# Eco-inspired Robust Control Design for Linear Time-Invariant systems with Real Parameter Uncertainty

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Rohan V. Chandavarkar, B.Sc Graduate Program in Aeronautical and Astronautical Engineering

\* \* \* \* \*

The Ohio State University

2013

Master's Examination Committee:

Dr. Rama K. Yedavalli, Adviser

Dr. Chia Hsiang Menq

© Copyright by

Rohan V. Chandavarkar

2013

### ABSTRACT

This thesis addresses the importance and issues of the robust control design of linear time-invariant (LTI) systems with real-time parameter uncertainties. It is known that most of the existing robust control techniques are fairly conservative when dealing with real-time parameter uncertainty. Also, majority of these existing techniques use control gains that are essentially functions of the perturbation information. The robust control design algorithm proposed in this thesis differs from these traditional techniques by focusing on the control design in achieving a specific structure of the closed loop system matrix that guarantees a maximum stability robustness index as possible without the using any of the perturbation information. The determination of this specific desired structure of closed loop system matrix forms the focal point of this algorithm and is inspired by already existing principles in the field of ecology. Using this ecological backdrop, the desired closed loop matrix is determined to contain self regulated species with predator-prev interactions among these species. In matrix nomenclature, such a set of matrices are labelled as Target Pseudo-Symmetric (TPS) matrices and hence form the the class of the desirable closed-loop system matrices. Based on these TPS matrices, which capture the maximum robustness index for any LTI system, a robust control design is carried out such that the final closed loop system possesses a robustness index as close to this maximum as possible. The robust control design algorithm presented is based on minimizing the norm of an implicit error and is supported with several illustrative examples. This eco-inspired robust control algorithm exemplifies the strong correlation that exists between natural systems and engineering systems. Hence, the main goal of this thesis is to aid in the revival of research in the field of robust control using insights from ecological principles. Dedicated to my family for their never-ending love and support.

### ACKNOWLEDGMENTS

Firstly, I would like to acknowledge my research advisor, Dr.Rama. K. Yedavalli, for giving me this fantastic opportunity. In addition to financial support, he has also provided me with wonderful research insights while allowing me the freedom to put forth my own ideas. For this, I would like to thank him for all the intense and productive meetings without which this thesis could not be completed. I would also like to acknowledge Dr.Chia Hsiang Menq for serving on my thesis defense committee and for his valuable help during my defense.

Finally, none of this work could be made possible without the financial, emotional and unconditional support provided by my family. Therefore I would like to express my full gratitude because of the instrumental role they have played toward this academic achievement.

# VITA

June 2012	.(B.Sc.) The Ohio State University,
	Columbus, Ohio, USA.
September 2012 - Present	Graduate Research Associate, DECS
	lab, The Ohio State University

## FIELDS OF STUDY

Major Field: Aeronautical and Astronautical Engineering

# TABLE OF CONTENTS

## Page

Abstract
Dedication
Acknowledgments
Vita
List of Tables
List of Figures
List of Symbols
Chapters:
1. Overview of robustness theory and ecological principles
1.1       Introduction
1.3       Quantitative and Qualitative robustness example       1         1.4       Chapter 1 summary       2
2. The Target matrix formulation
<ul> <li>2.1 Quantitatively most robust matrices</li></ul>

		2.3.2 Improved set of TSS matrices: The Target Pseudo-Symmetric matrix	20
	2.4	Chapter 2 summary	$\frac{30}{32}$
3.	Eco-	inspired robust control design algorithm	33
	3.1	1. (m=n) open-loop systems	35
	3.2	2. $(m < n)$ open-loop systems $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	37
		3.2.1 Stable open-loop systems	38
		3.2.2 Eco-inspired robust control algorithm summary	47
		3.2.3 Heuristic proof of closed-loop system stability	48
		3.2.4 Unstable open-loop systems	55
	3.3	Chapter 3 summary	58
4.	Eco-	inspired robust control design examples	59
	4.1	Aircraft flight control:	59
	4.2	Satellite Attitude Control:	65
5.	Num	nerical Implications, conclusions and future work	69
	5.1	Numerical Implications	69
		5.1.1 Numerical implication on Aircraft control example	70
	5.2	Conclusions and Future work	71
Bib	liograp	phy	73

# LIST OF TABLES

Table	P	age
1.1	Results: $\beta_i$ s' for S, K and H	19
1.2	Results: Stability degree $\alpha$ , condition number $\kappa$ , and $\mu_s$ for S, K and H	20
1.3	Results: Effect of $\beta$ on $\mu$ for S, K and H $\ldots \ldots \ldots \ldots \ldots$	21
3.1	Three cases for stable $(m < n)$ open-loop systems $\ldots \ldots \ldots$	50
3.2	$\mu$ results	53
4.1	Optimization phase results	61
4.2	Robustifying phase results	62
4.3	LQR results	63
4.4	Optimization phase results	66
4.5	Robustifying phase results	67
4.6	LQR results	68
5.1	Effect of parametric gridding on resolution of results	71

# LIST OF FIGURES

Figu	re	Page
1.1	Different types of uncertainties $[1]$	. 5
1.2	Types of interactions $[2]$	. 11
1.3	A qualitative matrix and its corresponding digraph $[2]$	. 12
1.4	Interactions: l-cycles $[2]$	. 13
1.5	Interconnections: k-cycles [2]	. 13
1.6	Sufficient condition for stability and instability $[2]$	. 16
1.7	Effect of $\beta_1$ on qualitative stability $(\beta_2 = \beta_3 = 0)$ [2]	. 17
1.8	Effect of $\beta_2$ on qualitative stability $(\beta_1 = \beta_3 = 0)$ [2]	. 17
1.9	Effect of $\beta_3$ on qualitative stability $(\beta_1 = \beta_2 = 0)$ [2]	. 18
2.1	Set of Target matrices [2]	. 27
2.2	Classification of target matrices [1]	. 31
3.1	Flowchart summary of the Robust Control design algorithm	. 46
3.2	Proof of guaranteed closed-loop system stability	. 51

## LIST OF SYMBOLS

$\mathbf{A}$	Open-loop sytem matrix
В	Control variables' dynamics matrix
â	State-variable vector
û	Control/input variable vector
G	Full-state feedback gain
${f A_{cl}}$	Hurwitz stable closed-loop matrix
E(t)	Error due to time-varying parametric variations (perturbations)
$\mu$	Stability degree-based quantitative robustness index
$\alpha$	Stability degree
$\kappa$	Condition number of modal matrix
i	Elemental row number
j	Elemental column number
$\mathbf{a}_{\mathbf{ij}}$	Elemental value located in row i and column j
$\beta_1$	Qualitative metric 1
$\beta_2$	Qualitative metric 2
$eta_{3}$	Qualitative metric 3
eta	Qualitative robustness index
$\mathbf{A}_{\mu}$	Set of quantitatively most robust matrices
$\mathbf{A}_eta$	Set of qualitatively most robust matrices
TSS	Target Sign-Stable
TPS	Target Pseudo-Symmetric
d	Diagonal element in a TPS matrix
$\mathbf{A_t}$	Design target matrix
$\mathbf{A_{T}}$	Optimized target matrix
n	Number of state variables
m	Number of control variables
$B^{\dagger}$	Pseudo-inverse of B
$E_{pseudo}$	Pseudo-inverse error
$E_{TPS}$	Pseudo-symmetric error
E	Total error
$E_{min}$	Minimized error
$\overline{G}$	Intermediate gain via optimization phase

$\delta$	Elemental design variables in the target matrix
$B_{aug}$	Product of $B$ and $B^{\dagger}$
Ι	Identity matrix
$B^+$	$B_{aug}$ subtracted from $I$
$A_{t_{ii}}$	The maximum elemental design variable range value
$\psi$	Number of off-diagonal pairs
$\mathbf{S}$	Number of sign patterns
$\mathbf{G}_{\mathbf{pp}}$	Robustifying pole-placement gain
$ G_{max} $	Maximum elemental gain value
$G_f$	Final closed-loop system gain
$\mu_{ol}$	Open-loop robustness index
$\overline{\mu}$	Intermediate robustness index
$\mu_T$	Optimized target robustness index
$\mu_{cl}$	Closed-loop robustness index
ho	LQR design variable
$J_u$	Control effort
au	Robustness to control effort index
$\Delta\delta$	Elemental step-sizing/Parmetric gridding index

### CHAPTER 1

## OVERVIEW OF ROBUSTNESS THEORY AND ECOLOGICAL PRINCIPLES

### 1.1 Introduction

As education has continued to grow over the past few decades, it has had a tremendous influence on many engineering principles and designs. Education has laid a common foundation allowing several different fields to learn from each other. The field of controls, for example, has benefited greatly from education because of the sharing of different ideas that span across various disciplines and industries. A very interesting feature of controls is that it is very different from many other engineering specializations because it can be implemented in almost any sense of the word *control*. Its uses can be extended from controlling the mass flow of air in an engine to controlling the cash flow in a business. It can even be used to identify exactly how natural systems work. In fact, this entire research hinges on the previous statement and this is all made possible because of the growth of the value of education.

Control is primarily important because it allows us to achieve a desired result. Stability is one of the most important features in a control design. If a system is unstable, it's behavior is hard to predict and hence it would not output a desired result (which is what control is all about). In fact, most control designs that focus on other performance specifications require to guarantee a system's stability before achieving any of these desired specifications. Another important specification in controls is the feature of robustness.

Most real-life engineering systems today are of a non-linear nature. This nonlinearity introduces several complexities that can only be solved through the use of different numerical methods and softwares. Hence, many of these systems that are non-linear in nature are linearized about a nominal state. The resulting linearized system is much easier to solve using control designs and techniques that do not necessarily require the use of numerical softwares. When a non-linear system is linearized, it can only withstand a certain magnitude of perturbation from that nominal state beyond which it becomes unstable. Hence it can be pointed out that the choice of the nominal state plays a huge role in this system's tolerance to perturbations [2]. This is where the feature of robustness ties into a control design. A system is deemed to be more robust if it can tolerate a higher perturbation from its nominal state. Hence by achieving a higher robustness index, a system is able to withstand a higher perturbation without losing stability. However, it is extremely important to keep in mind that robustness is an innate feature of stability. If a system is unstable from the start, then there can be no concept of robustness associated with it.

Many engineering systems adapt their ideas by examining the workings of natural systems that occur everyday. Previous research has been carried out by ecological scientists studying the variations and behaviors of natural systems. The research in those fields allowed an understanding of how and why some natural systems remain stable, decay or grow out of bound. These principles can be extended to engineering systems in order to gain a similar perspective in the engineering sense. Hence, this research deals with using these ecological principles and ideas to form a control design that maximizes the robustness index of linear systems while maintaining stability which is why it is called an *Eco-inspired robust control design*.

#### **1.2** Literary survey and motivation

As mentioned previously, it should be clear that the choice of a nominal system should be such that the bound on a perturbation is maximized [2]. Therefore it is essential that we choose the best nominal system that allows this maximized perturbation. The above definition of robustness is a conventional one that basically quantifies the value of the robustness. However, robustness can also be defined by its qualitative attributes. As a result, there are two perspectives of robustness: *Quantitative robustness* and *Qualitative robustness*. The qualitative aspect of robustness comes purely from the ecological principles derived from previous research while the quantitative aspect deals with the relationship between the eigenvalues and eigenvectors a system. Both robustness definitions are explained in detail as follows.

#### **1.2.1** Quantitative robustness:

Once an engineering system has been linearized about its nominal state, it is called a linear system. This research is primarily concerned with linear time invariant systems (LTIs); this basically means that the system's dynamics do not vary with time. An LTI system primarily consists of two types of variables: state variables and control variables. The state variables are a set of parameters that can completely describe a system. The control variables, on the other hand, are parameters that help control the system. Examples of control variables on an aircraft can consist of the elevators on the tail fins, or the flaps on a wing. Both the state and control variables along with their respective dynamics can be written in a state-space form given by:

$$\dot{x} = Ax + Bu \tag{1.1}$$

In equation 1.1, x denotes the state variables of the system and u denotes the control variables; A contains the dynamics of the state variables of the system; B contains the dynamics of the control variables.

Equation 1.1 refers to an open-loop system and thus can be regarded as a stable or unstable system. In order to achieve a guaranteed stable system, a full-state feedback is implemented wherein every state variable can be measured. Hence the gain needed to convert this open-loop system to a closed-loop stable system is given by:

$$\hat{u} = -Gx \tag{1.2}$$

As a result, the final state-space form of an LTI system with a full-state feedback can be written as:

$$\dot{x} = [A - BG]x \quad or \quad \dot{x} = A_{cl}x \tag{1.3}$$

Here,  $A_{cl}$  constitutes a Hurwitz stable matrix and is called the closed-loop system matrix. We are primarily concerned with the amount of perturbation this closed-loop system matrix can tolerate. There are primarily 4 different types of perturbations (errors) that can be associated with any engineering system that are broadly classified as [1]:

i. Error due to real parametric variations/uncertainties.

ii. Error due to neglected non-linearity.

- iii. Error due to external disturbances.
- iv. Error due to model-reduction (unmodeled dynamics).



Figure 1.1: Different types of uncertainties [1]

This research is only concerned with errors that are caused due to parametric variations/uncertainties. Hence such errors can be classified as perturbations, or parametric variations. Parametric variations can occur in almost any kind of system. For example, say one of the state variables in a turbine engine is the mass-flow of air through that engine. Now assume that due to some fatigue/failure, a hole was to form somewhere in that engine. This would change the mass-flow in that engine and this change from its original value is referred to as a parametric (state) variation. Therefore it is important to incorporate this perturbation into equation 1.3. This can be shown by:

$$\dot{x} = [A_{cl} + E(t)]x$$
 (1.4)

where: E denotes the perturbation/error.

As far as this research goes, the perturbation E is classified as one that is timevarying. The main question now is how large of an E can be tolerated before the system goes unstable. In addition, this research is only concerned with the unstructured (norm-bounded) uncertainty of E [3] and so it can be seen that the norm-bound of E is regarded as a measure of robustness of the system. Therefore, the higher the bound on E that can be tolerated, the more robust  $A_{cl}$  is [2].

It is now an appropriate time to define quantitative robustness. Quantitative robustness is based on the unstructured, norm-bounded uncertainty of E. It is denoted by  $\mu$  and depends on the stability degree of the system. It is because this robustness measure is calculated by quantitative information, it is labeled as quantitative robustness.

Note: There is another measure of quantitative robustness that is based on the Lyapunov matrix, however, it was never used in this research and hence shall not be addressed in much detail.

Again, recalling that robustness is a feature of stability,  $A_{cl}$  needs to be a Hurwitz stable matrix. This was guaranteed via a full-state feedback gain G given in equation 1.2. The quantitative robustness of  $A_{cl}$  can now be defined by [2]:

$$||E|| < \mu_{cl} = -\frac{\alpha_{cl}}{\kappa_{cl}} \tag{1.5}$$

where:  $\alpha_{cl}$  denotes the stability degree (most dominant eigenvalue of  $A_{cl}$ );  $\kappa_{cl}$  denotes the condition number of the modal matrix of  $A_{cl}$ . Therefore it can be seen that  $\mu_{cl}$  is basically a function of the closed-loop system's eigenvalues and eigenvectors.

Note: Since only robustness based on the 'stability degree' is used as the definition of quantitative robustness in this research, it shall be denoted by  $\mu_s$ .

Therefore the higher the value of  $\mu_s$ , the higher the perturbation it can tolerate. Quantitative robustness is important because it provides a numerical value, or quantity, that shows how large of a perturbation a system can be tolerated. Although quantitative robustness provides a measure of the amount of perturbation that can be tolerated (robustness index), it does not, however, provide any details as to the structure of the elements in  $A_{cl}$  that are needed to maximize this index. In fact it is very crucial to understand the internal workings of  $A_{cl}$ 's elements in order to make sure that the robustness bounds are maximized. Therefore a qualitative perspective of robustness was found that was adapted from research carried out in the field of ecology. This research addressed the signs, interactions and interconnections of the different elements in  $A_{cl}$  that bring about this qualitative information of robustness.

#### 1.2.2 Qualitative robustness

#### **1.2.2.1** Review of Ecological principles.

Ecosystems consist of many different types of species. These various species play different roles in an ecosystem. Mathematical models of ecosystems are constructed by studying the effect a species has on itself and its surrounding species [4], [5]. Moreover, the workings of an ecosystem depend on the sustenance or extinction of different species in that ecosystem. Every species obviously has an effect on itself based on how it feeds, how healthy it is etc. However it is also essential to understand the effect a species has on another species. Therefore the relationship between two species is characterized as an *interaction*. The type of interaction is very important to the understanding of the sustenance or extinction of those species. There are five primary classifications of interactions between any two different species in an ecosystem.

i. **Mutualism**: This interaction takes place when both species have a positive effect on one another. For example, an ant-plant interaction is a type of mutualism wherein the plant provides nectar to the ants, while the ants help keep away insects that would harm the plants.

ii. **Competition**: This interaction occurs when both species have a negative effect on one another. An example of this is a predator-predator interaction. Both predators target the same prey and hence induce a competition with one another.

iii. **Commensalism**: This sort of interaction takes place when one species has a positive effect on the other while the other species has a neutral effect on the former. For example, barnacles adhere to the skin of whales using them as a mode of transport to gain nutrients in richer water. Hence, the whales have a positive effect on the barnacles, but the barnacles do not harm the whales in any way and hence play a neutral role.

iv. **Ammensalism**: This interaction is similar to commensalism but in this case one of the species induces a negative effect on the other. An example of ammensalism is between humans and another species that is under extinction due to human actions such as pollution etc. Here, the species under extinction is affected negatively by humans, however humans, in return, aren't necessarily affected at all.

v. **Predation**: This is a very common interaction wherein one species has a negative effect on the other, while the other species has a positive effect on the former. A predator-prey interaction is an example of predation.

All five of these interactions can be illustrated with the use of signs. For example, if a species has a positive effect on another, it is denoted by a + sign; if a species has a negative effect on another, it is denoted by a - sign; and finally if a species has a neutral effect on another, it is simply denoted by a 0.

Just like engineering systems, ecosystems are predominantly non-linear in nature and hence they too are linearized about some nominal point. It makes intuitive sense that ecosystems only contain information about the type of interactions between species. Therefore there is an ambiguity in the quantitative information within these interactions and so when these systems are linearized, the resulting linear system does not contain numerical values, but instead, only contain the signs (+, -, 0) of the interactions between the species. Hence only the *qualitative* information is included in these systems [2].

Once an ecosystem has been linearized, a matrix can be formed that contains information about the interactions between he species that constitute the system. The manner in which this information is translated into a matrix form is by assigning every diagonal element in this matrix as a species and hence the off diagonal elements serve as an interaction between two species in this matrix. Every element in this matrix can be identified by its row (i) and column (j). For instance (1,2) denotes the element that is in the first row and second column. As mentioned previously, since this matrix only contains qualitative information (ie. signs), and hence no numerical values, an  $(i,j)^{th}$  element in this matrix basically denotes species j's effect on species i. Hence if the same (1,2) element is identified by a + sign, this only means that species 2 has a positive effect on species 1. The sign on the diagonal elements, on the other hand, denotes the species' effect on itself. Figure 1.2 below illustrates this concept formulation.

Interaction	Digraph	Matrix	
type	representation	representation	
Mutualism		$\begin{bmatrix} * & + \\ + & * \end{bmatrix}$	
Competition		[* _ ] _ *]	
Commensalism	1+2	$\begin{bmatrix} * & + \\ 0 & * \end{bmatrix}$	
Ammensalism		$\begin{bmatrix} * & - \\ 0 & * \end{bmatrix}$	
Predation (Parasitism)		[* _ + *]	

Figure 1.2: Types of interactions [2]

Figure 1.2 illustrates two ways of representing any interaction type through a *digraph* and *matrix* form. A digraph can be shown to consist of two terms: *nodes* and *paths*. Nodes (the numbered circles) indicate the species itself while paths (the arrows) indicate the interaction between the species. Figure 1.3 below gives us a better understanding of this concept.



Figure 1.3: A qualitative matrix and its corresponding digraph [2]

It is essential to define these representations in terms of linear algebra nomenclature. The product of paths (in matrix notation) with ordered indices are classified as follows [2] [6]:

i. The product of any pair of off-diagonal elements  $a_{ij}a_{ji}$  are known as *l-cycles*. This product indicates the interaction of only two distinct nodes in the matrix. Hence, the product  $a_{ij}a_{ji}$  and  $a_{ji}a_{ij}$  are the same.

ii. When products of three or more nodes  $(a_{ij}a_{ji}...a_{mi})$  are under consideration, they are known as k - cycles.

l-cycles are referred to as *interactions* because they occur only between two nodes (species). k-cycles are referred to as *interconnections* because they occur between three or more nodes. The l-cycles and k-cycles are illustrated in figures 1.4 and 1.5 respectively.

$$A = \begin{bmatrix} 0 & +a_{12} & +a_{13} \\ -a_{21} & -a_{22} & 0 \\ +a_{31} & +a_{32} & +a_{33} \end{bmatrix}$$

Figure 1.4: Interactions: l-cycles [2]



Figure 1.5: Interconnections: k-cycles [2]

#### 1.2.2.2 Ecological sign stability

Ecological sign stability is a qualitative property that is of huge importance that can be extended to engineering systems. As mentioned above, the ecological matrices formed were of a qualitative nature in that there were no numerical values associated with the elements in the matrix (only signs). Sign stability is defined as follows: A matrix that is 'sign stable' conforms to Hurwitz stability solely because of the nature of its sign pattern regardless of what the magnitudes of the individual elements in this matrix are [2].

This is a very fascinating feature because most engineering systems contain magnitudes in their matrices. Therefore if such an engineering system matrix conforms to a sign-stable matrix, it's elements can take on any values of magnitude (other than zero) and would still remain a Hurwitz stable matrix. In other words, the stability of the matrix solely depends on the sign pattern of the matrix!

It is fairly intuitive that there only exists a certain variety of sign patterns that would provide sign stability. Previous research [7] has shown that there is a set algorithm that stores all sign stable matrices of order 3 or more. For example the following matrices are considered to be sign stable matrices:

$$A = \begin{bmatrix} - & - & - \\ 0 & - & 0 \\ + & + & - \end{bmatrix} \qquad B = \begin{bmatrix} - & - & - & - \\ 0 & - & - & 0 \\ 0 & + & - & 0 \\ + & 0 & + & - \end{bmatrix}$$
(1.6)

If the above matrices are properly scrutinized, it can be shown that the product of the 'k-cycles' in both matrices equal to 0. For example the product  $a_{13}a_{32}a_{21}$  is 0 in matrix A and the product  $a_{14}a_{43}a_{32}a_{21}$  is also 0 in matrix B. Hence a general rule of thumb for sign stability is that if every k-cycle in a matrix is 0, the matrix is sign stable. Research in the past [8] [9] [10] produced many different models and algorithms that were used to determine the sign stability of matrices. It turns out that from all the different interaction types discussed earlier, only the 'predation' and 'commensalism/ammensalism' interactions (along with zeros placed in appropriate locations needed for a 0 product 'k-cycle') resulted in sign-stable matrices. Therefore 'mutualism' and 'competition' interactions, along with non-zero k - cycles are unfavorable to sign (qualitative) stability.

It is important to impart the feature of sign-stability into robustness theory. Since robustness is a feature of stability and since sign-stability is independent of magnitude, it allows these different stable sign patterns to have an important effect on robustness that can be characterized as a *qualitative robustness index*.

#### 1.2.2.3 Defining qualitative robustness

Ecosystems are known to be inherently robust to various disturbances and perturbations. Therefore it makes intuitive sense to try and adapt these ecological principles and apply them to engineering systems to increase their robustness. As mentioned earlier, there are some interactions that enhance the robustness of a system while the others are detrimental to robustness and stability. Hence, different frameworks/metrics that assess the qualitative robustness of a system are presented as follows. By defining [2]:

$$\beta_1 = \frac{no. of a_{ii} > 0}{total no. of a_{ii}}$$
$$\beta_2 = \frac{no. of l - cycles > 0}{total no. of l - cycles}$$
$$\beta_3 = \frac{no. of k - cycles = 0}{total no. of k - cycles}$$

Another cumulative metric  $\beta = \beta_1 + \beta_2 + \beta_3$  can also be formed. This cumulative metric is a measure of qualitative robustness. If this metric  $\beta$  is higher, then the less robust the system is qualitatively [2]. Also it should be noted that since each of these robustness indices that make up the cumulative  $\beta$  are ratios,  $\beta$  can have a minimum value of 0 and a maximum value of 3. Hence the most qualitatively robust matrix would have a  $\beta$  value of 0. If a matrix has a  $\beta$  value of 3, it not only means that it the least robust qualitatively but it also implies that the matrix is unstable.

Since ecological matrices only contain signs, it can be seen that each of the  $\beta_i$ (i=1,2,3) values can be obtained fairly easily. Hence, based on the definition of  $\beta$ above, a sufficient condition can be put forward:

A quantitative matrix is always Hurwitz stable if  $\beta$  is 0. [2]

Using the statement above, a reverse condition can also be invoked that is a quantitative matrix is always unstable if  $\beta$  is 3. Figure 1.6 below illustrates this sufficient condition.



Figure 1.6: Sufficient condition for stability and instability [2]

While Figure 1.6 exemplifies the effect of the cumulative  $\beta$  on stability, it is also important to see the individual contributions of each  $\beta_i$  to the qualitative stability of matrices. Figures 1.7, 1.8 and 1.9 show these individual contributions [2].



Figure 1.7: Effect of  $\beta_1$  on qualitative stability  $(\beta_2 = \beta_3 = 0)$  [2]



Figure 1.8: Effect of  $\beta_2$  on qualitative stability  $(\beta_1 = \beta_3 = 0)$  [2]



Figure 1.9: Effect of  $\beta_3$  on qualitative stability  $(\beta_1 = \beta_2 = 0)$  [2]

Figure 1.7 shows that when  $\beta_2$  and  $\beta_3$  are kept at zero, and as  $\beta_1$  is increased from 0 to 1, the probability of stability drops from 1 to 0. Further delving into the field of linear algebra, it can be shown that if every diagonal element in a matrix is positive then that matrix will contain at least one positive eigenvalue leading to an unstable matrix. Hence by referring to the definition of  $\beta_1$ , it can be shown that if  $\beta_1 = 1$  it serves as a sufficient condition for instability. Furthermore, if figures 1.7, 1.8 and 1.9 are closely observed, it can seen that  $\beta_1$  has the most critical effect on stability while  $\beta_3$  has the least effect on the same.

Up to this point, qualitative and quantitative robustness have been described in detail. It may seem that these two measures are independent of each other because qualitative robustness is independent of magnitude. However, that is not the case and in order to provide a better understanding of how these two robustness measures affect each other, an example is proposed as follows.

### 1.3 Quantitative and Qualitative robustness example

The following example was adapted from [2].

Consider the following three matrices:

$$S = \begin{bmatrix} -3 & -1 & -0.8 & -4.8\\ 0 & -1.2 & -1.5 & 0\\ 0 & 2 & -0.4 & 0\\ 2.6 & 0 & 4 & -5 \end{bmatrix} \qquad K = \begin{bmatrix} -3 & -1 & -0.8 & 4.8\\ 0 & -1.2 & -1.5 & 0\\ 0 & 2 & -0.4 & 0\\ 2.6 & 0 & 4 & -5 \end{bmatrix}$$
$$H = \begin{bmatrix} -3 & -1 & -0.8 & 4.8\\ 0 & -1.2 & -1.5 & 0\\ 0 & 2 & 0.4 & 0\\ 2.6 & 0 & 4 & -5 \end{bmatrix}$$

Note that all three matrices are Hurwitz stable and each of their elements have the exact same absolute magnitudes. The only difference in the three matrices are the signs of some of their individual elements. This can be summarized as follows:

1. S, K and H are all stable.

2. 
$$|S_{ij}| = |K_{ij}| = |H_{ij}|.$$

3.  $\operatorname{sign}(S) \neq \operatorname{sign}(K) \neq \operatorname{sign}(H)$ .

 $\beta_1,\,\beta_2,\,\beta_3$  and  $\beta$  for S, K and H are given in the table below:

	S	K	Η
$\beta_1$	0	0	0.25
$\beta_2$	0	0.1667	0.1667
$\beta_3$	0	0	0
$\beta$	0	0.1667	0.4167

Table 1.1: Results:  $\beta_i$ s' for S, K and H

Therefore it can be seen that S is the most qualitatively robust matrix (lowest  $\beta$ ) while H is the least qualitatively robust matrix (highest  $\beta$ ). However, it is interesting to see how this qualitative robustness index affects the quantitative robustness values  $(\mu_s)$ . The quantitative robustness index values  $(\mu_s)$  for each of the three matrices are shown in the table below. Again, note that  $\alpha_s$  is the stability degree (the real part of the most dominant eigenvalue) and  $\kappa_s$  is the condition number of the modal matrix of each of the three matrices.

	S	Κ	Η
$\alpha_s$	0.8	0.3285	0.3285
$\kappa_s$	2.7838	4.3525	5.7923
$\mu_s$	0.2874	0.0755	0.0567

Table 1.2: Results: Stability degree  $\alpha$ , condition number  $\kappa$ , and  $\mu_s$  for S, K and H

Recalling the definition of quantitative robustness,  $\mu_s$  is a measure of the amount of perturbation that can be tolerated from a system's nominal state. Hence a higher  $\mu_s$  value indicates a higher tolerance (and therefore more robust) to perturbations. Looking at the the table above, we can see that S has the highest  $\mu_s$  index indicating that it is yet again the most robust matrix while H is the least robust matrix (again).

The summarized results showing the inverse effect of  $\beta$  on  $\mu$  are shown in Table 1.3 below:

	S	Κ	Η
β	0	0.1667	0.4167
$\mu_s$	0.2874	0.0755	0.0567

Table 1.3: Results: Effect of  $\beta$  on  $\mu$  for S, K and H

Therefore it seems that the sign pattern of S that made it the most qualitatively robust matrix allowed it to have the maximum quantitative robustness index among the three matrices. In other words, the qualitative nature of S maximized the tolerance of S to perturbations, hence making it a more robust matrix than the other two. Hence it is obvious that both the qualitative and quantitative robustness indices are very closely tied in to each other.

### 1.4 Chapter 1 summary

Up to this point, two measures of robustness have been provided: quantitative robustness ( $\mu_s$ ) and qualitative robustness ( $\beta$ ). The first measure is important because it provides numerical information on the amount of perturbation that can be tolerated from the nominal state of any system. The latter measure provides an index pertaining to the sign structure of a matrix that is needed to maximize the former measure. Both measures are equally important and together provide the founding framework for this research.

The following chapter deals with narrowing down to exactly which set of sign and magnitude patterns in a matrix provide the best closed-loop nominal systems that can be used to maximize robustness.

### CHAPTER 2

### THE TARGET MATRIX FORMULATION

In the previous chapter, the importance of stability was addressed and it was pointed out how robustness is an innate feature of stability. It was also shown how qualitative and quantitative robustness are affected by each other and the goal of this chapter is to define the best qualitatively robust matrix (least  $\beta$ ) that would serve as a reference to maximize a system's quantitative robustness index (maximize  $\mu_s$ ).

Recall that the choice of the nominal state which a system is linearized about plays a huge role on the system's tolerance to perturbations. For convenience, equation 1.4 is reiterated below.

$$\dot{x} = [A_{cl} + E(t)]x$$

Here  $A_{cl}$  serves as the nominal closed-loop system matrix. Hence this chapter is going to focus on identifying a set of the most qualitatively and quantitatively robust matrices. Based on the properties of these matrices, the best selection for the nominal closed-loop system matrix that would maximize the system's tolerance to perturbations (hence maximizing the robustness of the system) is proposed [2].
### 2.1 Quantitatively most robust matrices

Referring to equation 1.5, it is observed that  $\mu_s$  is a function of the stability degree  $\alpha$  and the condition number  $\kappa$ . This equation is reiterated below for convenience:

$$\mu_s = -\frac{\alpha}{\kappa} \tag{2.1}$$

Now it is well known in linear algebra that the condition number of any matrix is always greater than or equal to 1. In addition, for any diagonal matrix D, the condition number is basically a ratio of the maximum and minimum diagonal values of that matrix. This can be summarized as follows [11]:

i.  $\kappa(A) \ge 1$ .

ii. For an diagonal matrix, D:  $\kappa(D) = \frac{\max |d_{ii}|}{\min |d_{ii}|}$  .

Hence, from Equation 2.1, it can be observed that a maximum value of  $\mu_s$  can be obtained if  $\kappa$  is minimized. Since the minimum value of  $\kappa$  is 1, this serves as the first condition for obtaining a quantitatively most robust matrix. Therefore:

$$Condition \ 1: \ \kappa = 1. \tag{2.2}$$

Matrices that conform to condition 1 are known as normal matrices [11]. Now by using condition 1 from above, condition 2 follows as a result of the former:

Condition 2: 
$$\mu_s = -\alpha.$$
 (2.3)

Hence it is clear from the two conditions above that matrices that are normal (2.2) and stable (2.3) form the set of quantitatively most robust matrices.

Now that the set of most quantitatively robust matrices has been determined, it is essential to find the corresponding set of qualitatively most robust matrices. This is addressed in the following section.

### 2.2 Qualitatively most robust matrices

In Chapter 1, a set of metrics  $(\beta_1, \beta_2 \text{ and } \beta_3)$  were defined that gave rise to a cumulative qualitative robustness measure,  $\beta$ . For convenience these metrics are listed below.

$$\beta_1 = \frac{no. of a_{ii} > 0}{total no. of a_{ii}}$$
$$\beta_2 = \frac{no. of l - cycles > 0}{total no. of l - cycles}$$
$$\beta_3 = \frac{no. of k - cycles = 0}{total no. of k - cycles}$$

Recall that: l - cycles are the interactions and k - cycles are the interconnections between different species (diagonal elements) in a matrix.

It was also stated that a matrix is most qualitatively robust if  $\beta = 0$ . In other words,  $\beta_1 = \beta_2 = \beta_3 = 0$  and any matrix satisfying all three of these measures is identified as the most qualitatively robust matrix.

In order to conform to  $\beta_1 = 0$ , every diagonal element in this matrix must be of a negative value. Hence,

$$Condition \ 3: a_{ii} < 0. \tag{2.4}$$

 $\beta_2 = 0$  can only be obtained if every pair of off-diagonal elements  $(a_{ij} \text{ and } a_{ji})$  form negative products. Hence,

$$Condition \ 4: a_{ij}a_{ji} < 0. \tag{2.5}$$

Now for  $\beta_3 = 0$ , every 'k-cycle' in the matrix would have to equal zero. For example in a 3x3 matrix, the product  $a_{12}a_{23}a_{31}$  would have to equal to 0. However, this can only be possible if at least one or more of these elements equals zero. Therefore, Condition 5 can be stated as follows:

$$Condition \ 5: a_{ij}a_{jk}...a_{li} = 0.$$

Therefore, using the 5 conditions listed in these previous two sections, a set of *Target Sign-Stable matrices* (TSS) can be formulated that serve as the best choice for a nominal closed-loop system matrix. This is further explained in the following section.

# 2.3 Choice of the best nominal closed-loop system: The Target Matrix

### 2.3.1 Target sign-stable matrices

A set of quantitatively most robust matrices has been narrowed down to through conditions 1 and 2. Furthermore, a set of qualitatively most robust matrices can also be identified through conditions 3, 4 and 5. By denoting the set of quantitatively most robust matrices as  $A_{\mu}$  and the set of qualitatively most robust matrices as  $A_{\beta}$ , the set of matrices that are a subset of both  $A_{\mu}$  and  $A_{\beta}$  need to be obtained. This set of target matrices can be illustrated thru the following figure.



Figure 2.1: Set of Target matrices [2]

These set of *Target Matrices* have to satisfy each of the conditions listed in the previous section. Conditions 1 and 2 bring about an interesting feature that can be explained as follows.

In order to satisfy, or rather guarantee, condition 2, every diagonal element in the matrix to be negative. However, the main issue arises in trying to satisfy condition 1. In order to guarantee this condition it is essential that in addition to every diagonal element in this matrix being negative, they are also to be of the same magnitude. Furthermore, every off-diagonal pair should be symmetric in that each of these elements only differ by their sign but not their magnitude.

Such a matrix that satisfies conditions 1-5 is known as a *Target sign-stable* (TSS) matrix and it is this set of matrices that act as the best nominal choice for the closed-loop system. A TSS matrix, S, can be summarized as follows:

i.  $s_{ii} = -d$ , where 'd' is a positive real constant.

ii. sign(S) is qualitatively stable.

iii.  $s_{ij} = -s_{ji}$ . (However, if  $s_{ij}$  is zero, then the corresponding  $s_{ji}$  must also be zero.)

Example of a TSS matrix is shown below:

$$S = \begin{bmatrix} -d & -s_{1,2} & 0 & 0\\ s_{1,2} & -d & s_{2,3} & 0\\ 0 & -s_{23} & -d & s_{3,4}\\ 0 & 0 & -s_{3,4} & -d \end{bmatrix}$$

Consider the following theorem [12],

**Theorem 1**: For an  $n \ge n$  ecological sign-stable matrix,

$$a_{ii_{min}} \le Re(\lambda_i)_{min} \le Re(\lambda_i)_{max} \le a_{ii_{max}}$$
(2.7)

Since all the diagonal elements in a target sign-stable matrix are equal, then according to the theorem above, the real part of every eigenvalue will possess the same stability degree that can simply be denoted by **d**, which is simply the magnitude of each diagonal element itself. It has already been proven [11] that target signstable matrices are normal matrices through condition 1 and therefore the quantitative robustness index,  $\mu_s$ , of a target sign-stable matrix is:

$$\mu_s = -\frac{-d}{\kappa(=1)} = d \tag{2.8}$$

Therefore, it is clear that the set of TSS matrices provide the best robustness measures both qualitatively and quantitatively. However, the following sub-section narrows down to an even better subset of matrices that exist within this set of TSS matrices that can be used as an even better choice for the nominal closed-loop system.

# 2.3.2 Improved set of TSS matrices: The Target Pseudo-Symmetric matrix

Condition 5 listed above required that every k-cycle in a Target sign-stable matrix was to be zero. Therefore, it can be seen that zeros need to exist in appropriate locations in this TSS matrix in order to fulfill Condition 5. However most engineering systems in nature do not really possess many zero elements in their matrix dynamics. This can be attributed to coupling that occurs between the different state-variables in an engineering system. Hence it is very rare to find a real engineering system that contains the zeros required to satisfy condition 5. Therefore it is essential to relax the constraint set by this condition in a manner that is justifiable.

In chapter 1, the effect of each  $\beta_i$  on stability was shown in figures 1.7, 1.8 and 1.9. It was determined that  $\beta_1$  affected stability the most and  $\beta_3$  affected stability the least. Therefore it is sufficient to say that  $\beta_3$  is the least critical of all the  $\beta_i$ 's in terms of affecting stability and hence robustness. Therefore the  $\beta_3$  constraint (condition 5) can be relaxed to reduce the restrictive nature of TSS matrices [2].

Condition 4, however, still requires that every interaction forms a negative product and so an improved TSS matrix known as the *Target Pseudo symmetric* (TPS) matrix is formed that adheres to all the conditions 1-4, but not 5. Note that condition 4 exemplifies an interaction that is purely predator-prey in nature and it can be seen how the ecological principles discussed in chapter 1 tie into the discussion of robustness. A TPS matrix, P, conforms to the following properties:

i.  $p_{ii} = -d$ , where 'd' is a positive real constant.

ii. sign(P) is qualitatively stable.

iii.  $p_{ij} = -p_{ji}$ .  $(p_{ij}, p_{ji} \neq 0)$ 

iv.  $P^T = -P$ .

An example of a TPS matrix is shown below.

$$P = \begin{bmatrix} -d & -s_{1,2} & s_{1,3} & s_{1,4} \\ s_{1,2} & -d & s_{2,3} & -s_{2,4} \\ -s_{3,1} & -s_{23} & -d & s_{3,4} \\ -s_{4,1} & s_{2,4} & -s_{3,4} & -d \end{bmatrix}$$

It was shown in the previous chapter that the effect of increasing  $\beta$  decreased the  $\mu$  of a system. However, because  $\beta_3$  is not very critical to stability and robustness, it is proved in [13] that **Theorem 1** holds for TPS matrices as well and so the quantitative robustness index,  $\mu_s$ , is still the magnitude of the diagonal elements of the TPS matrix. Hence it can be concluded that the final Target matrix,  $A_T$ , that serves as the best choice for the nominal closed-loop system matrix,  $A_{cl}$ , is the Target Pseudo-Symmetric matrix.

The figure below shows the different sets of matrices that have been detailed so far.



Figure 2.2: Classification of target matrices [1]

## 2.4 Chapter 2 summary

This chapter dealt with narrowing down to a set of TSS matrices that formed appropriate choices for the nominal closed-loop system,  $A_{cl}$ . However, due to the restrictive nature of these matrices (the zero elements) along with the fact that most engineering matrices do not conform to TSS matrices, a new improved set of matrices was formed known as Target Pseudo-symmetric (TPS) sign-stable matrices. These matrices form the final Target matrix used in this research. The importance of having such a matrix as the nominal closed-loop system allows the system to maximize its tolerance (maximize  $\mu_s$ ) to perturbations without losing stability.

The following chapter involves the problem formulation and algorithm performed using the TPS matrix in order to maximize the overall robustness of the final closedloop system for various types of open-loop systems.

## CHAPTER 3

# ECO-INSPIRED ROBUST CONTROL DESIGN ALGORITHM

In order to detail the robust control algorithm which is the focal point of this research, it is essential to recap a few of the equations and concepts that were put forward in the previous chapters. Recall that this research only dealt with the robustness measure of linear time-invariant (LTI) systems:

$$\dot{x} = Ax + Bu$$

The above equation is the open-loop representation of an LTI system and hence it was pointed out in chapter 1 that this open-loop system could be stable or unstable. In other words, A could be a stable or an unstable matrix depending on the dynamics of the open-loop system under consideration. Hence, it was essential to convert this open-loop system into a stable closed-loop system which was guaranteed via a fullstate feedback gain, G, that resulted in equation 1.3. For convenience, this equation is given below:

$$\dot{x} = [A - BG]x \text{ or } \dot{x} = A_{cl}x$$

The  $A_{cl}$  matrix is the nominal closed-loop system matrix whose robustness is to be maximized.

Now the open-loop system contains an 'A' and a 'B' matrix. If it is assumed that there are 'n' state-variables in any system, A would then be an  $n \ge n$  matrix. Further, assuming there are m control variables needed to control any system, B would then be an  $n \ge m$  matrix. If m were to equal n, B would be a square,  $n \ge n$ , matrix. However, most real engineering systems have (m < n) because it is relatively expensive to come up with just as many control variables as there are state-variables.

The robust control algorithm for an (m=n) and (m < n) system differs because of concepts that will be clear later. Hence, the (m=n) case shall be addressed first before moving onto the (m < n) case.

## 3.1 1. (m=n) open-loop systems

For any controllable pair A and B, a nominal closed-loop system matrix needs to be created,  $A_{cl}$ , whose robustness index,  $\mu_s$ , is maximized. Hence it is hoped that this  $A_{cl}$  exactly mimics a *Target Pseudo-symmetric* (TPS) matrix,  $A_t$ , for reasons discussed in chapter 2. It is known that,

$$A_{cl} = A - BG \tag{3.1}$$

Since this section assumes a (m=n) system, B is a square  $n \ge n$  matrix. Furthermore, if B is invertible,

$$G = B^{-1}(A - A_{cl}) (3.2)$$

However, in order to maximize the robustness, the desired closed-loop matrix needs to take the form of  $A_t$ . Therefore if there is a desired  $A_t$  in mind, a gain Gcan be obtained that is able to achieve this only because B is square and invertible. Thus,

$$G = B^{-1}(A - A_T) (3.3)$$

Therefore the G obtained in Equation 3.2 is plugged into 3.1 to achieve a final  $A_{cl}$  whose robustness index,  $\mu_s$ , is maximized. Therefore,

$$A_{cl} = A_t \tag{3.4}$$

Hence, for any (m=n) open-loop system, it is possible to achieve any desirable closed-loop TPS system only because B is a square and invertible matrix. This final closed-loop system,  $A_{cl}$ , is the most robust system whose tolerance to parametric variations is maximized.

However, in the case that B is not square (and hence not invertible), another algorithm needs to be performed. This is discussed in the following section.

# **3.2 2.** (m < n) open-loop systems

In the previous section it was stated that for any (m=n) open-loop system, there exists a gain, G, that allows the final closed-loop system,  $A_{cl}$ , to exactly mimic any desired TPS matrix  $(A_t)$ . This is only possible because in an (m=n) system, the B matrix is square (and invertible). However, in the case of an (m < n) system, B is not square (and hence not invertible) and so the goal is to come up with a matrix that is invertible and is directly related to B. Therefore the *Pseudo-inverse* matrix of B is introduced that is symbolized by  $B^{\dagger}$ .

The inclusion of  $B^{\dagger}$ , however, introduces an error known as a *pseudo-inverse error*,  $E_{pseudo}$ . In addition to this, there is another error that exists that is referred to as  $E_{TPS}$ . Therefore,

$$E = E_{pseudo} + E_{TPS}$$

It is the constrained feature of a TPS matrix that gives rise to  $E_{TPS}$ . Therefore, these two errors together can be simply referred to as E. It is this overall error that restricts  $A_{cl}$  in exactly following a desired  $A_t$  (TPS) structure. The goal of the robust control design algorithm is to minimize the norm of this error such that  $A_{cl}$  is able to mimic  $A_t$  as closely as possible.

The algorithm presented below is divided into two phases namely: The optimization phase and The Robustifying phase. The optimization phase is concerned with minimizing the norm of the error addressed above, while the robustifying phase concentrates on finding a final gain,  $G_{final}$ , that allows  $A_{cl}$  to be as close to  $A_T$  as possible. *Note:* Unlike the (m=n) case,  $A_T$  is not any desirable TPS matrix, but is an optimized matrix constrained by the final minimized error. Therefore, the TPS matrix is considered to be a design variable that is denoted by a lower-case t symbol  $A_t$ . The final optimized TPS matrix corresponding to the minimum norm of E is denoted by an upper-case T symbol  $A_T$ .

The algorithm presented below differs for stable and unstable open-loop systems. Therefore the stable open-loop case shall be addressed first before proceeding to the unstable case.

#### 3.2.1 Stable open-loop systems

Before proceeding to the optimization phase, it is essential to make sure that the  $\underline{A}$  and  $\underline{B}$  matrices under consideration form a controllable pair.

#### 3.2.1.1 Optimization phase

#### Norm-minimization of E

The *pseudo-inverse* matrix of  $\underline{B}$  discussed above is defined as follows:

$$B^{\dagger} = (B^T B)^{-1} B^T \tag{3.5}$$

It is important to note that the primary reason for the inclusion of  $B^{\dagger}$  is because it serves as a fictitious B matrix that is directly related to B itself. It is because it does not equal to the actual <u>B</u> of the system, the *pseudo-inverse* error,  $E_{pseudo}$ is introduced. Therefore the corresponding gain that uses  $B^{\dagger}$  to form a closed-loop system does not compensate for this error, rather, the error is an internal feature possessed within this gain. Hence this gain is not considered to be the final closedloop gain that maximizes robustness, but instead, it serves as an intermediate gain denoted by  $\overline{G}$ . Thus, the equation for  $\overline{G}$  is:

$$\overline{G} = B^{\dagger}(A - A_t) \tag{3.6}$$

Again, recalling that  $A_t$  is a design variable,  $\overline{G}$  can be used with the original  $\underline{A}$ and  $\underline{B}$  systems to minimize the norm of E and form the final optimized target matrix,  $A_T$ . This can be shown in the equation below:

$$A_t + E = A - B\overline{G} \tag{3.7}$$

Equation (3.7) forms the focal point of the optimization phase.

The goal is to minimize the norm of the error that can be denoted as  $E_{min}$ , and find the corresponding  $A_t$  matrix that shall serve as the optimized Target matrix,  $A_T$ . Hence it can be observed that the elements in  $A_t$  can be treated as design variables. Equation (3.6) further shows that  $\overline{G}$  is a function of  $A_t$  as well and so every time the elements in  $A_t$  are changed (known as an iteration),  $\overline{G}$  is also changed, and the corresponding norm of E is stored. After going through all the possible iterations, the minimum value of the norms of E that were stored is recorded and the  $A_t$  matrix that corresponds to this minimum norm of E is taken as the optimized  $A_T$  matrix. This process is better aided via a matrix-design representation. The design target matrix representation of a 2 x 2 system is provided below:

$$A_t = \begin{bmatrix} -\delta_1 & \pm & \delta_2 \\ \pm & \delta_2 & -\delta_1 \end{bmatrix}$$

Notice that since the target matrix is a *Pseudo-Symmetric* matrix, it contains two design variables:  $\delta_1$  and  $\delta_2$ . The first design variable,  $\delta_1$ , serves as the diagonal elements and the second design variable,  $\delta_2$ , serves as the off-diagonal elements. The  $\pm$  sign only reiterates the point that the pair of off-diagonal elements have to be of opposite signs.

#### Range determination for design variables

It is known by now that both these design variables are changed through a specified range and it is important to note what this range is. If  $\overline{G}$  is expanded in equation (3.7) above, then it can be written as:

$$A_t + E = A - BB^{\dagger}(A - A_t) \tag{3.8}$$

Now if  $BB^{\dagger}$  is denoted as  $B_{auq}$ , then:

$$E = (I - B_{auq})(A - A_t) \tag{3.9}$$

where: I is the identity matrix.

Hence, if  $(I - B_{aug})$  is given the symbol  $B^+$ , then equation (3.9) can be re-written as:

$$E = B^{+}A - B^{+}A_{t} \tag{3.10}$$

Note that both A and  $B^+$  are properties of the open-loop system and hence do not change with every iteration. Now the ideal case of equation (3.10) exists if E = 0. Therefore if this ideal case is implemented,  $A_t$  can be written as:

$$A_t = (B^+)^{-1} B^+ A \tag{3.11}$$

Since equation (3.11) characterizes the ideal case, it can be seen that the maximum value any element, or design variable, in  $A_t$  can obtain is:

$$A_{t_{ij\ max}} = [min(B^+)]^{-1}max(B^+A)_{ij}$$
(3.12)

This maximum value provides a conservative maximum for the range.

Equation (3.12) specifies the range through which every design variable in  $A_t$  will span. Hence, every off-diagonal design variable will span from 0 to  $\pm A_{t_{ij}\ max}$ , while every diagonal variable will span from 0 to  $-A_{t_{ij}\ max}$  (since we can only have negative off-diagonal elements for a TPS matrix). This discussion, however, only covers the magnitude aspect of every element in  $A_t$ . Another consideration is the sign pattern of  $A_t$  that is explained below.

#### Sign Pattern consideration

Since  $A_t$  has to take up a certain sign pattern in order to conform to a TPS matrix, there are only so many different ways it can be represented. This can be better visualized with the aid of a sign pattern representation of  $A_t$ . The sign pattern representations for a 2 x 2 target matrix are shown below:

$$\begin{bmatrix} -\delta_1 & -\delta_2 \\ +\delta_2 & -\delta_1 \end{bmatrix} \begin{bmatrix} -\delta_1 & +\delta_2 \\ -\delta_2 & -\delta_1 \end{bmatrix}$$

It can be seen that there are two ways to represent any 2 x 2 target matrix. Therefore it is clear that there is more than one way that a target matrix can be represented in terms of its sign patterns and so if the order of the system under consideration is increased, many more sign patterns for the same system can be visualized. Hence, it is important to pre-determine the number of different sign patterns associated with any  $n \ge n$  matrix. Let  $\psi$  represent the number of off-diagonal pairs in an  $n \ge n$  matrix.  $\psi$  can be calculated through the following equation:

$$\psi = \frac{n(n-1)}{2} \tag{3.13}$$

Since each element in an off-diagonal pair can take one of two signs (+ or -), the number of sign-patterns, S, that are associated with any  $n \ge n$  matrix is given by:

$$S = 2^{\psi} \tag{3.14}$$

Therefore it can be observed that a 2 x 2 target matrix can have S = 2 sign-pattern considerations while a 4 x 4 target matrix can have as many as S = 64 sign-pattern considerations!

Thus every sign-pattern associated with any  $n \ge n$  target matrix is considered in the algorithm before an optimized  $A_T$  is formed. This is important because every different sign pattern associated with  $A_t$  might have a different corresponding E and  $\overline{G}$  (through equation (3.7)) that could eventually lead to a different final robustness index,  $\mu_s$ , in the robustifying phase. Therefore it is vital to make sure that every sign-pattern is considered in order to guarantee the most robust final closed-loop system.

It should also be noted that when higher order matrices are under consideration, the number of off-diagonal predator-prey pairs increase and so the number of design variables also increase. As a result, the number of permutations and iterations also increase and this plays a role as a numerical implication which is elaborated on in the final chapter.

#### Optimization phase summary

Once equation (3.7) is formulated, the elements of  $A_t$  are used as design variables that span through a specified range and the corresponding norm of E of every iteration is stored. Once every iteration has run its course, the minimum of all the stored norms of E is found and the corresponding optimized target matrix,  $A_T$ , along with  $\overline{G}$  are recorded to be used in the next phase (the Robustifying phase) of the algorithm.

#### 3.2.1.2 Robustifying phase

This phase deals with finding a final gain that results in the most robust closedloop system. Recall that the  $\overline{G}$  found in the optimization phase did not compensate for the *pseudo-inverse error*, but rather, contained the  $E_{pseudo}$  implicitly. Therefore the task now is to find an additional gain that can be added to  $\overline{G}$  to compensate for  $E_{psuedo}$ .

The method used is to basically form a pole-placement gain using the eigenvalues of the optimized target matrix,  $A_T$ , found in the optimization phase. This gain shall be referred to as  $G_{pp}$ . Once  $G_{pp}$  is formed, the absolute value of the maximum magnitude element in this matrix is recorded and shall be referred to as  $|G_{pp_{max}}|$ . The same process is performed on  $\overline{G}$  and the absolute value of its maximum magnitude element is recorded and shall be referred to as  $|\overline{G}_{max}|$ .  $|G_{pp_{max}}|$  and  $|\overline{G}_{max}|$  are finally added together to form a maximum elemental gain value that shall simply be referred to as  $|G_{max}|$ . Hence,

$$|G_{max}| = |G_{pp_{max}}| + |\overline{G}_{max}| \tag{3.15}$$

It is known that the final aim of this control design is to form a closed loop matrix of the form:

$$A_{cl} = A - BG_f \tag{3.16}$$

where  $A_{cl}$  is the final most achievable robust closed-loop matrix that can exist for any (m < n) system. Therefore, in order to find  $G_f$  that makes the former statement possible, it is important to form a finite range of gain matrices within which  $G_f$ actually lies.  $|G_{max}|$  represents the maximum element that can exist in this gain space. This is true because it is obvious that  $A_T$  found in the previous phase is the most robust closed-loop matrix that can hope to be achieved with any given open-loop <u>A</u> and <u>B</u> matrix. Therefore by performing a pole-placement gain using the eigenvalues of  $A_T$  to form  $|G_{pp_{max}}|$ , adding that to  $|\overline{G}_{max}|$  to form  $|G_{max}|$ , and using  $|G_{max}|$  to create a limit for the maximum absolute value of any element that can exist in these range of gain matrices, the overall gain space within which the final gain exists is narrowed down to a finite vicinity. In other words, every element in  $G_f$  is used as a design variable spanning from  $-|G_{max}|$  to  $+|G_{max}|$  and the corresponding robustness index,  $\mu_s$ , of every  $A_{cl}$  formed is stored. Once every element in  $G_f$  has spanned over the specified range, the maximum stored robustness index value is recorded and the corresponding  $G_f$  is used to form the final closed-loop system that serves as the most achievable robust closed-loop system for any (m < n) open-loop system. Therefore, the final closed-loop system is given by:

$$A_{cl} = A - BG_f \tag{3.17}$$

#### Robustifying phase summary

Equation (3.17) guarantees to contain the maximum achievable robustness index,  $\mu_{cl}$ , for any (m < n) open-loop system. The robustifying phase is needed to compensate for the pseudo-inverse error,  $E_{pseudo}$ . However, it should be noted that the final closed-loop system,  $A_{cl}$ , is not equal to  $A_T$  because of the existence of  $E_{TPS}$  that was described at the start.  $E_{TPS}$ , unlike  $E_{pseudo}$ , just cannot be compensated for due to the constrained structure of  $A_T$  and hence the algorithm spits out a closed-loop system that is as close enough to, but not equal to,  $A_T$ .

Figure 3.1 below provides an illustrative summary of this algorithm.



Figure 3.1: Flowchart summary of the Robust Control design algorithm

## 3.2.2 Eco-inspired robust control algorithm summary

The above algorithm only works for stable open-loop systems and the reason behind this is described in further detail in a later section. Since a stable system is being considered, there exists an initial robustness associated with this open-loop. This initial robustness is referred to as  $\mu_{ol}$  and serves as the lower-bound for the final closed-loop robustness,  $\mu_{cl}$ . Therefore the algorithm guarantees that the final closed-loop system will have a robustness greater than  $\mu_{ol}$ .

At the end of the optimization phase,  $\overline{G}$  is formulated. Letting  $A - B\overline{G}$  denote an intermediate closed-loop system in the algorithm,  $\overline{A_{cl}}$ , the robustness measure of this intermediate system is referred to as  $\overline{\mu}$ , such that:

$$\mu_{ol} < \overline{\mu}$$

The optimization phase also formulates the optimized target matrix,  $A_T$ , whose robustness,  $\mu_T$ , is basically the magnitude of its diagonal elements. It should be noted that  $\mu_T$  is the desired robustness of the system. However, due to the existence of  $E_{TPS}$ ,  $\mu_{cl}$  cannot achieve this desired robustness, and hence  $\mu_T$  serves as an upperbound to  $\mu_{cl}$ . Therefore,

$$\mu_{ol} < \overline{\mu} < \mu_{cl} < \mu_T$$

Keep in mind that the algorithm discussed above only works for stable open-loop systems. In fact, for any stable open-loop system, this algorithm guarantees a stable closed-loop system. A heuristic proof of this statement is detailed as follows.

### 3.2.3 Heuristic proof of closed-loop system stability

Since the algorithm described above is used for stable open-loop systems, it is assumed that A is a stable matrix. Equation (3.7) needs to be revisited again and is repeated below for the sake of convenience.

$$A_t + E = A - B\overline{G}$$

Looking at this equation, three distinct cases can be known to exist.

#### Case i: When $E \rightarrow 0$

It is known by now that almost every (m < n) system will always contain an error. The word almost is emphasized because case *iii* forms an exception to the former statement. Therefore it is intuitive that if E were to  $\rightarrow 0$ , the final closed-loop system would tend towards the desired optimized target matrix,  $A_T$ . This can be summarized as follows:

$$as \ E \to 0, \ A_{cl} \to A_T \tag{3.18}$$

This forms the ideal case and therefore serves as one extreme limit for this proof.

#### Case ii: When $E \neq 0$

This comprises of the normal case that requires the use of the robust control algorithm. It is because  $E \neq 0$ , the closed-loop system cannot exactly mimic the optimized  $A_T$ . Therefore,

$$A \neq A_T \tag{3.19}$$

Case iii: When E doesn't exist

This is not a very intuitive case because it was stated throughout this chapter that every (m < n) system will always contain an error, E. However, what if the A matrix, for some coincidental reason (that could be attributed to the natural openloop dynamics of the system), is of a TPS structure itself. In this case, the open-loop system matrix, A, is already the most robust it can be. Therefore the optimized target matrix is the open-loop system itself and so adding any gain to this system would only serve to decrease its robustness. In fact, if an open-loop TPS matrix were to be fed to the robust control design algorithm, it would output  $A_{cl}$  as A itself.  $E_{TPS}$ , in this case is zero because the open-loop matrix is already of a TPS structure. Therefore the overall error, E, is equal to only  $E_{pseudo}$ . However, because the openloop system is the most robust it can be,  $E_{pseudo}$  (and hence E), does not even need to be considered and so it can be be assumed not to exist.

*Case iii* serves to state the importance and beauty of this research by reiterating that a TPS matrix is the most robust matrix that can be achieved. It also forms the other extreme limit for this proof and can be summarized by,

$$A = A_{cl} = A_T \tag{3.20}$$

Note: This section is only concerned with (m < n) systems, however, a (m=n) system can also be regarded as an example of *case iii*. because of the absence of E.

The three cases are summarized in the table below:

i. $E \rightarrow 0$	$A_{cl} \to A_T$
ii. $E \neq 0$	$A_{cl} \neq A_T$
iii. E doesn't exist	$A_{cl} = \mathbf{A} = A_T$

Table 3.1: Three cases for stable (m < n) open-loop systems

There is a subtle difference between *Cases i.* and *iii.*. *Case i.* is an ideal case wherein E exists but somehow is made to be 0. *Case iii.*, on the other hand, does not require E to even take form because the most robust system that can be achieved is the open-loop system itself. Therefore *case i.* can be thought of as the closed-loop system moving from A to  $A_T$  while *case iii.* can be thought of as the closed-loop system remaining at A itself.

Therefore the three cases prove to show that if an (m < n) open-loop system is stable, the resulting closed-loop system will also be stable and this can be illustrated through Figure 3.2 below:



Figure 3.2: Proof of guaranteed closed-loop system stability

Since *case iii*. best proves the essence of this research, a mathematical example is provided below.

#### 3.2.3.1 Numerical example of Case iii.

In order to prove *Case iii.*, an LQR control design shall be used for comparison. Consider the following A and B matrices below:

$$A = \begin{bmatrix} -2 & -4 & 3\\ 4 & -2 & 6\\ -3 & -6 & -2 \end{bmatrix} \qquad \qquad B = \begin{bmatrix} 0.5 & -1.6\\ 0.5 & 5\\ 2.1 & 4 \end{bmatrix}$$

Notice that the A matrix satisfies a Target Pseudo-Symmetric matrix. Therefore it has a robustness index equal to that of the magnitude of its diagonal elements. This open-loop robustness index shall be referred to as  $\mu_{ol} = 2$ .

The goal of this example is to show that  $\mu_{ol}$  is the maximum robustness index that this system can achieve. Therefore an LQR controller shall be used for comparison where the Q and R matrices are chosen as follows:

Here,  $\rho$  is used as a design variable taking up three values of ascending order:

$$\rho = [1, 1000, 2000]$$

Hence the three  $\rho$  values chosen result in 3 different LQR gains (G1, G2, G3) that in turn lead to 3 different closed-loop systems ( $Acl_1$ ,  $Acl_2$ ,  $Acl_3$ ) used for comparison.

#### When $\rho = 1$

$$G1 = \begin{bmatrix} 0.158 & 0.027 & 0.2923 \\ -0.198 & 0.784 & 0.3609 \end{bmatrix} \qquad Acl_1 = \begin{bmatrix} -2.396 & -2.758 & 3.431 \\ 4.914 & -5.937 & 4.049 \\ -2.537 & -9.195 & -4.0577 \end{bmatrix}$$

The robustness measure for this closed-loop system,  $\mu_1 = 1.6544$ .

#### When $\rho = 1000$

$$G2 = \begin{bmatrix} 0.00012 & 0.000125 & 0.00052 \\ -0.0003988 & 0.00124 & 0.000997 \end{bmatrix} \quad Acl_2 = \begin{bmatrix} -2.0007 & -3.998 & 3.001 \\ 4.0019 & -2.0063 & 5.9947 \\ -2.9986 & -6.0052 & -2.0051 \end{bmatrix}$$

The robustness measure for this closed-loop system,  $\mu_2 = 1.99967$ .

When  $\rho = 2000$ 

$$G3 = 10^{-3} \begin{bmatrix} 0.0625 & 0.0624 & 0.2623 \\ -0.1997 & 0.624 & 0.4994 \end{bmatrix} \quad Acl_3 = \begin{bmatrix} -2.0003 & -3.999 & 3.0006 \\ 4.0009 & -2.0031 & 5.9973 \\ -2.9993 & -6.0026 & -2.0025 \end{bmatrix}$$

The robustness measure for this closed-loop system,  $\mu_2 = 1.99983$ . The table below summarizes and compares the three LQR results to the open-loop:

$\mu_{ol}$	2
$\mu_1$	1.6544
$\mu_2$	1.99967
$\mu_3$	1.99983

Table 3.2:  $\mu$  results

In this example, it is observed that  $\mu$  increases as a function of  $\rho$ . There is a significant increase from  $\mu_1$  to  $\mu_2$ . However when  $\rho$  is increased further,  $\mu$  seems to plateau out to a value that is close to but never equal to  $\mu_{ol}$ . In addition, notice that as  $\rho$  increases, the corresponding gain tends to zero proving that the open-loop system itself is the most robust the system can get.

Therefore it can be inferred that if a system's open-loop is of a TPS form, it is the most robust it can be and hence there is *no existing gain* that could give the system a  $\mu$  greater than  $\mu_{ol}$ .

### **3.2.4** Unstable open-loop systems

Up to this point, only the stable open-loop case has been discussed in regard to (m < n) systems. The robust control algorithm described in the previous sections can only be used for the stable open-loop case and the reason behind this shall be elaborated on in this section.

In chapter 1, it was emphasized that robustness is an innate feature of stability. Robustness is basically a measure of the tolerance to a perturbation about a stable state. However, if this stable state did not exist in the first place, then there can be no notion of robustness associated with it. Therefore an explanation is provided that uses this relationship between stability and robustness to pinpoint the reason as to why the robust control algorithm does not work for unstable open-loop systems.

Consider the equation summarizing the optimization phase of the robust control design which is reiterated below for the sake of convenience.

$$A_t + E = A - B\overline{G}$$

Recall that during this phase, the individual elements of the  $A_t$  matrix are used as design variables that span a predetermined range that was originally derived from Equation (3.10) (repeated below).

$$E = B^+ A - B^+ A_t$$

Now the goal of the algorithm was to vary the elements in  $A_t$  through the predetermined ranges in order to find a set of elements that corresponded to the minimum norm of E. The maximum absolute value in this range is given by Equation (3.12). Hence every diagonal element in  $A_t$  spanned from 0 to the negative of this maximum absolute value,  $A_{t_{ij}\ max}$ . In other words, every diagonal element in  $A_t$  could only take a zero or negative real value.

Now consider the diagonal elements of A. When A was a stable open-loop matrix, it had to have had at least some negative diagonal elements that made the net value of all the diagonal elements negative. Therefore during every iteration, the negative diagonal elements in  $A_t$  were subtracted from the negative elements in A in the equation above. Therefore it can be seen that this led to a convergence of the error E. However, if A was an unstable open-loop matrix, the negative diagonal elements in  $A_t$  would then be subtracted form a net positive value of A's diagonal elements and this leads to a divergence of the error.

Recall that it was stated that every (m < n) stable open-loop system contains an optimized TPS matrix,  $A_T$ , that is determined by the A and B matrices which is why the error converges to a minimum value that corresponds to this optimized  $A_T$ matrix. However, in the unstable open-loop case, the error does not converge because there is no optimized  $A_T$  matrix associated with an unstable open-loop system. This in turns tracks back to the fact that unstable systems do not contain the concept of robustness in the first place.

#### Utilizing the Robust control algorithm for the unstable open-loop case

However, the algorithm can be used in a slightly different way for the unstable open-loop case. By realizing that the unstable open-loop system does not contain any concept of robustness, it first needs be pushed into a region of stability. This requires a certain degree of control effort,  $J_u$ . Once it is pushed into a stable region, it can then be used with the robust control design to create a robust closed-loop system. This closed-loop system, however, cannot be claimed to be the most robust system because it is inherently a function of the stable region it was initially pushed into before the algorithm was used. For instance if the unstable open-loop was pushed into a highly stable region, the final robustness associated with it through the algorithm would be greater, however, the control effort needed to push it into this stable region would also be greater. Therefore it is obvious that the least control effort required for this algorithm to work would be to push the unstable open-loop to a marginally stable region. In this way, the algorithm would produce the maximum robustness to control effort index,  $\tau = \frac{\mu_{cl}}{J_u}$ .

This concept works, however, the determination of what exact marginal stable region the unstable open-loop needs to be pushed into in the first place needs to be researched and hence can be attributed to as future work.

### 3.3 Chapter 3 summary

This chapter was mainly concerned with two types of systems: (m=n) and (m < n) open-loop systems.

It was made clear that a (m=n) open-loop system is able to mimic any desired TPS target structure as long as its B matrix is invertible. In other words, this type of system can achieve any desired robustness index,  $\mu_{cl}$ .

The (m < n) open-loop case consisted of two sub-categories: stable and unstable open-loop systems. The algorithm provides the maximum robustness index,  $\mu_{cl}$ , for stable open-loop systems. On the other hand, the algorithm provides the maximum robustness to control effort index,  $\tau$ , for unstable open-loop systems.

It is important to note that the key feature of the algorithm allowing it to produce a maximum  $\mu_{cl}$  and  $\tau$  is the *Target Pseudo-Symmetric matrix*. It is because the algorithm pushes the closed-loop system as close as possible to an optimized TPS structure is why  $\mu_{cl}$  and  $\tau$  are maximized. Also, the importance of the ecological principles discussed in chapters 1 and 2 that led to the determination of a TPS structure can now be realized, reiterating the beauty of this research.

Illustrative examples of the (m < n) cases for stable open-loop systems using the algorithm are provided in Chapter 4. The results from these examples are compared to another existing robust control technique, (The LQR controller), in order to prove that the eco-inspired robust control algorithm produces the maximized robustness results.

# CHAPTER 4

# ECO-INSPIRED ROBUST CONTROL DESIGN EXAMPLES

The examples given below illustrate the use of the eco-inspired algorithm and therefore only (m < n) systems are considered. It is known that the (m=n) case can achieve any desirable TPS matrix and therefore does not require the use of the algorithm.

# 4.1 Aircraft flight control:

The 2 x 2 open-loop system below contains the short period mode dynamics of an aircraft [14]. The pair A, B is controllable and the systems state and input dynamics are given in the A and B matrices below:

$$A = \begin{bmatrix} -0.334 & 1\\ -2.52 & -0.387 \end{bmatrix} \qquad \qquad B = \begin{bmatrix} -0.027\\ -2.6 \end{bmatrix}$$

Since a 2 x 2 system is under consideration, the number of off-diagonal pairs are:

$$\psi = \frac{n(n-1)}{2} = 1$$

and so,
$$S = 2^{\psi} = 2.$$

Hence, 2 sign patterns exist for a 2 x 2 system and the algorithm chooses the sign pattern that corresponds to the minimum error, E. The two sign patterns are given below.

$$A_{t_1} = \begin{bmatrix} -\delta_1 & \delta_2 \\ -\delta_2 & -\delta_1 \end{bmatrix} \qquad \qquad A_{t_2} = \begin{bmatrix} -\delta_1 & -\delta_2 \\ \delta_2 & -\delta_1 \end{bmatrix}$$

Open-loop eigenvalues:	$-0.3605 \pm 1.5872$ j
$\mu_{op}$ :	0.227
Optimized elemental values:	$\delta_1 = 1.25, \ \delta_2 = 1$
Optimized sign pattern:	$A_{t_1}$
Optimized Target Matrix, $A_T$ :	$\begin{bmatrix} -1.25 & 1 \\ -1 & -1.25 \end{bmatrix}$
$A_T$ eigenvalues:	$-1.25 \pm 1.00$ j
$\mu_T$ :	1.25
$\overline{G}$ :	$[0.5809 \ -0.3319]$
$\overline{A_{cl}}$ :	$\begin{bmatrix} -0.3183 & 0.991 \\ -1.0097 & -1.2499 \end{bmatrix}$
$\overline{A_{cl}}$ eigenvalues:	$-0.7841 \pm 0.8852$ j
$\overline{lpha}$ :	0.7841
$\overline{\kappa}$ :	1.6563
<u> </u>	0.4734

Table 4.1: Optimization phase results

$G_f$ :	[-0.967 -1.0477]
$A_{cl}$ :	$\begin{bmatrix} -0.3601 & 0.9718 \\ -5.0317 & -3.1067 \end{bmatrix}$
Closed-loop eigenvalues:	$-1.7355 \pm 1.73261$ j
$\alpha_{cl}$ :	1.7355
$\kappa_{cl}$ :	3.15
$\mu_{cl}$ :	0.55095

Table 4.2: Robustifying phase results

From the two tables above, it can be seen that:

$$\mu_{ol} < \overline{\mu} < \mu_{cl} < \mu_T$$

If  $A_{cl}$  and  $A_T$  are compared, the two matrices are very different from one another. However, in order to show that the  $A_{cl}$  derived with this eco-inspired algorithm provides the best achievable robustness index, it is compared to another existing controller - LQR.

### LQR controller comparison:

In order to compare two different controllers, a common basis must be stated. One common basis that can be used for this comparison is the stability degree (1.7355) of the closed-loop system.

Since this is a single input system, the Q and R matrices are given by,

$$Q = \begin{bmatrix} 1 & 0\\ 0 & 1 \end{bmatrix} \qquad \qquad R = \rho * \begin{bmatrix} 1 \end{bmatrix}$$

where:  $\rho$  is a design variable that is used to give the overall system the same stability degree as  $A_{cl}$  above.

The results using the LQR technique are given in the table below:

$\rho^*$ :	0.7789
$G_{LQR}$ :	[-0.2219 -1.0554]
$A_{cl}$ :	$\begin{bmatrix} -0.3400 & 0.9715 \\ -3.0968 & -3.1309 \end{bmatrix}$
Closed-loop eigenvalues:	$-1.7355 \pm 1.0302$ j
$\alpha_{LQR}$ :	1.7355
$\kappa_{LQR}$ :	3.6773
$\mu_{LQR}$ :	0.4720

Table 4.3: LQR results

Referring to the robustness index of the closed-loop via the eco-design controller as  $\mu_{eco}$ ,

# $\mu_{LQR} < \mu_{eco}.$

Another example of a marginally stable open-loop system is considered next and shall be compared to the LQR controller again in order to prove that the Eco-design robust design is the most robust controller to parametric variations.

# 4.2 Satellite Attitude Control:

The example below is adapted from [13]. It portrays the linear range dynamics of an axisymmetric satellite. The A and B matrices are given as:

$$A = \begin{bmatrix} 0 & -0.7199 & 0\\ 1.1479 & 0 & 0\\ 0 & 0 & 0 \end{bmatrix} \qquad \qquad B = \begin{bmatrix} 1\\ 2\\ 3 \end{bmatrix}$$

Since a 3 x 3 system is under consideration, the number of off-diagonal pairs are:

$$\psi = \frac{n(n-1)}{2} = 3$$

and so,

$$S = 2^{\psi} = 8.$$

Therefore 8 sign patterns exist for a 3 x 3 system and the algorithm chooses the sign pattern that corresponds to the minimum error, E. The eight sign patterns are given below.

$$A_{t_1} = \begin{bmatrix} -\delta_1 & \delta_2 & \delta_3 \\ -\delta_2 & -\delta_1 & \delta_4 \\ -\delta_3 & -\delta_4 & -\delta_1 \end{bmatrix} \quad A_{t_2} = \begin{bmatrix} -\delta_1 & -\delta_2 & \delta_3 \\ \delta_2 & -\delta_1 & \delta_4 \\ -\delta_3 & -\delta_4 & -\delta_1 \end{bmatrix} \quad A_{t_3} = \begin{bmatrix} -\delta_1 & \delta_2 & -\delta_3 \\ -\delta_2 & -\delta_1 & \delta_4 \\ \delta_3 & -\delta_4 & -\delta_1 \end{bmatrix}$$

$$A_{t_4} = \begin{bmatrix} -\delta_1 & \delta_2 & \delta_3 \\ -\delta_2 & -\delta_1 & -\delta_4 \\ -\delta_3 & \delta_4 & -\delta_1 \end{bmatrix} \quad A_{t_5} = \begin{bmatrix} -\delta_1 & -\delta_2 & -\delta_3 \\ \delta_2 & -\delta_1 & -\delta_4 \\ \delta_3 & \delta_4 & -\delta_1 \end{bmatrix} \quad A_{t_6} = \begin{bmatrix} -\delta_1 & \delta_2 & -\delta_3 \\ -\delta_2 & -\delta_1 & -\delta_4 \\ \delta_3 & \delta_4 & -\delta_1 \end{bmatrix}$$

$$A_{t_7} = \begin{bmatrix} -\delta_1 & -\delta_2 & \delta_3\\ \delta_2 & -\delta_1 & -\delta_4\\ -\delta_3 & \delta_4 & -\delta_1 \end{bmatrix} \qquad \qquad A_{t_8} = \begin{bmatrix} -\delta_1 & -\delta_2 & -\delta_3\\ \delta_2 & -\delta_1 & \delta_4\\ \delta_3 & -\delta_4 & -\delta_1 \end{bmatrix}$$

Open-loop eigenvalues:	$0, 0 \pm 0.9091 j$
$\mu_{op}$ :	0
Optimized elemental values:	$\delta_1 = 0.39, \ \delta_2 = 0.9, \ \delta_3 = 0.1, \ \delta_4 = 0.1$
Optimized sign pattern:	$A_{t_2}$
Optimized Target Matrix, $A_T$ :	$\begin{bmatrix} -0.39 & -0.9 & 0.1 \\ 0.9 & -0.39 & 0.1 \\ -0.1 & -0.1 & -0.39 \end{bmatrix}$
$A_T$ eigenvalues:	-0.39, -0.39 $\pm$ 0.911j
$\mu_T$ :	0.39
$\overline{G}$ :	$[0.0847 \ \ 0.09 \ \ 0.0621]$
$\overline{A_{cl}}$ :	$\begin{bmatrix} -0.0847 & -0.8099 & -0.0621 \\ 0.9785 & -0.18 & -0.1243 \\ -0.2541 & -0.27 & -0.1864 \end{bmatrix}$
$\overline{A_{cl}}$ eigenvalues:	-0.2033, -0.1239 $\pm$ 0.8614j
$\overline{\alpha}$ :	0.1239
$\overline{\kappa}$ :	1.801
$\overline{\mu}$ :	0.0688

Table 4.4: Optimization phase results

$G_f$ :	[-0.1568  0.3332  0.1632]
$A_{cl}$ :	$\begin{bmatrix} 0.1568 & -1.0531 & -0.1632 \\ 1.4614 & -0.6665 & -0.3265 \\ 0.4703 & -0.9997 & -0.4897 \end{bmatrix}$
Closed-loop eigenvalues:	-0.3337, -0.3328 $\pm$ 1.0496j
$\alpha_{cl}$ :	0.3328
$\kappa_{cl}$ :	3.4145
$\mu_{cl}$ :	0.0975

Table 4.5: Robustifying phase results

From the two tables above, it can be seen that:

$$\mu_{ol} < \overline{\mu} < \mu_{cl} < \mu_T$$

Again, in order to emphasize the importance of this eco-inspired algorithm, an LQR comparison shall be made.

#### LQR controller comparison:

Once again the stability degree (0.3328) of the closed-loop system serves as the common basis for the comparison.

Since this is a single input system, the Q and R matrices are given by,

where:  $\rho$  is a design variable that is used to give the overall system the same stability degree as  $A_{cl}$  above.

$\rho^*$ :	8.9
$G_{LQR}$ :	[-0.1312  0.4148  0.3352]
$A_{cl}$ :	$\begin{bmatrix} 0.1312 & -1.1347 & -0.3352 \\ 1.4104 & -0.8295 & -0.6704 \\ 0.3937 & -1.2443 & -1.0056 \end{bmatrix}$
Closed-loop eigenvalues:	-1.0383, -0.3328 $\pm$ 0.8304j
$\alpha_{LQR}$ :	0.3328
$\kappa_{LQR}$ :	5.1041
$\mu_{LQR}$ :	0.0652

The results using the LQR technique are given in the table below:

Table 4.6: LQR results

Once again, referring to the robustness index of the closed-loop through the ecodesign controller as  $\mu_{eco}$ ,

## $\mu_{LQR} < \mu_{eco}.$

It can be observed again that the LQR technique produces a lower robustness index value than that of the eco-inspired algorithm. This exemplifies the importance of the ecological and structural properties of the eco-design controller that make it a very robust controller to parametric variations.

### CHAPTER 5

# NUMERICAL IMPLICATIONS, CONCLUSIONS AND FUTURE WORK

## 5.1 Numerical Implications

The algorithm discussed in the previous chapter dealt with the iterations of design variables that spanned specific ranges. The optimization phase dealt with the ranges through which the design variables of the  $A_t$  matrix spanned while the robustifying phase dealt with the ranges across which the final gain matrix  $G_f$  elements spanned. This was made clear in the previous chapter. However, it was not discussed as to what elemental step-sizing each range consisted of. It makes intuitive sense that the smaller the step-sizing, the better the results. The tradeoff that exists is that if the step-sizing is made smaller, the results also take longer to converge. This issue is called *Parametric gridding* and it plays a vital role in achieving the correct results. In order to better understand the concept of *Parametric gridding*, the aircraft control example from the previous is repeated below.

### 5.1.1 Numerical implication on Aircraft control example

The A and B matrices are repeated below for convenience.

$$A = \begin{bmatrix} -0.334 & 1\\ -2.52 & -0.387 \end{bmatrix} \qquad \qquad B = \begin{bmatrix} -0.027\\ -2.6 \end{bmatrix}$$

Again, the design target matrix can take one of the following two forms:

$$A_{t_1} = \begin{bmatrix} -\delta_1 & \delta_2 \\ -\delta_2 & -\delta_1 \end{bmatrix} \qquad \qquad A_{t_2} = \begin{bmatrix} -\delta_1 & -\delta_2 \\ \delta_2 & -\delta_1 \end{bmatrix}$$

Recall from Equation 3.12, that every off-diagonal design variable spans the following range,

$$\delta_2 = [-[min(B^+)]^{-1}max(B^+A)_{ij} : + [min(B^+)]^{-1}max(B^+A)_{ij}]$$

Every diagonal design variable only spans one-half of this range since the diagonal elements in a TPS matrix can only be negative. Thus,

$$\delta_1 = [0 : + [min(B^+)]^{-1}max(B^+A)_{ij}]$$

However, what was not discussed in the previous chapter was each design parameter's step-sizing. In order to better illustrate this concept, it shall be talked on a macro-level.

In the previous chapter, a 0.01 step-sizing was used for both ranges in the optimization and robustifying phases. Therefore an  $A_T$  matrix was obtained that was accurate to 2 decimal places. However, if a 0.1 step-size was used, the resulting  $A_T$ matrix would only be accurate to one decimal place. By denoting the step-size by

 $\Delta \delta$ , the results of the corresponding  $A_T$  and final  $\mu_{cl}$  values from the two step-sizings are compared in the table below:

$\Delta \delta = 0.1$	$\Delta \delta = 0.01$
$A_T = \begin{bmatrix} -1.2 & 1\\ -1 & -1.2 \end{bmatrix}$	$A_T = \begin{bmatrix} -1.25 & 1\\ -1 & -1.25 \end{bmatrix}$
$\mu_{cl} = 0.5507$	$\mu_{cl} = 0.551$

Table 5.1: Effect of parametric gridding on resolution of results

It seems that the final closed-loop robustness index results does not vary by much with the change in step-sizing. However, keep in mind that this example only considers a 2 x 2 system and so the results get even more skewed as higher order systems are considered. This example only serves to show the difference that can occur on a macro-level. One might ask as to why not have a very small step-sizing fixed from the start itself. This is because the computation time is increased because the number of iterations are greatly increased and therefore the final result takes a lot longer to converge.

### 5.2 Conclusions and Future work

The eco-inspired robust control design adapts certain ideas and principles from ecology and applies them to matrix theory that can in turn be applied to engineering systems. The fascinating feature of this research is the manner in which it exemplifies the manner in which different ideas from different disciplines can work together to produce a unique design.

The eco-inspired control design guarantees any desirable robust matrix for (m=n) systems. In addition, it guarantees the maximum robustness index of stable openloop (m < n) systems. On the other hand, for unstable open-loop (m < n) systems, it guarantees the maximum robustness index to control effort index. However further research needs to be performed in order to determine what marginally stable region the unstable open-loop needs to be pushed into in a manner that is justifiable so that the eco-inspired control design can be implemented.

Lastly, the main goal of this research is to revive the enthusiasm in the field of controls toward eco-inspired design techniques. It was stated earlier that robustness, like stability, is a very important feature in any control design and it is important that significant progress be made toward this design specification in the future.

# BIBLIOGRAPHY

- N. Devarakonda. Eco-inspired robust control design for linear dynamical systems with applications. *Dissertation, The Ohio State University press, Columbus OH*, 2011.
- [2] R.K. Yedavalli and Nagini Devarakonda. Qualitative robustness and its role in the robust control of uncertain engineering systems. *Proceedings of ASME Dynamic Systems and Control conference*, 2011.
- [3] R.K. Yedavalli and P. Dorato. Recent advances in robust control. *IEEE Press*, 1990.
- [4] L. Edelstein-Keshet. *Mathematical models in Biology*. McGraw Hill, 1988.
- [5] J. Hofbauer and K. Sigmund. Growth rates and ecological models: The theory of evolutions and dynamical systems. *Cambridge University Press, London*, pages 29–59, 1988.
- [6] L. Hogben. Handbook of linear algebra. Chapman and Hall/CRC, 2007.
- [7] R.K. Yedavalli. Qualitative stability concept from ecology and its use in the robust control of engineering systems. *Proceedings of the American Control Conference, Minneapolis*, 2006.
- [8] C. Jeffries. Qualitative stability and digraphs in model ecosystems. *Ecology*, 55:1415–1419, 1974.
- [9] R. May. Stability and complexity in model ecosystems. *Princeton University Press*, 1973.
- [10] J. Quirk and R. Rupert. Qualitative economics and the stability of equilibrium. *Reviews in Economic studies*, 32:311–326, 1965.
- [11] D.S Bernstein. Matrix mathematics: Theory facts and formulas with application to linear systems theory. *Princeton University Press*, 2005.

- [12] R.K. Yedavalli and N. Devarakonda. Qualitative principles of ecology and their implications in quantitative engineering systems. AIAA journal of Guidance control and Dynamics, 33(2):333–346, 2010.
- [13] R.K. Yedavalli and N. Devarakonda. A new robust control design method for linear systems with norm-bounded time-varying real parameter uncertainty. Proceedings of ASME Dynamic systems and Controls conference, Cambridge MA, 2010.
- [14] Robert C. Nelson. Flight, stability and automatic control. McGraw Hill, 1998.