Scheduling Schemes using Connective Stability

by

Thilan Costa

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Large systems are often constructed using small subsystems which are connected. These interconnections can lead to complex behavior; for example, the entire system may become unstable even if each of the individual subsystems are stable by themselves. The unstable systems can be stabilized with the use of a shared feedback controller. The effects of one subsystem on the state of other subsystems (coupling) can be reduced if each subsystem has access to the state information of the subsystems that are affecting its state. However, this solution requires communication between the controller and the subsystems and between subsystems. If there are limited communication resources, management of this resource is also required. Hence there is a need for a scheduling policy that specifies which subsystem should use the communication resource at any given time.

We start our formulation by first investigating systems that contain only stable subsystems. If the connected system is unstable due to coupling, the system cannot be scheduled. Therefore, we first proceed to extend previous work on stability of connected systems in order to formulate computationally efficient schedulability checks for these systems. We provide sufficient and necessary conditions for certain topologies and results for scalar systems that are dependent on the number of subsystems.

Then we proceed to formulate a centralized scheduling policy based on results of connective stability. Here we constrain ourselves to first studying systems with only a single communication resource that restricts only one subsystem to transmit its state in a given time slot. We study the best input a subsystem may apply once it has knowledge of the state of another subsystem that is affecting its state. We also provide evidence from simulations to support the performance increase in using the proposed algorithm.

Finally, we extend these results to formulate a decentralized scheduling policy that supports multiple communication resources. We also analyze a possible way of improving the scheduling policy using similarity transformations and show that such a methodology does not guarantee performance improvement and in-fact may lead to worse performance.

Acknowledgments

First and foremost, I would like to give praise and thanksgiving to God the Father almighty, creator of heaven and earth, God the Son, and God the Holy Spirit without whose love and blessing I would not exist and this thesis would never have been completed.

I thank my loving mother and father who brought me in to this world and were always there for me every step of the way with prayers and words of encouragement. I also thank my sister, who was always there to lend her support through prayers and suggestions.

I thank my supervisor Prof. Shreyas Sundaram, for taking me under his wing as his student, for his patience, support, guidance, and advice.

Last but not least, the Blessed Virgin Mary for her loving prayers and intercessions on behalf of such an unworthy son.

Dedication

To my loving parents and sister.

Table of Contents

List of Figures

Chapter 1

Introduction

Most (if not all) large systems are composed of smaller subsystems that are interconnected together. Some of these systems exist naturally (as in the instance of eco-systems [23]) while others arise as a result of human design (as in the example of large scale power systems [22]). In such systems, the state of one subsystem can affect the other subsystems in the network as a result of being connected. We will refer to this form of interaction as coupling. Furthermore, the level of coupling between states of each subsystem can vary dynamically or could be static during the operation of the system. In such systems, the coupling effects could be reduced if a subsystem has prior knowledge of the dynamics of the coupling and the current state of the coupled subsystem.

In such systems that require the sharing of one's state information with other subsystems, the communication resource needs to be managed. Hence, the resource must be scheduled such that the communication data required to maintain stability is transferred to each subsystem within the time period in which it is required.

Our focus here will be on developing a scheduling scheme for connected dynamical systems of the above form that require information to be communicated, and therefore requires the communication resource to be managed. The scope of our work will be limited to the case where individual subsystems are assumed to be stable by themselves, which is a requirement of connective stability [24]. We start by extending previous work by Siljak on connective stability for discrete time systems [19] for different network topologies. We then consider the case where subsystems are stable without the presence of coupling and we discuss the formulation of a scheduling policy to minimize effects of coupling when the system is connected.

The thesis will address the following issues with respect to scheduling for such systems.

1. Develop a simplified schedulability tests for some common topologies
2. Develop a scheduling scheme for a centralized connected system to achieve stability
3. Develop a scheduling scheme for a decentralized connected system to achieve stability

1.1 Related Work

Many prior works have presented results on analyzing connected systems using concepts such as connective stability [19], [24], [22], [21], [20], [26], [27], [25], [6] or switching systems [30], [1], [9]. Loosely speaking if a system is connectively stable, the system will remain stable even as the coupling varies between its maximum or zero [19] [24]. However, connective stability requires each subsystem to be stable when the subsytem is isolated from any coupling effects or connections from other subsystems. The work based on switching system theory deals with systems that switch between various subsystems according to a switching rule [30], [1], [9].

In the area of real-time scheduling, there have been various scheduling schemes developed for distributed control systems. The work by Marti describes the problems in adapting scheduling theory in real-time theory to solving control theory problems dealing with stability [10]. There is also work by the same author on extending classical real time scheduling policies such as 'Earliest Deadline First (EDF)' to control systems [29]. The proposed 'LEF' policy deals with giving higher priority in terms of using the resource to the subsystem that has the largest deviation between its actual response and desired response at a given time. The issue of resolving packet drops due to contention was also addressed in a work by Zhang [31]. The work presented in Zhang's paper differs in its approach from the work presented in this thesis in that it works within the confines of an ALOHA network.

Work by Nilsson on problems with real time scheduling of control systems [13] delves in to the issues faced with jitter, and other lower level problems. These issues while important, were too specific for consideration in designing our scheduling policy.

We also considered the possibility of modeling the problem of scheduling the transmission resource using work on resource allocation in other areas such as processors for stability related functions. Work by Gupta on anytime control algorithms based on time varying processor availability [16], [3] was considered. The model under study in that paper required the resource availability to follow a distribution or function. Our work focuses on the structure of the interconnections and does not assume a distribution on the resource.

A work by Hespanha develops a methodology to construct a deadline, period and execution time for a control system [12]. After these quantities are defined and calculated for a given system, a standard technique such as EDF is used to schedule the system.

In the paper by Michel, Miller & Tang on connective stability [11], they develop a result based on the concept of stability preserving links. The result shows that if the interconnections satisfy the stability preserving conditions, then the system is connectively stable. While the result is based on a different approach than the use of Lyapunov methods, the analysis requires strongly connected groups of subsystems and requires transformation of the original system based on graph theoretic decompositions.

In work related to modeling a connected system as a packet transmitting network [12], [18], [2] the goal is to determine when such a system is mean square stable. In these works, the properties of the system under study are captured by a set of inequalities. These inequalities are constructed by considering the probabilities that each subsystem state would be of a particular form. If a solution exists to the set of inequalities, then the corresponding system is also guaranteed to be mean square stable [32], [5]. There is also work on developing a result for connective stability in such a packet network [6]. Here the goal is once again to find conditions under which the system would be connectively stable.

Another type of work on connected systems deal with using the network itself as a controller [15][14]. Each subsystem in such a network updates its state as a linear combination of the states of the neighboring subsystems. It is shown in [15] that such a strategy will cause the network itself to act as a controller. The problem of packet drops in such networks are handled in [4] [7] [18] [17]. All these works discuss the problem of maintaining mean square stability of the system.

Our work differs from these existing works in that

1. We extend current results for specific network topologies to obtain simpler results.
2. While previous results only address the issue of stability, we use the existing results to develop scheduling policies to achieve faster and more reliable convergence.

1.2 Notation and Previous Results

In this thesis, we will consider discrete-time linear systems composed of N subsystems, where each subsystem is of the form

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j=1}^{N} e_{ij}A_{ij}x_j(k) \tag{1.1}$$

with state vector $x_i \in \mathbf{R}^n$, $u_i \in \mathbf{R}^m$, (A_{ii}, B_{ii}) are system matrices with correct dimensions, and $A_{ij}x_j(k)$ represents the coupling from subsystem j on subsystem i. The scalar e_{ij} represents the coupling strength which varies between 0 (no coupling) and 1 (maximum coupling).

The norm of a matrix will be denoted by $\|\cdot\|$. For an example, $\|x\|_2$ denotes the 2-norm of x which is the root mean square of a vector x which is $\sqrt{\sum_{i=1}^{N} x_i^2}$. The p-norm is denoted by $\|\cdot\|_p$. For a positive definite matrix P, the P-norm is defined as $(x^T P x)^{1/2}$.

We now summarize the important results on connective stability and few other relevant results. These results will be used in our derivations and analysis in the thesis.

1.2.1 M-matrices

A matrix with non-positive off diagonal elements, if it satisfies certain properties, is called an M-matrix [24]. Properties of M-matrices that will be used frequently in this thesis are given below

Theorem 1.1 *If W is an M-matrix, then the following conditions are satisfied and equivalent [24]*

1. *There exist a vector $d \in \mathbb{R}_+^N$, $d_i > 0$, $i \in N$, such that vector $c \in \mathbb{R}_+^N$ defined as*

$$c = Wd \tag{1.2}$$

   is positive, that is $c_i > 0$ for all $i \in N$.

2. *All leading principal minors of W are positive.*

3. *W is a positive quasi-dominant diagonal matrix, that is $w_{ii} > 0$, $i \in N$, and there exists numbers $d_i > 0$ such that*

$$d_i w_{ii} > \sum_{j=1, j\neq i}^{N} d_j w_{ij} \tag{1.3}$$

   $\forall i \in N$.

4. The real part of each eigenvalue of W is positive.

The properties of an M-matrix are useful in determining whether a system is connectively stable.

1.2.2 Connective Stability

Let E denote a matrix where each entry at position i,j represents the strength of the coupling from subsystem j on i (e_{ij}) at a given time. Since the coupling strength $e_{ij} \in [0, 1]$, there are many possible unique instances of E that have different values of e_{ij}. Let $\bar{E}$ denote the set of all such possible E. Then one can define connective stability as follows [24]

Definition 1.1 *A system is connectively stable if it is stable in the sense of Lyapunov for all $E \in \bar{E}$*

The above definition of connectively stability implies that the system without any coupling must also be stable (since $\mathbf{0} \in \bar{E}$). This would be satisfied if and only if each subsystem is stable by itself. Therefore, for a system to be connectively stable A_{ii} must also have all its eigenvalues inside the unit circle, for each i.

If the coupling terms are bounded, it is possible to arrive at a sufficient condition for connective stability [19] [24]. In the case of the subsystem in 1.1, it is clear that the coupling can be bounded using the inequality

$$\|\sum_{j=1}^{N} e_{ij} A_{ij} x_j(k)\|_2 \leq \sum_{j=1}^{N} e_{ij} \xi_{ij} \|x_j\|_2 \tag{1.4}$$

where

$$\xi_{ij} \ = \|A_{ij}\|_2 \ . \tag{1.5}$$

Given that A_{ii} is stable, it is also possible to find a Lyapunov function of the form

$$v_i(x_i) = (x_i^T H_i x_i)^{1/2}, \forall i = 1, 2, .., N \tag{1.6}$$

where H is a positive definite matrix that satisfies

$$A_{ii}^T H_i A_{ii} - H_i = -G_i \tag{1.7}$$

where G_i is also a positive definite matrix. The condition in 1.7 guarantees that the Lyapunov function in 1.6 is decreasing for all $t \geq 0$.

Definition 1.2 *The* ***robustness bound*** *of such a system is shown [19] to be given by*

$$\xi_v(G_i) = \frac{\sigma_m(G_i)}{\sigma_M^{1/2}(H_i)\sigma_M^{1/2}(H_i - G_i) + \sigma_M(H_i)} \tag{1.8}$$

where σ_m and σ_M are the minimum and maximum eigenvalues of the matrix in their respective arguments. The robustness bounds of each subsystem can be used to derive an upper-bound for the Lyapunov function of the system [19], [24]. Therefore, if the robustness bound is more exact, the derived upper-bound will be much tighter. It has also been shown [19] that the robustness bound is more maximized when G_i is the identity matrix and more exact when A_{ii} is in diagonal form.

To analyze stability of the overall system, it is natural to consider a Lyapunov function that is constructed in terms of the Lyapunov functions for each of the individual subsystems as [19], [24]

$$v[x(k)] = \sum_{i \in N} d_i v_i(x_i) \tag{1.9}$$

where d_i is positive for each i.

Using the above definition, Siljak derives the following results [19].

Theorem 1.2 *The rate of decrease of the Lyapunov function given in 1.9, where each subsystem's Lyapunov function is of the form of 1.6, can be upper-bounded as*

$$\triangle v[x(k)] \leq -m^T W \bar{w}(x) \ \ \forall x \in \mathbb{R}^n \tag{1.10}$$

where

$$m = [d_1 \sigma_M^{1/2}(H_1), d_2 \sigma_M^{1/2}(H_2), ..., d_N \sigma_M^{1/2}(H_N)]^T \tag{1.11}$$

$$W_{ij}(x) = \begin{cases} \xi_v(G_i) - e_{ij}\xi_{ij} & \text{if } i = j \\ -e_{ij}\xi_{ij} & \text{if } i \neq j \end{cases} \tag{1.12}$$

$$\bar{w}(x) = [\|x_1(k)\|, \|x_2(k)\|, ..., \|x_N(k)\|]^T \tag{1.13}$$

Theorem 1.3 *A discrete-time connected system is connectively stable if W is an M-matrix, where W is defined as*

$$W_{ij}(x) = \begin{cases} \xi_v(G_i) - e_{ij}\xi_{ij} & \text{if } i = j \\ -e_{ij}\xi_{ij} & \text{if } i \neq j \end{cases} \tag{1.14}$$

for the case when $e_{ij} = 1$

The system matrix A_{ii} of each subsystem i determines the robustness bound $\xi_v(G_i)$ from 1.7, 1.8 and the strength of the coupling and its dynamics are factored in through $e_{ij}\xi_{ij}$.

1.2.3 Input to State Stability

In a connected system, the coupling can be seen as an incoming input to each subsystem. It can be useful to be able to predict the stability of a system based on information available about its inputs and dynamics of that system. The following theorem gives such a result [8]

Theorem 1.4 *A system is input-to-state stable if there exists a KL-function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and a K-function γ such that, for each input u and initial state ξ, it holds that*

$$|x(k, \xi, u)| \leq \beta(|\xi|, k) + \gamma(\|u\|) \tag{1.15}$$

for each $k \in \mathbb{Z}_+$

In the above, the function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a $K - function$ i.e. it is continuous, strictly increasing and $\gamma(0) = 0$ [8]. The function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a K *L-function* i.e. for each fixed $t \geq 0$, the function $\beta(\cdot, t)$ is a *k=function*, and for each $s \geq 0$ the function $\beta(s, \cdot)$ is decreasing and $\beta(s, t) \to 0$ as $t \to \infty$ [8].

This theorem allows us to conclude that a system is stable given that the inputs to a system are bounded and the system is stable without any inputs.

The rest of the thesis is organized as follows. In Chapter 2, we use the results on connective stability to analyze systems with different interconnection topologies. In Chapter 3, we develop a scheduling algorithm for a centralized system where only one subsystem can transmit its state to others in the network at each time step. In Chapter 4, we extend the scheduling algorithm for a decentralized system and consider a system where multiple subsystems can transmit their state at once.

Chapter 2

Connective Stability of Common Topologies

2.1 Introduction

In this chapter we study different topologies and how previous work on connective stability [19], [24] applies to these structures. In Chapter 1, we presented Theorem 1.3 which was a sufficient condition for connective stability. The goal of the analysis we present in this Chapter is to identify simplified tests and necessary conditions for connective stability given specific topologies. These simplified stability checks would act as a schedulability check for the scheduling policy we will develop in subsequent sections. There exists previous work [21] [23] which analyzes different matrix types (such as Metzler matrices and interval matrices) to develop stricter or less conservative results for connective stability. But Metzler matrices differ from the matrices that occur in our work in that it requires off-diagonal elements to be non-negative. In the situations we consider, from the definition of the W matrix in Equation 2.1, and e_{ij} in Equation 1.5, the off-diagonal elements of W will always be non-positive i.e., W is a "Z-matrix". Our work also deals with the variation of coupling anywhere in the interval of $[0, 1]$ and therefore gives a more general result than with interval matrices.

2.2 Modeling the Interconnection of the System as a Graph

The interconnection topology of each system discussed is captured within the W matrix (given in (2.1)). The W matrix acts as an adjacency matrix in that

$$W_{ij} = \begin{cases} \text{non zero} & \text{if state of subsystem } j \text{ affects } i \\ 0 & \text{if no coupling.} \end{cases} \tag{2.1}$$

This also means that the system can be treated as a graph where the coupling strength between two plants would be captured as a weighted edge. We can therefore refer to graph properties such as cycles when we refer to the system topology. A system (or equivalent graph) that has no cycles will be referred to as an *acyclic* topology. Since $W_{ij} \neq W_{ji}$, the adjacency matrix (W matrix) of such systems will represent that of a directed graph and will therefore be referred to as a *directed* topology.

Examples of an interconnected system with a directed cycle topology is shown in figure 2.1 and a directed acyclic topology is shown in figure 2.2.

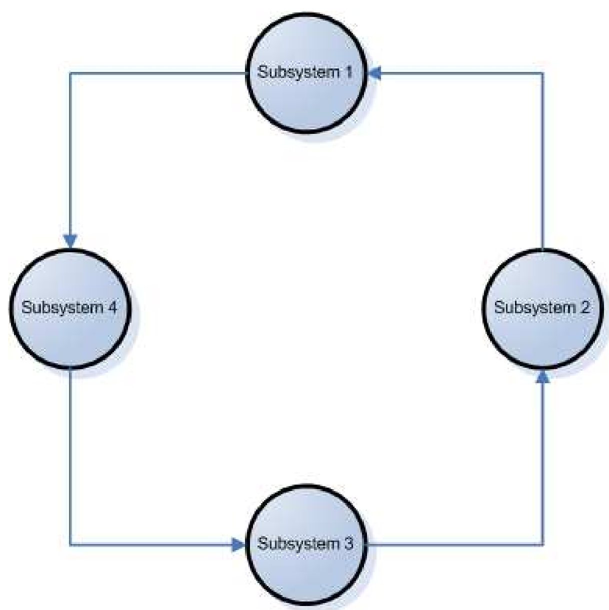

Figure 2.1: Example of a Directed Cycle Topology

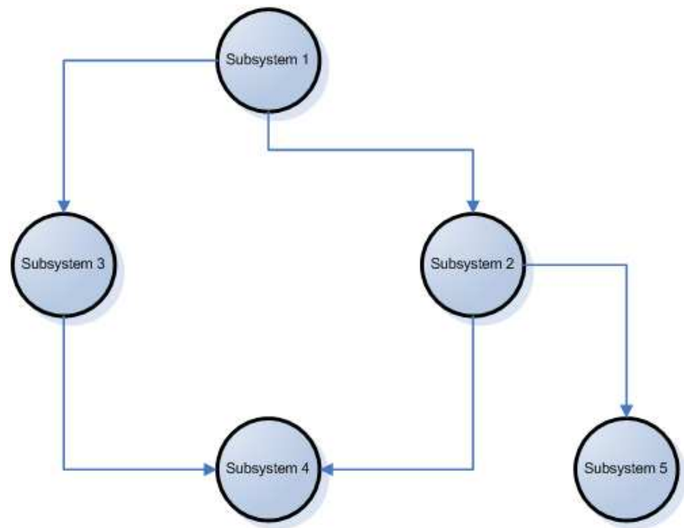

Figure 2.2: Example of a Directed Acyclic Topology

2.3 Directed Acyclic Topology

Theorem 2.1 *If a coupled system has a directed acyclic topology, then the system is connectively stable if and only if each subsystem by itself (with no coupling) is stable.*

Proof: The necessity of this is clear from the definition of connective stability which requires that the subsystems are stable.

Then we prove sufficiency by induction as follows.

In a directed acyclic topology, there must be subsystems with no incoming connections. These subsystems we label as level 1 subsystems. The subsystems only having incoming connections from level 1 subsystems we label as level 2. The ones that have incoming connections from level 1 and level 2 subsystems we call level 3 and so on.

Now each subsystem at level k can only have incoming connections from subsystems that belong to level $k-1$ and less since the topology is directed and acyclic. So we proceed to prove the sufficiency condition by induction as follows.

If a subsystem is a level 1 subsystem, these subsystems are by-themselves stable (the condition of the theorem). The inputs to the level 2 subsystems, which are a linear combination of the states of level 1 subsystems, are asymptotically approaching zero since level 1 subsystems are stable. Since the inputs to the level two subsystems are approaching zero, and subsystems by themselves are stable, we can also know from the results of input to state stability (Theorem 1.4) that level 2 subsystems are also stable.

Let us now assume subsystems at level k are stable. Then the subsystem from level one to level k must also be stable. Subsystems at level $k+1$ will only have incoming connections from subsystems at level 1 to level k. Since all these subsystems from level 1 to k are stable, the inputs to level $k+1$ subsystems are also approaching zero. Therefore, from input to state stability, we see that subsystems at level $k+1$ are also stable since inputs are approaching zero and the subsystems by themselves are also stable.

Thus, we have proved by induction that if the subsystems are stable by themselves and are connected according to a directed acyclic topology, then the system is connectively stable. □

The above result also implies that when we deal with directed acyclic systems, we need not worry about the dynamics of the interconnections but only of the stability of each subsystem. The result also suggests that it might be possible to refine the connective stability test presented by Siljak [19], [24].

Before we provide a refined connective stability test, the following lemma must be developed since it is useful for proving the validity of the refined test.

Lemma 2.1 *If in a connected system, the subsystems by themselves are stable, then the corresponding robustness bound of each subsystem is positive.*

Proof: The Robustness bound of a system i was given by Equation 1.8 as

$$\xi_v(G_i) = \frac{\sigma_m(G_i)}{\sigma_M^{1/2}(H_i)\sigma_M^{1/2}(H_i - G_i) + \sigma_M(H_i)} \ .$$

By definition, G_i and H_i are positive definite matrices which only exist if the system is stable. Since they are positive definite, the eigenvalues are all positive. Therefore all their eigenvalues $\sigma_m(G_i)$, $\sigma_M(H_i)$ are all positive as well.

With respect to $\sigma_M^{1/2}(H_i)$, it appears that it can be either negative and positive. But the origin of this quantity in the expression for the robustness bound comes from the upper-bounding of the Lyapunov Function in terms of its norm [19] i.e.,

$$\sigma_m^{1/2}(H_i)\|x_i\|_2 \leq (x_i^T H_i x_i)^{1/2} \leq \sigma_M^{1/2}(H_i)\|x_i\|_2 \ .$$

Since the Lyapunov function is a positive decreasing function, the upper-bound of the function cannot be negative. So the quantity $\sigma_M^{1/2}(H_i)$ in this context must be positive.

Also, H_i and G_i satisfy

$$A_i^T H_i A_i - H_i = -G_i$$
$$\Leftrightarrow A_i^T H_i A_i = H_i - G_i \ .$$

Now diagonalizing H_i as $P^T \lambda P$ where λ is a matrix with all the eigenvalues of H_i on the diagonal, we get

$$A_i^T P^T \lambda P A_i = H_i - G_i$$
$$\Leftrightarrow (PA_i)^T \lambda (PA_i) = H_i - G_i \ .$$

Using the above expression on the LHS for $H_i - G_i$, we get that for any vector y

$$y^T(H_i - G_i)y = y^T(PA)^T \lambda (PA_i)y$$
$$= (PA_i y)^T \lambda (PA_i y) \ .$$

Let $(PA_i y) = y'$, then

$$y^T(H_i - G_i)y = y'^T \lambda y$$
$$= y'^2(1)\lambda(1,1) + ... + y'^2(N)\lambda(N,N)$$

Since $y'^2(n) \geq 0, \forall n = 1, 2, ..N$ and λ is a diagonal matrix with positive values, the above sum is also non-negative. Hence $y^T(H_i - G_i)y > 0$ for all y. Therefore, $(H_i - G_i)$ is positive semi-definite. Therefore, all eigenvalues of $(H_i - G_i)$ are non-negative.

Since $\sigma_M^{1/2}(H_i - G_i)$ is non-negative and all the other terms in the expression for the robustness bound are positive, the robustness bound itself must be positive. $\square$

Extending the connective stability test in Theorem 1.3, we can also say the following

Theorem 2.2 *A connected system with directed acyclic topology is connectively stable if and only if the W-matrix is an M-matrix.*

Proof: From the work shown previously, it is clear that if the W-matrix of a system is an M-matrix, then *any* system is connectively stable. What remains therefore to be shown is that if a system with a directed, acyclic topology is stable, then the W-matrix must also be an M-matrix.

This can be shown as follows:

For a directed acyclic topology with N sub-systems, the subsystems can be ordered such that $W_{i,j=0}$ for $j > i$ in the W-matrix since $e_{ij} = 0$. Also, for the system to be connectively stable, by definition, each sub-system by itself must be stable. If each subsystem by itself is stable, then the robustness bound is positive (from Lemma 2.1).

If the robustness bounds are positive, all the elements in the diagonal of the W-matrix are positive (it is assumed that $e_{ii} = 0$). Since W is a triangular matrix and principal minors are therefore the product of diagonal elements, all the principal minors of the W-matrix are also positive. Thus, the W-matrix must also be an M-matrix. □

2.4 Scalar Systems with a Directed Cycle Topology

For this analysis, we will use the following system which is an equivalent scalar system to the system in 1.1

$$x_i(k+1) = a_i x_i(k) + \gamma_i u_i(k) + \sum_{j=1}^{N} e_{ij}\beta_{ij}x_j(k). \tag{2.2}$$

We will consider the case where $\gamma_i u_i(k) = 0$ since we will not discuss how to determine the optimal input term $u_i(k)$ to apply in this section. For this chapter, we will limit our analysis to determining only whether a system is connectively stable.

2.4.1 Heterogeneous Systems

In this section, we consider general systems with non-homogeneous dynamics i.e. $a_1 \neq a_2 \neq .... \neq a_N$

Theorem 2.3 *If a connected network of subsystems (where each system is stable itself) has a topology of a directed cycle, then the system is connectively stable if the determinant of the corresponding W-matrix (given in Equation 2.1) is positive.*

Proof: If the W matrix is an M-matrix, the system is connectively stable. According to Theorem 1.1, a matrix is an M-matrix if all its principal minors are positive.

If the graph is directed and cyclic, one can construct the W matrix for the system with N subsystems of the form in 2.2 as

$$W_{ij} = \begin{cases} \zeta_{ij} & \text{if } i = j \\ -\|\beta_{ij}\|_2 & \text{if } i = j-1 \text{ or } i = 1, j = N \\ 0 & \text{others} \end{cases}$$

where ζ_{ij} is the robustness bound. Therefore, the W-matrix will be lower triangular except for the non-zero element $-\|\beta_{1,N}\|_2$.

All principal minors up to $N-1$ x $N-1$, will simply be the product of the diagonals (since $-\|\beta_{1,N}\|_2$ is not part of any of these principal minors, each principal minor is a triangular matrix). Since the robustness bounds are positive, the product of robustness bounds (ζ_{ij}) is also positive, making the principal minors up to the $N-1$th positive. So the only principal minor that needs to be checked for is the N x N principal minor. This is just the determinant. Therefore, if the determinant is also positive, then the W-matrix has all positive principal minors, and is therefore an M-matrix, which then from Theorem 1.3 gives connective stability. □

We can also derive an expression for the robustness bound in terms of the subsystem scalar a_i as follows.

Theorem 2.4 *If a connected network of N scalar subsystems (which are stable by themselves) is of the form*

$$x_i[k+1] = a_i x_i[k] + b_i x_j[k], \forall i, j = 1, 2, ..N$$

where $b_i x_j[k]$ is the coupling due to interconnections, then the robustness bound of each subsystem i is given by $1 - |a_i|$.

Proof: To calculate the robustness bound using Equation 1.8, we require H_i and G_i such that Equation 1.7 is satisfied. Setting $G_i = 1$ (it is shown in [19] that $G_i = I$ maximizes the robustness bound) in Equation 1.7, we obtain

$$\begin{aligned} a_i^2 H_i - H_i &= -1 \\ H_i &= \frac{1}{1-a_i^2} \end{aligned}$$

Substituting the above values into the robustness bound Equation 1.8 gives

$$\begin{aligned} \xi_v(G_i) &= \frac{1}{\frac{1}{(1-a_i^2)^{1/2}}(\frac{1}{(1-a_i^2)} - 1)^{1/2} + \frac{1}{(1-a_i^2)}} \\ &= \frac{1}{\frac{1}{(1-a_i^2)^{1/2}}\frac{|a_i|}{(1-a_i^2)^{1/2}} + \frac{1}{(1-a_i^2)}} \\ &= \frac{1}{\frac{|a_i|+1}{(1-a_i^2)}} \\ &= \frac{1}{\frac{|a_i|+1}{(1-|a_i|)(1+|a_i|)}} \\ &= \frac{1}{\frac{1}{(1-|a_i|)}} \\ &= 1 - |a_i| \end{aligned}$$

as required. □

2.4.2 Homogeneous Systems

In this section, we consider the systems where the dynamics a_{ii}, b_{ij} are the same. The goal of analyzing such systems is to investigate the possibility of arriving at a necessary and sufficient condition for connective stability.

Theorem 2.5 *If a connected network (with a directed cycle topology) of N homogeneous scalar subsystems (where each system is stable by itself without coupling), is of the form*

$$x_i[k+1] = ax_i[k] + bx_j[k],\ \forall i = 1, 2, .., N,\ j = \begin{cases} i+1 & \textit{if } i < N \\ 1 & \textit{if } i = N \end{cases}$$

where N is an even number, the system is connectively stable if and only if $|a| + |b| < 1$

Proof: For such a system, the W-matrix will be of the form

$$W_{ij}(x) = \begin{cases} \alpha & \text{if } i = j \\ -\beta & \text{if } i = j-1 \text{ or } i = 1, j = N \\ 0 & \text{else} \end{cases}$$

where α is the robustness bound according to the previous theorem, $\alpha = 1 - |a|$ and $\beta = |b|$. In Theorem 2.3 (on determining connective stability of a system with directed cyclic topology), it was shown that we only require the determinant of the W-matrix to be positive. Using the theorem gives us

$$\begin{aligned} \alpha^N - (\beta)^N &> 0 \\ \Leftrightarrow \alpha^N &> (\beta)^N \\ \Leftrightarrow \alpha &> \beta \\ \Leftrightarrow 1 - |a| &> |b| \\ \Leftrightarrow |a| + |b| &< 1 \end{aligned}$$

as required.

We prove necessity as follows. According to the condition of the theorem, we have that

$$|a| + |b| < 1 \ .$$

So to show that this condition is necessary for stability, we must show that if $|a|+|b| > 1$, the system is unstable.

Now the system matrix A consisting of all N subsystems will be of the form

$$A_{ij} = \begin{cases} a & \text{if } i = j \\ b & \text{if } i = j-1 \text{ or } i = 1, j = N \\ 0 & \text{else} \end{cases}$$

Since a and b can be positive or negative, we must consider all possibilities.

If $a > 0, b > 0$ the characteristic equation for calculating the eigenvalues λ would be

$$(\lambda - |a|)^N - (-|b|)^N = 0$$

to which $\lambda = |a| + |b|$ is a solution. So if $|a| + |b| > 1$, then one eigenvalue of the system would be greater than 1 making the system unstable.

If $a < 0, b > 0$ the characteristic equation for calculating the eigenvalues λ would be

$$(\lambda + |a|)^N - (-|b|)^N = 0$$

to which $\lambda = -|a| \pm |b|$ is a solution, which means $\lambda = -(|a| + |b|)$ is a solution. So if $|a| + |b| > 1$, then one eigenvalue of the system would have magnitude greater than 1 making the system unstable.

If $a > 0, b < 0$ the characteristic equation for calculating the eigenvalues λ would be

$$(\lambda - |a|)^N - (|b|)^N = 0$$

to which $\lambda = |a| + |b|$ is a solution. So if $|a| + |b| > 1$, then one eigenvalue of the system would have magnitude greater than 1 making the system unstable.

If $a < 0, b < 0$ the characteristic equation for calculating the eigenvalues λ would be

$$(\lambda + |a|)^N - (|b|)^N = 0$$

to which $\lambda = -(|a| + |b|)$ is a solution. So if $|a| + |b| > 1$, then one eigenvalue of the system would be greater than 1 making the system unstable. □

In the case of having odd number of plants (N is odd), it was not possible to construct a sufficient and necessary condition. But a simplified sufficient condition for stability can still be presented as follows.

Theorem 2.6 *If there are N subsystems in a connected system (in a directed cycle topology), where N is odd, then the system is connectively stable if $|a| + |b| < 1$.*

Proof: Follows the same form as proof of Theorem 2.5 $\square$

It is however possible to produce a necessary condition for connective stability for systems with an odd number of subsystems as well. The result, unlike in the case with an even number of subsystems, depends on the whether a, b are positive or negative. We summarize the results in the theorem below.

Theorem 2.7 *If there are N subsystems, where N is odd, then the system is unstable*

1. *If $a > 0$, $b > 0$, and $|a| + |b| > 1$*
2. *If $a > 0$, $b < 0$, and*
$$N > 2\pi(2k+1)/\cos^{-1}[(1 - |a|^2 - |b|^2)/2|a||b|]$$
3. *If $a < 0$, $b > 0$, and*
$$N > 2\pi(2k)/\cos^{-1}[-(1 - |a|^2 + |b|^2)/2|a||b|]$$
4. *If $a < 0$, $b < 0$, and*
$$N > 2\pi(2k+1)/\cos^{-1}[-(1 - |a|^2 - |b|^2)/2|a||b|]$$

where $k = -(N-1)/2, ... - 1, 0, 1, ..., (N-1)/2$.

Proof:

If $a > 0, b > 0$ the characteristic equation for calculating the eigenvalues λ is

$$(\lambda - |a|)^N + (-|b|)^N = 0$$

to which $\lambda = |a| + |b|$ is a solution. So if $|a| + |b| > 1$, then one eigenvalue of the system would be greater than 1 making the system unstable.

If $a > 0, b < 0$ the characteristic equation for calculating the eigenvalues λ is

$$(\lambda - |a|)^N + (|b|)^N = 0$$

to which $\lambda = |a| + |b|e^{j2\pi(2k+1)/N}$ where $k = -(N-1)/2, ... -1, 0, 1, ..., (N-1)/2$. One can find the specific N for which the system is unstable by solving as before

$$\begin{aligned}
||a| + |b|e^{j2\pi(2k+1)/N}| &> 1 \\
|a|^2 + 2|a||b|\cos[2\pi(2k+1)/N] + & \\
|b|^2\cos^2[2\pi(2k+1)/N] + & \\
|b|^2\sin^2[2\pi(2k+1)/N] &> 1 \\
2|a||b|\cos[2\pi(2k+1)/N] &> [1 - (|a|^2 + |b|^2)] \\
\cos[2\pi(2k+1)/N] &> [1 - (|a|^2 + |b|^2)]/2|a||b|
\end{aligned}$$

Therefore, if RHS of the inequality is less than -1, then any N satisfies the requirement. More generally, note that the RHS is strictly less than 1, and the argument of the LHS goes to zero as N increases. Thus, for N sufficiently large, this condition will be satisfied. i.e., for

$$N > 2\pi(2k+1)/\cos^{-1}[[1 - (|a|^2 + |b|^2)]/2|a||b|]$$

all k between $k = -(N-1)/2, ... -1, 0, 1, ..., (N-1)/2$.

If $a < 0, b > 0$ the characteristic equation for calculating the eigenvalues λ would then be

$$(\lambda + |a|)^N + (-|b|)^N = 0$$

to which $\lambda = -|a| + |b|e^{j2\pi k/N}$ is a solution. The system would be unstable if $|\lambda| = |-|a| + |b|e^{j2\pi k/N}| > 1$. Therefore, using the same procedure as above we obtain

$$N > 2\pi(2k)/\cos^{-1}[-(1 - |a|^2 + |b|^2)/2|a||b|] \ .$$

If $a < 0, b < 0$ the characteristic equation for calculating the eigenvalues λ would then be

$$(\lambda + |a|)^N + (|b|)^N = 0$$

to which $\lambda = -|a| + |b|e^{j2\pi(2k+1)/N}$ where $k = -(N-1)/2, ... -1, 0, 1, ..., (N-1)/2$. One can then find the specific N for which the system is unstable by using the same procedure as above to show that $N > 2\pi(2k+1)/\cos^{-1}[(1 - |a|^2 - |b|^2)/2|a||b|]$.

$\square$

2.5 Summary

In this section, we have formulated simplified connective stability tests for various topologies. We also presented conditions under which the system could be stabilized. These results will act as simplified schedulability tests for our scheduling policy when the underlying network topology matches those considered in this section. It will also help in restructuring the network or modifying the network to achieve stability.

Chapter 3

Scheduling Scheme based on Connective Stability

3.1 Introduction

In this chapter we extend the previous work on connective stability [19] to develop a scheduling algorithm. The system analyzed will consist of subsystems of the form given in (1.1). We assume that in the connected system, each subsystem is stable without any coupling i.e. A_{ii} has all eigenvalues inside the unit circle. We also assume that there is a centralized scheduler that can observe the state of each subsystem. The connected system will allow a single subsystem to transmit its state to all other subsystems at a given time slot. The centralized controller will determine which subsystem should transmit based on its impact in increasing the rate of decrease of an appropriate Lyapunov function of the entire system. Such restricted access to the communication resource may be present in systems where there is a single resource that is shared by all subsystems. [1] Since the Lyapunov function only gives an upper-bound on the actual evolution of the state, the scheduling algorithm will be based on an approximation. In this section, we will show that knowing the state of another subsystem and using that information to reduce the effects of coupling on another subsystem is beneficial. Then we will determine the optimal input that a controller should apply to the subsystem to reduce the effects of coupling by another subsystem. Using the results from our analysis, we will devise a scheduling algorithm.

[1] In the next Chapter of this thesis, we will remove this restriction to address a general model where the communication resource can be shared between more than one subsystem at a time.

Finally, we will compare the scheduling algorithm against other possible schemes through simulations.

3.2 Effect of Reducing Coupling

In a connected system of N subsystems, let $\alpha_{ij} \in [0, 1]$ be the additive reduction in coupling e_{ij} (recall that e_{ij} is the coupling by subsystem j on subsystem i). The resultant coupling will therefore be $e_{ij} = 1 - \alpha_{ij}$. The resultant W matrix (defined in (2.1)) W' can then be defined as

$$W'_{ij}(x) = \begin{cases} \xi_v(G_i) - (1 - \alpha_{ij})\xi_{ij} & \text{if } i = j \\ -(1 - \alpha_{ij})\xi_{ij} & \text{if } i \neq j \end{cases} \tag{3.1}$$

Using Theorem 1.2, the change in the upper-bound on the rate of decrease in the Lyapunov function is defined as

$$\begin{aligned} -m^T W w(x) - (-m^T W' w(x)) &\triangleq -m^T (\triangle W'(x)) w(x) \\ -m^T (\triangle W'(x)) w(x) &= -\sum_{i=1}^{N} m_i (\sum_{j=1}^{N} \alpha_{ij} \xi_{ij} \|x_j\|_2) \end{aligned} \tag{3.2}$$

According to the above, when the coupling is reduced, the upper bound on the decrease in the Lyapunov function is larger compared to having a coupling strength of 1. So it can be concluded that reducing the coupling may be beneficial in increasing the rate of convergence toward stability.

We can therefore state the following theorem using the above result.

Theorem 3.1 *An estimate for the increase in the overall system's Lyapunov function's rate of decrease, as a result of reducing the coupling effects from a subsystem j is* $\sum_{i=1}^{N} m_i (\sum_{j=1}^{N} \alpha_{ij} \xi_{ij} \|x_j\|_2)$

3.3 Optimal Input to Apply

From the above section, it is clear that reducing the coupling may increase the rate of decrease of the Lyapunov function for the overall system. If the states of other subsystems are known, it should be possible for that subsystem to apply an input which would act to reduce or negate the effect of coupling from other states. This section analyzes this possibility.

We will analyze the system described in (1.1) where the input term is given by $B_{ii}u_i(k)$. The matrix B_{ii} is the input matrix of the system while $u_i(k)$ is free to be appropriately chosen. The question we seek to answer is what would be the optimal input $u(k)$ to apply.

To constrain the problem, we will limit our analysis to a system where only one subsystem may transmit its state to all other subsystems at a given time step k. It is also assumed that each of the subsystems are aware of the nature of the coupling matrices A_{ij}. Thus, when the other subsystems receive the state that has been transmitted, each of them can apply an input to compensate or negate the effects of the coupling. Given that subsystem j transmitted its state to all other subsystems, the problem of finding the best input u_i for subsystem i to apply in order to minimize the coupling effects from subsystem j can be formulated as

$$\text{Optimal } u_i(k) = \text{argmin}_{u_i(k)} \|B_{ii}u_i(k) + A_{ij}x_j\|_2 \tag{3.3}$$

$$= -(B_{ii}^T B_{ii})^{-1} B_{ii}^T A_{ij} x_j \tag{3.4}$$

In order to calculate the reduction in coupling achieved by applying this input, we can substitute this optimal $u_i(k)$ back into (1.1) and obtain an expression for the new coupling which we denote A'_{ij}, as

$$A'_{ij}x_j = [-B_{ii}(B_{ii}^T B_{ii})^{-1} B_{ii}^T + I]A_{ij}x_j \ .$$

Bounding both sides of the equation above in terms of the 2-norm one gets the following

$$\|A_{ij}\| = \| - B_{ii}(B_{ii}^T B_{ii})^{-1} B_{ii}^T + I\| \|A_{ij}\|$$

Then by comparing the the LHS and RHS of the above equation, we obtain an expression for the new reduced coupling e'_{ij} as

$$e'_{ij} = \| - B_{ii}(B_{ii}^T B_{ii})^{-1} B_{ii}^T + I\| \ . \tag{3.5}$$

Interestingly, the above result shows that the maximum possible reduction in coupling is independent of the state of the subsystem. This makes it possible to calculate these values prior to system operation for all subsystems.

Given the above results, one can now formulate a scheduling policy that can determine which subsystem state to transmit at a given time slot. We assume for now that a centralized monitor is aware of the state of each subsystem. So the goal of this scheduling process is to best schedule the transmission of state information so that when each subsystems applies this received state information to reduce the coupling, it will result in the best decrease in the overall Lyapunov function for that time step.

In the scheduling policy we propose in this section, the possible decrease in the Lyapunov function is estimated by using the M-matrix based results from the previous Chapter. The subsystem state whose transmission would lead to the largest decrease is chosen to be transmitted. We particularly use the result of Theorem 3.1 to estimate the possible increase in the rate of decrease of the Lyapunov function for a system. Since only one subsystem will be transmitting at each time step, the difference in rate of decrease due to transmission of a particular subsystem z can be estimated as $-\sum_{i=1}^{N} m_i e'_{iz} \xi_{iz} \|x_z\|$. This reduction of the result in according to Theorem 3.1 is possible since all the α_{ij} terms other than for $j = z$ will be zero. So only the state of the plant that is being transmitted affects the change in the upper bound of the rate of decrease in the Lyapunov function.

3.4 Scheduling Algorithm

Combining the results above, we can formulate the following algorithm

Algorithm 3.1 *The Precomputations:-*

1. *Compute the* ξ_{ij} *terms using* $\xi_{ij} = \|A_{ij}\|_2$.
2. *Compute the robustness bounds* $\xi_v(G_i)$ *using Equation* (1.8) *for each plant* i.
3. *Compute the W-matrix as given in* (2.1).
4. *Check if* W *is an M-matrix. If so, the system is connectively stable. Find a positive row vector* d *such that* dW *is a positive vector.*
5. *Calculate the maximum possible reduction in coupling* e'_{ij} *using* (3.5).

The scheduling policy:-

1. *At each time step, and for each subsystem j, evaluate $P_j = -\sum_{i=1}^{N} m_i e'_{ij} \xi_{ij} \|x_j\|$.*

2. *Pick the subsystem j whose transmission leads to the maximum decrease in the Lyapunov function (i.e. $argmin_j P_j$).*

3. *Each subsystem $i \neq j$ will then apply the input u_i as given in Equation (3.4).*

As can be seen above, the steps that require the bulk of the computation can already be calculated off-line and be stored. The run-time computations or the scheduling policy itself require relatively simple operations of multiplications and additions.

3.5 Simulation

To evaluate the above scheduling policy, we evaluated it against four other policies via simulations.

1. Normal Coupling: Here the system is left to evolve naturally. No states are transmitted and no reduction in coupling takes place. This provides the natural evolution of the system state to compare against other solutions.

2. H-norm Reduction: Applies the scheduling algorithm formulated in Algorithm 1.

3. Random: Randomly picks which state to send to the other plants at each time step.

4. Greedy: Picks the state to send by evaluating all the possibilities to send and picking the one that gives the most reduction in the Lyapunov function defined in (1.6). This is different from the 'H-norm Reduction' since the 'H-norm Reduction' is based on an upper bound of the Lyapunov function given in Theorem 1.2. The 'Greedy' algorithm evaluates the resulting state of each subsystem after a time step and the resulting value of the Lyapunov function. In this way, it evaluates all the possible decreases in Lyapunov function as a result of transmitting the state of each subsystem j. Then it chooses the subsystem that results in the maximum decrease.

5. Brute Force: This method operates in the same way as the Greedy Policy except it looks ahead to multiple time steps to decide on which current transmission will give the best reduction of the overall system state (as computed by the Lyapunov function) in the future.

Note: The 'Brute Force' and 'Greedy' require the exact decrease in the Lyapunov function (unlike the 'H-norm policy' that operates using an upper bound'). To perform such an evaluation, the scheduler needs access to the current state of the system as well as all the possible resulting states of the system due to reduced coupling. Evaluating all the possible resulting states of the system is a computationally heavy procedure and cannot be supported by all systems. This makes 'Brute Force' and Greedy' less desirable options compared to the 'H-norm reduction' policy.

The reason for simulating the system with coupling and no scheduling algorithms is that it allows us to measure the effect of reducing the coupling. Now as described in the above sections, even though the scheduling policy requires only simple computations for its operation, it still does need some computation. Therefore, it is also useful to compare it against a Random scheduling policy where no computation is required and the subsystem to transmit is arbitrarily selected. If a random scheduling policy performs as well as the proposed scheduling policy, then it would be meaningless to implement the scheduling policy on a system.

While a Random scheduling policy and 'Normal Coupling' were included to observe whether the new scheduling scheme is justified, the 'Brute-Force' and 'Greedy' gives us a measure of how well our scheduling policy performs against computationally heavy policies that require future state information. Both 'Greedy' and 'Brute Force' policies are computationally intensive to be implemented for a system but can give insight as to how well one can perform if the scheduler were able to have more knowledge and resources. Since the brute-force evaluates the possible improvement of the system state over multiple time steps to the future, it can make a better decision on the subsystem state to transmit at a given time such that it will have the best possible impact a certain number of time-steps in to the future. The 'Greedy' policy on the other hand, is simply the 'Brute-Force' policy that looks ahead to only by one time step. In other words, it transmits the state of the subsystem that can have the best impact on the overall system in the next immediate time step.

The simulations were conducted for systems with various number of subsystems, different number of state variables, and different interconnection topologies.

Figures 3.1 and 3.2 show a system with 5 subsystems where each contains 2 state variables (the data for the system used in the simulation is provided in the Appendix).

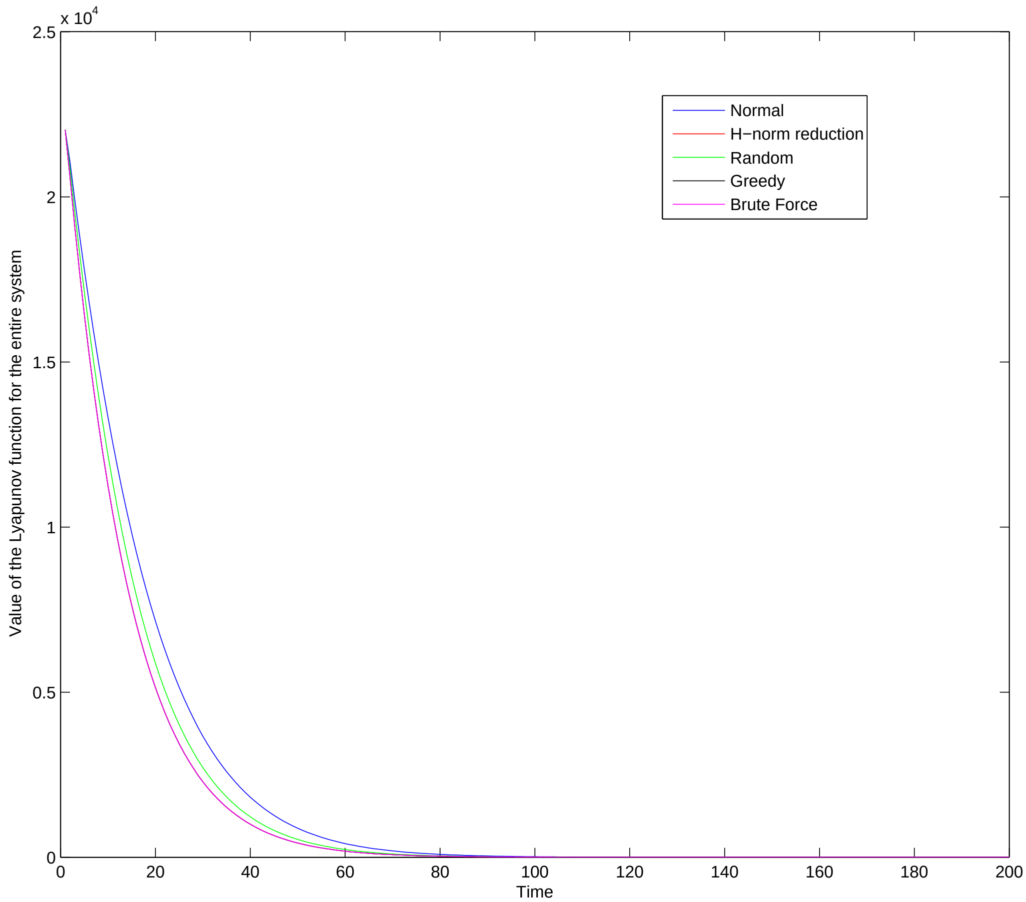

Figure 3.1: Value of Lyapunov function under different scheduling policies

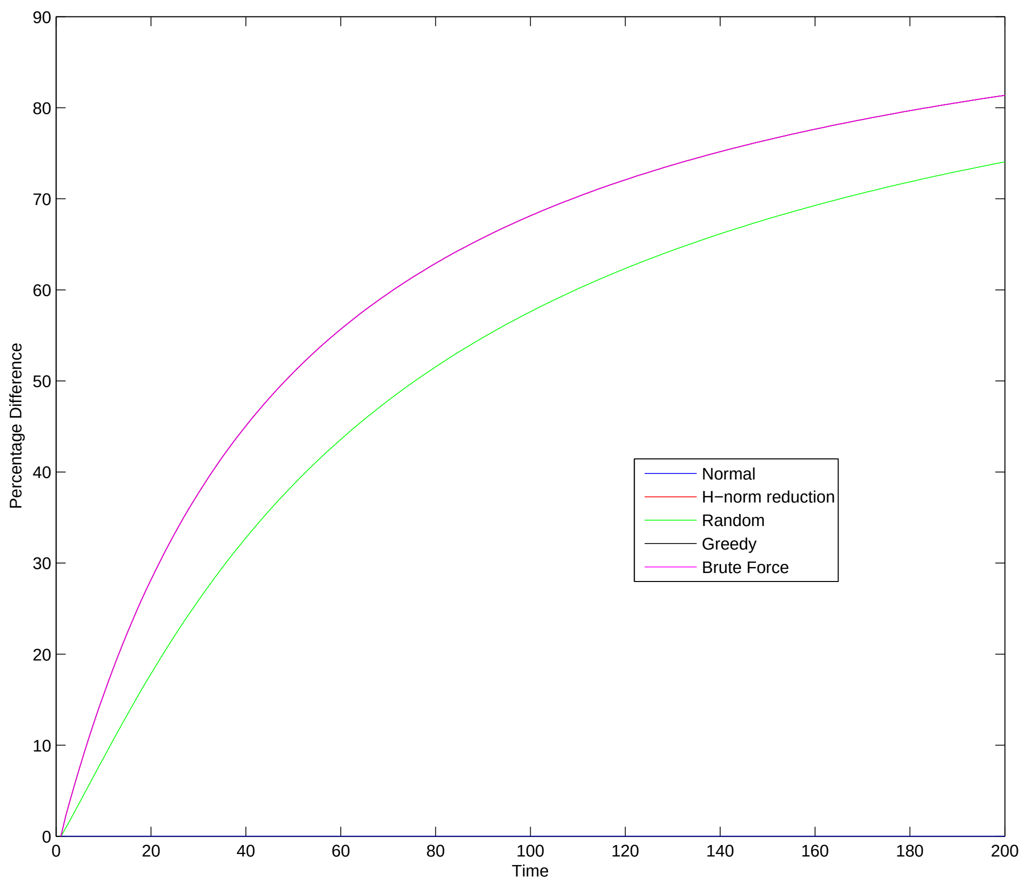

Figure 3.2: Comparison of improvement in performance by each scheduling policy

Figure 3.1 shows the evolution of the state of each system under different scheduling policies. The Figure 3.2 plots the percentage improvement of the four transmission policies compared to the 'Normal Coupling' scenario.

Note that the 'Brute-Force', 'Greedy' and 'H-norm reduction' (our proposed scheduling policies) perform equally in this case and therefore MATLAB plots it as one line. Also, the lookahead was set to 2 time steps in the above simulation to save computation time. A lookahead of more than 2 time steps resulted in a simulation times reaching more than an hour.

As seen from the above plot, the scheduling algorithm proposed in the previous sections does perform better than the random policy and 'Normal Coupling'. The performance difference can be high as 20% during certain periods. In general, while the proposed scheduled policy does perform significantly better than the random scheduling policy and 'Normal Coupling', the 'Greedy' and 'Brute-Force' does perform better in systems with certain number of states and combination of initial values. But this has not been by a significant amount compared to the improvement over 'Normal Coupling' and the random scheduling policy.

Therefore, the simulation results can be considered to validate the scheduling policy as a competitive and reliable policy in minimizing the effects of coupling on the state of the system.

From the Figure 3.2, it can be seen that the proposed algorithm gives more than a 50 percent improvement in performance compared to 'Normal Coupling' starting from around 40 time steps onwards. Also important is that the proposed algorithm performs as well as a greedy algorithm and has a significant 15 percent improvement over random scheduling.

It must be noted that in some cases, with complex interconnections between subsystems, random performs much worse than the proposed algorithm.

3.6 Summary

In this section, we have developed a centralized scheduling policy based on connective stability. We have also provided simulation results that show the performance improvement of the scheduling policy compared to competing policies. In the next chapter we will extend the Algorithm 1 in this section by removing the constraint of transmitting one subsystem state per time step.

Chapter 4

Extensions

4.1 Introduction

In the previous chapter, we considered a centralized system where only one subsystem is allowed to transmit at a given time. Here we relax this constraint by first studying a decentralized system and then introducing simultaneous multiple transmissions.

4.2 Decentralized Control

We consider now the extension of the above scheduling policy in the absence of a centralized controller. In such an arrangement, it is desirable that each subsystem performs part of the computation required to decide which subsystem state should be transmitted at a given time step.

The possible improvement of the decrease in the Lyapunov function given by Equation 3.2 can be simplified to $-\sum_{i=1}^{N} m_i e'_{ij}\xi_{ij}\|x_j\|$ as we indicated in the previous section. To evaluate the expression, one only requires knowledge of the W-matrix, the d_i and the eigenvalues of H_i, all of which are computed off-line. Then the other piece of information required is the state of the subsystem that will be transmitted.

Since each subsystem is aware of its own state, each subsystem j can evaluate the expression $-\sum_{i=1}^{N} m_i e'_{ij}\xi_{ij}\|x_j\|$. Then the subsystem that has the highest decrease should be allowed to transmit its state.

Therefore the scheduling policy for a decentralized system will be as follows:-

1. At each time step, each subsystem j evaluates P_j for $P_j = -\sum_{i=1}^{N} m_i e'_{ij}\xi_{ij}\|x_j\|$, where $j \in \{1, 2, ..N\}$
2. The subsystem with the smallest P_j (largest negative P_j) will transmit for that time step.

The first step above differs from the previous centralized policy in that it only requires each subsystem to have information about its own state and dynamics of the other subsystems. Since the information on the dynamics of each subsystem can be stored within each subsystem, the first step above can proceed without any communication between the subsystems during runtime.

Step 2 would now be different from the centralized policy in that there is no centralized scheduler that can determine who has the smallest P_j. We will address the question of dealing with this problem in a decentralized system when we formulate the scheduling algorithm later in this Chapter.

4.3 Multiple Transmissions per Time Slot

In this section we consider a system where more than one subsystem may transmit per time step. We will consider the number of transmissions to be limited to L where $L \leq N$. The goal will be to devise a scheduling policy that efficiently picks the number of plants to transmit.

Let $W'(x)$ be the corresponding W matrix (as given in (3.1)) of the system. Recall that in the case of a single subsystem transmitting per time-step, $\triangle W'(x)$ in Equation (3.1) will have only one non-zero column corresponding to the subsystem that is transmitted. When multiple subsystems are transmitted, there will be multiple non-zero columns in $W'(x)$.

Let $\triangle W_z$ be the $\triangle W'(x)$ for the case where only one subsystem z will be transmitting in the time slot. Then the matrix $\triangle W_z$ has only one non-zero column at column z corresponding to the decrease due to transmission of subsystem z's state. The $\triangle W'(x)$ can now be written as a sum of $\triangle W_z$ corresponding to each subsystem z that is transmitted. Therefore, the increase in the decrease of the Lyapunov function per time step due to L transmissions by subsystems $z \in \{1, 2, ..., L\}$ can be rewritten as

$$-m^T(\triangle W'_{ij}(x))w(x) = -m^T(\triangle W'_1(x) + ... + \triangle W'_L(x))w(x)$$

which can be simplified to

$$= -m^T(\triangle W_1'(x))w(x) + ... + -m^T(\triangle W_L'(x))w(x)\ . \tag{4.1}$$

What the above result indicates is that to find the maximum attainable decrease in the Lyapunov function due to multiple transmissions, one only needs to consider the decrease by transmitting each subsystem state. So after evaluating the possible increase at each time step, the N subsystems that give the maximum increase in the rate of decrease of the Lyapunov function can be selected for transmission.

The scheduling policy for an interconnected system with L transmissions per time-step can therefore be stated as follows:-

1. At each time step, the controller evaluates P_j for $P_j = -\sum_{i=1}^{N} m_i e'_{ij}\xi_{ij}\|x_j\|$, where $j \in \{1, 2, ..N\}$
2. The L subsystems with the smallest P_j (largest negative P_j) will transmit for that time step.

For a decentralized system with L transmissions, the work of the controller in the above scheduling policy will simply be done by each of the subsystems just as in the case for $L = 1$ discussed in the previous section. But for the step 2, a mechanism is needed that allows all of the L subsystems to know that they may transmit in a given time slot. We now turn our attention to proposing a solution to this problem.

4.4 Decentralized Scheduling Algorithm

In the previous sections, it was shown that in a decentralized system, the computations for determining the impact of transmitting one's own state can be computed by that subsystem itself. But we still require a self-determining mechanism that allows the optimal subsystem to transmit its state based on its computations. Since there is no centralized scheduler that would have access to the possible impact that transmitting each subsystem state would have, each subsystem has to determine by itself whether to transmit at a given time slot.

Previous work on decentralized systems do propose solutions to this problem. A work by Tang [28] discusses the possibility of spreading information by flooding the network with the information that is required to be transmitted. Such a system is pointed out as infeasible since it requires a lot of energy to transmit all information to every part of the network, requires the subsystems to be aware of the system topology, and is prone to a single point failure. The same work goes on to discuss a "Controlled Hopwise Averaging (CHA)" scheme as an alternative. The fundamental idea of this scheme is to use a function that maps the decrease in the Lyapunov function to the next time to transmit. So each subsystem can use its estimated decrease to determine when it should transmit next. For this scheme to work, it is assumed that there is no propagation delay in the system and no errors in transmission. The scheme is also asynchronous.

This makes the CHA scheme ideal for our current application as well. We will assume that each subsystem has an internal timer that can be used to keep track of time. The original centralized algorithm given in Algorithm 3.1 can now be rewritten as follows for a decentralized system with L transmissions per time slot.

Algorithm 4.1 *The Precomputations:-*

1. *Compute the* ξ_{ij} *terms using* $\xi_{ij} = \|A_{ij}\|_2$.
2. *Compute the robustness bounds* $\xi_v(G_i)$ *using Equation 1.8 for each plant* i.
3. *Compute the* W*-matrix as given in 2.1.*
4. *Check if* W *is an* M*-matrix. If so, the system is connectively stable. Find a positive row vector* d *such that* dW *is a positive vector.*
5. *Calculate the maximum possible reduction in coupling* e'_{ij} *using Equation 3.5.*

The scheduling policy:-

1. *At each time step, and for each subsystem* j *that is a candidate for transmission, each subsystem* j *evaluates* $P_j = -\sum_{i=1}^{N} m_i e'_{ij} \xi_{ij} \|x_j\|$.
2. *All subsystems then find the next time to transmit using* $f(P_j)$. *The function* f *maps the value of* P_j *(the rate of decrease in the Lyapunov Function) to a time delay after which the subsystem should transmit.*
3. *Each subsystem waits for the period of time given by the function* f *and then transmits its own state.*

4. If a subsystem receives L transmissions before transmitting its own state, then it no longer waits to transmit it's own state within that time slot. It will then use the received state information in its input and moves back to step 1 *in the scheduling policy.*

5. If a subsystem receives $L-1$ transmissions after transmitting its own state, then it will use the received state information in its input and moves back to step 1 *in the scheduling policy.*

The function f in this scenario will have to be defined such that the probability of two systems having the same 'next time to transmit' is unlikely. It would also need to take into account the granularity of the unit of time perceived by each subsystem. The choice of f will also depend on the dynamics of each subsystem and the interactions within the subsystem. If the subsystems are similar in their dynamics and have similar coupling interactions between subsystems, the probability of obtaining the same time delay is higher. Similarly, if the dynamics of the subsystems are different or the coupling strength between subsystems are different, there would be a greater freedom in choosing f. Therefore, we did not investigate the possible functions that may be used for f since it is closely dependent on the system and specific application. The problem is also a mapping problem which is outside the scope of the scheduling problem discussed in this Thesis.

4.5 Transformed System

It would be interesting to consider at this point the question of how well the scheduling policy performs on a system that has been transformed using a similarity transformation. A previous work by Siljak [19] tells us that the estimate of the robustness bound of a subsystem is exact when the system matrix A_{ii} is of diagonal form. Thus, if we can reduce the subsystems given in Equation 1.1 to have a diagonal system matrix, then hypothetically, the corresponding robustness bound estimate will be more exact and hints toward a possible performance improvement of the scheduling policy.

However, the situation is complicated by the fact that, although the robustness bound is now exact, the other terms in the W-matrix given in Equation 1.3 have changed (due to the transformation of the coupling matrices).

The analysis that follows is an attempt to study the possible effect that transforming the system might have on the effectiveness of the scheduling policy by using simulations. For simplification of the analysis, we will only consider a system containing homogeneous subsystems. It must also be noted that this is a simulation based analysis rather than a rigorous mathematical analysis. The reason for this form of analysis is due to the fact that the effects on the coupling matrices due to diagonalizing the system matrix using a similarity transformation is not straightforward when there are no direct restrictions on the matrices involved.

4.5.1 Diagonalization of the System Matrix

We will consider a system where $A_{ii} = A_{jj}\ \forall i, j \in \{1, .., N\}$ and assume the case where A_{ii} is diagonalizable as $P_i^{-1}\Lambda_{ii}P_i$ where P_i is the matrix of eigenvectors of A_{ii} and Λ_{ii} is a diagonal matrix containing the eigenvalues of A_{ii}. We then diagonalize the system given by 1.1 using a similarity transformation as follows

$$P_i^{-1}x_i(k+1) = \Lambda_{ii}P_i^{-1}x_i(k) + P_i^{-1}B_{ii}P_iP_i^{-1}u_i(k) + \sum_{j=1}^{N} e_{ij}P_i^{-1}A_{ij}P_iP_i^{-1}x_j(k)\ . \tag{4.2}$$

We may now denote $P_i^{-1}x_i[k]$ by $\bar{x}_i[k]$, thus simplifying the above equation as

$$\bar{x}_i(k+1) = \Lambda_{ii}\bar{x}_i(k) + P_i^{-1}B_{ii}P_i\bar{u}_i(k) + \sum_{j=1}^{N} e_{ij}P_i^{-1}A_{ij}P_i\bar{x}_j(k) \tag{4.3}$$

The following result follows immediately from the definition of connective stability

Lemma 4.1 *If a system is connectively stable, the system obtained by the similarity transformation of each of its subsystems is also connectively stable*

We will now study the above system using simulations.

4.5.2 Simulation Results for Transformed System

Figures 4.1 and 4.2 below show the results of simulating a scheduling policy based on a given system and its transformed system.

It must be noted that in the simulation for the transformed system, the transformed system was run in parallel with a version of the regular system. However, the regular system was scheduled by using the scheduling policy on the data from the transformed system. Both the transformed system and its corresponding version of the regular system were updated using the state transmission that was determined for the transformed system. Another separate version of the regular system was also run in parallel for comparison of performance.

The reason for such a procedure is that it gives us the ability to compare the performance effect by using the transformed system in the scheduling policy by measuring its state size. In this arrangement, the state can be estimated using the same H-norm for both systems. While the transformed system itself would have a different H-norm making the comparison meaningless, the regular version run in parallel will have the same H-norm.

Figure 4.1 gives us an example of where the scheduling policy based on the transformed system leads to worse performance compared to the scheduling policy based on the regular system. Figure 4.2 gives the difference in performance as a percentage of the regular system calculated as

$$\frac{\text{Regular system's state size} - \text{Transformed system's state size}}{\text{Transformed system's state size}} \times 100\% \tag{4.4}$$

From the simulations, we can conclude that running the scheduling policy based on a transformed system does not always lead to better performance since the algorithm seems to perform better on the untransformed system compared to the transformed system. Therefore we can conclude that in general, it is not guaranteed that we will have a performance improvement by transforming the system to a diagonal form.

4.6 Summary

In this section, we have reformulated the centralized scheduling policy for a decentralized system with multiple communication resources. We have also studied the possibility of improving the performance of the scheduling policy by using a transformed system for scheduling and shown that such a transformation does not guarantee an improvement in performance.

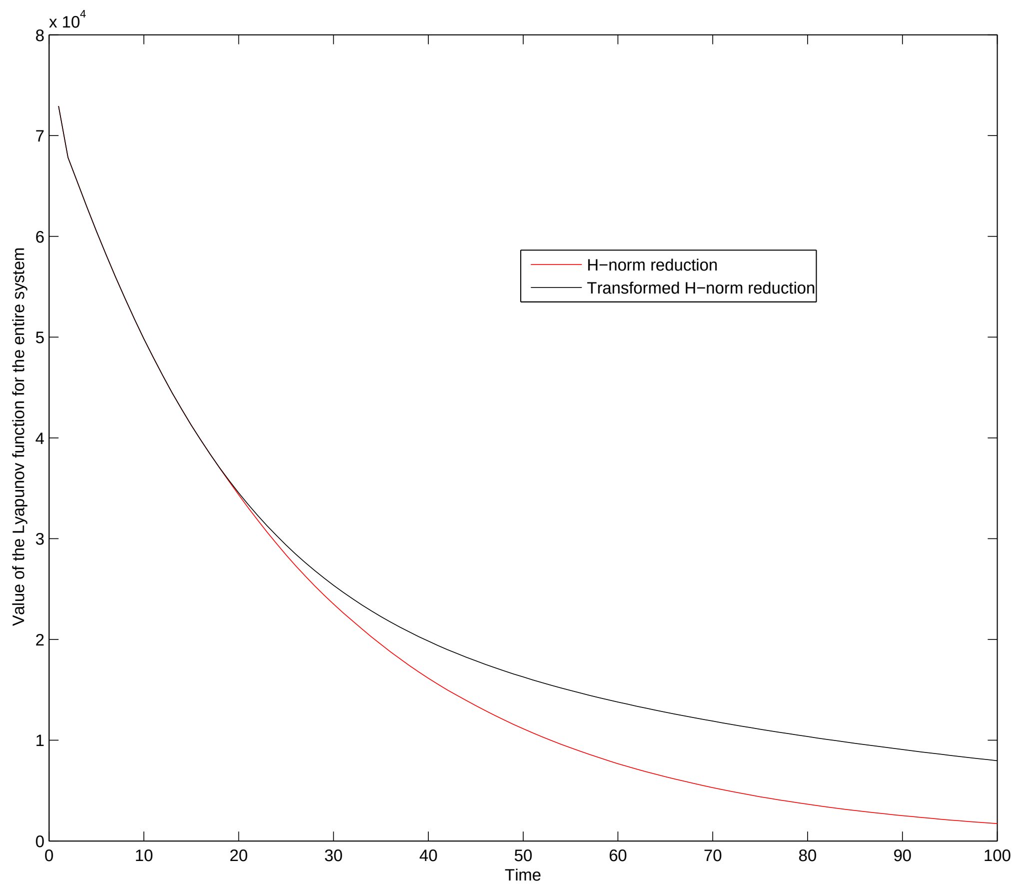

Figure 4.1: Comparison of state evolution

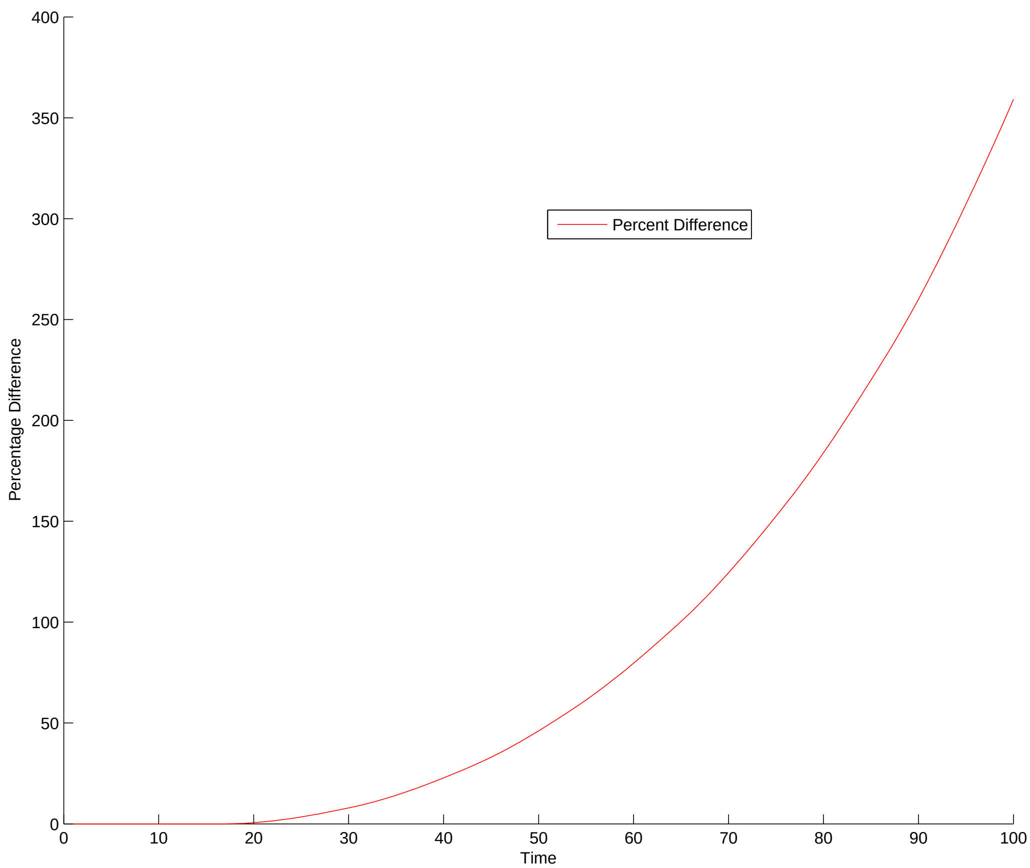

Figure 4.2: Performance Difference

Chapter 5

Conclusions

In this thesis, we have considered the problem of formulating a reliable scheduling policy for classes of connected networks. The connective stability of a system guarantees that the system will remain stable independent of the variations in effects due to interconnections. This makes the proposed scheduling scheme reliable (with respect to maintaining stability) since it operates by varying the interconnection strengths on connected systems. We began by extending the previous results on connected systems to common network topologies in order to obtain more simplified tests for connective stability of such interconnected systems. These tests would then be the schedulability tests for the proposed scheduling policy. Then using insights from this analysis and existing results, we proceeded to construct a centralized scheduling policy for interconnected systems that were connectively stable. The scheduling policy was then modified for a decentralized system and to support multiple transmissions per time step. We also delved into the possible effects that transforming the system using a similarity transformation might have on the performance of the proposed scheduling algorithm. It was shown that applying the proposed scheduling algorithm to a transformed system does not necessarily lead to an increase in performance and may cause it to perform worse compared to an untransformed system.

For future research, one possible direction would be to extend the work on the possible mapping functions that may be used for the decentralized algorithm. Such a function is tightly dependent on the properties of the application and the dynamics of the system and will therefore require careful study of specific systems in order to recommend possible mapping functions.

APPENDIX

The data used for simulating the interconnected network for comparison between various scheduling policies.

$$A = \begin{pmatrix} 0.37 & 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.55 & 0.14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.54 & 0.13 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.28 & 0.46 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.05 & 0.01 & 0.08 & 0.23 & 0.25 & 0 & 0 & 0 & 0 \\ 0.10 & 0.09 & 0.07 & 0.02 & 0.76 & 0.63 & 0 & 0 & 0 & 0 \\ 0.03 & 0.07 & 0.08 & 0.09 & 0.08 & 0.07 & 0.59 & 0.40 & 0.09 & 0.03 \\ 0.04 & 0.05 & 0.07 & 0.03 & 0.04 & 0.07 & 0.61 & 0.12 & 0.08 & 0.04 \\ 0 & 0 & 0.08 & 0.00 & 0 & 0 & 0.03 & 0.00 & 0.69 & 0.67 \\ 0 & 0 & 0.01 & 0.03 & 0 & 0 & 0.02 & 0.01 & 0.22 & 0.24 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.45 & 0.33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.54 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.38 & 0.96 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.02 & 0.27 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.09 & 0.67 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.32 & 0.36 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.19 & 0.44 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.67 & 0.65 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.92 & 0.52 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.84 & 0.86 \end{pmatrix}$$

$$X_0 = \begin{pmatrix} 50 \\ 50 \\ 50 \\ 50 \\ 50 \\ 50 \\ 50 \\ 50 \\ 50 \\ 50 \end{pmatrix}$$

$$\text{Adjacency Matrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Note: A, B are the system matrices for the interconnected system and X_0 is the initial state of the system. The adjacency matrix shows the topology of the system.

References

[1] J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883 – 1887, Nov 2002.

[2] M. E. M. B. Gaid, A. Cela, and Y. Hamam. Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system. *IEEE Transactions on Control Systems Technology*, 14(4):776 – 787, Jul 2006.

[3] V. Gupta. On a control algorithm for time-varying processor availability. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, HSCC '10, pages 81–90, New York, NY, USA, 2010. ACM.

[4] C. N. Hadjicostis and R.Touri. Feedback control utilizing packet dropping network links. *41st IEEE Conference on Decision and Control*, pages 1205–1210, 2002.

[5] L. Hetel, J. Daafouz, and C. Iung. Equivalence between the Lyapunov Krasovskii functionals approach for discrete delay systems and that of the stability conditions for switched systems. *Nonlinear Analysis: Hybrid Systems*, 2(3):697 – 705, 2008.

[6] H. Huang, W. Qing, and Y. Di. Connective stability of distributed interconnected systems. In *2010 29th Chinese Control Conference (CCC)*, pages 1016 –1020, July 2010.

[7] O. C. Imer, S. Yuksel, and T. Basar. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9):1429–1439, 2006.

[8] Z. Jiang, E. Sontag, and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857 – 869, 2001.

[9] H. Lin and P. J. Antsaklis. Persistent disturbance attenuation properties for networked control systems. In *43rd IEEE Conference on Decision and Control*, volume 1, pages 953 –958, Dec. 2004.

[10] P Marti. Analysis and design of realtime scheduling in control systems. *Seminari de Treball en Automatica Robotica i Percepcio (STAR99)*, pages 135–141, Oct 1999.

[11] A. Michel, R. Miller, and W. Tang. Lyapunov stability of interconnected systems: Decomposition into strongly connected subsystems. *IEEE Transactions on Circuits and Systems*, 25(9):799 – 809, Sept 1978.

[12] P. Naghshtabrizi and J. P. Hespanha. Analysis of distributed control systems with shared communication and computation resources. In *American Control Conference*, pages 3384 –3389, Jun 2009.

[13] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Some topics in real-time control. In *Proceedings of the 1998 American Control Conference*, volume 4, pages 2386 –2390, Jun 1998.

[14] M. Pajic, S. Sundaram, G.J. Pappas, and R. Mangharam. Topological conditions for wireless control networks. *IEEE Conference on Decision and Control and European Control Conference*, pages 2353–2360, 2011.

[15] M. Pajic, S. Sundaram, G.J. Pappas, and R. Mangharam. The wireless control network: A new approach for control over networks. *IEEE Transactions on Automatic Control*, 56(10):2305–2318, 2011.

[16] D. E. Quevedo and V. Gupta. Stability of sequence-based anytime control with Markovian processor availability. In *Australian Control Conference (AUCC), 2011*, pages 56 –61, Nov. 2011.

[17] C. L. Robinson and P. R. Kumar. Optimizing controller location in networked control systems with packet drops. *IEEE Journal on Selected Areas in Communications*, 26(4):661–671, 2008.

[18] L. Schenato, B. Sinopoli, B. Sinopoli, K. Poolla, and S. S. Sastry. Foundations of control and estimation over lossy networks. *Proc. of the IEEE*, 95(1):163–187, 2007.

[19] M. E. Sezer and D. D. Siljak. Robust stability of discrete systems. *International Journal of Control*, 48(5):2055–2063, 1988.

[20] M. E. Sezer and D. D. Siljak. A note on robust stability bounds. *IEEE Transactions on Automatic Control*, 34(11):1212 –1215, Nov 1989.

[21] M. E. Sezer and D. D. Siljak. On stability of interval matrices. In *American Control Conference*, pages 3108 –3111, Jun 1993.

[22] D. D. Siljak. Stability of large-scale systems under structural perturbations. *IEEE Transactions on Systems, Man and Cybernetics*, 2(5):657 –663, Nov. 1972.

[23] D. D. Siljak. When is a complex ecosystem stable? *Mathematical Biosciences*, 25(1-2):25 – 50, 1975.

[24] D. D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, 1250 Sixth Avenue, San Diego, CA 92101, 1991.

[25] D. D. Siljak. Decentralized control and computations: status and prospects. *Annual Reviews in Control*, 20:131 – 141, 1996.

[26] D.D. Siljak. Connective stability of competitive equilibrium. *Automatica*, 11(4):389 – 400, 1975.

[27] D. M. Stipanovic and D. D. Siljak. Robust stability and stabilization of discrete-time non-linear systems: The LMI approach. *International Journal of Control*, 74(9):873–879, 2001.

[28] C. Y. Tang and J. Lu. Controlled hopwise averaging: Bandwidth/energy-efficient asynchronous distributed averaging for wireless networks. In *American Control Conference*, pages 1561 –1568, Jun 2009.

[29] J. Yepez, P. Marti, J. Ayza, and J. M. Fuertes. LEF closed-loop scheduling policy for real-time control systems. In *10th IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 4–280, Sept. 2005.

[30] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel. Stability analysis of switched systems with stable and unstable subsystems: an average dwell time approach. In *Proceedings of the American Control Conference*, volume 1, pages 200 –204, Sep 2000.

[31] W. Zhang. Stabilization of networked control systems over a sharing link using ALOHA. In *42nd IEEE Conference on Decision and Control*, volume 1, pages 204 – 209, Dec. 2003.

[32] W. Zhang and L. Yu. Output feedback stabilization of networked control systems with packet dropouts. *IEEE Transactions on Automatic Control*, 52(9):1705 –1710, Sept. 2007.