1 f[1, 2] - 6 g[1] +
 s (-6 + 5 q1 f[1, 1] + f[1, 2] + 2 f1 f[1, 2] + 6 q1 f[2, 1] - 2 f[2, 2] -
    3 f1 f[2, 2] + g[1]) + s^3 (4 + q1 f[2, 1] + (1 + f1) f[2, 2] + g[1]) +
 s^2 (1 + (1 + f1) f[1, 2] + q1 (f[1, 1] + 5 f[2, 1]) + f[2, 2] + 2 f1 f[2, 2] + 4 g[1])
```

According to the set of desired closed-loop poles Γ, the closed-loop characteristic polynomial can be also given as

$$p_c = (s+4)(s+5)(s+1-i)(s+1+i) = s^4 + 11\ s^3 + 40\ s^2 + 58\ s + 40 \qquad (42)$$

Matching the coefficients of like powers of 's' in equation (42) and `out[9]` gives a set of linear equations, which can be solved to obtain the parameters of the compensator. The associated code is

```
in[10]:=  equations = Map[(# == 0)&,CoefficientList[(pc - polyclsD),s]];
% polyclsD is (s+4)(s+5)(s+1-i)(s+1+i)
```

With the command `LinearEquationsToMatrices`, we can transfer the set of linear equations into the form of equation (10).

```
in[11]:= {matrixE, columnh}=
LinearEquationsToMatrices[equations,columnC]
out[11]= {{{-6,6,q1,-2-3 f1,0,0},
           {1,5 q1,1+2 f1,6 q1,-2-3 f1},
           {4,q1,1+f1,5 q1,1+2 f1},
           {1,0,0,q1,1+f1}},{40,64,39,7}}
```

where `columnC`= {`g[1],f[1,1],f[2,1],f[1,2],f[2,2]`} is the vector containing the variables to be determined. It is known that $q = 1$, $n = 3$ and $l = 2$, so $ql + q + l > q + n = 4$, thus one free variable is available from $\mathbf{m}^{\mathrm{t}}$. `g[1]` can be selected as the free variable so that we can choose the pole of the compensator freely, thus `columnC` changes to {`f[1,1],f[2,1],f[1,2],f[2,2]`}. The matrix $\boldsymbol{E}$ and vector $\mathbf{h}$ should be modified accordingly by dropping the first element of vector $\mathbf{c}$

```
in[12]:= {matrixE, columnh}=
LinearEquationsToMatrices[eq,Drop[columC,1]];
in[13]:= matrixE
out[13]= {{6 q1,-2-3 f1,0,0},
           {5 q1,1+2 f1,6 q1,-2-3 f1},
           {q1,1+f1,5 q1,1+2 f1},
           {0,0,q1,1+f1}}
in[14]:= columnh
out[14]= {40+6 g[1],64-g[1],39-4 g[1],7-g[1]}
```

The command `SymFullRankQ` [10] is used to find the conditions for the symbolic matrix $\boldsymbol{E}$ to have full rank

```
in[15]:= SymFullRankQ[matrixE]
out[15]= f1q1≠ 0
```

Then the solution to equation (10) is found

```
in[16]:= solv=Solve[equations, columC]
```

The Dyadic output-feedback compensator is the outer product of vectors $\mathbf{f}$ and $\mathbf{m}^t$, i.e. `f` and `compensator` in the `MATHEMATICA` environment. Finally, substitute `solv` into the expression of the compensator

```
in[17]:= comp = Simplify[Outer[Times, Flatten[f],
Flatten[compensator]]];
         compensator /. solv
```

which returns the following result for the compensator

$$\begin{pmatrix} \frac{8\,(-1+s)+5\,f1^2\,(-1+s)+f1\,(-2+27\,s+2\,g[1]-2\,s\,g[1])}{2\,f1\,q1\,(s+g[1])} & -\frac{5\,f1\,(2+s)+8\,(3+s)}{2\,f1\,(s+g[1])} \\ \frac{8\,(-1+s)+5\,f1^2\,(-1+s)+f1\,(-2+27\,s+2\,g[1]-2\,s\,g[1])}{2\,q1\,(s+g[1])} & -\frac{5\,f1\,(2+s)+8\,(3+s)}{2\,(s+g[1])} \end{pmatrix}$$

The compensator obtained above is a general parametric solution. The extra degrees of freedom are remained as a symbolic variables ,`f1` and `g[1]` in the compensator. Now it remains to check the position of the poles of the resulting closed-loop system which is

```
in[18]:= Solve[Det[s IdentityMatrix[3]-Amx+Bmx.comp.Cmx] == 0, s]
out[15]= {{s -> -5}, {s -> -4}, {s -> -1-i}, {s -> -1 + i}}
```

and confirms that they are placed in the exact position required from the specification given in the example.

The developed package for the Dyadic method provided an efficient way to get the general output-feedback matrix $\boldsymbol{F}$ in terms of uncertain parameters (q_1) and free parameters ($\mathbf{g}[1]$ and f_1) based upon over-parameterization along with the conditions for the compensator to exist, i.e. the union of all the conditions found during the design process such that all the closed-loop poles could be exactly assigned to the prescribed positions.

6.2 Full-rank method package

The purpose of this package development is to find general compensators for parametric uncertain systems via Munro's full-rank method [6] based on output-feedback pole assignment. As in the previous package case, the first step is to load the associated commands of the current package using

```
in[1]:= <<OFBFullRank‘
 OFBFullRank toolbox has been loaded.
```

```
available commands:
DegreeDetermine,MunroFullRankCheck,MunroFullRank,poleCombinFullRank.
```

Full Rank Compensator Example.

Lets consider the system given from equation (40), as in the case of the Dyadic compensator example. The command `DegreeDetermine[n_,l_,m_]` was developed based on the principles discussed in section 4 to find the minimum order of the compensator capable of assigning all closed-loop poles arbitrarily. To determine the order of the compensator we need the following,

```
in[2]:= DegreeDetermine[3,2,2]
out[1]= 0
```

This implies that a constant compensator (order 0) is sufficient to assign all the closed-loop poles for the system having 3 states, 2 inputs and 2 outputs. Next we choose the set of prescribed closed-loop pole locations $\Gamma = \{-4, -1 \pm i\}$. The controllability indices of the system are $\mu_c = \{2, 1\}$ under the condition that $q_1 \neq 0$. Also checking the observability of the system

```
in[3]:= ObservabilityIndices[Amx,Cmx]
out[2]= {{{2,1},False},{{1,2},True}}
```

returns two possibilities, i.e. (i) the situation of having 2 states observable from the first output and one state observable from the second output is impossible `{{2,1},False}`, while (ii) the case to have one state observable from the first output and two states observable from the second output is true `{{1,2},True}`. Overall, the above simply means that the system is fully observable although the command lists all possible cases of observability indices. Next we need to partition the set of desired closed-loop poles to m subsets, $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \cdots \Gamma_m$, according to Assumption 1. Command `poleCombinFullRank[poles_List,m_]` was developed for that particular operation returning several subset combinations. It is required that each subset should be closed under complex conjugations, which implies that complex pairs can only be assigned in the final step. Note that different compensators will be obtained under the consideration of different combinations. Designers have the freedom to choose any possible

combination. However, in the case of this example only one combination is possible, i.e.

```
in[4]:= polescls= poleCombinFullRank[{-4,-1+I,-1-I}]
out[3]= {{-4},{-1+I,-1-I}}}
```

Next, construct the full-rank compensator according to equation (11) as

```
in[5]:= Fc = Table[If[i >= j, fc[i, j], 0], {i, 2}, {j, 2}]
        Fo = Array[fo, {2, 2}]
out[4]= {{fc[1,1],0},{fc[2,1],fc[2,2]}}
out[5]= {{fo[1, 1], fo[1,2]}, {fo[2, 1], fo[2, 2]}}
```

Now, let the first element in matrix `Fc`, `fc[1,1]`, be equal to 1

```
in[6]:= Fc[[1,1]]= 1
```

then the assignment will be carried out in two steps as follows:

Step 1.

The first input of the system `b` is chosen by `Fc[[1]]={1,0}`

```
in[7]:= b = Partition[Bmx.Fc[[1]], 1]
out[6]= {{1},{0}}
```

Next `Fo[[1]]={fo[1, 1],fo[1, 2]}` is to be determined to assign one pole to $\Gamma_1 = \{-4\}$. `Fo[[1]]` can provide one free variable, say, `fo[1,1]` and it is required that `fo[1,1]`$\neq 0$. The corresponding **c** vector is simply `{fo[1,2]}` and a similar procedure as in the Dyadic method is used to obtain the solution.

$$\{\texttt{fo[1,2]} \rightarrow 1 - \tfrac{1}{5}\ \texttt{q1}\ \texttt{fo[1,1]}\}$$

Step 2.

in this step the system under consideration is

$$[\boldsymbol{A}_{\mathrm{cl}}^{(1)}, \boldsymbol{B}, \boldsymbol{C}] \tag{43}$$

where $\boldsymbol{A}_{\mathrm{cl}}^{(1)}$ is the closed-loop system state matrix, i.e. the result of pole assignment in the first step

$$\boldsymbol{A}_{\mathrm{cl}}^{(1)} = \boldsymbol{A} - \boldsymbol{B}\ \mathbf{f}^{(1)}\ \mathbf{m}^{(1)}\ \boldsymbol{C} \tag{44}$$

The related code for calculating (44) is given below

```
in[8]:= Acl=Amx-Apply[Plus, Table[Bmx.Outer[Times,Fc[[1]],Fo[[1]]].C{i, 1, j}]
            /. Flatten[Append[solvo, solc]]]
```

Now we can determine `Fc[[2]]={fc[2,1],fc[2,2]}` which renders the pole at position $\{-4\}$ uncontrollable by substituting $s = -4$ into equation (12). Next, one free variable is provided by `Fc[[2]]`, say, `fc[2,2]` and it is also required that `fc[2,2]`$\neq 0$. Due to the fact that the adjoint matrix **Adj**(.) is of rank one, matrix equation (44) can be solved by using just one nonzero row. In `MATHEMATICA` this is structured using

```
in[9]:= pp=AdjointMatrix[s IdentityMatrix[n] - Acl].B.Fc[[2]]/. s->-4;
          For[i=1, i<=2, i=i+1,   % find one nonzero row
              If [pp[[i]]=!= {0},
                   t=i; Break[]
                 ]
              ];
          equation=pp[[t]]==0;
          Solve[equation, Drop[Fc[[2]],-1]]
```

`out[7]= {fc[2,1]->`$\frac{1}{10}$`(-5fc[2,2]+q1fc[2,2]fo[1,1])`

The above command `AdjointMatrix` according to the mathematical definition of adjoint matrices. Finally, `Fo[[2]]={fo[2,1],f[2,2]}` is to be determined in order to assign the remaining portion of the closed-loop poles to $\Gamma_2 = \{-1+i, -1-i\}$ using the same procedure as in the dyadic design method. The resulting general full-rank compensator `F`, which is the outer product of `Fc` and `Fo`, is obtained as

$$\left\{\left\{\frac{25 + 5\,q1\,fo[1,\,1]}{20\,q1 - 4\,q1^2\,fo[1,\,1]},\ \frac{5}{4}\right\},\ \left\{-\frac{5\,(25 - 15\,q1\,fo[1,\,1] + 4\,q1^2\,fo[1,\,1]^2)}{2\,q1\,(-5 + q1\,fo[1,\,1])^2},\ \frac{5 + 4\,q1\,fo[1,\,1]}{2\,(-5 + q1\,fo[1,\,1])}\right\}\right\}$$

`F` is a general parametric solution with one extra freedom represented by the variable `fo[1,1]`. Finally, check the poles of the resulting closed-loop system

```
in[10]:= Solve[Det[s IdentityMatrix[3] - Amx + Bmx.F.Cmx]== 0, s]
out[7]:= {{s->-4},{s->-1+i},{s->-1-i}}
```

confirms that all of the poles are correctly assigned to the desired locations using the output-feedback compensator `F`.

6.3 Partial pole assignment package

The partial pole assignment package for implementing the algorithms by Soylemez and Munro [15, 14] to find the general output-feedback compensator for parametric uncertain systems. Again we first need to load the package,

```
in[1]:= << PartialPoleAssign‘
 PartialPoleAssign toolbox has been loaded.
 available commands:
 compenDegree,poleArrange,inputPermutation,ModCond,ModObs,
 PartailConstant,PartialDynamic.
```

Example 3. Consider the following system

$$\boldsymbol{A} = \begin{bmatrix} -1+q_1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 3+q2 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1+q_3 \\ 1 & 0 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{45}$$

where q_1, q_2 and q_3 are uncertain parameters (additional uncertain parameters compared to the previous cases). The matrices in MATHEMATICA are structured as

```
in[2]:= Amx={{-1+q1,0,0},{0,-2,0},{0,0,3+q2}};
         Bmx={{1,0},{0,1+q3},{1,0}};
         Cmx={{1,0,0},{0,1,1}};
```

The command `compenDegree`, developed particularly for this package, returns the least order of the compensator required to assign the poles of the system.

```
in[3]:= compenDegree[Amx, Bmx, Cmx, 1]
out[1]= 0
```

The result illustrates that a constant compensator assigns all the closed-loop poles arbitrarily. The required set of closed-loop pole locations is $\Gamma = \{-4, -3+I, -3-I\}$. According to assumption (2), Γ can be partitioned as $\{\{-4\}, \{-3+i, -3-i\}\}$, being the only possible combination.

Next, it is important to check both controllability and observability indices of the system by using

```
in[4]:= ControllabilityIndices[Amx,Bmx]
out[2]= {{{2,1},(-4+q1-q2)(1+q3)!= 0},{{1,2},False},{{3, 0},False}}
in[5]:= ObservabilityIndices[Amx,Cmx]
out[3]= {{{2,1},False},{{1,2},5 + q2!=0}}
```

According to the controllability indices, we can only choose to assign one pole, $\Gamma_1 = \{-4\}$, through the second input in the first step and then assign the rest of the closed-loop poles, $\Gamma_2 = \{-3+i, -3-i\}$, via the first input. The observability indices reveal that the system is fully observable as far as $q_2 \neq -5$. Next the assignment problem is solved in two steps,

step 1. Take the second column of the input matrix $\boldsymbol{B}$

```
in[6]:= b = TakeColumns[Bmx, {2}]; out[4]= {{0},{1 + q3},{0}}
```

The system given by `[Amx,b]` is under consideration. To find the lower-triangular Toeplitz matrix according to equation (21), we use command `XMatrix` built as

```
XMatrix[poly_] := Module[ {co, a}, co =
Reverse[CoefficientList[poly, s]]; a = Table[Which[ i == j,1,
                 i > j,k=i-j;co[[k + 1]],
                 i < j, 0 ],
     {i,Length[co]-1},{j,Length[co]-1}
   ]];
```

Then we can find the lower-triangular Toeplitz matrix for the open-loop characteristic polynomial by using

```
In[7]:= n = Length[Amx];
        Po = Det[s IdentityMatrix[n] - Amx];
        X = XMatrix[Po]
out[5]= {{{1,0,0},{-q1-q2,1,0},{-7+ q1-3q2+q1q2,-q1-q2,1}}
```

The controllability matrix of the open-loop system is found by [$\boldsymbol{A}$ $\mathbf{b}$]

```
in[8]:= W = ControllabilityMatrix[StateSpace[Amx,b,Cmx]]
out[6]={{0,0,0},{1+q3,-2(1+q3),4(1+q3)},{0,0,0}}
```

Note that command `ControllabilityMatrix` is available in the `Control Systems Professional` package of `MATHEMATICA`. Matrix $\boldsymbol{Z}$ in equation (16) can be calculated as

```
in[9]:= Z = X.Transpose[W].Transpose[Cmx]
```

Partition matrix $\boldsymbol{Z}$ according to equation (32)

```
in[10]:= p=1;
         l=Length[Cmx];
         Z1=TakeColumns[Z, l-p];
         Z2=TakeColumns[Z, -p];
```

Matrix $\boldsymbol{D}_p$ is calculated based on the mathematical definition in equation (30)

```
in[11]:= Pd = s+4 ;
         Dp = Transpose[Table[Reverse[Flatten[Append[CoefficientList
              [Pd s^(n-p-i),s], Table[0, {i - 1}]]]], {i, n - p}]]
out[6]= {{1, 0},{4, 1},{0, 4}}
```

and the matrix $\widehat{\boldsymbol{X}}$ in equation (33) is obtained by

```
in[12]:= XX = Simplify[AppendRows[Z2, -Dp]]
out[7]= {{1+q3,-1,0},
         {-(2+q1+q2)(1+q3),-4,-1},
         {(-1+q1)(3+q2)(1+q3),0,-4}}
```

Because inverse operation is used on matrix $\widehat{\boldsymbol{X}}$, it is important to check for full rank conditions of matrix $\widehat{\boldsymbol{X}}$

```
in[13]:= SymFullRankQ[XX]
out[8]= (3+q1)(7+q2)(1+q3)!= 0
```

For the output-feedback compensator vector, given in (35), let $\mathbf{k}_{y1}$ initially be described in symbolic form as

```
in[14]:= Ky1=TakeRows[Array[ky, {l, 1}], l - p] out[9]= {{ky[1, 1]}}
```

According to equation (35-37), $\mathbf{k}_{y1}$ is obtained

$$\mathbf{k}_{y1} = [-\frac{2(3+q_1)}{(7+q_2)(1+q_3)}] \tag{46}$$

therefore the second row of the feedback matrix $\mathbf{K}_y = \begin{bmatrix} \mathbf{k}_{(1)}^{\mathrm{t}} \\ \mathbf{k}_{(2)}^{\mathrm{t}} \end{bmatrix}$, is determined by equation (35) as

$$\mathbf{k}_{(2)}^{\mathrm{t}} = \begin{bmatrix} -\frac{2(3+q_1)}{(7+q_2)(1+q_3)} & \frac{2}{(1+q_3)} \end{bmatrix} \tag{47}$$

Step 2. In this step, the rest of the closed-loop poles, $\Gamma_2 = \{-3+i, -3-i\}$, are assigned. The system considered in this step is $[\ \boldsymbol{A}_{\mathrm{cl}}^{(1)},\ \ \mathbf{b}_1,\ \ \boldsymbol{C}\ ]$, where $\boldsymbol{A}_{\mathrm{cl}}^{(1)} = \boldsymbol{A} - \mathbf{b}_2\mathbf{k}_{(2)}^{\mathrm{t}}$, $\mathbf{b}_i$ is the i^{th} column of matrix $\boldsymbol{B}$. Since this is the final step, pole retention is no longer needed. There are two remaining poles to be assigned, thus $\mathbf{k}_{(2)}$ contains no extra degrees of freedom. Therefore, $\mathbf{k}_{(2)}$ is determined from equation (15) as

$$\mathbf{k}_{(1)}^{\mathrm{t}} = \begin{bmatrix} \frac{99+29q_2+2q_2^2+4q_1(5+q_2+q_1^2(5+q_2)}{(-4+q_1-q_2)(5+q_2)}, & -\frac{(7+q_2)(37+12q_2+q_2^2)}{(-4+q_1-q_2)(5+q_2)} \end{bmatrix} \tag{48}$$

Therefore the final compensator matrix `F` is

$$\{\{\frac{99 + 29\,q2 + 2\,q2^2 + 4\,q1\,(5 + q2) + q1^2\,(5 + q2)}{(-4 + q1 - q2)\,(5 + q2)},\ -\frac{(7 + q2)\,(37 + 12\,q2 + q2^2)}{(-4 + q1 - q2)\,(5 + q2)}\},\ \{-\frac{2\,(3 + q1)}{(7 + q2)\,(1 + q3)},\ \frac{2}{1 + q3}\}\}$$

In the above example, the general compensator obtained through the partial pole placement method doesn't have extra degrees of freedom. Finally checking the poles of the resulting closed-loop system

```
in[10]:= Solve[Det[s IdentityMatrix[3] - Amx + Bmx.F.Cmx]== 0, s]
% F is the final output feedback compensator
out[7]:= {{s->-4},{s->-3+i},{s->-3-i}}
```

shows that all poles are successfully moved to the desired locations.

Note that single commands can be used to get the final compensator for all of the above packages,although the details aim to present the way of package development in MATHEMATICA and also to illustrate the usefulness of incorporating symbolic algebra to the output-feedback assignment problem. Finally, a more complex example is to be considered.

Example 4. The open-loop system is

$$a = \begin{pmatrix} -2.12 & 1.31 & -2.8 & 2.23 & 0.1 & 6.8 & 1.41 & 4.623 \\ 3.2 & 0.21 & 5.6 & -4.55 & 1.94 & -8.57 & 1.13 & -5.32 \\ 0 & 0.2 & -3.25 & -6.5 & 0 & 0 & -5.6 & 0 \\ 3.23 & 0 & 5.7 & -4.21 & 0 & 0 & 3.42 & -5.23 \\ 5 & 1.31 & -2.7 & 2.23 & -0.825 & 0.51 & 5.54 & -1.375 \\ 0 & 0 & 0 & 0 & 3.23 & -6.8 & 0 & -3.23 \\ -4.3 & 0 & 0.5 & 0 & 0 & 0 & -6.77 & 4.13 \\ 4.7 & 1.29 & -2.75 & 2.13 & 0.09 & 7.1 & 5.61 & -2.125 \end{pmatrix} \tag{49}$$

$$b = \begin{pmatrix} 1 & 0.5 \\ 0 & 0 \\ 0 & 2 \\ -2 & -1 \\ -2.2 & -1.5 \\ 2.7 & -1 \\ 3 & 1 \\ 0 & -0.5 \end{pmatrix} \quad c = \begin{pmatrix} 1 & 3 & 0 & -5 & 0 & 3 & -2 & 1 \\ 5 & -1 & -2 & 0 & 0 & 8 & -1 & -5 \\ 1 & 2 & 2 & 0 & 2 & 3 & 1 & 1 \\ 0 & 0.3 & 0 & 0 & -2 & 2 & 1 & 2 \end{pmatrix} \tag{50}$$

To assign all the poles of the system by the partial pole assignment method, a first-order dynamic feedback compensator is needed. And it is required to assign the closed-loop system poles to

$$\Gamma = \{-10, -2+I, -2-I, -3+2I, -3-2I, -5+3I, -5-3I, -8+5I, -8-5I\} \tag{51}$$

Use the command `PartialDynamic` in package `PartialPoleAsssign`

```
In[1]:= F = PartialDyanmic[a,b,c,{{-7 + 5I,-7-5I},{-9,-1+I, -1-I,
        -3+2I,-3-2I, -5+3I, -5-3I}},1,{2, 1},0][[1]]
```

The poles $\Gamma_1 = \{-7+5I, -7-5I\}$ is assigned first through the second input and the poles $\Gamma_2 = \{-9, -1+I, -1-I, -3+2I, -3-2I, -5+3I, -5-3I\}$ is assigned through the second input. Thus, the general compensator `F` is obtained.

There are two extra degrees of freedom in the compensator, `f[1,1]` and `g[1]`. Because floating point numbers are used in the open-loop system, the numbers in the resulting compensator are of `machineprecision`, which is the machine-number precision.

$$\left\{\left\{\frac{f[1,1]+s\left((5363.24+8.71944\times10^{-11}\,i)+(0.0865883+5.19888\times10^{-16}\,i)\,f[1,1]-(0.00255141-2.53989\times10^{-16}\,i)\,g[1]\right)}{s+g[1]},\right.\right.$$

$$\frac{1}{s+g[1]}\left((13862.7+1.74222\times10^{-10}\,i)-(1.05333+2.28224\times10^{-15}\,i)\,f[1,1]+\right.$$

$$s\left((-1386.69+3.38041\times10^{-11}\,i)-(0.137778+3.80434\times10^{-16}\,i)\,f[1,1]-(0.00803306-5.98476\times10^{-17}\,i)\,g[1]\right)-$$

$$(0.00111019-3.78602\times10^{-16}\,i)\,g[1]\Big),\ \frac{1}{s+g[1]}\left((-24095.6+6.56051\times10^{-10}\,i)-(0.104135-2.2173\times10^{-15}\,i)\,f[1,1]+\right.$$

$$s\left((-3287.16-9.21637\times10^{-11}\,i)-(0.0411932-8.22994\times10^{-17}\,i)\,f[1,1]+(0.00470786-1.95118\times10^{-16}\,i)\,g[1]\right)-$$

$$(0.011259-1.65725\times10^{-15}\,i)\,g[1]\Big),\ \frac{1}{s+g[1]}\left((-38779.9+1.0486\times10^{-9}\,i)-(1.32175-2.17566\times10^{-15}\,i)\,f[1,1]+\right.$$

$$s\left((-26858.-7.51373\times10^{-12}\,i)+(0.19888+4.23888\times10^{-16}\,i)\,f[1,1]-(0.064528+8.56861\times10^{-17}\,i)\,g[1]\right)-$$

$$\left.\left.(0.390106-2.45173\times10^{-15}\,i)\,g[1]\right)\right\},$$

$$\left.\{0.866714-1.11716\times10^{-15}\,i,\ -0.516536-6.93889\times10^{-17}\,i,\ 0.0251655+7.53825\times10^{-17}\,i,\ -0.592127+1.65114\times10^{-15}\,i\}\right\}$$

Although Symbolic Algebra techniques offer advantages, it is worth noting that the computational effort increases as the dimensions of the system (the number of system states n, number of inputs m, and number of outputs, l) increases. Moreover, the number of uncertain parameters involved and the number of free variables undoubtedly have a large influence on the speed of computation and related memory usage. A large number of free variables and of uncertain parameters in the system will result in large intermediate expressions, having a detrimental effect on the speed. When numerical computations are also involved, this problem usually becomes more difficult with numerical tolerance and accumulating error issues. Advancing computer systems technology, i.e. faster processor capability, multi-processor systems, increased memory capability, can accommodate this. Another solution is to carefully setup the problem, reduce system size and complexity, in cases of large systems if possible, or to further manipulate the generalised algorithms to accommodate the computational burden for the large systems.

7 Conclusion

The paper presented a way of using symbolic algebra to efficiently implement three output-feedback pole assignment algorithms for parametric uncertain systems: dyadic method, full-rank method and partial pole placement. The objective is to obtain general over-parameterized compensators in a symbolic computation environment. It is seen that, based on symbolic computations, the algorithms can be realized in a

straightforward way when translated into the MATHEMATICA environment. The useful merit is the elimination of round-up errors by employing symbols with infinite precision such that the closed-loop poles can be assigned to the exact desired positions. The final compensators are obtained in terms of their free (symbolic) variables and the uncertain (symbolic) parameters of the system. The generic form of the resulting compensators is very useful for optimization procedures over the free variables usually to satisfy certain specifications associated with robustness of closed-loop systems.

8 Acknowledgement

This paper is written in memory of Professor Neil Munro who, very unfortunately, passed away in July 2004. For many years Professor Munro had dedicated his research in Computer Aided Control Systems Design, in particular robust control design for linear systems using symbolic techniques with their novel applications to control systems design including pole assignment algorithms. This is indeed the area originated by him and has received a wide-spread attention in our control system research community. In the late days of his life he was working very hard to formulate a toolbox for use with MATHEMATICA.

For many years Professor Munro had been the director of the UMIST Control Systems Centre. He made significant contributions to the development of the centre and was highly respected by the international control research community. His contributions and papers on Multivariable Control Theory, Computer-Aided Control Systems Design and Robust Design Methods have been highly regarded by the community and his papers have been very widely cited. He will be remembered as an excellent scientist, a wise mentor, a conscientious colleague and a reliable friend.

References

[1] Deliverable 4.1: Survey of existing tools for formal mkm. *Mathematical Knowledge Management Network,http://monet.nag.co.uk/mkm//index.html*.

[2] F. M. Brash and J. B. Pearson. Pole placement using dynamic compensators. *IEEE Transaction on automatic control*, 15(34):43, 1970.

[3] C. T Chen and C. H Hsu. Design of dynamic compensators for multivariable systems. *Proceedings of Joint automatic control conference*, pages 893–900, 1971.

[4] H.Seraji. An approach to dynamic compensator design for pole assignment. *International Journal of Control*, 21(6):955–966, 1975.

[5] N. Munro. Symbolic algebra tools for control teaching. *IEE Colloquium on Symbolic Computation for Control (Digest No: 1996/078)*, pages 1–7, 1996.

[6] N. Munro and S. Novin-Hirbod. Pole assignment using full-rank output-feedback compensators. *International Journal of systems science*, 10(3):741–748, 1975.

[7] Rajnikant V. Patel and N.Munro. *Multivariable system theory and design.* Pergamon, Oxford, 1982.

[8] B. Pearson and C. Y. Ding. Compensator design for multivariable linear systems. *IEEE Trans. Automat.Contr.*, 14:130–134, 1969.

[9] J. B. Pearson. Compensator design for dynamic optimization. *International Journal of Control*, 9(4):473–482, 1969.

[10] M. T. Soylemez. Robust pole assignment using symbolic algebra. *Control systems center, UMIST,UK*, 1994.

[11] M. T. Soylemez. *Pole assignment for uncertain systems.* Research studies press Ltd. Baldock, Herfordshire, England, 1999.

[12] M. T. Soylemez and N. Munro. Development of a robust eigenvalue toolbaxi using a kharitonov based approach. *IEE Colloquium*, (380), 1997.

[13] M. T. Soylemez and N. Munro. Pole assignment and symbolic algebra: a new way of thinking. *Control '98. UKACC International Conference on*, 2:1306–1310, 1998.

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

[15] M.T. Soylemez and N. Munro. A new technique for partial pole placement using constant output-feedback. *Proceedings of the 37th IEEE Conference on Decision and Control*, 2:1722–1727, 1998.

[16] Eno Tonisson. Step-by-step possibilities in different computer algebra systems. *Proceedings of ACDCA Summer Academy: Recent Research on DERIVE/TI-92-Supported Mathematics Education*, page `http://www.acdca.ac.at/kongress/goesing/g_toniss.htm`, 1999.

[17] P. C. Young and J. C. Willems. An approach to the linear multivariable servomechanism problem. *International journal of control*, (5):961–979, 1972.

Citation: ZHENG, X., ZOLOTAS, A.C., WANG, H., 2006. Mathematica implementation of output-feedback pole assignment for uncertain systems via symbolic algebra. International Journal of Control, 79 (11), pp. 1431-1446.