QoC AND QoS BARGAINING FOR MESSAGE SCHEDULING IN NETWORKED CONTROL SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SİNAN ŞENOL

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2012

Approval of the thesis:

QoC AND QoS BARGAINING FOR MESSAGE SCHEDULING IN NETWORKED CONTROL SYSTEMS

submitted by **SİNAN ŞENOL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical-Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical-Electronics Engineering**

Prof. Dr. Kemal Leblebicioğlu
Supervisor, **Electrical-Electronics Eng. Dept., METU**

Assoc. Prof. Dr. Ece G. Schmidt
Co-Supervisor, **Electrical-Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Semih Bilgen
Electrical-Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu
Electrical-Electronics Engineering Dept., METU

Prof. Dr. Ömer Morgül
Electrical-Electronics Engineering Dept., Bilkent University

Assoc. Prof. Dr. Halit Oğuztüzün
Computer Engineering Dept., METU

Assoc. Prof. Dr. Klaus Werner Schmidt
Mechatronics Engineering Dept., Çankaya University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:

Signature :

ABSTRACT

QoC AND QoS BARGAINING FOR MESSAGE SCHEDULING IN NETWORKED CONTROL SYSTEMS

Şenol, Sinan
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Kemal Leblebicioğlu
Co-Supervisor : Assoc. Prof. Dr. Ece G. Schmidt

June 2012, 144 Pages

Networked Control Systems (NCS) are distributed control systems where the sensor signals to the controllers and the control data to the actuators are enclosed in messages and sent over a communication network. On the one hand, the design of an NCS requires ensuring the stability of the control system and achieving system response that is as close as possible to that of an ideal system which demands network resources. On the other hand, these resources are limited and have to be allocated efficiently to accommodate for future system extensions as well as applications other than control purpose. Furthermore the NCS design parameters for the control system messages and the message transmission over the network are interdependent. In this thesis, we propose "Integrated NCS Design (INtERCEDE: Integrated NEtwoRked Control systEm DEsign)" a novel *algorithmi*c approach for the design of NCS which ensures the stability of the control system, brings system response to that of an ideal system as close as desired and conserves network bandwidth at the same time. The core of INtERCEDE is a bargaining game approach which iteratively calculates the message parameters and network service parameters. Our experimental results demonstrate the operation of INtERCEDE and how it computes the optimal design parameters for the example NCS.

Keywords: Bargaining, Message Scheduling, Networked Control, Real-time System

ÖZ

AĞ TABANLI KONTROL SİSTEMLERİNDE MESAJ ZAMANLAMA İÇİN KK VE SK PAZARLIĞI

Şenol, Sinan
Doktora, Elektrik-Elektronik Mühendisliği
Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu
Ortak Tez Yöneticisi : Doç. Dr. Ece G. Schmidt

Haziran 2012, 144 Sayfa

Ağ Tabanlı Kontrol Sistemleri (ATKS), kontrolcülere gidecek algılayıcı sinyallerinin ve eyleyicilere gidecek kontrol değerlerinin mesajlar içine yerleştirilerek bir iletişim ağı üzerinden gönderildiği dağıtık kontrol sistemleridir. Bir taraftan, ATKS'nin tasarımı kontrol sisteminin kararlılığının sağlanmasını ve sistem tepkisinin ağ kaynaklarına ihtiyaç duyan ideal sisteme olabildiğince yakın olmasını gerektirir, diğer bir taraftan, bu kaynaklar sınırlıdır ve gelecek sistem eklentilerine olduğu kadar kontrol amacı dışındaki uygulamalara verimli bir şekilde ayrılmalıdır. Kontrol sistem mesajları için ATKS tasarım parametreleri ve ağ üzerinden mesaj iletimi birbirine bağımlıdır. Bu tezde, kontrol sisteminin kararlılığını sağlayan, sistem tepkisini ideal sisteme olabildiğince yaklaştıran ve aynı zamanda ağ kaynaklandıran tasarruf eden "Bütünleşik ATKS Tasarımı (INtERCEDE: Integrated NEtworked Control systEm DEsign)" adını verdiğimiz özgün bir algoritmik yaklaşım önerilmektedir. INtERCEDE'in merkezinde, mesaj parametrelerini ve ağ servis parametrelerini iteratif olarak hesaplayan pazarlık oyunu yaklaşımı bulunmaktadır. Deneysel sonuçlarımız, INtERCEDE'in çalışmasını ve örnek ATKS için optimal tasarım parametrelerini nasıl hesapladığını göstermektedir.

Anahtar Kelimeler: Pazarlık, Mesaj Çizelgeleme, Ağ Tabanlı Kontrol, Gerçek Zamanlı Sistem

ACKNOWLEDGEMENTS

The author wishes to express his best gratitude to his supervisor, Prof. Dr. Kemal Leblebicioğlu and co-supervisor, Assoc. Prof. Dr. Ece G. Schmidt for their infinite help and guidance throughout the thesis study.

The author also wishes to express great thanks to thesis jury members, Prof. Dr. Semih Bilgen, Prof. Dr. Ömer Morgül, Assoc. Prof. Dr. Halit Oğuztüzün and Assoc. Prof.. Dr. Klaus Schmidt for their helpful comments and guidance throughout the thesis study.

The author wishes to thank Aselsan Inc. for the support of his Ph.D. education.

Finally, the author wishes to thank his family members for their sharing of countless experiences during the education.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

ABBREVIATIONS

ACK	Acknowledgement
CAN	Controller Area Network
EDF	Earliest Deadline First
FIFO	First-In-First-Out
FlexRay	FlexRay Communication System
INtERCEDE	Integrated NEtwoRked Control systEm Design
LQC	Linear Quadratic Controller
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
MAC	Media Access Layer
MADB	Maximum Allowable Delay Bound
MATI	Maximum Allowable Transfer Interval
MIQP	Mixed Integer Quadratic Programming
MLD	Mixed Logical Dynamical
MODBUS	Modicon Bus
MPC	Model Predictive Controller
MSD	Minimum Slot Distance
NCS	Networked Control System
NWCOD	Negotiated Worst Case Overall Delay
PROFIBUS	Process Field Bus
RM	Rate Monotonic
QoC	Quality-of-Control

QoS	Quality-of-Service
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TOD	Try-Once-Discard
TTCAN	Time Triggered CAN
TTP/C	Time Triggered Protocol
UDP	User Datagram Protocol
ZOH	Zero Order Hold

CHAPTER 1

INTRODUCTION

The modern control systems are changing to be more complex and comprehensive. The integration of computing, communication and control into different levels of machine/factory operations and information processes results in this trend. The control applications are designed and implemented in a distributed fashion where the sensors, actuators, and controllers are at different physical locations. Different from classical implementations where these components exist on the same physical device and can share information with direct connections, a distributed implementation necessitates a *communication network* to transmit sensor data (observed plant states) to controllers and control data to the actuators. According to the above knowledge, a control system where the feedback loop is closed over a communication network is called a *Networked Control System* (NCS) [1], [2]. Examples for NCS include military vehicles, nuclear reactors, manufacturing systems and vehicle electronics [3], [4].

In NCS the sensors, actuators and controllers represent *nodes* that are sending *messages* to each other [1]. The control applications that run on these nodes are often safety control systems that are time-critical systems such that violating the operational requirements might lead to loss of human life and property. Hence, a family of special Industrial Communication Protocols that are also called

Fieldbus Protocols is developed to meet these requirements [5], [6]. Industrial communication protocols are mostly of shared-medium type that operates on medium access layer where the nodes get access to the network through priority-based access, event-triggered arbitration or through time division multiple access (TDMA) mechanisms. The most well-known event-triggered (event-based) standard is the Controller Area Network (CAN) [7] [8], [9], [10] while TTCAN [11], FlexRay [12], TTP/C [7], [13] are prominent time-triggered (TDMA) standards. Time-triggered networks are found to be a better suited architecture for NCS as it is deterministic in temporal performance, that is, satisfies transmission deadlines better and supports the periodical operation of distributed control systems [14]. There are also hybrid structures (CAN/ATM) that integrate more than one technology as in [15].

When the operational problems are considered, multiple nodes share the communication network to transmit their messages. So, the messages are transmitted with *delays* that occur because of the infrastructure and affect the performance of the control system. In particular, the time that elapses from the generation of a control system signal until its reception by the destination node should be bounded such that the control system is stable. Moreover, the response of the NCS should be as close as possible to a classical implementation in performance without any adverse effects of the network. In the networked control system design, message properties such as message generation *frequency* and the maximum *message delay* have to be formed to specify these performance requirements.

Also, the network has to meet these requirements by providing guaranteed services. The access of the nodes to the shared medium in a deterministic way is provided by conforming to a pre-computed *schedule* that the messages are transmitted according to for both for time-triggered and event-triggered industrial communication networks. For event-triggered networks, the schedule determines the priority for the messages and time-triggered networks the schedule determines time-slot assignment for the messages. The message schedule has the purpose to effectively allocate the network bandwidth as the main resource to the messages. Bandwidth is an important resource for industrial communication networks that determines the main aspects of the NCS design. Therefore, the upgrade and replacement of network components and the network interface in fieldbus networks and industrial environments are not straightforward tasks different from home and office environments. The new nodes and new messages that are added to the network to extend the control system consume more bandwidth. Furthermore, new applications such as multimedia data transfer (real-time video, audio, etc.) and remote factory management [16], [17] that are different from control system operation principles are continuously introduced to NCS. These applications often require higher data rates to transmit data over the network shared by control nodes. Therefore, the schedule has to be designed to allocate bandwidth to the messages efficiently and to meet their performance requirements.

To this end, designing an NCS involves two parts. The first part is based on considering the performance of the control system and determining message parameters for the control nodes. The second part is deciding the amount of bandwidth allocation and the remaining bandwidth and computing the message transmission schedule according to the message parameters. *"Here it must be noted that the design objectives for these two components trade of each other while the design parameters depend on each other" [18]*. On the one hand, assuring that control system performance is better by determining message parameters demands more bandwidth, but the bandwidth is physically constrained. On the other hand, conserving network resources by designing a message schedule that spares bandwidth gives rise to longer delays and even unstable system behavior.

A large amount of previous work in the literature focuses only on one component leading to designs with infeasible or inefficient implementations. There are studies that propose NCS design with both message parameters and schedule computation. However, due to the complexity of the problem and the interdependency of the parameters mentioned above, the proposed approaches are based on unrealistic assumptions to be able to achieve results or require infeasible amount of computation [19], [20], [21], [22], [23].

In this study, we propose "Integrated NCS Design (INtERCEDE: Integrated NEtwoRked Control systEm DEsign)" a novel *algorithmi*c approach for the design of NCS that ensures the stability of the control system, achieves optimal performance for the control system and conserves network bandwidth. INtERCEDE is constructed for shared medium, time-triggered networks where the schedule allocates time slots to messages.

The control system parameters and the physical bandwidth of the network are the inputs to INtERCEDE. The outputs are the message parameters and the corresponding message schedule. We resolve the dependency of the design parameters and the contradiction between improving control system performance and conserving network bandwidth by a *bargaining game* approach. The message parameters and the schedule are iteratively evaluated according to Quality of Control (QoC) and Quality of Service (QoS) metrics and updated such that the design converges to an optimal point. QoC measures the difference between the response of the networked control system and an ideal control system that is not affected by the delays of the network. QoS measures the difference between the bandwidth allocated to the control system and the desired amount of allocation such that there is a certain amount conserved.

The relatively low computational complexity of the bargaining game approach is its first advantage. In addition it enables the parameterization of the NCS design such that designers can emphasize the performance of the control system or the conservation of network bandwidth according to the application and the available network architecture. INtERCEDE does not involve designing the control system which makes it applicable to any given NCS. However, it takes the performance of the control system into consideration with the QoC metric. The output message schedule can be directly applied to most of the time-slotted protocols such as TTP/C or the static segment of FlexRay hence INtERCEDE generates realistic results.

We investigate INtERCEDE viability through simulation studies: We investigate 4 control systems with different properties and resource requirements. Our results show that INtERCEDE can successfully design message parameters and schedule to achieve optimal QoC and QoS values within feasible computation times.

The thesis is organized as follows. In Chapter 2, we present our assumptions, models and definitions for the control system parameters, message parameters, and network service parameters. We discuss the effects of the networked implementation on the control system behaviour and define the stability condition for an NCS. We also introduce our QoC and QoS performance metrics. Chapter 2 concludes with a comparative study of previous work on NCS design. Chapter 3 summarizes the survey about the controller design methodologies for Networked Control Systems. In a similar way, Chapter 4 summarizes the survey about the scheduling methods for Networked Control

Systems. The proposed NCS design method utilizes a stability test for deriving the controller stability bounds of the physical systems. Chapter 5 explains the derivation of the stability test for the proposed method. Chapter 6 describes the mathematical modeling of QoC and QoS definitions. Chapter 7 is dedicated to the description of INtERCEDE along with the bargaining game approach. We also present the novel features of our algorithm with respect to previous work. Chapter 8 presents our simulation study to demonstrate the operation and the results of INtERCEDE. Our conclusions are presented in Chapter 9. We present our two additional algorithms that compute the amount of network service achieved from a given schedule in Appendix A. The details of the bargaining game are presented in Appendix B. The additional simulation results that demonstrate features of the bargaining game are provided in Appendix C.

This thesis work is partially presented in [18].

CHAPTER 2

PRELIMINARIES AND FORMAL PROBLEM DEFINITION

2.1. Networked Control System (NCS)

In this work we define a *control system* as a *plant* that is controlled by *a controller*. The controller computes the required control values according to the state information extracted from the measurements of the *sensor* and sends the control values to the *actuator* which applies actions extracted from these values to the plant. In this work we consider *distributed control applications* where controller, sensor and actuator devices are implemented in different physical locations and the feedback loop of the control system is closed over a *shared medium communication network.* Multiple control systems can communicate over the same network as seen in Figure 1.

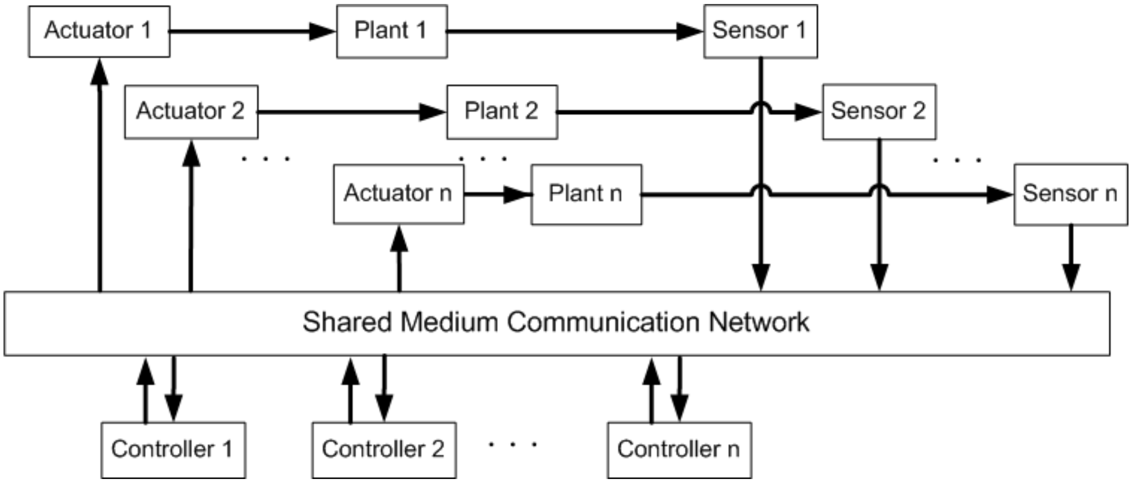

Figure 1. Multiple control systems in a distributed control application.

For this purpose, *Networked Control System (NCS)* is defined as a set of nodes which communicate over a shared medium network. The controllers, sensors and actuators on NCS are denoted as *control nodes* (CN). In addition, we assume that there are other nodes that communicate on the network which carry out tasks different than control tasks such as monitoring and maintenance, or transmit large amounts of data for multimedia applications.

The rest of this section presents our models, definitions and assumptions for the control system and the network. When the control system is implemented in a distributed fashion, the information can be shared by the components freely but has to be transmitted over a network. The impact of this networked operation on the control system is also discussed at the end of the Section.

2.2. Description of the Control System

We assume linear time-invariant plants:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

where $x(t)$ is n-by-1 state vector at time t, $u(t)$ is input at time t, $y(t)$ is output at time t, A is an n-by-n real matrix, B is an n-by-1 real matrix, C is an 1-by-n real matrix and D is a real constant. In order to simplify modeling, we assume that the states of the plant are directly measurable by the sensors. The sensor sample at the k^{th} sampling instance is as follows:

$$x_c[k] = x(kT)$$

where $x_c[k]$ is the state sample at time kT where T is the sampling period. $x_c[k]$ is transmitted to the controller enclosed in a s*ensor message* for the computation of the control value. The samples may not be synchronized to the communication network's slot time and frequency. We assume any controller can be used with a corresponding stability test in NCS design. For only the simulation studies, we model the controller as a discrete Linear Quadratic controller with static law:

$$u[k] = Kx_c[k]$$

where $u[k]$ is the control value at time instant kT and K is a 1-by-n real matrix. K is the linear static controller for the linear time invariant system.

The control value is encapsulated in *a control message* and sent to the actuator. The actuator extracts the control value from the message and reconstructs the control signal with a zero-order-hold (ZOH) function:

$$u(t) = u[k] \ , kT \le t < (k+1)T \ .$$

This one message sequence stated above completes one feedback loop transmission of one control system.

2.3. Description of the Network Service

The distributed control applications running on the NCS are mostly real-time applications that require signal transmission with certain delay and bandwidth guarantees. For this purpose, many time-triggered communication protocols such as TTCAN [11], FlexRay [12] [24], TTP/C [7] are developed. Time-triggered protocols are generally TDMA based and provide network access to the nodes based on synchronous *time-slots* which are exclusively assigned to nodes and their messages. Contrary to event-triggered protocols such as CAN [7], time-triggered protocols support deterministic message delays and periodic message transmission rather than just bounds.

In this work, we assume a shared medium time-slotted communication network in which message transmission is governed by a pre-computed *schedule* before the NCS starts to run [18]. In the rest of this section, we present our definitions that describe the operation of the NCS.

Definition 1: Message Schedule

Let the bandwidth of the network be C bps that is allocated to the control systems over N slots. Allocating $a(k)$ slots to control system k corresponds to allocating $\frac{a(k)}{N}C$ bps of network bandwidth.

To this end, $S_k = \{s_{k,1}, s_{k,2}, \ldots, s_{k,a(k)}\}$ where $S_k \subset \{1,2,\ldots,N\}$ is the *schedule* for control system k. Note that the schedules for different control systems do not share same slots and are mutually exclusive. Hence, $S_k \cap S_m = \varnothing$ for $k \neq m$. Once the schedules are computed they are repeated periodically every N slots. In this paper we only investigate the schedules for the control nodes where for other nodes schedules also have to be constructed using the remaining slots out of N.

Definition 2: Sampling frequency and Sample Losses

We have the following two assumptions related to message transmission. Firstly, only a single instance of any given message can be buffered. Hence, if the next instance of a message is created before its allocated time slot, it overwrites the previous value resulting in a *loss* for the previous value. Secondly, the controller responds with a control signal message immediately after sensor message in the same slot and time slots of the network are large enough to accommodate the single sensor reading and control value of a sensor node and the controller node, respectively. The adjacent slot sharing assumption of our model is valid when the practical situation of the controller technology is considered. Modern controller schemes are implemented on electronic chips which are fast enough to response immediately after receiving sensor message. According to these assumptions, L_i denotes the *maximum number of consecutive sample losses* under a certain schedule that is the achieved number of consecutive losses by the network, and $Loss_i$ denotes the *maximum number of consecutive sample losses* for the sampling frequency that is the requested number of consecutive losses by the control system. The sampling frequency for a sensor reading sample that is generated by control system i is denoted by $frequency_i$

Definition 3: Maximum Network Access Delay

We define the maximum time elapsed for a sample from the time it is written in the buffer until it is transmitted on the network as the maximum network access delay D_i for control system i. Note that $D_i < \frac{1}{frequency_i}$ such that the sample is not lost. After first L_i samples are overwritten in the buffer, the last sample gets transmitted after a maximum delay of D_i.

Definition 4: Minimum Effective Rate

The *minimum effective rate* for control system i is $Rate_i$ where the maximum time interval between the generation of any two successfully transmitted samples is 1/ $Rate_i$. Note that: $\frac{L_i + 1}{Rate_i} = \frac{Loss_i + 1}{frequency_i}$ (See Figure 2).

Definition 5: Maximum Network Delay

The *maximum network delay* for control system i is $Delay_i$ where:

$$Delay_i = \frac{L_i + 1}{Rate_i} + D_i$$

It is the total time for control system i to transmit a new sample successfully accounting for the maximum number of consecutive losses.

Note that our definition of the network delay conforms to the definition in [25] which is our reference for the stability criterion. Figure 2 demonstrates an example with $Loss_i$=2 where the lost samples are indicated by dashed arrows. Note that for this example $Rate_i$ is $frequency_i/3$ for L_i=0.

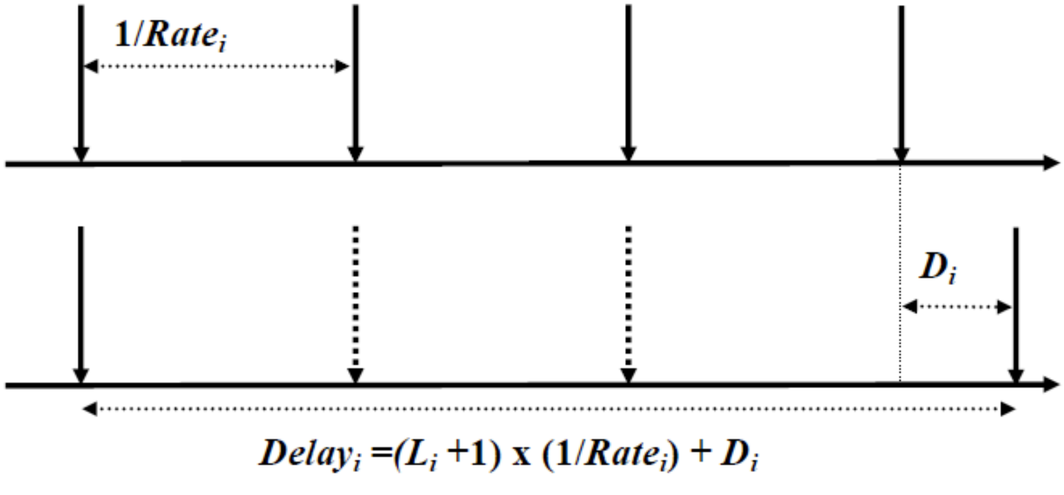

Figure 2. Example sample generation and transmission scenario for $Loss_i$ = 2.

Hence, for a given control system i with *frequency*$_i$ the service provided to the message by the network with respective message transmission schedule can be defined by *Delay*$_i$, L_i , and *Rate*$_i$.

2.4. Impact of Network Delays and Losses on the Control System

The analysis of the performance of the control system with the above assumptions is modified as follows considering the message delays and losses due to the communication over the network. The propagation and computation delays are fixed and small enough to ignore, hence, we only take the network access delay into consideration.

Let $\tau_{sc}(k)$ be network access delay for the k^{th} sensor-to-controller message. Then the state values received by the controller are affected as follows:

$$x_c[k] = x\big(kT - \tau_{sc}(k)\big) \quad .$$

The control value that reaches the actuator is:

$$u(t) = Kx\big(kT - \tau_{sc}(k)\big)$$

where $u(t)$ is the control value at the actuator at time instant *t*.

When the message loss is modeled, we assume that the controller does not compute and send new control values to the actuator when a sensor message is lost, simply, the actuator will use the previously received value. So, the value at the actuator will be:

$$u(t) = \begin{cases} u[k] \ , kT \le t < (k+1)T, \ if\ kth\ sensor\ message\ is\ not\ lost \\ \qquad u[k-1], \quad if\ kth\ sensor\ message\ is\ lost \end{cases}$$

2.5. Evaluating NCS Performance

In classical feedback control theory, the primary metrics for the closed-system response depends on transient-response, steady-state error and stability [26]. The system stability must be guaranteed by restricting the delay in the feedback loop with a delay bound. In order to asses the transient response and minimize steady-state error, we introduce the Quality of Control (QoC) metric to compare the response of the control system implemented as NCS with respect to a monolithic implementation without any adverse effects of the network namely message delay and loss. We also define a new metric Quality of Service (QoS) to measure the efficiency of network bandwidth allocation. We define QoS as a composite metric that consists of bandwidth, delay and packet loss rate parameters. For a given message schedule QoS measures the distance to the target service values for bandwidth and maximum network delay.

Our model and stability test in this paper conforms to the model in [25]. This model and stability test is chosen, because of the conformance of both the model and the test to the requirements of the problem. We provide a short survey of stability tests for NCS in Section 2.6.

If we consider a single control system that consists of a plant, a sensor, a controller and an actuator, the model can be simplified in representation. The physical system (plant) is assumed be linear and time-invariant with continuous dynamics. The control law is assumed static complying with the stability test of [25]. The assumed control system is given below:

$$\dot{x}(t) = A_C x(t) + B_C u(t)$$

$$y(t) = Cx(t) + Du(t)$$

$$x(t_0) = x_0$$

$$u(t^+) = Kx(t - \tau_k^{sc}),\ t \in \{kh + \tau_k^{sc}\},\ k = 1,2,\ldots$$

where, A_C, B_C, C and D are the physical system parameters. K is the static control law and disturbance effects are ignored. $x(t)$, $u(t)$ and x_0 are the state value, control value at time *t* and the initial state value at time t_0, respectively. τ_k^{sc} is the sensor-to-controller delay for the k^{th} sample and h is the sampling period. This model is expanded into the following model for the calculation of the control values which accounts for the message losses:

$u(t^+) = Kx(t - \tau_k)\,,\ i_k h + \tau_k \leq t < i_{k+1} h + \tau_{k+1},\ k = 1,2, \ldots$

where i_k is the index of the sensor sample which is successfully transmitted to the controller and the corresponding control value is transmitted to the actuator successfully, and

$\tau_k = \tau_k^{sc},\ k = 1,2, \ldots$

It is assumed that there is no synchronization between the sampler and the digital controller. So, the control value is no longer constant within a sampling period.

We chose to adopt the stability test in [25] because of its tight bounds close to the theoretical bounds compared to the other studies in the literature. We further discuss the stability test and the selection of our particular test in detail in Chapter 5. The assumptions for the stability test are as follows:

1) The sensor is clock-driven (periodic), the controller and actuator are event-driven.
2) The following inequalities hold:
   $(i_{k+1} - i_k)h + \tau_{k+1} \leq \eta,\ k = 1,2, \ldots.$
   $\tau_k \geq 0,\ k = 1,2, \ldots.$
   η is the worst case maximum delay in the feedback loop.
3) We ignore the disturbance effects.
4) We assume no parameter uncertainties.

Under these assumptions, the following matrix inequality should be satisfied if the system is stable:.

$$\boldsymbol{\Gamma_{11}} = \begin{bmatrix} \boldsymbol{T_1} & \boldsymbol{T_2} & \boldsymbol{R_1} & \boldsymbol{R_2} & \boldsymbol{0} \\ \boldsymbol{T_2}^{\boldsymbol{T}} & \boldsymbol{T_3} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{0} \\ \boldsymbol{R_1} & \boldsymbol{S} & \boldsymbol{T_4} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R_2} & \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{T_5} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\in \boldsymbol{I} \end{bmatrix} < \boldsymbol{0}$$

where

$T_1 = PA + {A_C}^T P + Q_1 + Q_2 - R_1 - R_2 + {A_C}^T H A_C$

$T_2 = PBK + {A_C}^T H B_C K$

$T_3 = (B_C K)^T H B_C K - 2S$

$T_4 = -Q_1 - R_1 - S$

$T_5 = -Q_2 - R_2 - S$

$H = \eta^2 (R_2 + S)$.

In the above expression, η denotes the *maximum allowable delay bound* (MADB) for the system to be stable. R_1, R_2, S, Q_1, Q_2, P are *n-by-n* real matrix variables, $\in > 0$ must be satisfied and A_C, B_C, K are the linear time-invariant plant parameters.

Definition 6: MADB for system *i*

η_i is *maximum allowable delay bound* (MADB) for system *i*. $Delay_i < \eta_i$ must be satisfied to ensure the stability where $Delay_i$ is the maximum network delay as in **Definition 5**.

Definition 7: Quality of Control (QoC)

Consider a control system i with sampling frequency f_i implemented as an NCS with a maximum network access delay of d_i in the feedback loop. The QoC for the control system i is defined as:$QoC(s(t)) = \frac{1}{\int_0^{\infty}|s(t)-d(t)|dt+\varepsilon} \times \frac{1}{RefQoC}$

$$QoC_i(f_i, d_i, l_i) = \frac{1}{\int_0^{\infty} \left| e_i(t, f_i, d_{total,i}) \right| dt + \varepsilon} \times RefQoC_i(f_i, \eta_i)$$

where,

$d_{total,i} = \frac{l_i + 1}{f_i} + d_i$, $\int_0^{\infty} \left| e_i(t, f_i, d_{total,i}) \right| dt$ (Integral of the Absolute Error) function at the denominator is the error between the ideal system response without any delay or loss and the response of the system that communicates over the network and $RefQoC_i(f_i, \eta_i) = \frac{1}{\int_0^{\infty} \left| e_i(t, f_i, 0.1\eta_i \right| dt + \varepsilon}$ is the reference QoC as suggested in [27] and normalizes the QoC value to 1 for any given system if the sampling period is 0.1 times the η_i. Here, ε is a small positive constant.

Our definition of Quality-of-Control (QoC) is similar to [26, 28] where we combine both transient response and steady state error. However, in those studies, the QoC is computed by considering best and worst cases of transmission sequences. Control systems are mostly real-time systems with strict performance requirements such as nuclear reactors, military vehicles, transportation vehicles, etc. hence, we always consider the worst case for computation. Note that this also reduces the amount of computation.

Definition 8: Quality of Service (QoS)

Let bw denote the desired bandwidth reservation for all control system messages. The desired maximum delay for control system i is $d_{\max,i}$ and $schedule = \bigcup_i S_i$.

The QoS for the NCS is defined as:

$$QoS(schedule, bw, d_{\max}) = \exp^{-(RateCost + MessageSetCost)}$$

where,

$$RateCost = \left| \frac{\sum_i Rate_i - bw}{bw} \right|^2$$

$$MessageSetCost = \sum_i \max\left(\frac{Delay_i - d_{\max,i}}{d_{\max,i}}, 0 \right).$$

$d_{\max}$ in $QoS(schedule, bw, d_{\max})$ represents the combined effect of $d_{\max,i}$ in $MessageSetCost$. $Rate_i$ and $Delay_i$ are the service parameters achieved for control system i through the slot assignment of the $schedule$.

2.6. Previous Work on NCS Design

In [29] a detailed survey about current situation of the NCS literature exists. There are different research problems integrated into the NCS research area related to various disciplines. Implementing the control system in a distributed way where the components exchange information over a communication network has impact on the performance of the control system, most importantly on its stability. Hence, a large number of studies explore the stability of NCSs. In [2], a related detailed survey may be found. The study in [30] is one of the pioneering works for the analysis of stability of NCS with delay. In [31], the time delay system modeling approach is examined for NCSs with details. [32] develops its analysis similar to [31], but the packet losses are not considered. In

[33], the delay bound model conforms to periodic sampling and constant delay, and the stability of the system is analyzed. In [34], the authors assume periodic sampling and variable delay, but small delays are assumed and no multiple packet losses is considered. In [35], the stability analysis in [34] is extended to multiple packet loss case. The method in [35] is developed based on the work in [36]. In [37], the stability for variable sampling and variable delay is considered, but the use of time-stamps is required for the correct working of the system. In [38], also the clocks of controller and sensor are assumed synchronized which is a restrictive assumption. In [39] the stability for a specific network protocol namely TCP is investigated. In [25] the packet losses are also considered, while stability for delay is investigated, so this test is also a candidate for our stability test. It is also shown in [25] that the stability test found is less conservative (more close to theoretical limit) than the previous works in [40], [41], [32], [42], [33]. [40], [32] and [42] Since our model conforms to periodic sampling and variable delay, and packet losses are possible, we choose to base our stability test in [25]. In [43], [44], [45], integrated controller and scheduling schemes are proposed. In [43] and [46] NCS controller schemes with scheduling for wireless networks are introduced, but the complexity of the methods is a disadvantage. In [47], a scheduling method for NCSs without considering control system performance is proposed. In [48] and [49], a scheduling mechanism providing stability is considered, but improving control performance is not considered. Authors of [50] consider the integrated controller and scheduling problem, but network assumptions are simple.

We classify the previous studies for the NCS design in three main classes. The first two classes are the studies that only consider either the performance of the control system or resource allocation of the network where as the third class studies consider both issues. We present a summary of these studies in Table 1.

Table 1. Classification of previous work in NCS design

Class	Ref	Approach	Further Comments
Control System Optimized, Network Performance is not considered	[26], [28], [41], [44], [46], [43],	Control system performance analysis dependent scheduling	The network is assumed to be dedicated only to the control systems.
	[51], [30], [31], [32], [33], [34], [35], [36], [37], [52], [39], [25], [40], [49]	Stability analysis based delay bound derivation	
Network optimized, Control System Performance is not considered	[53], [54], [55],	Bandwidth sharing	
	[56] [57] [58], [47]	Optimal schedule computation, schedulability analysis	Specific protocols for in-vehicle networks

Table 1. Classification of previous work in NCS design (cont'd).

Class	Ref	Approach	Further Comments
Integrated Control System and Network Resource Allocation	[59], [60]	QoS is defined a system design constraint and related to control system performance.	Network protocol is not considered. Communication network assumptions are unrealistic and oversimplifying.
	[19], [21], [22], [23], [50], [48], [45]	The controller of the control systems and message scheduler for the communication network are jointly designed	Not realistic, many simplifying assumptions on the operation of the communication network. Still computationally intensive.
	[20]	Both control system and communication network performance	Only used bandwidth is optimized. Real-time computation. High computational complexity
	[61]	Message scheduling method for multiple control systems	Distributes the available bandwidth of the communication network

Table 1. Classification of previous work in NCS design (cont'd).

Class	Ref	Approach	Further Comments
Integrated Control System and Network Resource Allocation	[62]	The controller of the control systems and message scheduler for the communication network are jointly designed	The network is assumed to be dedicated only to the control systems.
	[63]	Message scheduling method for a single control system	Priority-based access network
	[64]	Control system performance based communication protocol performance evaluation	Network protocol is not considered

CHAPTER 3

NCS CONTROL METHODOLOGIES SURVEY

This chapter serves as the introduction of the control systems in the NCSs. Their main problems arise with the network induced delay affecting control systems. Various control strategies exist to preserve the stability and improve control system error performance. The three groups of studies in Table 1 in Chapter 2 show that there are different approaches to controller and scheduler designs. While the integrated controller and scheduler approach is appealing at first sight, it is not really necessary to use this approach as this survey chapter will show. This survey chapter deals with only controller strategies to validate our predesigned controller strategy and establish the background for our message set and message scheduler design for any controller. There are so many different control methodologies and each one fits a different NCS structure. Some control strategies depend on delay compensation [65], some depend on controlling system motion [66], and some depend on fuzzy control strategies [67]. All the advantages and disadvantages of the existing control strategies are stated and the control strategies surveyed are a guide for developing and validating our model. Since our model assumes no specific control strategy, it must suit all the existing strategies to be general enough. The survey also shows that the integrated controller and scheduler approach is unrealistic as controller design approach may be changed in the future.

This survey for the NCSs is, mainly, carried out for the controller designs for the NCSs under communication constraints. Among these, there are various approaches such as different controller assumptions, different communication constraints assumptions, different topology assumptions, etc. as will be explained next.

In [68], the different control methodologies for Networked Control Systems are surveyed, mainly, under delay constraint and impacts of delay for NCS. The first approach is to summarize the present NCS technologies and emphasize the importance of the delay on the performance of NCS, for which the plant may be dragged to unsatisfactory performance or even to instability. In the first section, the NCS configurations are stated. There are 2 types of NCS configurations: Direct [69] and Hierarchical [70]. In the direct configuration, the controller is, completely, separated from the plant via a data network. This is the most commonly assumed configuration for the most researches. In the hierarchical configuration, the controller is divided into two parts: Main Controller and Remote Controller [70]. The main controller calculates the reference signal for the remote system, and on the remote system plant and remote controller, the system operates with closed-loop operation without data network and calculates sensor measurements and sends them to the main controller via data network to calculate new reference signal. The delays that occur in the NCS are explained next. Mainly, there are two delays that affect the performance: sensor-to-controller delay, τ_{sc} and controller-to-actuator delay, τ_{ca}. Both of these delays are composed of three kinds of delays: waiting-time delay, τ_w, frame-time delay, τ_f, and propagation delay, τ_p [71]. The waiting-time delay is the delay that is the result of waiting for the queuing and network availability. The frame-time delay is the delay during which the source puts the frame on the network. Finally, propagation delay is the result of transmission on the physical media. All of these delays are affected by the plant and data network specifications and configurations [71]. Delays have two kinds of characteristics according to the types of networks: cyclic networks and random access networks [72]. For cyclic networks, the delays are periodic and for random access networks, the delays are random and are, generally, modeled based on the probability and the characteristics of the sources and destinations. Examples are Poisson process and Markov Chain process. The basic effect of delays is performance degradation [68]. This is observed by the unwanted response characteristics on the plant output. The further effect of delays is the destabilization of the system, which may be observed in a system with a PI controller by the reduced stability margin in a root-locus of the system [73].

For the controller design, some assumptions are usually accepted. Among these, the network transmissions are error-free, frame or packet sizes are constant, skew times and computation times are ignorable, the network is not overloaded, the sensor outputs can be packed in a single frame or packet [68]. The controller methodologies are stated next. In Augmented Deterministic Discrete Time model, the state space model of the discrete time system is used [72]. The plant state, plant output, plant state estimate in the controller, and the control values are combined and an augmented state-space equation is formed and the delays are treated in this augmented state space modeling. For a linear system, the stability criterion is obtained by [72] for a periodic delay system. The next controller design method is the queuing methodology [74]. In this model, a FIFO queue is placed in the controller input for the sensor measurements and a FIFO queue is placed in the actuator input for the controller outputs, both of which are constant length. Also, the predictors are used in the controller and so the delays become deterministic, so the system operates as a time-invariant system. So, with the mathematical model and the predictors, the overall system specification is formed (Figure 3).

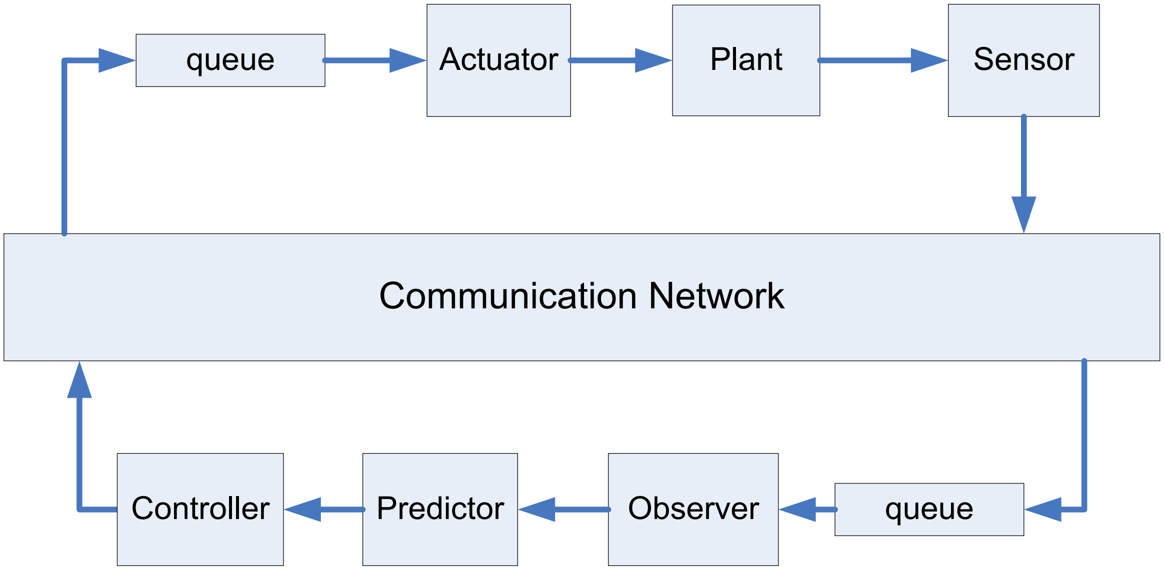

Figure 3. Queueing methodology.

Also, there is the probabilistic predictor-based delay compensation methodology for the queuing method [75].

The next controller design method is the optimal stochastic control methodology [76]. This method treats the random delays and the random delays are treated as Linear-quadratic-Gaussian (LQG) problem. This methodology assumes the delay is less than sampling period, T. The system equations are written according to this model and uncorrelated Gaussian white noises are added to state and output equations. And a stochastic cost function is formed that the expected values of the components are used to calculate the cost function value [76]:

$$J(k) = E[x^T(N)Q_N x(N)] + E\left[\sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}\right] \tag{1}$$

where $x(N)$ and $u(k)$ are the state and control values at time N and k, respectively. Q_N, Q are suitable sized square matrices and E is the expectation operator.

Finally, the control law for the optimal state feedback is calculated by dynamic programming. For the lack of full state feedback, the Kalman filter method can be applied to the controller law calculation conforming to the model in Figure 4.

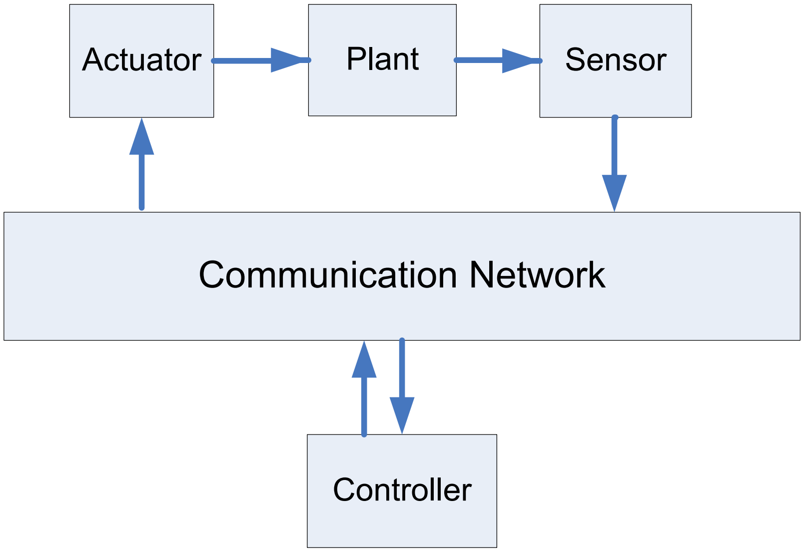

Figure 4. NCS Model.

The next control methodology is perturbation methodology [51]. In this method, the difference in the plant output and controller input (delayed plant output) is modeled as a perturbation and so, the perturbation is tried to be minimized, that is, vanished. The plant state vector and controller state vector is combined to form an augmented state vector and the error in the plant output and controller input is assumed to have certain dynamics and so a bound on delay is calculated to guarantee stability [51]. Another method is the sampling time scheduling methodology [77]. In this method, the plants are assumed to adjust sampling times, and according to the worst case delay bound, a new sampling time for the

NCS components are calculated. In this method, single dimensional NCS is assumed and M different NCSs operating on the same network are assumed. A condition for the optimality is given in terms of sampling times:

$$2\sum_{i=1}^{M}\frac{T_1}{T_i} = r \qquad (2)$$

where T_i is the i_{th} sampling period and r is the total number of messages. Later, the single dimensional sampling time scheduling method is extended to multi-dimensional NCSs.

The robust control methodology is, also, applied to NCS controller design. [65] designed a networked controller in the frequency domain using robust control theory. The novelty of the research is that it does not require a priori information about the delay's probability distributions and the delays are modeled as multiplicative perturbations [65]. The formulation of the network delay is approximated with a first order Padé approximation. And a perturbation function, Δ and a multiplicative uncertainty weight, W_m in the frequency domain form the uncertain network delay multiplicative component:

$$1 + W_m(s)\Delta \qquad (3)$$

The block diagram of the control methodology is given below:

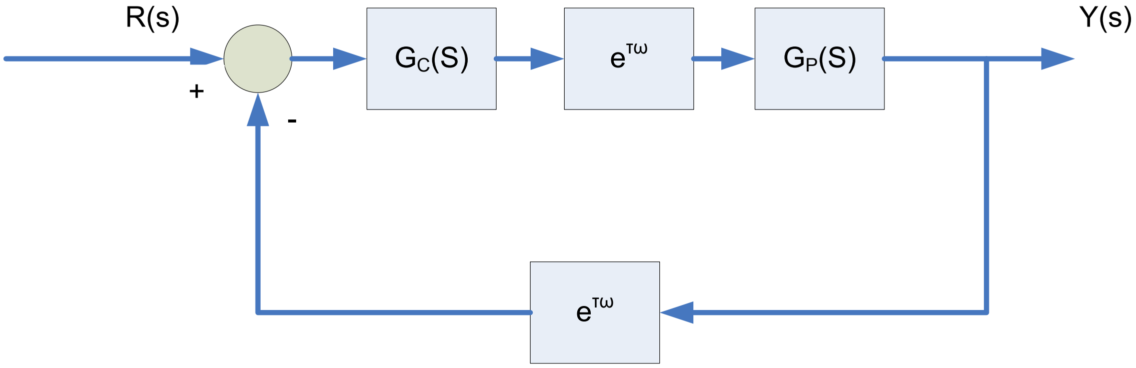

Figure 5. Frequency domain controller methodology.

where in Figure 5, $e^{\tau\omega}$ is the uncertain network delay component.

The next control methodology is fuzzy logic control methodology [67]. In this method, the network delay effects are handled according to fuzzy logic modulation. Two membership functions are assumed according to the error in the reference signal and the received plant output, and according to the size of the error, actual controller output is modulated with a coefficient whose size is calculated with the assumed membership functions. The shape of the membership functions are fine-tuned with a cost function according to the type of the optimization type (online or offline). The block diagram of this type of control is given below [67]:

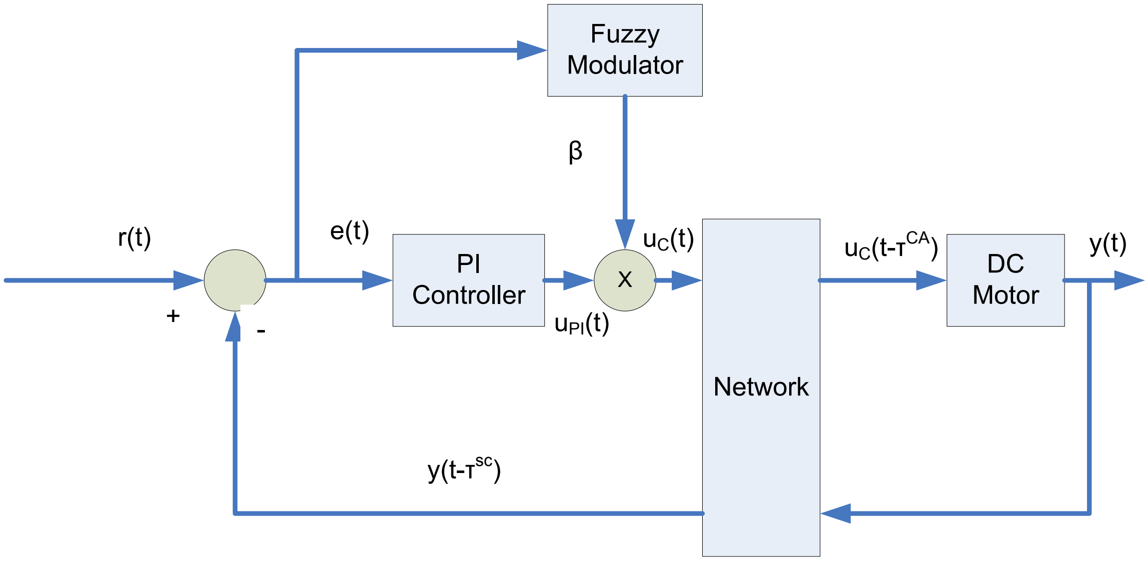

Figure 6. Fuzzy controller approach.

In the event based methodology, an original approach is stated and this method is developed for hierarchical structure and can be applied for direct structure, also [66]. Instead of using time, this method uses a system motion and calculates the reference of the system as the controller input according to this system motion. The updated reference of the system compensates delay effects [66]. Since no assumption on the network delay is assumed, any kind of delay can be compensated with this type of method. The block diagram of the event based method is given below [66]:

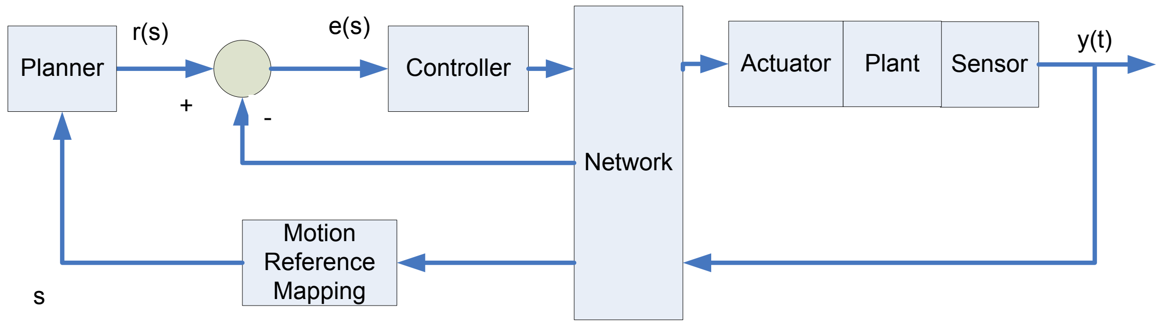

Figure 7. Controller approach based on system motion.

In the final control method, the end-user control adaptation methodology, is to adapt the controller parameters according to the current QoS measured from the network. The system controller is simulated with various possible QoS values and for each according to the integral of the error function between the system output and the reference output, various controller parameters are obtained, and when the corresponding QoS values are measured the pre-computed controller parameters are used to operate the system [78].

Another survey on NCSs, [1] investigates NCSs at a different perspective. In the first section, this paper gives information about the current controller design studies and states that, mostly, the linear models for the NCS models is investigated, whereas nonlinear models has received less attention [1]. Next, the studies on the delay that occurs in NCS which is control-induced delay and network-induced delay are stated and the importance is stressed. Then the NCSs are divided into two groups: Class A problems and Class B problems [1]. In Class A problems, the NCS is the simple model where plant and controller is separated with a data network:

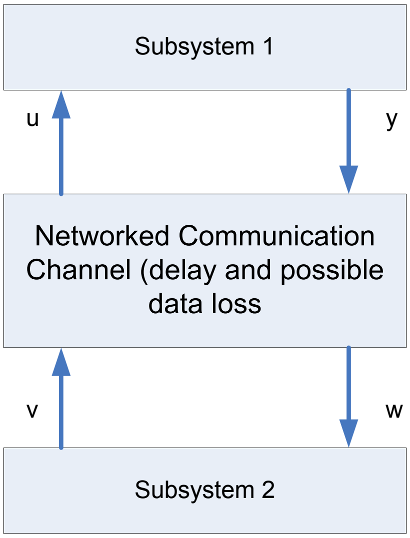

Figure 8. Class A topology approach.

In Class B problems, the network usage is two-leveled and level 1 operates with a data network which is the main control loop operations are carried out and level 2 is the high level network which carries out high level operations such as controller parameter adaptation, etc [1]:

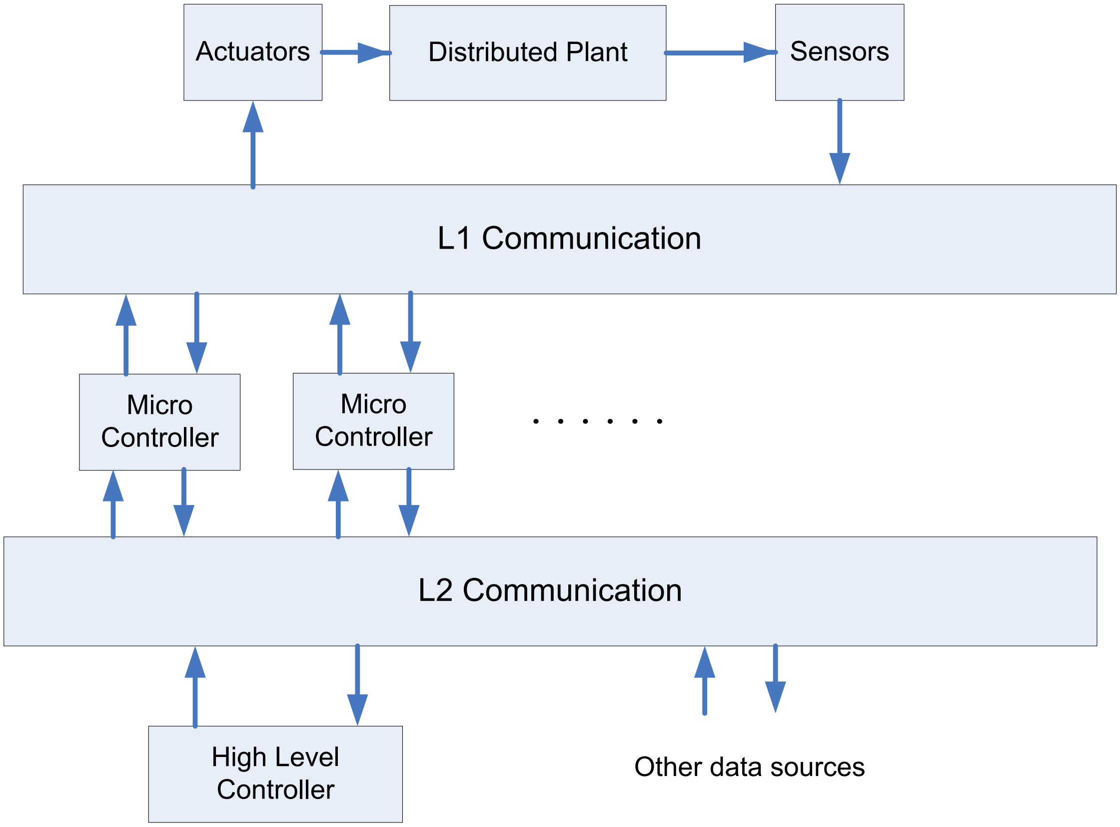

Figure 9. Hierarchical topology approach

Next, the NCS analysis approaches are stated. Also, NCS current architecture, protocols and packet scheduling in NCSs are explained. Then, the experimental and simulation studies are investigated. Also, the deterioration of stability and performance limits imposed by decentralized systems and the effect of NCSs to improve the performance of large-scale system control methodologies is stated. The evolution from centralized to decentralized control and from decentralized control to quasi-decentralized control and, finally, the evolution to NCSs is investigated [1].

In [51], a network protocol, try-once-discard (TOD) is offered, and an analytic proof of global exponential stability of the new protocol compared with the common static scheduling protocol, where classic TDMA techniques are used to arbitrate network access, is given. Firstly, the general model of the system is given where the LTI model is used to model the plant. Then the network model is defined and an error is defined where error is reset to zero when network transmission of the related control component message is transmitted and at other times the error is increasing with time. For this error, the error bounds are given for the two protocols, which is the maximum error [51]. Then by using these bounds, the bound on the maximum allowable transfer interval (MATI) for each control message is derived which guarantees global exponential stability [51]. The bounds are given for both the general NCS and the NCS with single-packet transmission. For the proof, the Bellman-Gronwall Lemma is utilized.

In [79], a LQR for a NCS is designed. In the modeling, firstly, the delay from the sensor to controller is assumed less than the sampling period and constant, and also the controller to actuator delay is assumed constant. Then, the LQR is designed. For the time-varying delays, fuzzy logic method is used to estimate the delays online. In the estimation algorithm, a window of past delays of size m, and the current sampling time of the controller and sampling period of sensor is used. According to the estimated values of delays, the fuzzy set of delays is chosen and pre-computed linear controller matrix is chosen that will preserve stability of the system [79]. The simulation results validate the method [79].

In [80], the NCS controller design is focused on state estimation under quantized signal transmissions and via a digital data-rate limited channel and under these conditions the control calculation problem. The problem is divided into two parts: the up-link design and the down-link design. The up-link design is the sensor-to-controller transmission problem and the down-link design is the controller-to-actuator transmission problem. The up-link design is the state estimation problem in controller with the sensor measurements received. The problem is firstly modeled as a stochastic problem and Monte-Carlo estimation technique is used to solve the state estimation [80]. Then, a deterministic formulation that will give state estimations under the same communication constraints is given that deals with the quantization regions formed from the transmissions of sensor values. Then, down-link design is investigated and quantized control values are calculated and in order to reduce bandwidth usage, the increments in the control values are calculated to be transmitted. And, an optimal control approach is used to design the controller with prediction of the next control value from the receding horizon of the control values [80]. The stability of the model is shown with a proof [80]. Simulation results and experimental results are given to validate the results obtained [80].

In [81], a NCS is modeled that is comprised of a shared data network with limited number of input channels (sensor-to-controller) and limited number of output channels (controller-to-actuator) [81]. Then, a mathematical model is formed that includes the ordered sequences of the nodes on the network that access the network. The model is formed with a variable which is 0 for no-access and 1 for access to network. With the sequences of this variable, the communication sequences are formed [81]. Finally, the proper order of the communication sequences is shown to exist for preserving the reachability and observability of the system [81]. Finally, an observer is added to the NCS with the above communication sequences that stabilize the NCS [81].

In [32], a memoryless NCS controller and the value of the maximum allowable network-induced delay are obtained by solving a set of linear matrix inequalities (LMIs) for a continuous time system. In modeling, both message loss and delay in network is modeled. By using a Lyapunov functional, firstly, a theorem for the exponentially asymptotical stability is stated from the derived model [32]. Then, by using the stated theorem, an algorithm is given to obtain the maximum allowable network induced delay value [32]. Finally, a numerical example is given for the validity of the results [32].

In [82], the controller design for the LTI systems that operate over unreliable links such as Internet or wireless medium is investigated. The controller design is carried out by optimal control approach by mean square stabilizing the system by minimizing a quadratic performance criterion as below [82]:

$$J_\pi = E[x_N^T F x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + \alpha_k u_k^T R u_k] \tag{4}$$

where x_k is the state and u_k is the control and $F \geq 0$, $Q \geq 0$, $R > 0$. The NCS model is suitable to the Class A type NCS in [1]. Two types of communication protocols is assumed, where in the first one an ACK for the transmitter is present and in the second one no ACK is present. Specifically, the Internet protocols TCP and UDP are assumed as the two protocols [82]. The idea depends on the fact that, for the TCP, the source has the knowledge of the lost packets on the network, whereas for the UDP this knowledge is not present [82]. Finally, sufficient conditions for the existence of the stabilizing controllers are given [82]. The conditions are used to obtain stability regions for the NCSs and a numerical example is given illustrating the achievement of the results [82].

In [83], a NCS model, of which controller has a model of the plant to estimate the plant state, and the communication medium that is between sensors and the controller is assumed. Then a model of the system is formed by state feedback to the controller with model uncertainty. Firstly, the Lyapunov stability condition for the NCS, of which communication medium results in delays in a constant interval, is derived [83]. Later, the stochastic delay is assumed and this delay is assumed as uniformly distributed or Markov chain-driven [83]. For the two cases, almost sure asymptotic stability where the stability is achieved with probability 1, and mean square asymptotic stability conditions are derived [83]. For the mean square asymptotic stability, the system state, z satisfies the following condition:

$$\lim_{t\to\infty} E[\|z(t, z_0, t_0\|^2] = 0 \tag{5}$$

and for almost sure asymptotic stability

$$\lim_{\delta \to \infty} P\{sup_{t \geq \delta} \|z(t, z_0, t_0)\| > \varepsilon\} = 0 \tag{6}$$

where *z(t,.,.)* is the state at time *t*, z_0 is the initial state and t_0 is the initial time [83]. It is shown that both types of stochastic stability allows larger network induced delays in the system compared to Lyapunov stability [83].

In [84], the same NCS model as in [83] is used, and this time the delay is constant and the effect of quantization is investigated and the sufficient stability conditions for the NCS under quantization are obtained. The investigated quantization methods are static and dynamic [84]. The static quantization methods are uniform quantization and logarithmic quantization [84]. It is shown that the minimum data rate needed for stability coincides with the well-known minimal theoretical rate for stability for the dynamic quantizer [84].

In [35], a similar model to [83] is assumed, but this time the disturbance effect is added to the model as below:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \tag{7}$$

$$y(t) = Cx(t) \tag{8}$$

where *x(t)* is the state , *u(t)* is the input, *d(t)* is the disturbance signal and *y(t)* is the output at time *t*. A, B, C and E are appropriate sized matrices.

The system is designed as a switched system and the delay and data message loss is modeled firstly in the system. The delay is modeled by assuming a sampled data system and by using ZOH method during the network delay [35]. The delay and data message loss modeling results in a switched system of a number of modes, which is determined with a relation between maximum number of consecutive message losses and maximum number of delay samples [35]. Then a condition for stability of the system is derived [35]. Finally, the attenuation of the disturbance on the system is investigated and an algorithm to calculate a robust performance bound for the disturbance effect [35].

In [85], a NCS model such as the one below is assumed:

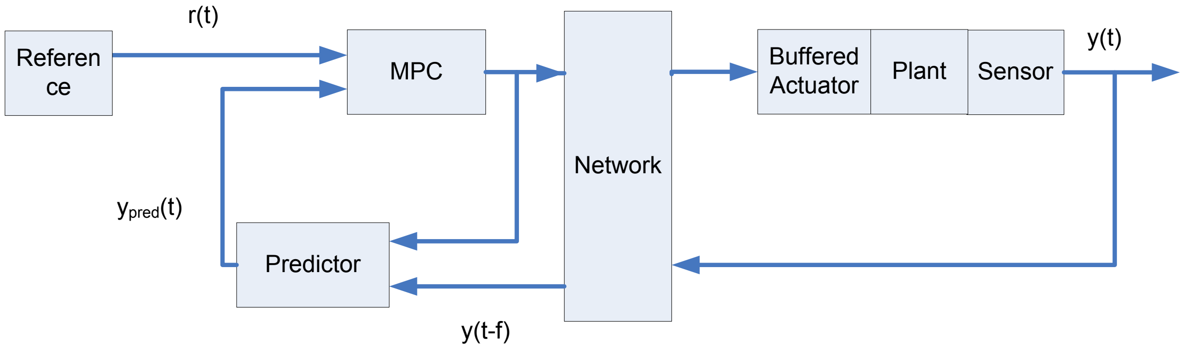

Figure 10. Controller based on prediction.

In this model, at controller side a predictor for the current output value is used [85]. Then, this value and the reference input is passed into model predictive controller (MPC) to generate a series of control values for the subsequent times [85]. The idea is to send all these control values to the plant, and by choosing the correct control values from the sequence with the known delay calculated from the timestamp, the expected control values are reached [85]. The sufficient conditions for the stability of this scheme are obtained that preserves the stability [85]. Simulations and experiments are done to validate the results [85].

In [86], the state estimation problem in a NCS, where delay, data message loss, out-of-order delivery and corrupted data exists in the system is treated. The aim is to find the minimum variance state estimate of the state in the controller by model predictive control techniques by dynamically scheduling the sensors to transmit their observations [86]. The paper assumes that there is a feedback mechanism from controller to sensors to adjust the sensor transmission schedule [86]. A minimum variance state estimator is designed under the above constraints [86]. In [87], a similar NCS model is assumed, but this time, instead of sensor scheduling, an optimal control approach is used to design the controller and a recursive state estimator is used in the controller to estimate the state [87]. The proofs for the state estimation method's validity and the controller's validity are given [87].

In [88], a NCS model similar to [86] is assumed. The optimal control approach is used to design the controller, the topological entropy for the NCS is defined and the method to determine the topological entropy is given [88]. It is shown that the topological entropy defined for the NCS determines the solvability of the optimal control problem for the NCS. In [89], a NCS model similar to [88] is assumed. And, the state estimation via Kalman filtering method is proposed [89]. Then, the strongly robustly observable system criterion is defined and the condition for the criterion to hold is given [89]. Finally, the coder in the sensor and decoder in the controller pair is given that will provide optimal state estimation in the controller, and a numerical example is given to illustrate the obtained results [89].

This chapter deals with the controller strategies in the NCSs. The main problem is eliminating the network-induced delay that is the total result of the message set parameters message period, message deadline and message loss. Each controller strategy suits a specific type of control system and integrating it with a specific scheduler is not a straight forward task since the communication network protocol dynamics is another system with different requirements. Also, the different topological structures in the control methodologies invalidate the integrated controller and scheduler approach. The survey shows that the best and simplest approach is to predesign the controller under the constraints of the control system firstly, and integrate the scheduler to provide the service requirements of the control system.

CHAPTER 4

NCS MESSAGE SCHEDULING SURVEY

In this chapter, we present different approaches to the message scheduling problem in NCSs and categorize them into three classes. Class A is defined as the class that includes restricted network dynamics. Class B is defined as the class that includes restricted network topology assumptions, e.g. communication network is only between controllers and actuators. Class C is defined as the class that includes restricted performance measures for optimal design, e.g. the solution optimizes control system performance metrics such as stability, minimum output error, however, the network utilization or message set requirements are discarded. We remove all these simplifications in our model leading to a more general and realistic framework.

In [90], a survey on real-time scheduling for embedded systems is presented. These scheduling techniques are also applicable to the NCSs as other real-time systems, where possible benefits can be gained. The paper is, mainly, focused on two scheduling techniques: Fixed-Priority Scheduling and Dynamic-Priority Scheduling. In real-time scheduling theory, priorities are given to jobs to access shared resources such as communication channels. In fixed-priority scheduling, all jobs are given fixed priorities and access the resources in the order of decreasing priorities (ex: highest priority job accesses the resource firstly). In the pioneering work on fixed-priority scheduling by Liu and Layland [91], an analysis is discussed. Rate Monotonic approach is described and it is stated that the scheduling with this approach is feasible under a set of assumptions if:

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(2^{\frac{1}{n}} - 1) \tag{9}$$

where C_i is the computation time of job i, T_i is the period of job i, and the n is the number of jobs. The summation on the left of the given inequality is the utilization of the resource. In the study, also it is stated that the feasibility analysis above is pessimistic and more realistic feasibility analysis is given. It is also stated that in Liu and Layland model, the jobs are assumed independent and periodic, which may not be realistic. The job interaction and aperiodicity are also investigated. Several protocols for solving the job interaction problem are outlined.

In [90], dynamic-priority scheduling is also investigated. The basic approach for this type of scheduling technique is the Earliest Deadline First (EDF) algorithm. In this type of scheduling, the priorities are assigned in the order of decreasing deadlines of the jobs (ex: Nearest deadline is given the highest priority). Liu and Layland showed that a set of jobs are schedulable under the same assumptions above if:

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1 \tag{10}$$

where C_i is the computation time of job i, T_i is the period of job i. Also, the Least Laxity First (LLF) algorithm is discussed and it is stated that both EDF and LLF are optimal algorithms. The processor demand criterion approach to analyze EDF is given. The aperiodic job scheduling under EDF and the total bandwidth server approach for scheduling aperiodic jobs is stated. Also, the constant bandwidth server approach for scheduling jobs which need resource reservations is stated. These scheduling techniques are also applicable to the NCSs as other real-time systems, where possible benefits can be gained. Resource sharing and overload management problems under EDF are investigated as well. These problems emerge also in NCSs. Resource sharing affects directly jitter in NCSs and overload management is required when network bandwidth is not sufficient for the current message scheme.

In [92], the design of NCS is aimed to unify both the controlled system design and network design issues. The separate control and network performance measures are combined to obtain a schedule by assigning priorities for the processes with the RM policy. The authors discuss the Liu Layland analysis about the RM policy optimality that: No fixed priority schedule exists for a set of processes if they can not be scheduled with RM policy.

The cost function in [92] is:

$$J(h_i) = \sum_{i=1}^{n} J_i(h_i) \tag{11}$$

where

$$J_i(h_i) = \frac{\bar{a}}{a_i} e^{a_i h_i} \tag{12}$$

under the RM schedulability and NCS stability constraints:

$$h_1 \leq \cdots .. \leq h_N \tag{13}$$

$$\frac{c_1}{h_1} + \cdots .. + \frac{c_i}{h_i} + \frac{b_i}{h_i} \leq i(2^{\frac{1}{i}} - 1) \tag{14}$$

$$h_i \leq h_{suff,i} - b_i \tag{15}$$

where c_i is the computation time, h_i is the period of the task, b_i is the blocking time, a_i is a positive number, N is the number of processes, $h_{suff,i}$ is the maximal period. This cost function optimizes transmission periods by accounting for output error in the plant outputs with the $\bar{a}$ and a_i error parameters. In other words, the cost function minimizes plant output errors with respect to transmission periods. Then, the effects of delay and packet message losses are investigated separately. A bound on message loss rate is derived. A scheduling policy which uses this message loss rate bound is derived and, when the utilization of the system is greater than 1, by dropping some packets not greater than the derived bound, it is stated that both the system has become schedulable and the stability is assured. This is a class ABC work. Class A emerges from the fact that real network dynamics is not included. Class B emerges from the fact that communication is only between sensors and controllers. Class C emerges from the fact that only message loss as the QoS is considered, where delay is more effective in NCS.

In [93], the scheduling of NCS, when message loss governed by Markov chain exists in the system, is investigated. The model assumes average message loss rate instead of assuming a true switched system. The model of the NCS with data message losses is given below:

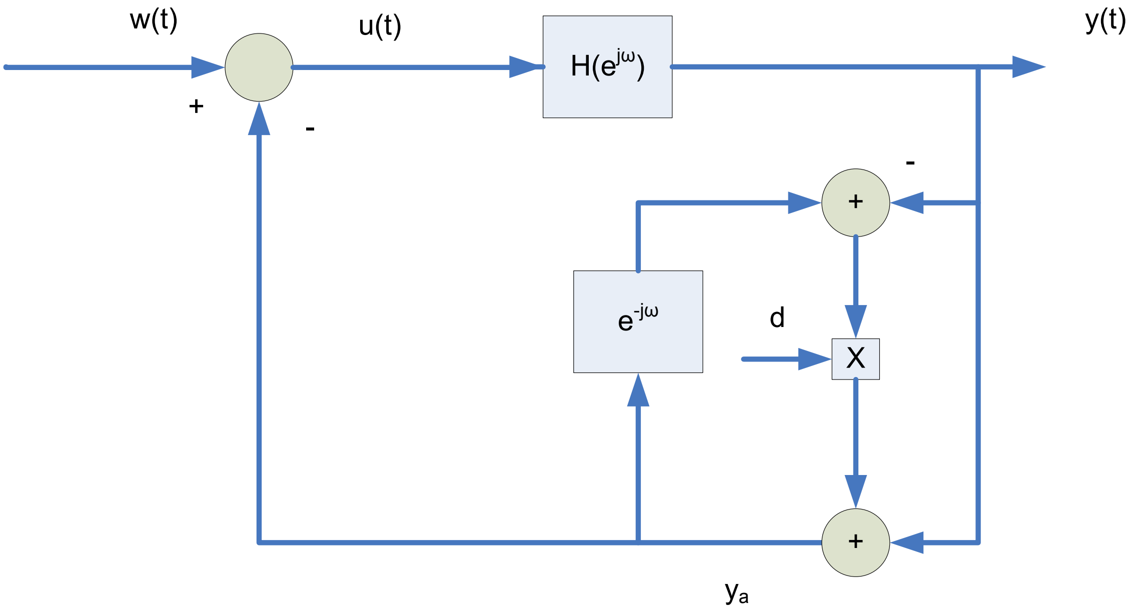

Figure 11. Scheduling approach based on message loss modeling.

Under white noise assumption, the theorem for the stability of the system is given for the assumed model. The optimal message loss policy under message losses is given and, also, the soft message loss policy with message losses governed by Markov chain is shown to be stable by allowing some of the message losses and improving schedulability. The performance measure is the output power's semi-norm. This is a class ABC work. Class A emerges from the fact that real network protocol dynamics are not included. Class B emerges from the fact that communication is assumed very simple topology to include its real effect. Class C emerges from the fact that only message loss as the QoS is considered where delay is more effective in NCS.

In [28], the co-design of adaptive controllers and feedback scheduling policies to achieve overall Quality-of-Control (QoC) is treated. Firstly, the limitations of the existing models are stated and then, the new model is proposed. The new model accounts for the dynamics of both the application and executing system, and achieves dynamic management of the QoC by message scheduling. Firstly, the system model with time delay is formed. Then, the system model with time varying delay is modeled. The assumed system model is given below:

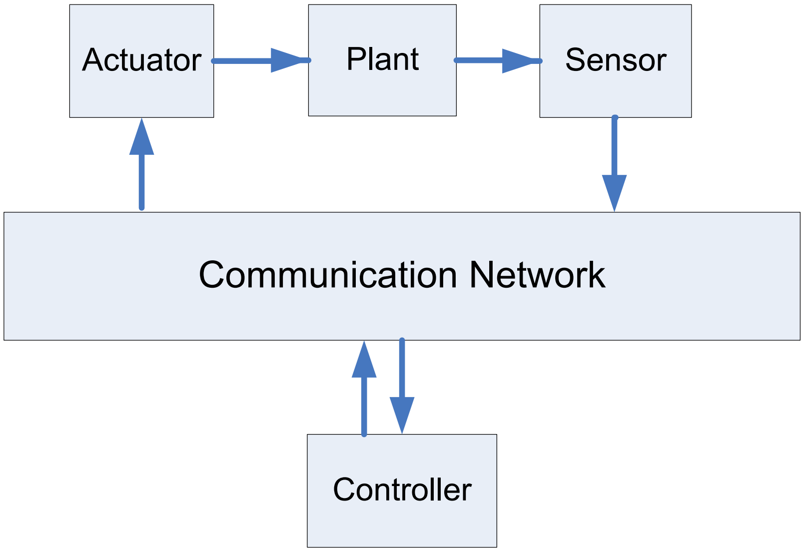

Figure 12. Basic NCS model.

The model evaluation is done with a response error calculation which is the integral of the absolute error:

$$IAE = \int_0^\infty |e(t)|dt \tag{16}$$

where

$$e(t) = y_d(t) - y(t) \tag{17}$$

and $y(t)$ is the measured system output and $y_d(t)$ is the desired system output.

We mentioned in Chapter 1 that our control system performance metric is QoC. In [28], an alternative definition of the QoC is stated as:

$$QoC\left(y_{act}: seq < \tau_{\sigma(k)} >\right) = \frac{\frac{1}{IAE(y_{act}:seq<\tau_{\sigma(k)}>)} - \frac{1}{IAE(y_{act}:seq<\max(\tau_\sigma)>)}}{\frac{1}{IAE(y_{act}:seq<\min(\tau_\sigma)>)} - \frac{1}{IAE(y_{act}:seq<\max(\tau_\sigma)>)}} \tag{18}$$

where $\tau_\sigma(k)$ is the sampling-actuation delay at the *kth* instant, $seq<\tau_\sigma(k)>$ is the sequence of delays applied to system messages, and $seq<max(\tau_\sigma)>$ and $seq<min(\tau_\sigma)>$ is the system performance measured with longest and shortest message delay. This QoC measure is the basic performance measure for obtaining the optimal message schedule.

In [28], it is shown that: it is true in every condition that, the smaller the delay in each message, the better the system response. Finally, an optimization problem for determining the bandwidth allocation to the nodes is formulated by maximizing QoC. This is a class AC work. Class A emerges from the fact that real network protocol dynamics are not included. Class C emerges from the fact that only QoC is considered, while QoS or bandwidth utilization is not considered.

In [19], the problem of the optimal control and scheduling of networked control systems over limited bandwidth deterministic networks is investigated.

Multivariable linear systems subject to communication constraints are modeled in the Mixed Logical Dynamical (MLD) framework, Then, the model is translated into the mixed integer quadratic programming (MIQP) problem. Then the solving this problem is described, and arrived to the modification of this solution to obtain an online scheduling algorithm, which is a compromise of the original solution which must be solved offline because of computational complexity.

The system model assumed in [19] is given below:

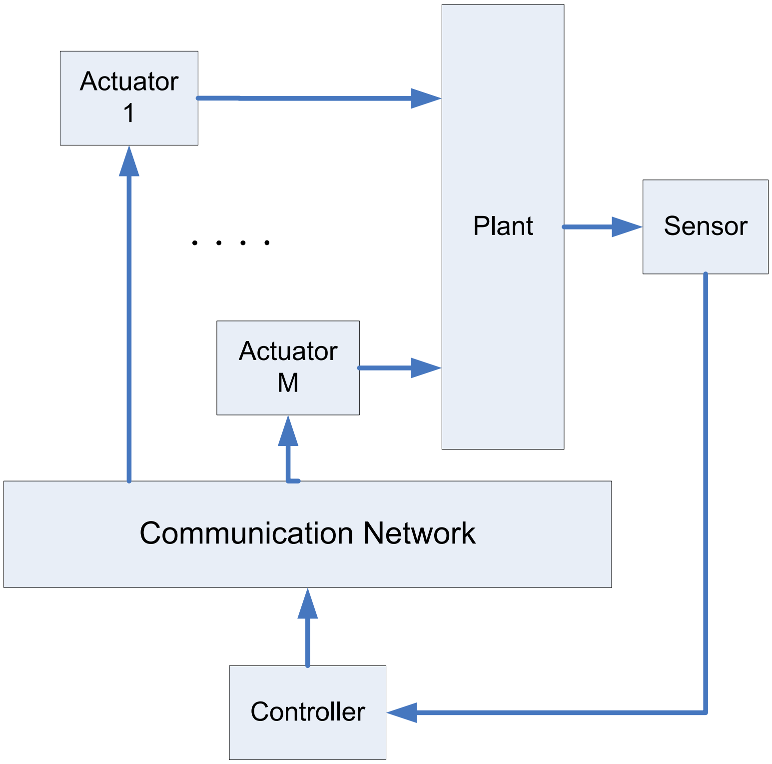

Figure 13. NCS model with multiple actuators.

The model assumes zero-delay between sensors and controllers, and assumes the communication is between the controllers and the actuators. This is simplification of the model, which may be generalized. MLD problems are solved with Model Predictive Controller (MPC) techniques and the high computational complexity is the result of this technique [19]. By simplifying the technique with some assumptions which are trade-off of performance, the reach Optimal Pointer Placement (OPP) algorithm which may be run online to schedule messages. It is shown that OPP algorithm is better than static scheduling and worse than the solution by MPC technique by simulations. This is a class ABC work. Class A emerges from the fact that actual network dynamics are not included. Class B emerges from the fact that communication is only between controllers and actuators. Class C emerges from the fact that the problem is formulated as an optimal control problem, and communication performance is ignored.

In [94], which is later corrected in [56], the scheduling algorithm for Controller Area Network (CAN) is given by applying classical real-time scheduling to CAN. This work gives some insights to us about evaluating the network performance for the quantifying the delay in the feedback loop of the control systems. It is shown that the developed CAN analysis can give worst-case latency of a given message queued for transmission on CAN. The analysis is applied to a specific CAN controller, namely, Intel 82527. The scheduling analysis is used to show that the SAE benchmark for class C automotive systems is run on Intel 82527 and all messages are schedulable, although the 125 Kbit/s bus speed is chosen for the bandwidth capacity of CAN. The analysis is carried out to find the response time of each message and it is compared to the deadline of the message. The key point in finding the response time is finding the queueing delay:

$$w_m = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{w_m + J_j + \tau_{bit}}{T_j} \right\rceil C_j \tag{19}$$

where

w_m: queueing delay

B_m: blocking time by lower priority messages

J_j: jitter time of message j
C_j: transmission time of message j
T_j: period of message j
τ_{bit}: bit time
hp(m): messages higher priority than m

Then response time is:

$$R_m = J_m + w_m + C_m \tag{20}$$

In [55], the above analysis is carried out on more general cases other than CAN, and enriched with possible other cases. The analysis is first carried out by ignoring jitter time. Then, by adding jitter, the above results are obtained. The jitter analysis is developed to the case of precedence constrained distributed tasks:

$$J_i = \max_{\forall k \in dpred(i)} J_k + r_k + M_{k,i} \tag{21}$$

where
dpred(i): is the set of tasks that precede task I
$M_{k,i}$: is the worst case transmission time of message from task k to task i

The updated jitter is used to analyze the response time of messages. Also, the tick driven scheduling and the sporadically periodic message cases are investigated. A sample task set is used to exhibit the effectiveness of the analysis.

The survey in this chapter gives important insights to develop the performance measurement metrics and algorithms when the network protocol is known. Among these contributions are developing our scheduling method, measuring the delay in the feedback loop of the control systems, and the construction approaches of our QoC and QoS metrics.

CHAPTER 5

STABILITY TEST

The delay in the feedback loop of the networked control system has a stability bound. So, in order to design a message set and message schedule, this stability bound must be determined with a stability test. The stability concept in NCSs is explored in several ways. Our model conforms to periodic sampling and variable delay, and packet losses are possible. Next we briefly summarize the literature and explain the stability test that we selected for our approach.

The study in [30] is one of the pioneering works for the analysis of stability of NCS with delay and in [2], a detailed survey about stability in NCSs may be found. In [31], the time delay system modeling approach is detailed for NCSs. [32] develops its analysis similar to [31], but the message losses are not considered. In [33], the delay bound model conforms to periodic sampling and constant delay, and the stability of the system is analyzed. In [34], the authors assume periodic sampling and variable delay, but assume small delays and no multiple packet losses are concerned. In [35], the stability analysis in [34] is extended to multiple packet loss case. The method in [35] is developed based on the work in [36]. In [37], the stability for variable sampling and variable delay is considered, but the use of time-stamps is required for the correct working of the system. In [38], also the clocks of controller and sensor are assumed to be synchronized which is a restrictive assumption for our case. In [39] the stability for a specific network protocol, namely TCP, is considered while our aim is not to depend the stability test on network dynamics. In [25] the packet losses are also considered, while stability for delay is investigated, so this test is also a candidate for our stability test. It is also shown in [25] that the stability test found is less conservative than the previous works in [40], [41], [32], [42], [33]. [40], [32] and [42] also express the delay bounds based on LMIs, but are found to be more conservative than [25]. We choose to base our stability test on [25] as it fits to the properties of our system model best.

This part forms the details of the mathematical model of the control part by conforming to the model in [25]. This model and stability test is chosen, because of the conformance of both the model and the test to the requirements of the problem. This model is applied to a single control system consisting of a plant, a sensor, a controller and an actuator as in Figure 4.

The physical system (plant) is assumed to have continuous dynamics. The plant is assumed to be linear and time-invariant. The control law of the control system is assumed to be static to apply the stability test of [25] to the analysis. As most of the control system literature treats the linear time-invariant control systems [27] and most of the nonlinear systems can be suitably linearized, we assume a linear time-invariant control system. The assumed control system is given below:

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{22}$$

$$y(t) = C_c x(t) + D_c u(t) \tag{23}$$

$$x(t_0) = x_0 \tag{24}$$

$$u(t^+) = Kx(t - \tau_k^{sc}),\ t \in \{kh + \tau_k^{sc}\},\ k = 1,2,\ldots \tag{25}$$

where, in the above equations, A_c, B_c, C_C and D_c are the physical system parameters. K is the static control law and disturbance effects are ignored. $x(t)$, $u(t)$ and x_0 are the state value, control value at time *t* and the initial state value at time t_0, respectively. τ_k^{sc} is the sensor-to-controller delay. h is the sampling period. k is the index of sensor samplings. This model is expanded into the following model for the calculation of the control values which accounts for the message losses:

$$u(t^+) = Kx(t - \tau_k)\,,\ i_k h + \tau_k \le t < i_{k+1} h + \tau_{k+1},\ k = 1,2,\ldots \tag{26}$$

where i_k is the index of the sensor sample which is successfully transmitted to the controller and the corresponding control value is transmitted to the actuator successfully, and

$$\tau_k = \tau_k^{sc},\ k = 1,2,\ldots \tag{27}$$

The control part of the mathematical model is modeled as conforming to the system model in [25]. For the model, it is assumed that there is no synchronization between the sampler and the digital controller. So, the control value is no longer constant within a sampling period [25]. We will model the message losses and delays in the system with the sampling frequency of the sensor conforming to the model in [25].

For application of the stability test, we will use the following assumptions [25]:

The sensor is clock-driven, the controller and actuator are event-driven.

We assume two inequalities:

$$(i_{k+1} - i_k)h + \tau_{k+1} \le \eta,\ k = 1,2,\ldots. \tag{28}$$

$$\tau_k \ge 0,\ k = 1,2,\ldots. \tag{29}$$

We ignore the disturbance effects.

We assume no parameter uncertainties.

So, the stability test is the result of the test whether there is a feasible solution to the following linear matrix inequality given in [25] transformed into our case:

$$\Gamma_{11} = \begin{bmatrix} T_1 & T_2 & R_1 & R_2 & 0 \\ {T_2}^T & T_3 & S & S & 0 \\ R_1 & S & T_4 & 0 & 0 \\ R_2 & S & 0 & T_5 & 0 \\ 0 & 0 & 0 & 0 & -\in I \end{bmatrix} < 0 \tag{30}$$

where

$$T_1 = PA + {A_C}^T P + Q_1 + Q_2 - R_1 - R_2 + {A_C}^T H A_C \tag{31}$$

$$T_2 = PBK + {A_C}^T H B_C K \tag{32}$$

$$T_3 = (B_C K)^T H B_C K - 2S \tag{33}$$

$$T_4 = -Q_1 - R_1 - S \tag{34}$$

$$T_5 = -Q_2 - R_2 - S \tag{35}$$

$$H = \eta^2 (R_2 + S) \tag{36}$$

for $R_1,\ R_2, S, Q_1, Q_2, P$ are *n-by-n* real matrix variables, $\in > 0$ must be satisfied given η as the maximum allowable delay bound to be tested. $A_C, B_C\ K$ are the linear time-invariant plant parameters.

We assume time slots of the network are large enough to accommodate the single sensor reading and control value of a sensor node and the controller node, respectively. Since modern controller technologies are implemented on fast chips, this assumption is valid. If we call state-vector and control value vector dimension *n* and *m*, respectively, slot size T_S *sec.*, total bandwidth *BW bps* and the quantization of each state variable with *Q bits*, then a slot size choice of

$$T_S > \frac{(n+m)Q + a}{BW} \tag{37}$$

will guarantee that sensor messages and control messages fit into slots where *a bits* is the protocol overhead.

For each feedback loop, the sampling frequency f, the maximum delay d and maximum consecutive losses l of the sensor readings can be transformed into the worst case maximum delay in the system as

$$d_{max} = \frac{l+1}{f} + d < \eta \tag{38}$$

which is the inequality that must be satisfied in the feedback loop [25]. This equation directly derives from the fact that the stability test imposes a worst-case delay for stability in the feedback loop that corresponds to no message transmission in the NCS for this period in the feedback loop. No message transmission period in the feedback loop can be modeled in terms of maximum consecutive losses, sampling frequency and single message delay.

Finally, with the stability test maximum allowable delay bound in the feedback loop (η) will be determined for each control system. By using this bound, the message set parameters of the single control node as worst case overall delay (η), message frequency, message deadline and message loss.

In Table 2, we summarize our assumptions for the NCS model.

Table 2. NCS model assumptions.

COMPONENT	**ASSUMPTIONS**
Control System	Linear Time Invariant
Sensor	Periodic State Sampler
Controller	Linear Quadratic Controller
Actuator	Asynchronous Actuator (Applies the control value immediately it receives it)
Stability Test	Derived from [25]. Disturbance effects and uncertainties ignored.
Message Set	Message frequency (period), Message Deadline, Message Loss
Sensor-to-Controller Delay	Considered
Sensor-to-Actuator Delay	Assumed to be considered in Sensor-to-Controller Delay

CHAPTER 6

QoC AND QoS DEFINITIONS

The message sets and the message schedule that are designed must be quantitatively evaluated to optimize and trade off for better results. The message sets belong to control systems, so, as a control system measure we define Quality-of-Control (QoC) as a measure of the performance of the message set that belong to a specific control system. The schedule determines how network resources are shared, so, as a measure of the performance of the network resource sharing, we define Quality-of-Service (QoS). These definitions provide us with the metrics to integrate control system behavior and communication network behavior in the next chapter.

6.1. QoC Definition

In classical feedback control theory, several properties are used for closed-loop performance measurement [26]. The primary performance metrics for the closed-system response depend on transient-response, steady-state error and stability. We guarantee the system stability by assuming worst-case overall delay in the system to be less than Maximum Allowable Delay Bound (MADB). So, in order to improve the transient response and minimize steady-state error, we choose to calculate normalized step response of control systems rather than to consider all response time to measure steady state error. We use the Quality-of-Control (QoC) metric to measure the performance for the control system implemented as an NCS. Our definition of Quality-of-Control (QoC) is similar to [26, 28]. To this end, we calculate the normalized step response and measure the error between the actual response with the network delay incurred and the ideal response with no delay and loss. In [95], an alternative QoC metric definition is proposed, which is vector representation of various parameters. But, since in a single metric, we combine both transient response and steady state error, we base on the definition in [26, 28].

To sum up QoC metric measures how close the control system response is to the ideal communication case which is non-networked case with no delay and loss between the communicating components (i.e. sensors, controllers and actuators).

In [26, 28], the QoC is computed by considering best and worst cases of transmission sequences. Since we consider in each case the worst case for computation, we eliminate best case and worst case sequences, we compute QoC for the worst case delay in a single case. This enables us to reduce computational complexity of the QoC computation.

At the sequel, we will call *maximum allowable delay bound (MADB)* η_{max} and, when we refer to a specific control system, $\eta_{max,i}$ for *ith* control system. We will call η as the general feedback loop delay of the control system.

In this work we define QoC as follows:

Definition 1: *For a control system i, the system response s(t, η) for the worst case overall delay η in the feedback loop and ideal response d(t) the QoC is defined as:*

$$QoC = \frac{1}{\int_0^{\infty} |s(t,\eta) - d(t)| dt + \varepsilon} \tag{39}$$

ε is a small positive constant.

The reciprocal of the IAE (Integral of the Absolute Error) function serves to increase QoC for the parameters that result in worst case system response that is closer to the ideal response. This QoC definition is frequently used in the literature for measuring NCS performance [20].

The QoC values versus *k* for this definition for 5 different control systems with 5 different MADBs given in Table 3 are sketched in Figure 14 with the dynamical system parameters A_i, B_i, C_i, D_i, K_i and feedback loop delay η with sampling frequency calculated as:

$$sampling\ frequency = \frac{k}{\eta_{max}} \tag{40}$$

where $k=1,2,\ldots,10$, $i=1,\ldots.,5$.

Table 3. Control systems.

Component	**Parameters**
Control System 1	A1=[6 -3;0.1 -0.02] B1=[1; 0.4] C1=[1 0] D1=0.1 η_{max} =0.0894 sec
Control System 2	A2=[-2 8;-1 3] B2=[-1; 2] C2=[0 1] D2=1.3 η_{max} =0.1363 sec
Control System 3	A3=[-0.03 0.02;0.02 0.09] B3=[-0.4; 0.3] C3=[1 0.2] D3=2 η_{max} =3.8400 sec
Control System 4	A4=[-0.08 -0.004;0.1 0.02] B4=[0.88;0.3] C4=[1 1] D4= 0.3 η_{max} =1.4400 sec
Control System 5	A4=[-0.8 -0.004;0.1 0.4] B4=[0.88;0.3] C4=[1 1] D4= 0.6 η_{max} =0.9606 sec

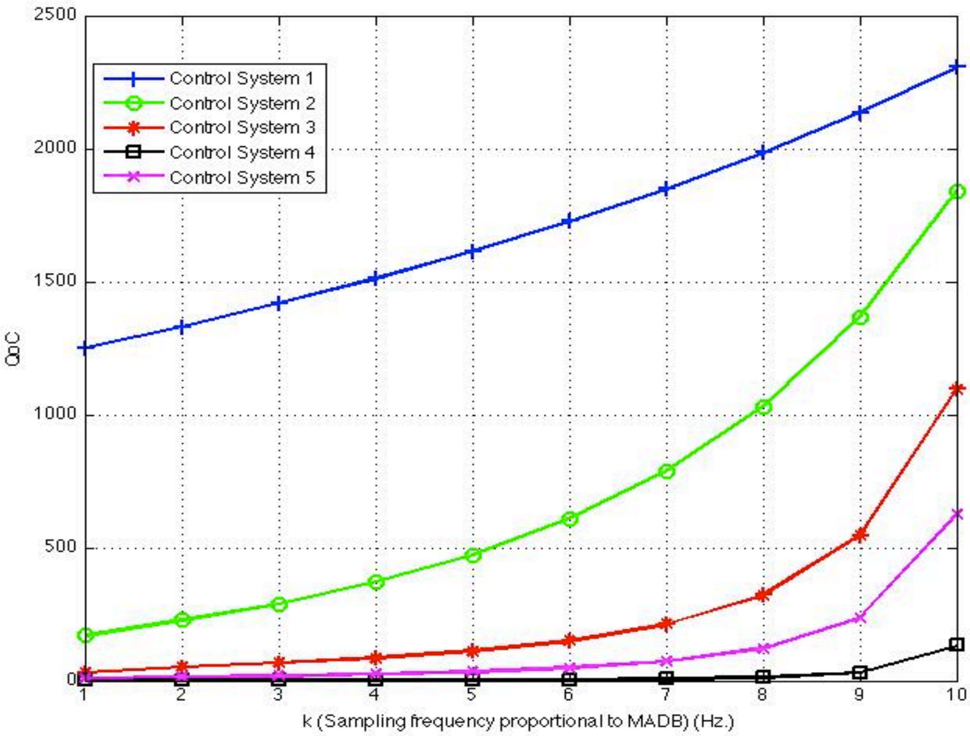

Figure 14. QoC values for sampling frequencies proportional to each control system's MADB.

Since, this representation of the QoC are different in values to reflect the same performance for the same desired response (at the point 0.1 times of the MADB that corresponds to 10 Hz. in Figure 14), we normalize this QoC value with the value obtained at the desired response point that is reference QoC value at the desired response, $s(t,\eta)$ at $k=10$:

$$RefQoC = QoC(s(t, 0.1\eta))$$

Definition 2: *For a control system i, the system response s(t, η) for the worst case overall delay η in the feedback loop and ideal response d(t) the QoC is defined as:*

$$QoC = \frac{1}{\int_0^{\infty} |s(t,\eta) - d(t)| dt + \varepsilon} \times \frac{1}{RefQoC} \qquad (41)$$

where ε is a small positive constant. RefQoC is the reference QoC for the control system sampled at the period of 0.1 times the MADB as suggested in [27] and computed according to Definition 1.

The normalized QoC values for Definition 2 for 5 different control systems with 5 different MADBs are sketched in Figure 15.

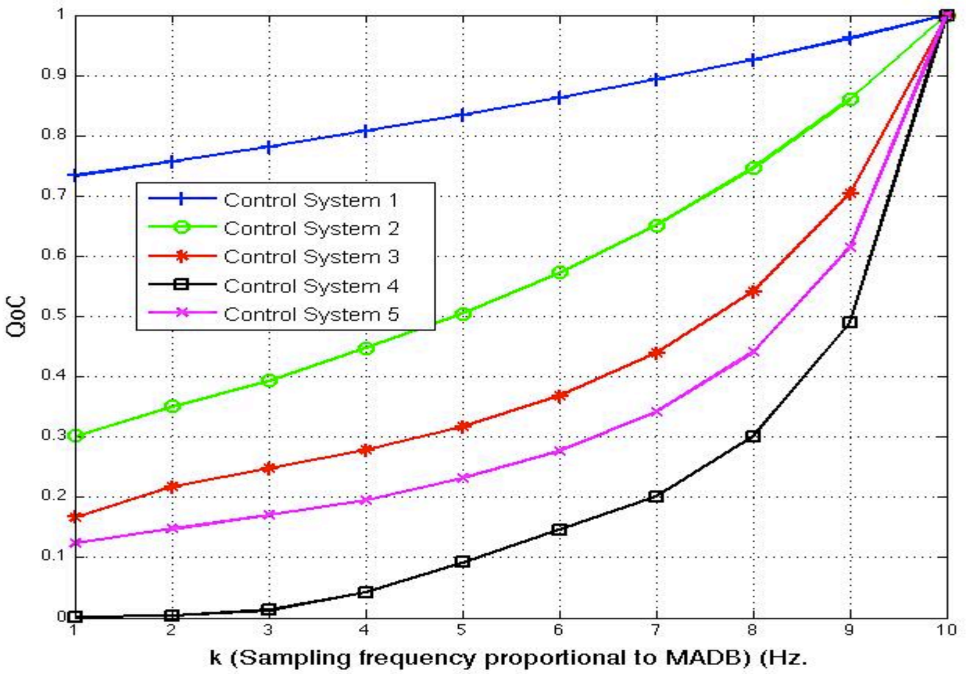

Figure 15. Normalized QoC values for sampling frequencies proportional to each control system's MADB.

The normalized QoC values serve to interpret the QoC values as the same performance at the suggested points in the design. In other words, at *k=10* we can say that all systems are interpreted to have the same performance. At the sequel, we call the normalized QoC as QoC.

6.2. QoS Definition

The performance metric for the performance of the network is Quality-of-Service (QoS). QoS aims to fulfill the control system's requirements with minimum resources, so the other possible nodes in the system benefit from the remaining bandwidth. Another criterion for the QoS metric is satisfying the message set constraints. QoS metric is therefore measures the remaining bandwidth in the NCS after allocating the necessary resources to the control systems and the how much message set constraints are satisfied. Since our aim is to satisfy the negotiated bandwidth after scheduling the NCS, the other possible control or non-control communicating components will fulfill their (possibly real-time) requirements. This will provide a better NCS performance.

In this work we define the aggregation of bandwidth, delay and packet loss rate parameters and the remaining bandwidth as Quality-of-Service (QoS). We consider the design of the network in our performance evaluation algorithms given previously. We consider blocking probability as part of packet loss in our model. As presented in the introduction, previous works model allocated bandwidth as QoS in NCSs, since real network protocols are not considered. We model delay and packet loss as part of QoS in NCSs.

In [96], a vector-based representation of QoS metric is introduced. We consider this vector based representation in our model, and calculate the worst case values of bandwidth, delay and packet loss as a representation of QoS values. As a single parameter, we compute the worst case overall delay that is computed from the worst case values of the bandwidth, delay and packet loss. In our design we aim to be as close as possible to the negotiated service with the control systems, when we consider our bargaining based message set and message schedule design method. So, we compute the overall distance to the negotiated service values that are negotiated bandwidth and negotiated delay. We add the total distance with appropriate weighting factors that provides both distance metrics to be minimized. Finally, we compute negated exponential of this distance metric to make it as close as to the impulse function to guarantee feasibility.

We define QoS as follows:

Definition 3: *For a communication network, if the reserved bandwidth for the control systems is* bw_{negot} *after a scheduling process which is an iteration in our negotiation process is completed, QoS is defined as below:*

$$QoS = exp^{-(RateCost+MessageSetCost)}$$

where,

$$RateCost = \left|\frac{\sum_i Rate_i - bw_{negot}}{bw_{negot}}\right|^2$$

$$MessageSetCost = \sum_i \max\left(\frac{\left(\frac{Loss_i}{Rate_i} + Deadline_i\right) - \eta_{negot,i}}{\eta_{negot,i}}, 0\right)$$

and $\eta_{negot,i}$ *is the negotiated worst case overall delay. Above,* $Rate_i(Frequency_i)$*, Deadline*$_i$*, and Loss*$_i$ *are the message set parameters that the network provides in the worst case to control system i.* $RateCost$ *models the remaining bandwidth quality in QoS and* $MessageSetCost$ *models the service contract quality in QoS. In order to give the same emphasis,* $\max(RateCost) = \max(MessageSetCost)$.

The higher the value for the above QoS definition, the better is the satisfaction of the communication contract.

CHAPTER 7

PROPOSED APPROACH: INtERCEDE ALGORITHM

The *design of the NCS* has two components. Firstly the message set that is a set of parameters has to be constructed for the control systems. For this purpose, for every control system *i*, the *message parameters* (*$frequency_i$ $Deadline_i$, $Loss_i$*) have to be determined where *$Deadline_i$*, *$Loss_i$* denote the maximum values set by the control system for D_i and L_i respectively as defined in Section 2.3. Secondly, given the message parameters, a schedule S_i has to be computed to provide network service to control system *i*. ($Rate_i$, D_i ,L_i) are *network service parameters* which can be algorithmically computed for a message with *$frequency_i$* and S_i (See Appendix A). The schedule computation is governed by a certain *target bandwidth allocation* to conserve bandwidth for communication of nodes that have non-control functions as well as for the future extension of the system.

It is important here to note that the choice of the message parameters (*$frequency_i$*, *$Deadline_i$*, *$Loss_i$*) and the network service parameters ($Rate_i$, D_i and L_i) indirectly depend on each other and affect both QoC and QoS metrics as defined in Section 2.5. Hence the integrated NCS design is computationally intensive. We overcome this issue by separating the NCS design problem into control and network parts and employing a bargaining game approach designating these two parts as players. One of the most appealing features of the bargaining game is that the bargaining game [97], [98] has low computational complexity and a highly parametric structure to satisfy different NCS designers' demands such as a high quality control system response and/or high remaining bandwidth.

To this end, we propose "Integrated NCS Design (INtERCEDE: Integrated NEtwoRked Control systEm DEsign)" a novel *algorithmi*c approach for the design of NCS which aims to ensure the stability of the control system while achieving optimal values for the QoC and QoS metrics.

The core of INtERCEDE is the bargaining game with two players, *control player* and *network player* which act their moves according to nonzero-sum cooperative bargaining game rules. The control player aims to achieve an optimal QoC, such that the effect of the delay induced by the network on the system response is as small as possible. Accordingly, *the control player designs the message sets* for each control system *i*, by determining the message parameters, $frequency_i$, $Deadline_i$, and $Loss_i$. The network player aims to achieve an optimal QoS, such that bandwidth is efficiently allocated to the control player as close as possible to a target bandwidth. Accordingly, *the network player designs message schedules* S_i which in turn determines $Rate_i$, D_i and L_i for each control system *i*. Both players respect the constraints set by the other player when determining their own parameters. The bargaining game proceeds iteratively until the arrival at an agreement on the message set and the transmission schedule for the communication network. In each step of the bargaining, the message set and the corresponding schedule are updated according to the outcomes of the previous step. For both players of the game, the respective parameters are calculated to achieve an optimal QoC and QoS by using a genetic optimization algorithm to fulfill the constraints of the other player in the sense that all players are satisfied. The genetic algorithm choice is due to the fact that genetic optimization algorithms handle any kind of problems including nonlinear problems, converge relatively fast and produce good results for complex problems.

Our approach is applicable to any number of control systems. The analysis for multiple sensors considers the worst case of each sensor-to-controller message transmission and assumes the controller-to-actuators transmission is multicast.

In the remaining part of this section, we first provide the definitions of the parameters that are involved in the bargaining. Then we present computation of message set parameters and the corresponding schedule to achieve optimal QoC and QoS respectively. After having built the basis to all the players we present INtERCEDE Algorithm followed by a discussion of its merits with respect to the previous work.

7.1. Bargaining Game Model

Definition 9: Delay parameters for the bargaining

1. $\eta_{negot,i}(k)$ is t*he negotiated maximum network delay (NMND)* at the k^{th} step of bargaining. For a given message set, let *frequency*$_i$(*k*), *Deadline*$_i$(*k*) and *Loss*$_i$(*k*) denote the message set parameters for control system *i* at the k^{th} step of bargaining, respectively. Then $\eta_{negot,i}(k)$ is ;

$$\eta_{negot,i}(k) = \frac{Loss_i(k)+1}{frequency_i(k)} + Deadline_i(k) \ .$$

The NMND serves as a constraint for the network player that must be satisfied by the designed schedule. $\eta_{negot,i}(k)$ is initialized to some value and then updated during the bargaining step of INtERCEDE.

2. The *target maximum network delay* $t_{\eta,i}$ denotes the final value of $\eta_{negot,i}(k)$ that the control player aims to reach. It is used in updating $\eta_{negot,i}(k)$ as detailed in Appendix B. Note that this value is not mandatory to agree on, but simplifies the decisions for $\eta_{negot,i}(k)$ to converge to a desired value. $t_{\eta,i}$ is initialized to the desired value and kept constant during the run of the INtERCEDE. The desired value is chosen by the system designer to fulfill the NCS design criteria.
3. We define the *maximum reserved network delay* for i^{th} control system as $\eta_{\max,i} = \eta_i - \varepsilon$, where ε is a positive constant and η_i is the MADB to be satisfied for the stability test.

Definition 10: Bandwidth parameters for the bargaining

1. *The negotiated bandwidth* $bw_{negot}(k)$ is the total reserved transmission bandwidth for all of the messages and serves as a constraint for the control player for designing the message set.
2. The *target bandwidth* t_{bw} denotes the final value of $bw_{negot}(k)$ that the network player aims to reach. It is used in updating $bw_{negot}(k)$ as detailed in Appendix B. Similar to *target maximum network delay* this value is not mandatory to agree on, it simplifies the decisions for $bw_{negot}(k)$ to converge to a desired value.
3. $bw_{\max}$ is the physical transmission bandwidth for the communication network.

Below, some definitions use the parameter cr that is capacity ratio. Capacity ratio, cr, is less than 1, and determines the bandwidth demand from the network to satisfy sampling periods while the remaining bandwidth is to compensate for delays and losses in the transmission.

Definition 11: The model of the bargaining game

For the k^{th} step of the bargaining game of NCS design, the set of possible agreements MS_i for $i = 1, .., N$ where N is the number of control systems is defined as below while cr is the capacity ratio defined in the next section:

$$MS_i = \{Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k)\} :$$

$$\left| \sum_i frequency_i - cr \times bw_{negot(k)} \right| < \varepsilon$$

$$\sum_i Rate_{final,i}(k) \leq bw_{negot}(k) \leq bw_{max}$$

Constraint 1 Bargaining game: Bandwidth satisfaction

$$\frac{L_{final,i}(k) + 1}{Rate_{final,i}(k)} + D_{final,i}(k) \leq \eta_{negot,i}(k) \leq \eta_{\max,i}$$

Constraint 2 Bargaining game: Delay satisfaction

$$\left| \eta_{negot,i}(m) - \eta_{negot,i}(k) \right| < \varepsilon, \quad for\ m > k$$

$$\left| bw_{negot}(m) - bw_{negot}(k) \right| < \varepsilon, \quad for\ m > k$$

where $Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k)$ are the final proposed values of the network player for $Rate_i, D_i, L_i$ as defined in Section 2.3 under the schedule formed at the k^{th} bargaining step. These final values are achieved after an optimization as discussed in Section 7.3. A NCS design satisfying above inequalities return the set of MS_i containing the message set values for the control systems for ∀i and the schedule that fits to this message set. At the beginning of bargaining, the stability test is run once to find $\eta_{max,i}$.

7.2. Fair QoC Optimization and Message Set Construction

Let $QoC_{i,k}$ and QoS_k denote the values of QoC for control system *i* and QoS for *N* control systems at the bargaining iteration *k*.

Definition 12: QoC Optimization

Maximize:

$Min(w_i \times QoC_{i,k}(frequency_i(k), Deadline_i(k), Loss_i(k)))$

for i=1,…,N subject to:

$$\eta_{negot,i}(k) = \frac{Loss_i(k)+1}{frequency_i(k)} + Deadline_i(k)$$

Constraint 3: QoC NMND

$$\left|\sum_i frequency_i(k) - cr \times bw_{negot}(k)\right| < \varepsilon$$

Constraint 4: QoC negotiated bandwidth

$\sum_i w_i = 1$ and $w_i > 0$

The weights w_i in the QoC optimization enables providing service to the physical systems according to the user desired levels hence introducing a fairness. The weights are chosen by the NCS designer.

We call the optimal $(frequency_i(k), Deadline_i(k), Loss_i(k))$ triplet that solves the above problem $(frequency_{final,i}(k), Deadline_{final,i}(k), Loss_{final,i}(k))$. These parameters are calculated for all control systems and form the message set that will be negotiated with the network player.

7.3. QoS Optimization and Forming a Schedule

Definition 13: QoS Optimization

Maximize:

$QoS_k(schedule_k, bw_{negot}(k), \eta_{negot,i}(k))$

for $i = 1,2,\ldots, N$ subject to:

$$\frac{L_i(k)+1}{Rate_i(k)} + D_i(k) \le \eta_{negot,i}(k)$$

Constraint 5: QoS NMND

$$\left|\sum_i Rate_i(k) - bw_{negot}(k)\right| < \varepsilon$$

Constraint 6: QoS Bandwidth

Where ε is a very small number.

The QoS function measures how closely the service parameters satisfy the contract between the control nodes and the network, and how much the message set constraints are satisfied. As QoS value is higher, the designed schedule is better.

The optimal values are $Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k)$ for the control system i for the given constraints.

The optimal schedule for a given message set is computed by a genetic algorithm which proceeds as follows:

```
Inputs: Fitness Function plus population of schedules of size PS (schedule_population)
Initalization: new_schedule_population ← empty set,
schedule_population←initial_population
1) For 1 to PS
        a. X ← RandomScheduleSelection(schedule_population, fitness_function)
        b. Y ← RandomScheduleSelection(schedule_population, fitness_function)
        c. Child ← ReproduceSchedule (X,Y)
        d. If (small_probability) then Child ← MutateSchedule(Child)
        e. Add Child to new_schedule_population
2) schedule_population ← new_schedule_population
3) If (schedule_population) is fit enough Goto step 6, Else Goto step 2
4) Return the best schedule in schedule_population according to fitness_function
```

Definition 14 Genetic Algorithm Features

1. **fitness_function:** The fitness function is the QoS in **Definition 8**) where $Rate_i$ and $Delay_i$ are computed for each control system's schedule using the algorithms in Appendix B in the population. For the fitness_function, larger the value of QoS(schedule), the better the fitness.

2. **RandomScheduleSelection:** The schedules will be chosen randomly with this function by giving higher probability to the schedule with higher fitness value.
3. **ReproduceSchedule:** A uniformly random point will be chosen in the schedule, and the two schedules X and Y will produce a child by taking the slot schedule up to the random point of schedule X, and appending the rest of the slot schedule from the random point to the end of slot schedule Y.
4. **MutateSchedule:** A random slot in the schedule will be chosen and it will be replaced by a random control node that has message to transmit.

When the schedule computation is finalized corresponding $Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k)$ will be computed and returned as the network service parameters for each control node *i*.

7.4. INtERCEDE Algorithm

The INtERCEDE algorithm is the iterative computation of the message parameters (*frequency$_i$*, *Deadline$_i$*, *Loss$_i$*) and the network service parameters (*Rate$_i$*, *D$_i$* and *L$_i$*). To this end, $\eta_{negot,i}(k)$ and $bw_{negot}(k)$ that are derived from these parameters are progressively updated in the bargaining step of INtERCEDE that is executed in each iteration [18]. Each iteration *k* of the algorithm aims to improve the NCS design to an optimal point according to both QoC and QoS metrics defined in Section 2.5. The bargaining game between the control player and the network player proceeds until an agreement on the message parameters of the control systems and the service parameters for the communication network is achieved.

We present the flowchart of the INtERCEDE in Figure 16. The steps in the flowchart are explained below.

(1) Initialization

For each control system *i*, the stability test is performed as described in Section 2.5 to determine η_i. The negotiation parameters are initialized to $\eta_{negot,i}(1)$ and $bw_{negot}(1)$ for *k*=1. In addition the fixed parameters $bw_{\max}$, $\eta_{\max,i}$, $t_{\eta,i}$ t_{bw} are set. $\eta_{negot,i}(1) < t_{\eta,i}$ and $bw_{negot}(1) < t_{bw}$ are selected. The bargaining step brings the negotiated values close to target values in each iteration of INtERCEDE.

(2) Compute message set parameters

The control player computes the optimal $(frequency_{final,i}(k), Deadline_{final,i}(k), Loss_{final,i}(k))$ for each control system *i* as described in Section 7.2.

(3) Is the message set feasible?

For each control system i, the control player checks if $(frequency_{final,i}(k), Deadline_{final,i}(k), Loss_{final,i}(k))$ satisfy Constraint 3 and Constraint 4. Furthermore, for each control system *i* the control player checks if the following constraint holds to ensure stability:

$$\frac{Loss_{final,i}(k)+1}{frequency_i(k)} + Deadline_{final,i}(k) \le \eta_{\max,i} = \eta_i - \varepsilon$$

Constraint 7: Final service parameters feasibility

(4) Compute message schedule

If all control systems i provide message parameters that satisfy Constraint 3, Constraint 4 and Constraint 7 then the network player computes S_i corresponding and for each of these control systems. The network service parameters corresponding to S_i are computed using the algorithms in Appendix A. $(Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k))$ are the optimal values for the network service parameters achieved as described in Section 7.3.

(5) Is the message schedule feasible?

The network player checks if $(Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k))$ satisfy **Constraint 5** and **Constraint 6** as described in Section 7.3 for each control system *i*.

(6) Update message set

QoC for each control system *i* is computed using the respective network service parameters supplied by network player. These updated values represent the offer of the network player in the bargaining. The game proceeds until control player also agrees on these values.

(7) Bargaining

The details of the bargaining game are presented in Appendix B. Bargaining game step of the INtERCEDE algorithm is executed by both the control player and the network player. Briefly, the inputs to the bargaining at iteration *k* are:

- Negotiated parameters $bw_{negot}(k), \eta_{negot,i}(k)$ for all *i*
- $Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k)$
- $QoC_{i,k}(Rate_{final,i}(k), D_{final,i}(k), L_{final,i}(k))$,

  $QoS_k(schedule_k, bw_{negot}(k), \eta_{negot,i}(k))$
- First and second derivatives of $QoC_{i,k}$:

  $\frac{\partial}{\partial k} QoC_{i,k} = QoC_{i,k-1} - QoC_{i,k-2}$, $\frac{\partial}{\partial^2 k} QoC_{i,k} = \frac{\partial}{\partial k} QoC_{i,k-1} - \frac{\partial}{\partial k} QoC_{i,k-2}$
- First and second derivatives of QoS_k:

$$\frac{\partial}{\partial k}QoS_k = QoS_{k-1} - QoS_{k-2}, \quad \frac{\partial}{\partial^2 k}QoS_k = \frac{\partial}{\partial k}QoS_{k-1} - \frac{\partial}{\partial k}QoS_{k-2}$$

- Fixed parameters: $bw_{\max}, \eta_{\max,i}, t_{\eta,i}$ t_{bw}

The outputs are $bw_{negot}(k+1), \eta_{negot,i}(k+1)$ for all i which are obtained as follows:

$\eta_{negot,i}(k+1) = \eta_{negot,i}(k) + \delta\eta_{negot,i}(k+1)$

$bw_{negot}(k+1) = bw_{negot}(k) + \delta bw_{negot}(k+1)$

where $\delta\eta_{negot,i}(k+1)$ and $bw_{negot}(k+1)$ are the update values for the next step of bargaining game for control player and network player, respectively. The computation of these values depends on $n_i(k+1)$ and $b(k+1)$ which are the *step sizes* for the control player's control system i and network player, respectively. Note that in standard 2-player bargaining game players determine their limits and the targets over one specific commodity (parameter that the bargaining is proceeded on). On the contrary INtERCEDE bargaining is carried out over two different commodities (delay and bandwidth) which depend on each other. The details for determining are presented in Appendix B.

(8) Step Sizes Converged?

The step sizes are initiated from 0 and they change during the iterations of INtERCEDE algorithm. The algorithm stops when $n_i(k)$ and $b(k)$ converge to 0. Later in our simulation study in Chapter 8 we demonstrate the change of the step sizes as the bargaining proceeds in Figure 17 and Figure 18 respectively. Note that, it is possible that the step sizes do not converge if no feasible message parameters and service parameters can be computed as the bargaining proceeds. Currently, we implement a heuristic decision which decides that the step sizes will not converge when number of iterations exceed 50 and step sizes are still not below 0.4.

(9) Continue?

If the step sizes have not converged yet and the iteration number is less than 50, we continue to iterate the bargaining. The next iteration begins with the control player and control player calculates new message set and form its next proposal. If the iteration number exceeded the maximum value, we return failure of the algorithm.

(10) Successful Return

The bargaining arrives at an agreement when the control player accepts the service parameters of the network. This agreement indicates that the maximum network delay satisfies the desired value of the control player that is computed considering QoC as follows:

$$Delay_{final,i} = \frac{L_{final,i} + 1}{Rate_{final,i}} + D_{final,i} < \frac{Loss_{final,i} + 1}{frequency_{final,i}} + Deadline_{final,i}$$

Constraint 8: Agreement constraint

The final values are adopted for the message set parameters and network service parameters.

(11) Unsuccessful Return

No feasible schedule could be constructed satisfying the desired QoC and QoS.

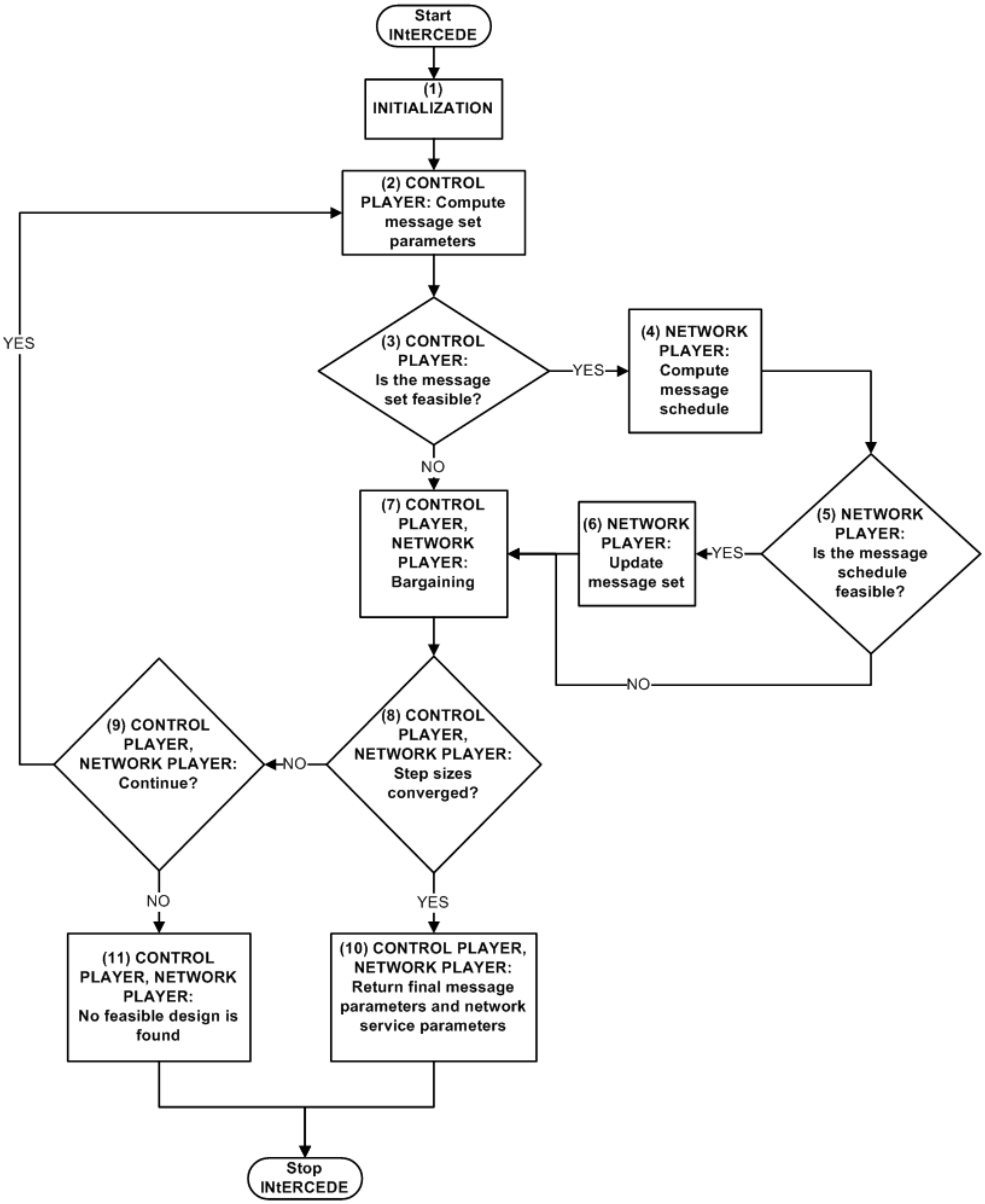

Figure 16. INtERCEDE Algorithm Flow Chart.

7.5. Novelty and Comparison with Previous Work

A performance comparison of INtERCEDE and other works is in Section 8.3. For a comparison of design principles of INtERCEDE with other works, we add the novelties of INtERCEDE in this section.

INtERCEDE is a novel approach to NCS design which integrates control system performance metrics and network performance metrics by a bargaining game which enables parameterization of these metrics according to the system requirements [18]. INtERCEDE considers network protocol dynamics with realistic assumptions that are not considered in previous integrated control and scheduling works [19], [21], [23], [61], [63], [62], [59], [64], [20], [60]. It is based on a time-slotted realistic network assumption and our results are directly applicable to existing protocols such as TTP/C [7] and the static segment of FlexRay [12]. INtERCEDE is an offline method, which does not require infeasible amount of computing resources such as [61], [63], [62], [19], [21], [23], [20], [60].

A very important advantage of INtERCEDE is that it does not depend on any specific controller design method as in [19], [21], [23], [61], [63], [62], [59], [64], [20], [60]. In simulation study, we assume a specific type of controller, but, in general, any pre-designed controller method with its own stability test can be integrated to INtERCEDE. Since we eliminate controller design, the NCS design has reduced computational complexity, but still the control system performance is taken into consideration in the bargaining design approach. The solution of the design are one of the many possible solutions in which control systems are stable and communication network schedules are feasible, which satisfy user demands. We design not only a feasible system, but also one that is optimized at the user desired level.

CHAPTER 8

SIMULATION-BASED STUDY of INtERCEDE

8.1. Simulation set-up

We investigate the performance of the INtERCEDE algorithm by a simulation study of a NCS that is composed of 4 control systems communicating over a time slotted network. The related parameters for each control system are computed assuming a Linear Quadratic Regulator (LQR) controller as a static control law and listed in Table 4. The control system parameters are chosen to simulate an NCS, of which control systems require different bandwidths in order to reach the fair QoC distribution of resources as defined in Section 7.2 where $w_1 = w_2 = w_3 = w_4$. Control systems 1 and 2 represent systems with tight delay bounds to ensure stability, hence they require relatively higher bandwidth with respect to control systems 3 and 4 with larger MADB. Note that TDMA network access isolates the communication of different control systems from each other. Hence, the 4 control systems in our simulation represent 4 different types of control systems with different parameters and requirements. In Chapter 6, we used 5 systems to show the calculation of QoC performance metric. In simulation study, 4 of the control systems that are enough for showing different bandwidth requirements of the control systems are chosen.

Table 4. Control system parameters for the simulation.

Control System (*i*)	A_i	B_i	C_i	D_i	$\eta_{\max,i}$
1	[6 -3;0.1 -0.02]	[1; 0.4]	[1 0]	0.1	0.0894
2	[-2 8;-1 3]	[-1; 2]	[0 1]	1.3	0.1363
3	[-0.03 0.02;0.02 0.09]	[-0.4; 0.3]	[1 0.2]	2	3.8400
4	[-0.08 -0.004;0.1 0.02]	[0.88;0.3]	[1 1]	0.3	1.4400

The slot duration is $T_{slot} = 600\mu\, sec$ which leads to an NCS where initially no feasible schedule can be found. Hence, how INtERCEDE achieves the schedule iteratively through bargaining is demonstrated. Note that the number of bits that can be transmitted per slot determines the network bandwidth. Assuming that the underlying slotted network is 10Mbps Ethernet, this slot duration 750Bytes/slot. This number is quite high for only sensor samples and the corresponding responses from the controllers. In our model for the NCS we assume that there is additional traffic that is not related to the control applications. Hence, the size of the data produced by these applications should be taken into consideration when determining the number of bits per slot. If the non-control application generates large data frames, slots that are big enough to accommodate these frames without fragmentation simplifies the operation of the network. However, increasing the slot size, while keeping the bandwidth constant decreases the number of slots/second which in turn decreases the schedulability of the system.

The simulation is carried out using MATLAB running on a Windows computer with 2.4 GHz QuadCore Processor and 4 GB RAM. One simulated run of INtERCEDE on the NCS described above lasts approximately one hour.

The control system performance measure metric in Chapter 6 Section 1, QoC, depends on the calculation of the ideal step response measure that is one of the main conventional performance measure methods for control systems in literature [27]. Since QoC depends on this method, the simulation results are valid in terms of measuring control system performance. The network performance metric in Chapter 6 Section 2, QoS, depends on necessary condition evaluating algorithms given in Appendix A. Since QoS metric depend on the basic scheduling literature [99], [55], [100], the simulation results are valid in terms of measuring network performance.

8.2. Results: Control System and Network Performance

The run of INtERCEDE algorithm for the four control systems return the following message parameters and the network service parameters displayed in Table 5 and Table 6. Total number of slots in the periodic cycle is 30, and the simulation is complete after 30 bargaining iterations.

Table 5. Message parameters determined by INtERCEDE.

Control System (i)	$frequency_{final,i}$ (Hz)	$Deadline_{final,i}$ (sec)	$Loss_{final,i}$ (msgs)	$\eta_{final,i}$	$\eta_{\max,i}$
1	45.5593	0.0073	0	0.0292	0.0894
2	41.9310	0.0105	0	0.0344	0.1363
3	1.9579	0.1702	0	0.6810	3.8400
4	4.4922	0.1207	0	0.3433	1.4400

Table 6. Network service parameters determined by INtERCEDE.

Control System (i)	$Rate_{final,i}$ (msg/sec)	$D_{final,i}$ (sec)	$L_{final,}$ (msgs)	$Delay_{final,i}$ (sec)	S_i (slot number)
1	70.4512	0.0138	0	0.0280	1, 5, 12, 14, 18, 30
2	124.5184	0.0078	1	0.0239	2, 9, 15, 16, 20, 25, 29
3	29.0816	0.0010	13	0.4824	23, 24
4	54.0672	0.0169	11	0.2389	13, 28

In the tables above;

$$Delay_{final,i} = \frac{L_{final,i}+1}{Rate_{final,i}} + D_{final,i}$$

$$\eta_{final,i} = \frac{Loss_{final,i}+1}{frequency_{final,i}} + Deadline_{final,i}.$$

In Table 5, control systems have low frequencies that have no loss durability due to the conservative choice of capacity ratio. We observe in Table 6, the conservative choice of capacity ratio is compensated by assigning positive loss values. As expected, first two control systems are assigned high number of slots due to their high resource requirements. Also, all systems satisfy the constraints of control player and network player.

The following Figure 17 demonstrates the progressive updates of the negotiated maximum network delay $\eta_{negot,i}(k)$ with respect to the number of iterations that INtERCEDE algorithm runs for $i = 1,2,3,4$. For each control system the respective target maximum network delay $t_{\eta,i}$ and maximum reserved network delay $\eta_{\max,i}$ are also plotted. For $i = 1,2,3,4$, $\eta_{negot,i}(k)$ converges to a value that is correlated to $t_{\eta,i}, \eta_{\max,i}$. We observe that the bargaining achieved a final agreement that is closer to the target value than the reserved value. $t_{\eta,i}$ is chosen high in order to present the fairness concept of QoC optimization. The convergence of 4th control system is observed not to violate fair distribution of resources. Similarly, $bw_{negot}(k)$ converges to a value that is closer to t_{bw} than $bw_{\max}$ as seen in Figure 18. Figure 18 shows that bandwidth is not sacrificed so much that negotiated value converges to the target value.

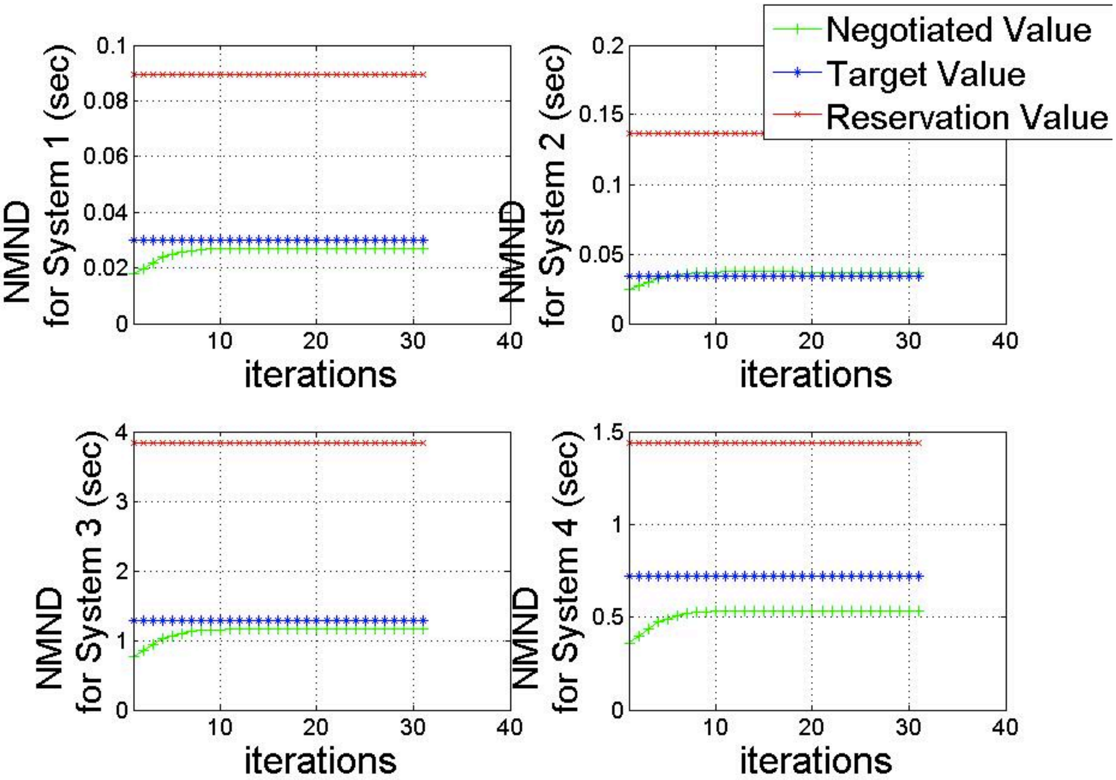

Figure 17. Converging of $\eta_{negot,i}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $\eta_{negot,i}(k), t_{\eta,i}, \eta_{\max,i}$ respectively.

Next, we investigate the desired QoC levels of the control player expressed in message parameters which are (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$) in comparison to the achieved QoC levels from the network expressed in network service parameters (*Rate*$_i$, D_i and L_i) in Figure 19.

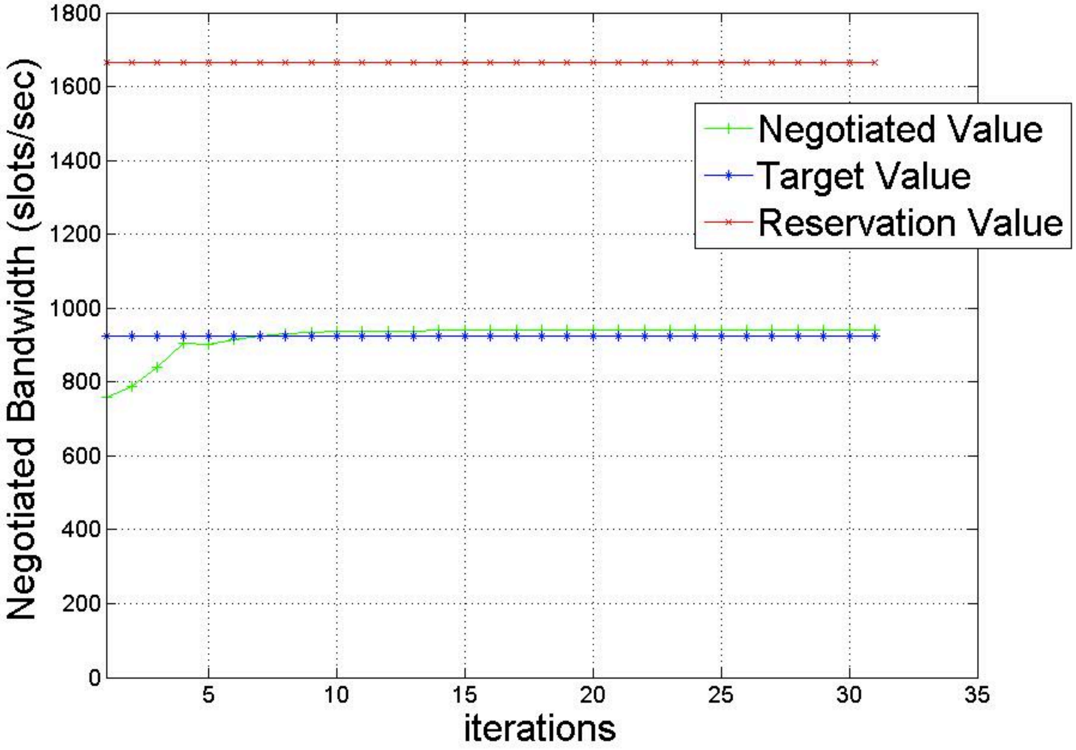

Figure 18. Converging of $bw_{negot}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $bw_{negot}(k)$, t_{bw}, $bw_{\max}$ respectively.

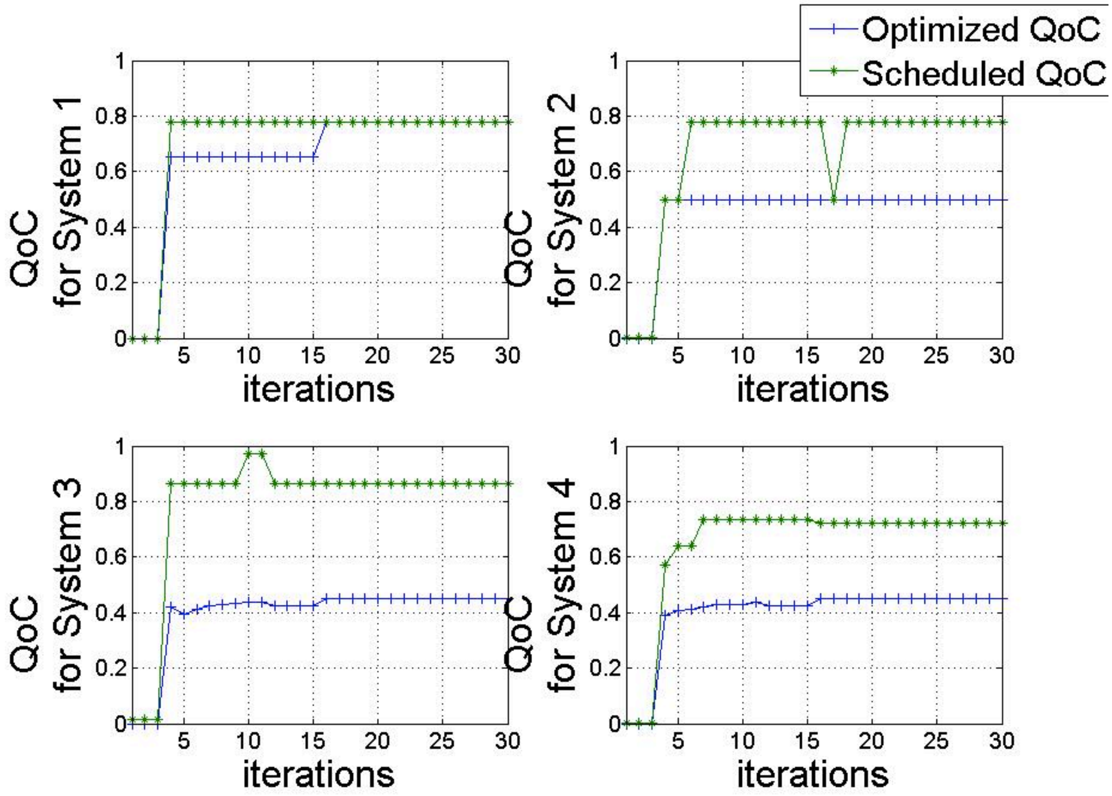

Figure 19. Optimized Control System's QoC (computed according to (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$)) and Scheduled QoC (computed according to (*Rate*$_i$, D_i and L_i))

Our first observation for Figure 19 is related the perturbations in the QoC curves and the convergence. The control player's and network player's respective QoC and QoS computations are carried out by iterative optimizations. We restricted the number of iterations in these optimizations to fixed numbers to decrease the runtime of the INtERCEDE. This restricted optimization may produce suboptimal results leading to perturbations in the control player's and network player's output. However at the end of the bargaining the results converge. The QoC values of the network player for control system 3 and 4 are higher respect to that of the control player due to their low resource demands with respect to the other control systems.

Next we investigate the achieved QoC values in Figure 19. Control systems 1 and 2 have tighter MADB requiring more bandwidth compared to control systems 3 and 4. Hence the control player's demands for systems 1 and 2 are harder to meet for the network player. This leads to an agreement at a relatively closer QoC value for these systems with respect to control systems 1 and 2. Since the control system 3 is the least bandwidth demanding control system, the control system 3 has significant difference between control player's demand and network player's proposal than the other systems in network resource distribution.

The bargaining finishes when Constraint 8 is satisfied. Accordingly, the control player is expected to get higher QoC values than its demands. However satisfying Constraint 8 does not necessarily achieve ($frequency_i$,= $Rate_i$), ($Deadline_i$= D_i), ($Loss_i$= L_i). Furthermore, the relation between the message parameters ($frequency_i$, $Deadline_i$, $Loss_i$), network service parameters ($Rate_i$, D_i and L_i) and QoC as in Definition 7 is not linear. So, there are differences between the resource demands and proposed resources with respect to the control system performances. But, as a desirable solution, control systems receive significantly or gently higher performance than they demand.

Figure 20 shows the percentage remaining bandwidth of the communication network as bargaining proceeds with the value of the QoS metric. The remaining bandwidth curve has a discrete form because QoS is calculated in terms of the slots in the schedule. The design of the bargaining leads to gradual allocation of bandwidth to the control systems in each iteration. Note that a general practice in design of networks for real-time embedded systems is a conservative bandwidth assignment in one step which leads to inefficient use of bandwidth resources. In contrast, INtERCEDE bargaining aims to allocate bandwidth that is just enough to provide the required service for desired QoC. Since there are plenty of resources for the network player, the bandwidth sacrifice for the network player is small.

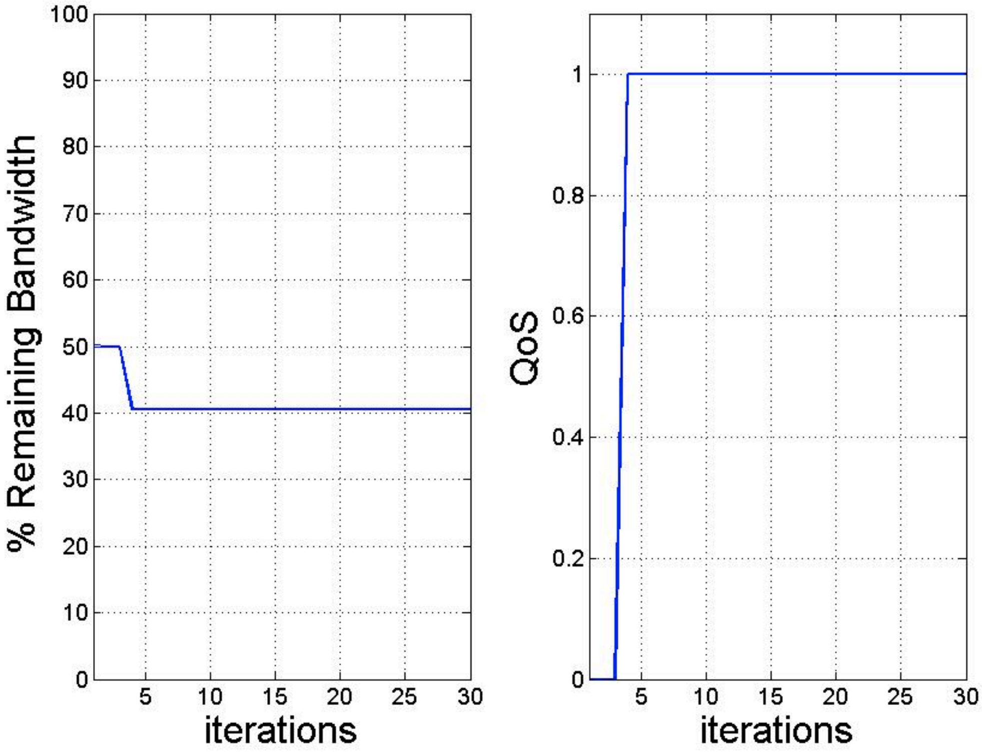

Figure 20. Remaining bandwidth after allocation to control systems and the change of QoS.

The quick convergence of the QoS performance shows that the network resources are plenty enough to satisfy control systems' demands.

The simulation study is repeated for two other case studies. In the first case, the simulation is run for 4 control systems which require higher bandwidth and comprised of first two control systems and their replicated two others:

Indices 1,3: Control System 1

Indices 2,4: Control System 2

This case represents a NCS with control systems requiring higher bandwidth for message scheduling. Other simulation parameters are the same with the parameters given in Table 15. Specific bargaining parameters. The second case to be simulated is a NCS that require lower bandwidth for message scheduling. The 3rd and 4th control systems in Table 4 and their two replicated ones are simulated to see the progress of bargaining:

Indices 1,3: Control System 3

Indices 2,4: Control System 4

Similarly, the same bargaining parameters are used as in other cases.

The run of INtERCEDE algorithm for the four control systems for case 1 return the following message parameters and the network service parameters displayed in Table 7 and Table 8. Total number of slots in the periodic cycle is 30, and the simulation is complete after 30 bargaining iterations.

Table 7. Message parameters determined by INtERCEDE for case 1.

Control System (i)	$frequency_{final,i}$ (Hz)	$Deadline_{final,i}$ (sec)	$Loss_{final,i}$ (msgs)	$\eta_{final,i}$	$\eta_{\max,i}$
1	27.1168	0.01956	0	0.0564	0.0894
2	39.2266	0.00858	0	0.0341	0.1363
3	23.8208	0.0159	0	0.0586	0.0894
4	42.0157	0.011	0	0.0348	0.1363

Table 8. Network service parameters determined by INtERCEDE for case 1.

Control System (i)	$Rate_{final,i}$ (msg/sec)	$D_{final,i}$ (sec)	$L_{final,}$ (msgs)	$Delay_{final,i}$ (sec)	S_i (slot number)
1	77.0048	0.0126	1	0.0433	2, 4, 13, 15, 26
2	82.7392	0.0112	1	0.0538	5, 7, 17, 19, 22, 24, 25, 27
3	49.1520	0.0151	1	0.0441	9, 11, 16, 21
4	64.7168	0.0150	0	0.0535	3, 6, 8, 10, 23, 28, 30

In Table 7, control systems have lower frequencies that result in no loss durability due to increasing bandwidth demand in the system. Since all four control systems require higher bandwidth, more sacrifice occurs in the negotiation. All systems satisfy the constraints of control player and network player.

Figure 21 shows the converging of the progressive updates of the negotiated maximum network delay $\eta_{negot,i}(k)$ with respect to the number of iterations. The negotiated values have more sacrifice in bargaining due to the increasing demand for bandwidth of the 3rd and 4th control systems.

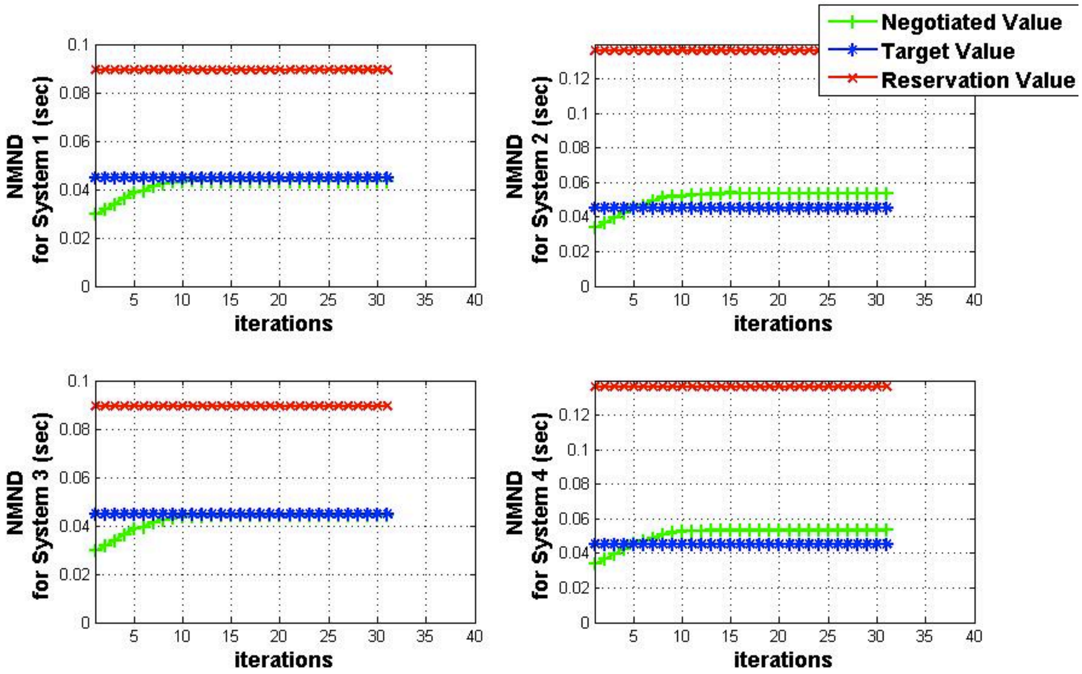

Figure 21. Converging of $\eta_{negot,i}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $\eta_{negot,i}(k), t_{\eta,i}, \eta_{\max,i}$ respectively for case 1.

Figure 22 demonstrates the progress of the negotiated bandwidth with respect to the bargaining iterations. Since the bandwidth demand in the system increases with respect to the previous simulation case, more bandwidth is sacrificed. The bargaining keeps the performance of the control systems by sacrificing more bandwidth.

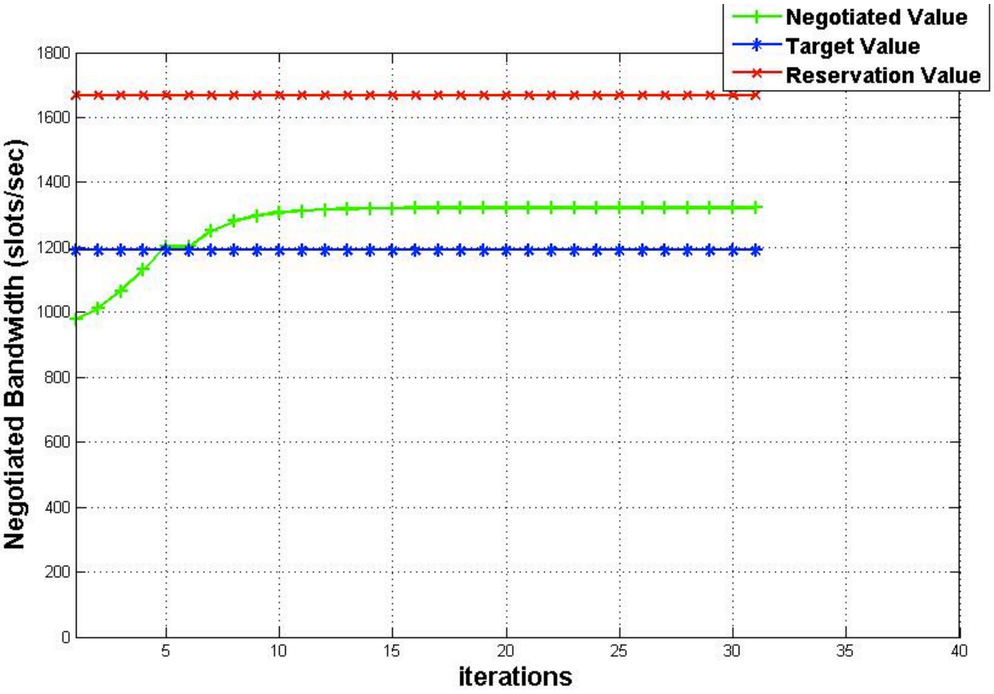

Figure 22. Converging of $bw_{negot}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $bw_{negot}(k), t_{bw}, bw_{\max}$ respectively for case 1.

Figure 23 demonstrates QoC performance of the control systems. The figure shows that control system 1 has lower QoC served by the network player than the QoC requested by the control player. The rate, delay and loss parameters are feasible in Table 8, but due to the discrete nature of the assigned slots, QoC served by the network player is less than the optimized QoC. Another reason is the numeric problems of the optimization algorithm, since we observe that QoC served to the other control nodes are strictly followed by the network player.

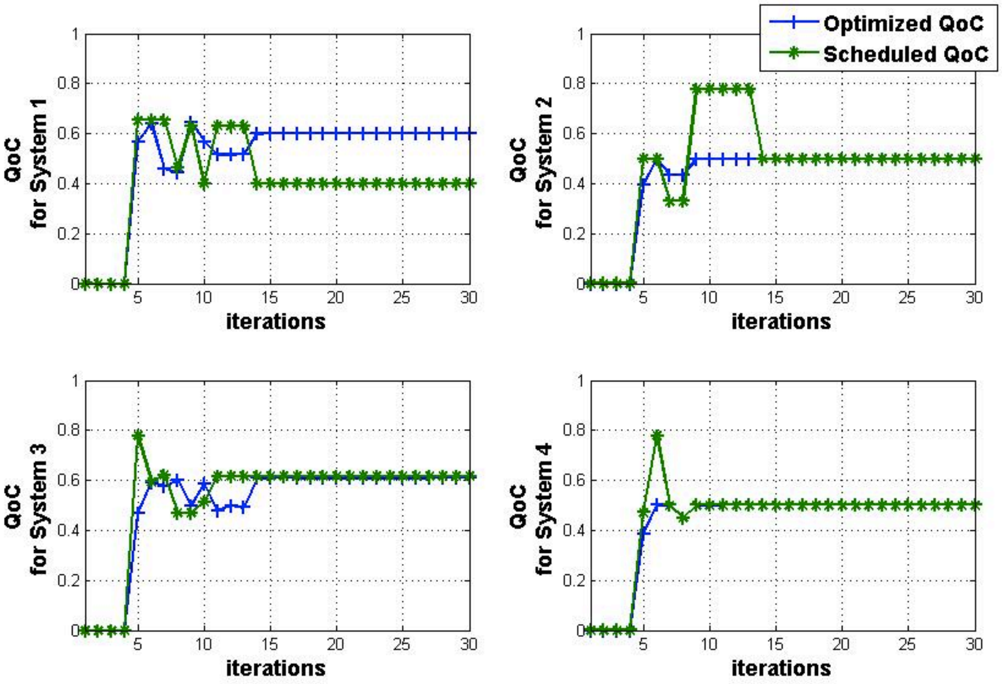

Figure 23. Optimized Control System's QoC (computed according to (*frequency$_i$*, *Deadline$_i$*, *Loss$_i$*)) and Scheduled QoC (computed according to (*Rate$_i$*, *D$_i$* and *L$_i$*)) for case 1.

Figure 24 demonstrates the QoS patterns of the bargaining against the bargaining iterations. We observe the effects of the same increased bandwidth demand by the control systems.

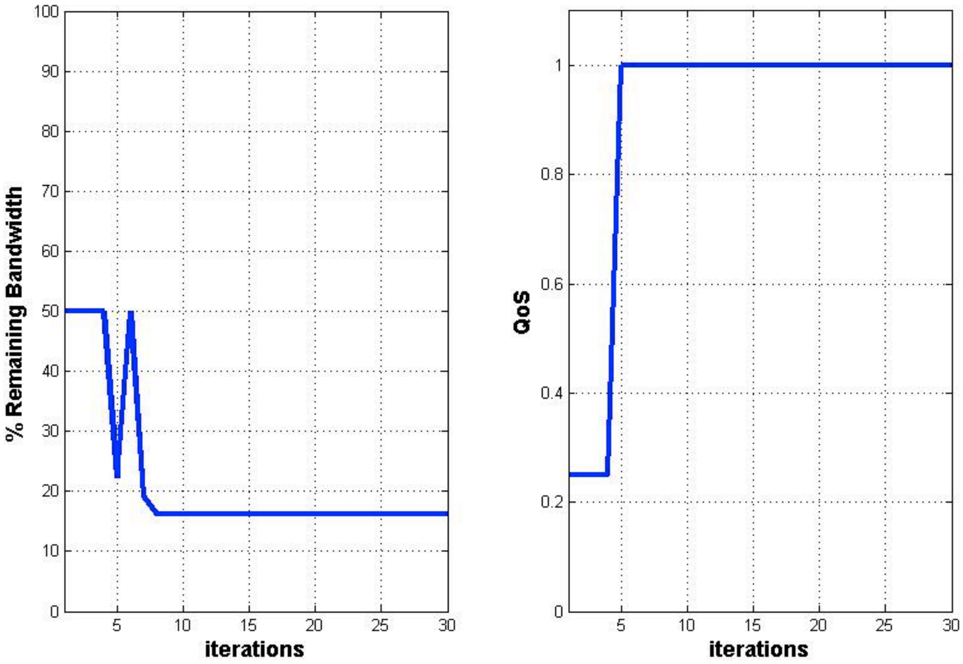

Figure 24. Remaining bandwidth after allocation to control systems and the change of QoS for case 1.

The run of INtERCEDE algorithm for the four control systems for case 2 return the following message parameters and the network service parameters displayed in Table 9 and Table 10. Total number of slots in the periodic cycle is 30, and the simulation is complete after 30 bargaining iterations.

Table 9. Message parameters determined by INtERCEDE for case 2.

Control System (i)	$frequency_{final,i}$ (Hz)	$Deadline_{final,i}$ (sec)	$Loss_{final,i}$ (msgs)	$\eta_{final,i}$	$\eta_{\max,i}$
1	7.9179	0.0509	4	0.6824	3.8400
2	31.3096	0.0132	10	0.3645	1.4400
3	5.5033	0.0937	2	0.6388	3.8400
4	29.3055	0.0136	10	0.3889	1.4400

Table 10. Network service parameters determined by INtERCEDE for case 2.

Control System (i)	$Rate_{final,i}$ (msg/sec)	$D_{final,i}$ (sec)	$L_{final,}$ (msgs)	$Delay_{final,i}$ (sec)	S_i (slot number)
1	39.3216	0.0100	23	0.7977	14, 26, 28
2	39.3216	0.0100	12	0.3891	4, 5, 21, 25
3	47.5136	0.0144	24	0.8184	16, 23, 29
4	64.7168	0.0045	23	0.4073	2, 3, 12, 19

In Table 9, control systems have higher frequencies that result in high loss durability due to plenty of bandwidth in the system. Since all four control systems require lower bandwidth, network player can reserve bandwidth for future extensibility of the NCS. All systems satisfy the constraints of control player and network player.

Figure 25 shows the converging of the progressive updates of the negotiated maximum network delay $\eta_{negot,i}(k)$ with respect to the number of iterations. The negotiated values need no sacrifice in bargaining due to the low bandwidth demand of the 1st and 2nd control systems.

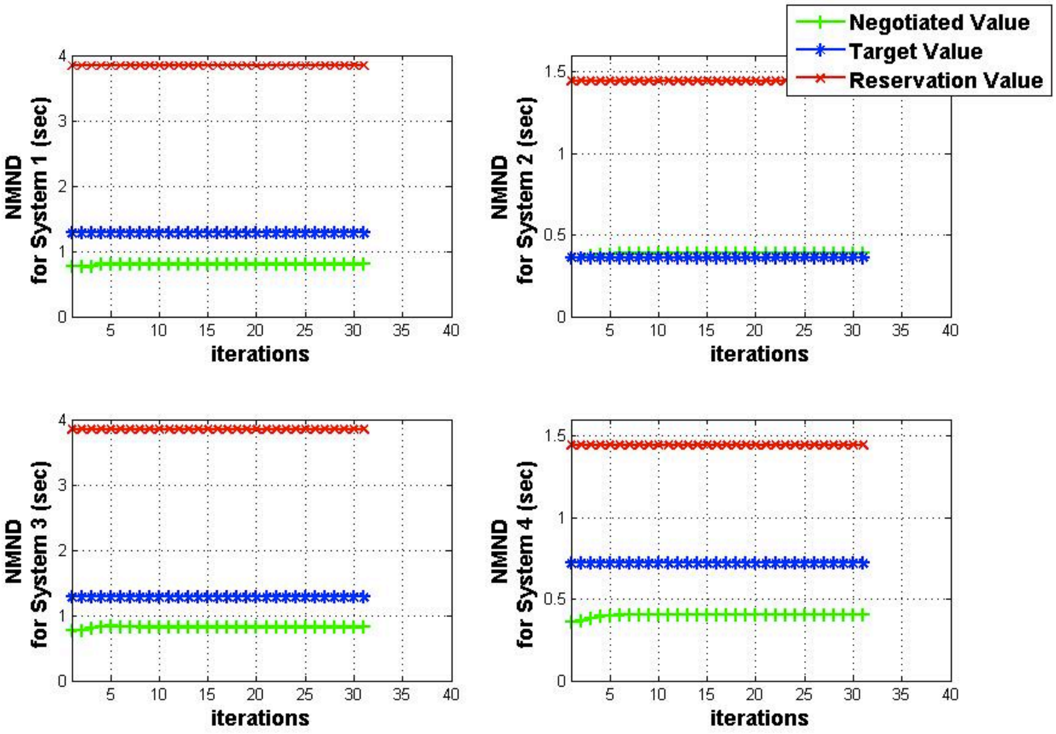

Figure 25. Converging of $\eta_{negot,i}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $\eta_{negot,i}(k), t_{\eta,i}, \eta_{\max,i}$ respectively for case 2.

Figure 26 demonstrates the progress of the negotiated bandwidth with respect to the bargaining iterations for case 2. Since the bandwidth demand in the system by the control nodes decreases with respect to the first simulation study, negotiated bandwidth decreases as the negotiation progresses.

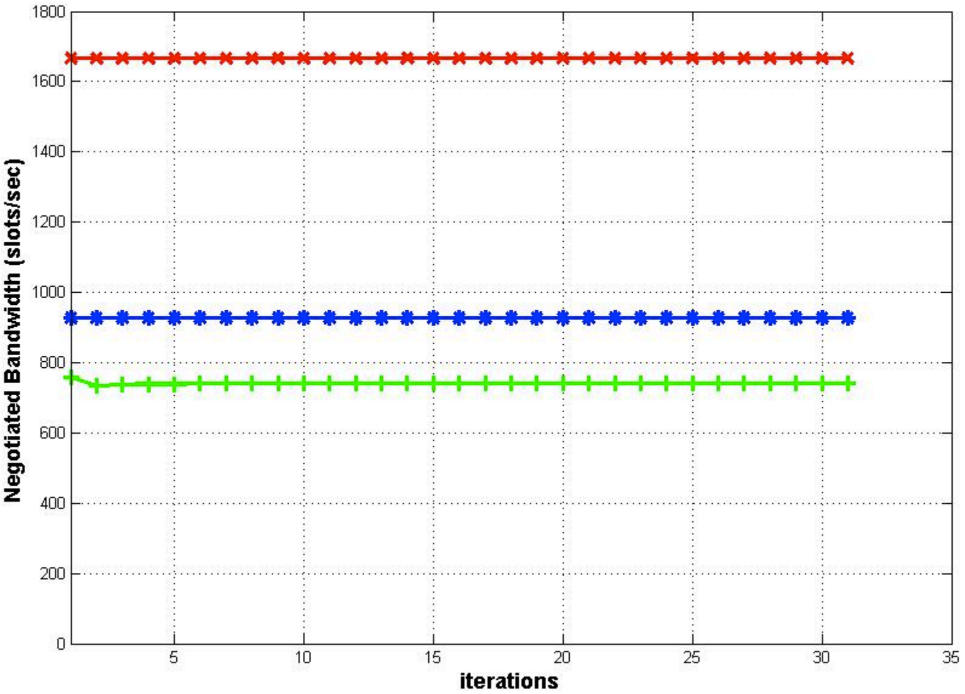

Figure 26. Converging of $bw_{negot}(k)$ with iterations, *k*. Negotiated Value, Target Value, Reservation Value show $bw_{negot}(k)$, t_{bw}, $bw_{\max}$ respectively for case 2.

Figure 27 demonstrates QoC performance of the control systems for case 2. All control systems are sufficiently served by the network player corresponding to the demands of the control player. We observe that while all control systems' performance are preserved, some bandwidth is saved for future extensibility.

Figure 28 demonstrates the QoS patterns of the bargaining against the bargaining iterations. We observe the fast convergence in the QoS pattern by finding a good schedule quickly due to the low bandwidth demand of the control systems.

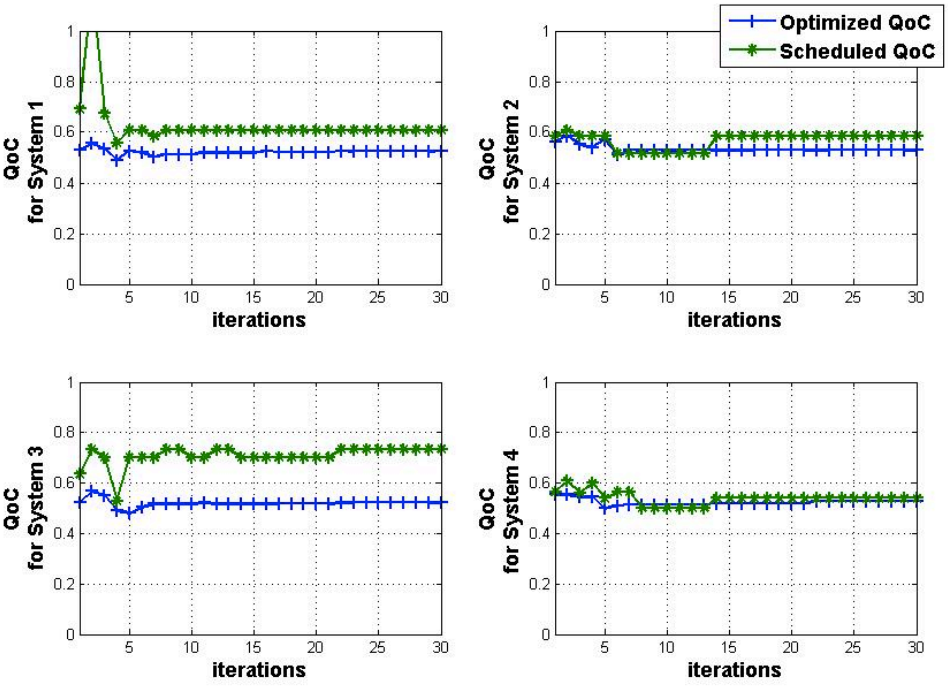

Figure 27. Optimized Control System's QoC (computed according to (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$)) and Scheduled QoC (computed according to (*Rate*$_i$, D_i and L_i)) for case 2.

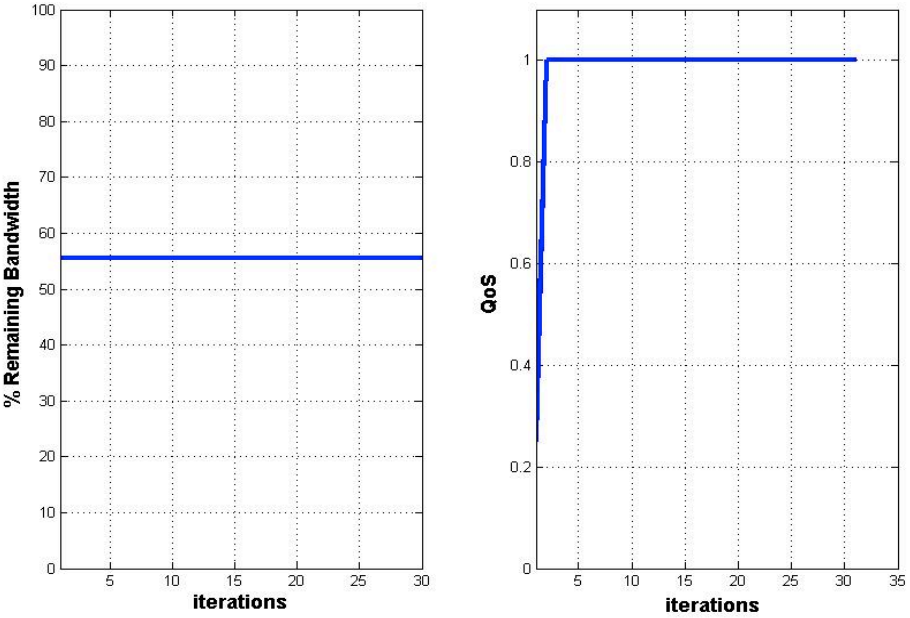

Figure 28. Remaining bandwidth after allocation to control systems and the change of QoS for case 2.

INtERCEDE algorithm is run for a case where all parameters are the same with the initial simulation study except the bandwidth is reduced to demonstrate the performance behavior of the same control systems under lower bandwidth. The run of INtERCEDE algorithm for this case return the following message parameters and the network service parameters displayed in Table 11 and Table 12. Total number of slots in the periodic cycle is 30, and the simulation is complete after 30 bargaining iterations.

Table 11. Message parameters determined by INtERCEDE for case 3.

Control System (i)	$frequency_{final,i}$ (Hz)	$Deadline_{final,i}$ (sec)	$Loss_{final,i}$ (msgs)	$\eta_{final,i}$	$\eta_{\max,i}$
1	56.3373	0.0076	0	0.0253	0.0894
2	27.0587	0.0201	0	0.0571	0.1363
3	1.4472	0.228	0	0.9188	3.8400
4	3.0539	0.1715	0	0.4989	1.4400

Table 12. Network service parameters determined by INtERCEDE for case 3.

Control System (i)	$Rate_{final,i}$ (msg/sec)	$D_{final,i}$ (sec)	$L_{final,}$ (msgs)	$Delay_{final,i}$ (sec)	S_i (slot number)
1	100.7616	0.0094	0	0.0372	1, 3, 4, 5, 9, 10, 13, 14, 15, 20, 23, 25, 26, 27
2	54.0672	0.0180	0	0.0480	7, 11, 12, 17, 21, 22, 24, 28
3	25.1904	0.0193	16	1.5990	8, 16, 18
4	25.1904	0.0193	7	0.7314	19, 29

In Table 11, control systems have numerically close frequencies to the first simulation study. However, in this case bandwidth is limited due to the 1 msec slot time compared to the 0.6 msec slot time in the first simulation study. This constraint is reflected into the behavior of the network player as in Table 12, and the communication slots are assigned to the control systems in increased number.

Figure 29 shows the converging of the progressive updates of the negotiated maximum network delay $\eta_{negot,i}(k)$ with respect to the number of iterations. The negotiated values sacrifice in bargaining significantly compared to the first simulation case due to the lower bandwidth in the NCS.

Figure 30 demonstrates the progress of the negotiated bandwidth with respect to the bargaining iterations for case 3. Since the total bandwidth in the system is decreased with respect to the first simulation study, most of the bandwidth is reserved for the control systems.

Figure 31 demonstrates QoC performance of the control systems for case 3. All control systems are sufficiently served by the network player corresponding to the demands of the control player. We observe that while all control systems' performances are preserved, the convergence takes more iterations.

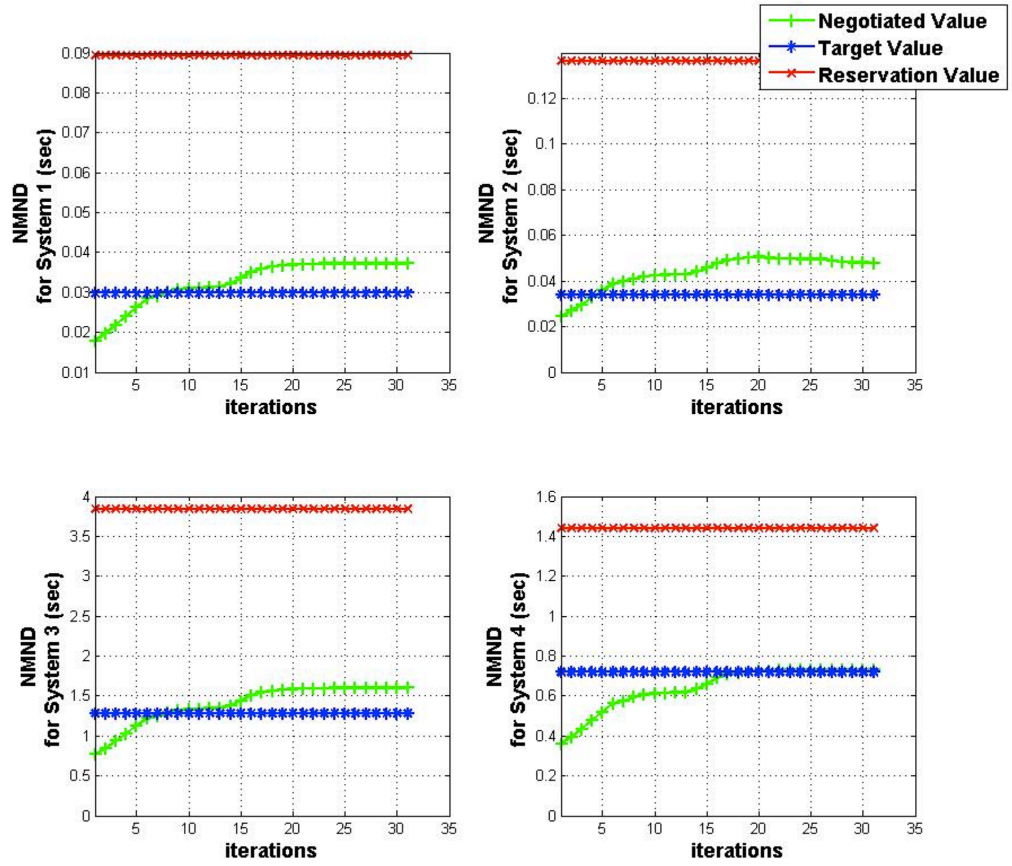

Figure 29. Converging of $\eta_{negot,i}(k)$ with iterations, k. Negotiated Value, Target Value, Reservation Value show $\eta_{negot,i}(k), t_{\eta,i}, \eta_{\max,i}$ respectively for case 3.

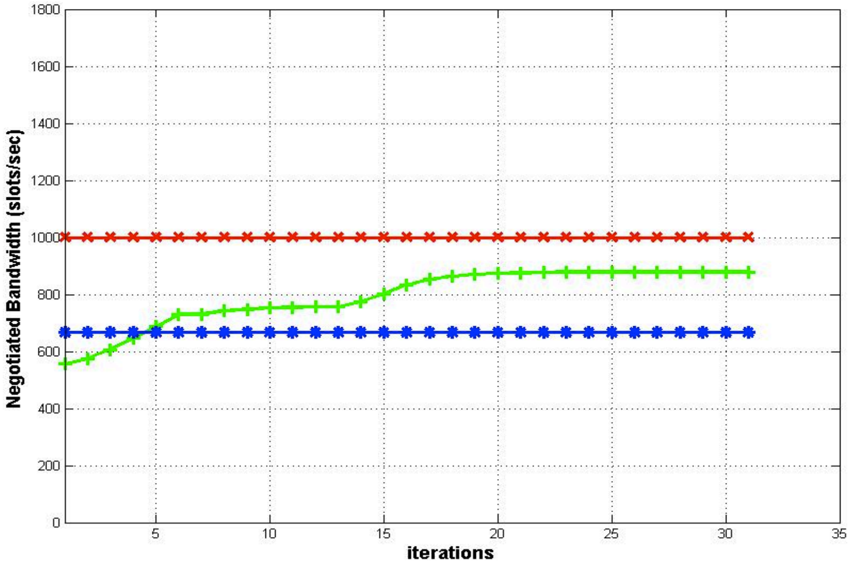

Figure 30. Converging of $bw_{negot}(k)$ with iterations, k. Negotiated Value, Target Value, Reservation Value show $bw_{negot}(k)$, t_{bw}, $bw_{\max}$ respectively for case 3.

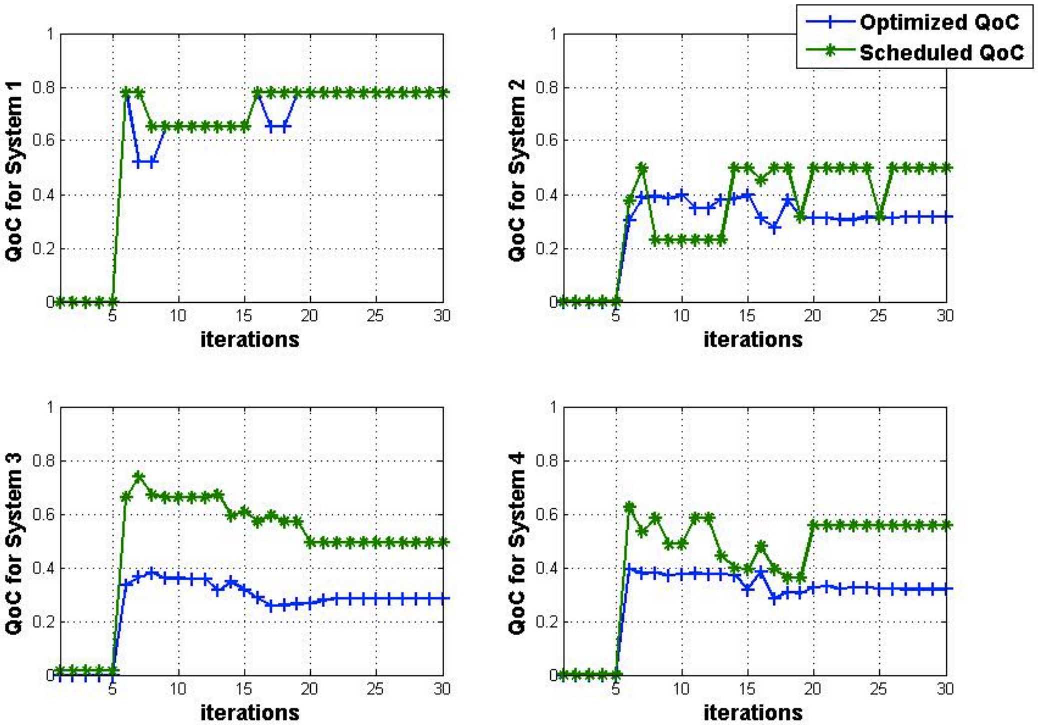

Figure 31. Optimized Control System's QoC (computed according to (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$)) and Scheduled QoC (computed according to (*Rate*$_i$, D_i and L_i)) for case 3.

Figure 32 demonstrates the QoS patterns of the bargaining against the bargaining iterations. We observe that the convergence in the QoS pattern takes a bit longer and remaining bandwidth is smaller compared to the previous cases due to the amount of available bandwidth.

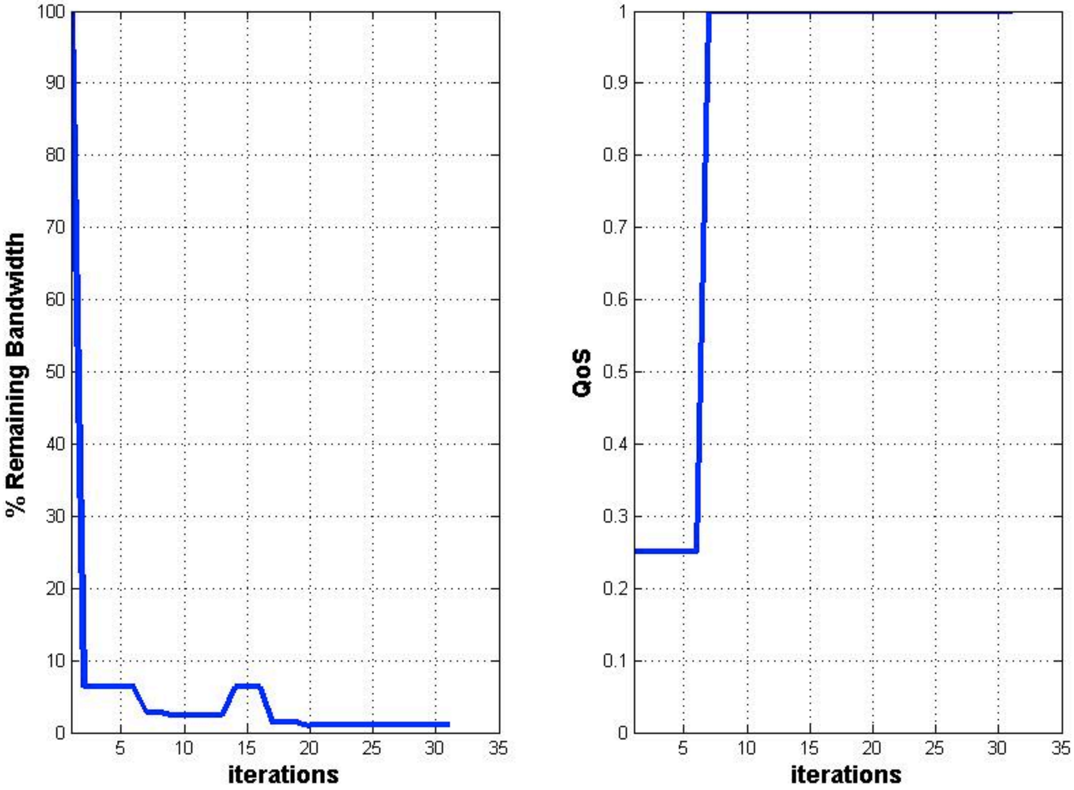

Figure 32. Remaining bandwidth after allocation to control systems and the change of QoS for case 3.

8.3. Comparison with Other Studies

A literature survey about similar studies shows that there are no complete studies that assume realistic NCS assumptions. However, there are studies that consider bandwidth management studies for reserving to the control systems with respect to control system performance. Among these studies, [20] and [101] are the studies that have the best performance in terms of control system output error.

So, we compare our study with these two studies. The studies assume simplified network assumption, so we adapt the studies to our case. Also, the two studies are online bandwidth management algorithms, so an adaption of the methods to offline design is implemented. Both studies are shown to be better than fixed bandwidth allocation, as in our case. The study in [20] assumes a fuzzy bandwidth management proceeded with respect to output error. The study in [101] reaches an optimal bandwidth management by allocating the all network resources to the control system with the highest output error. The study in [20] shows that it is better slightly than [101]. We call the studies in [20] and [101] as FBM and OBM, respectively.

The simulations for the two strategies are plotted for QoC performance and bandwidth convergence.

Figure 33 demonstrates the QoC performance of the four control systems under the FBM bandwidth management strategy. We observe that convergence in terms of QoC performance is not established. Also, the distribution of the bandwidth is not fair. The lower bandwidth demanding control systems receive lower performance in terms of both scheduled and optimized QoC. Figure 34 demonstrates the bandwidths assigned to the four control systems. We observe that bandwidth assignments do not converge, where we can interpret the reason for the fluctuating QoC performances.

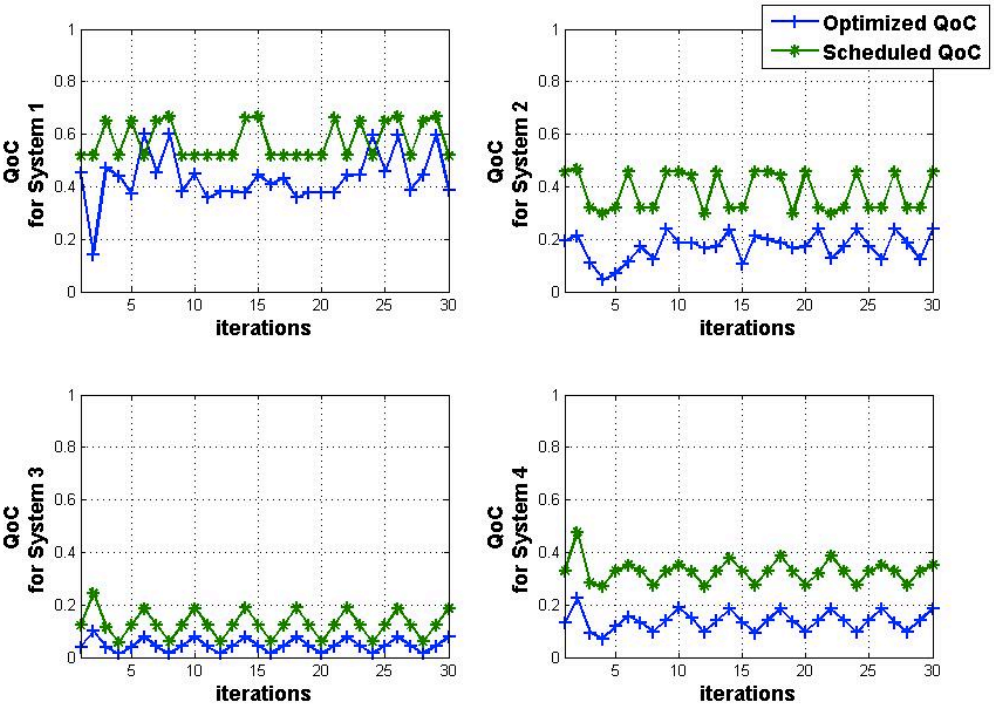

Figure 33. Optimized Control System's QoC (computed according to (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$)) and Scheduled QoC (computed according to (*Rate*$_i$, D_i and L_i)) for FBM strategy.

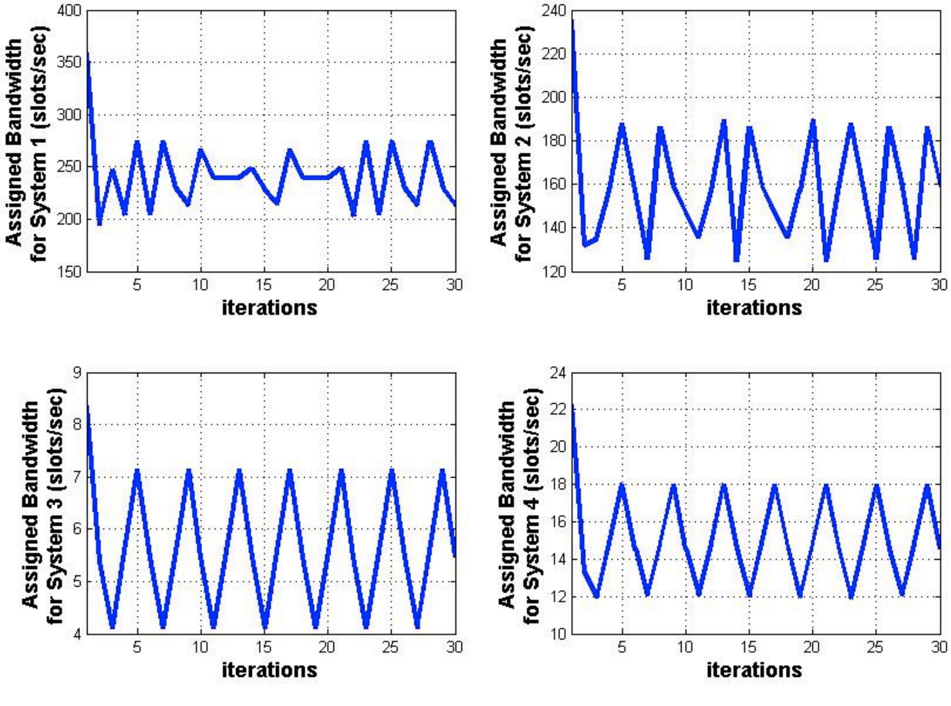

Figure 34. Assigned bandwidth to the four control systems for FBM strategy.

Figure 35 demonstrates the QoC performance of the four control systems under the OBM bandwidth management strategy. We observe that convergence in terms of QoC performance is not established as in the case of FBM strategy. The distribution of the bandwidth is not fair, also. The lower bandwidth demanding control systems receive lower performance in terms of both scheduled and optimized QoC. Figure 36 demonstrates the bandwidths assigned to the four control systems. We observe that bandwidth assignments converge for lower bandwidth demanding nodes, but other nodes receive fluctuating service. Additionally, the nodes that have converging bandwidth receive lower QoC performance from the NCS.

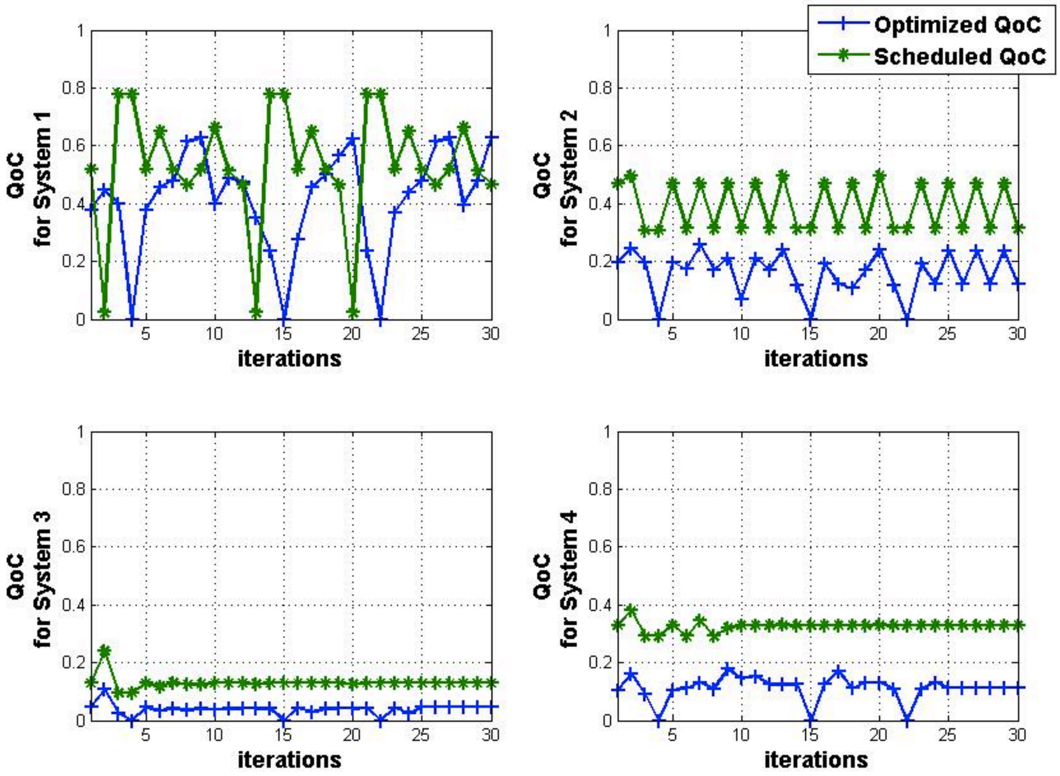

Figure 35. Optimized Control System's QoC (computed according to (*frequency$_i$*, *Deadline$_i$*, *Loss$_i$*)) and Scheduled QoC (computed according to (*Rate$_i$*, *D$_i$* and *L$_i$*)) for OBM strategy.

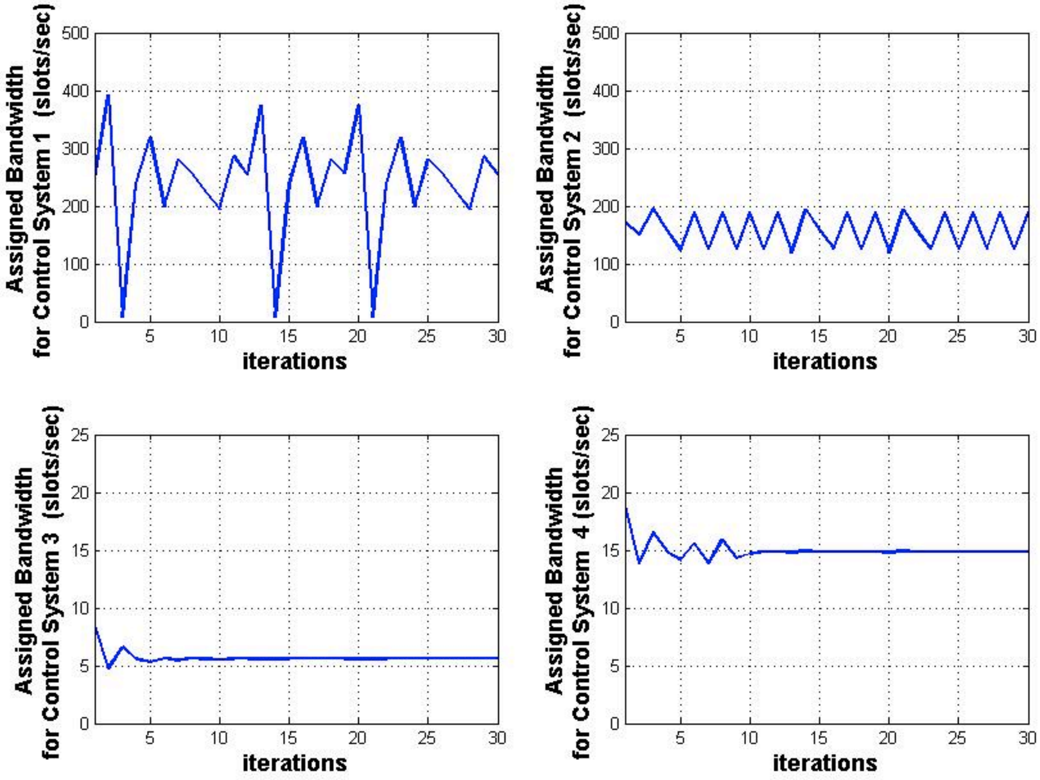

Figure 36. Assigned bandwidth to the four control systems for OBM strategy.

8.4. Scalability of the INtERCEDE

In order to observe the performance of INtERCEDE for NCSs which have different number of control systems, various simulations are demonstrated to measure different aspects of performance. The simulations are performed for the different numbers of one control system that exist on the same NCS. In order to have a clear picture of the INtERCEDE, we choose the most bandwidth demanding control system in the previous simulations that is control system 1. While all the simulation parameters are the same as in the first simulation study, the replication number of the control system 1 existing on the same NCS takes different values.

Figure 37 demonstrates the average QoC computed corresponding to control player's (Optimized QoC) and network player's service (Scheduled QoC). Except a fluctuation in scheduled QoC, we observe that average QoC that the control systems receive from the NCS decrease exponentially after a specific number of control systems. When the number of control systems is 6, the bandwidth in the NCS becomes scarce. So, network player has some difficulty to fully satisfy control player demands. In order to satisfy control system demands, more bandwidth is assigned for some control systems to satisfy constraints, while others receive less bandwidth. This reservation decreases average Scheduled QoC more than average Optimized QoC. When the number of control systems is 8, the control systems start to oscillate at the control system output since the bandwidth is scarcer. In this case, no control systems receive sufficient bandwidth from the network player and average Optimized QoC and average Scheduled QoC both decrease and have close QoC values. After 8 control systems in the NCS, the bandwidth provided by network player is not sufficient to guarantee stability and QoC in the system vanishes. In other words, after some specific number of control systems existing on the NCS, the NCS is not schedulable.

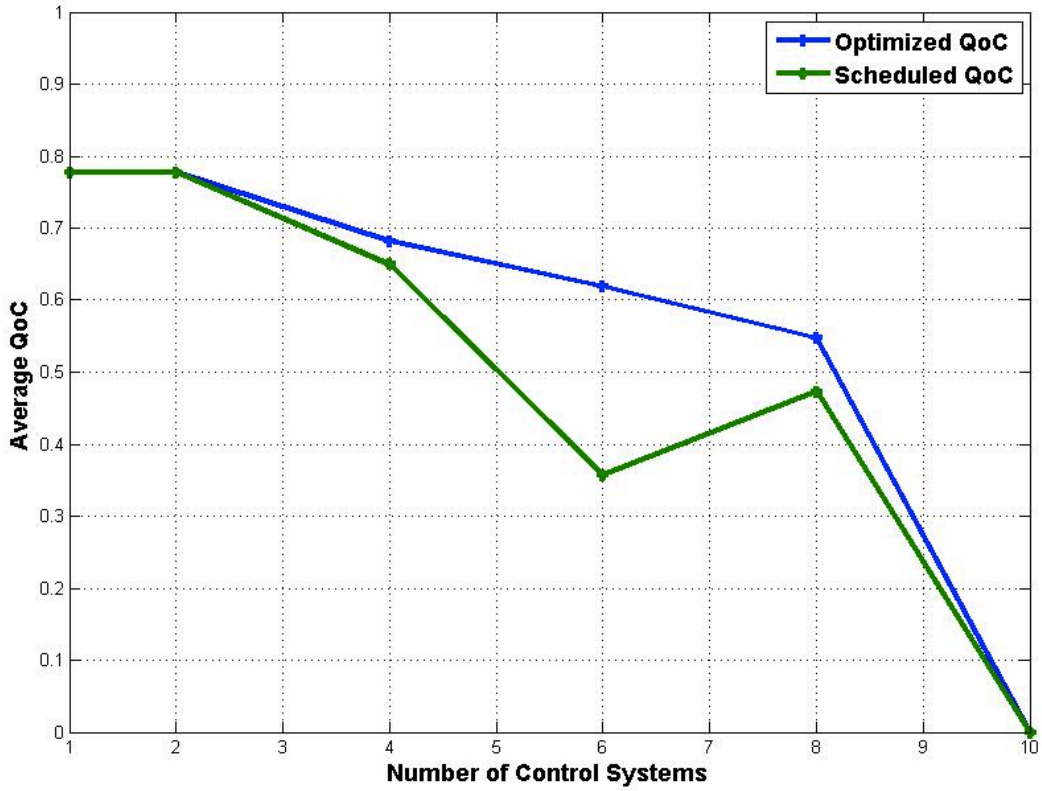

Figure 37. Average Optimized Control System's QoC (computed according to (*frequency*$_i$, *Deadline*$_i$, *Loss*$_i$)) and Average Scheduled QoC (computed according to (*Rate*$_i$, D_i and L_i)) versus number of control systems.

Figure 38 demonstrates QoS in terms of remaining bandwidth in the NCS against number of control systems. We observe that after 6 of the control system 1, the remaining bandwidth vanishes. We also observe from Figure 37 that the system is schedulable until 10 of the control system 1, where QoC reaches zero. So, we can say that after first allocation of all bandwidth to the control systems with INtERCEDE, there is a region of number of control systems that can be added to the NCS that can preserve schedulability.

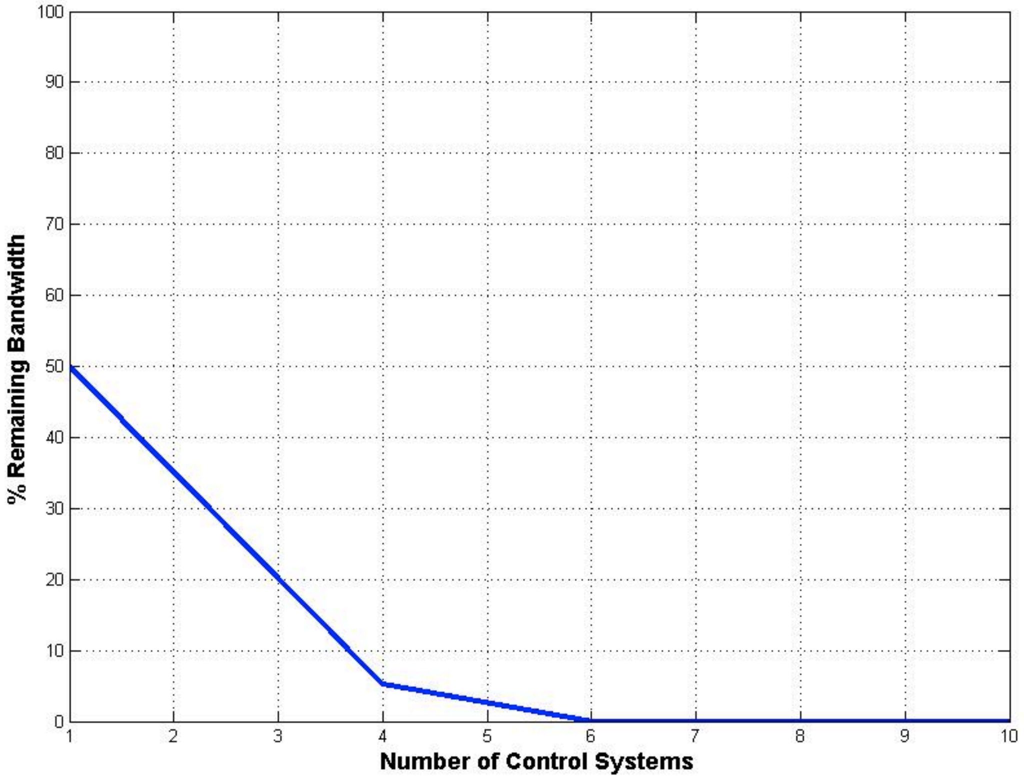

Figure 38. Remaining bandwidth after allocation to control systems versus number of control systems.

Figure 39 demonstrates the average NMND against number of control systems. Similar to the previous figures, the average NMND reaches to its reservation value at 8 of the control system 1. So, we can say that 8, not inclusively, is the limit for the stability of the control system 1 in the given simulation parameters in Table 4 and Table 15. Also, we observe a linear relationship between average NMND values and the number of control systems on the NCS.

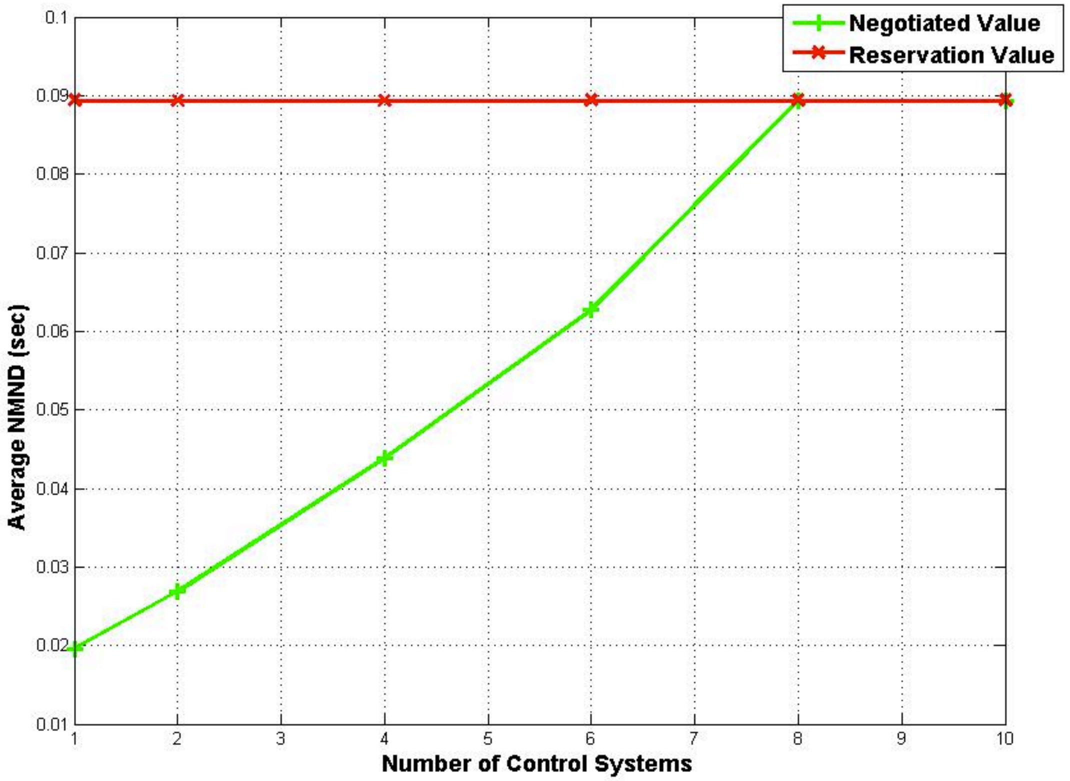

Figure 39. Negotiated Value, Reservation Value show $\eta_{negot,i}(k), \eta_{\max,i}$ respectively versus number of control systems.

Figure 40 demonstrates the negotiated bandwidth value against number of control systems. Similar to the remaining bandwidth QoS measure, after 6 of the control system 1, the total bandwidth is consumed by the control nodes. We also observe an exponential decrease in remaining bandwidth after a specific number of control systems existing on the NCS.

Simulations show that the NCS bargaining ends with a feasible and optimal solution in the sense that the user desired parameters play the significant roles in the bargaining. In fact, the flexibility in the design approach makes possible any different solution as an optimal solution with respect to the user desired choices of parameters. Different choices in the parameters may encompass infinitely possible choices of system design. The designer will choose parameters according to the system design constraints.

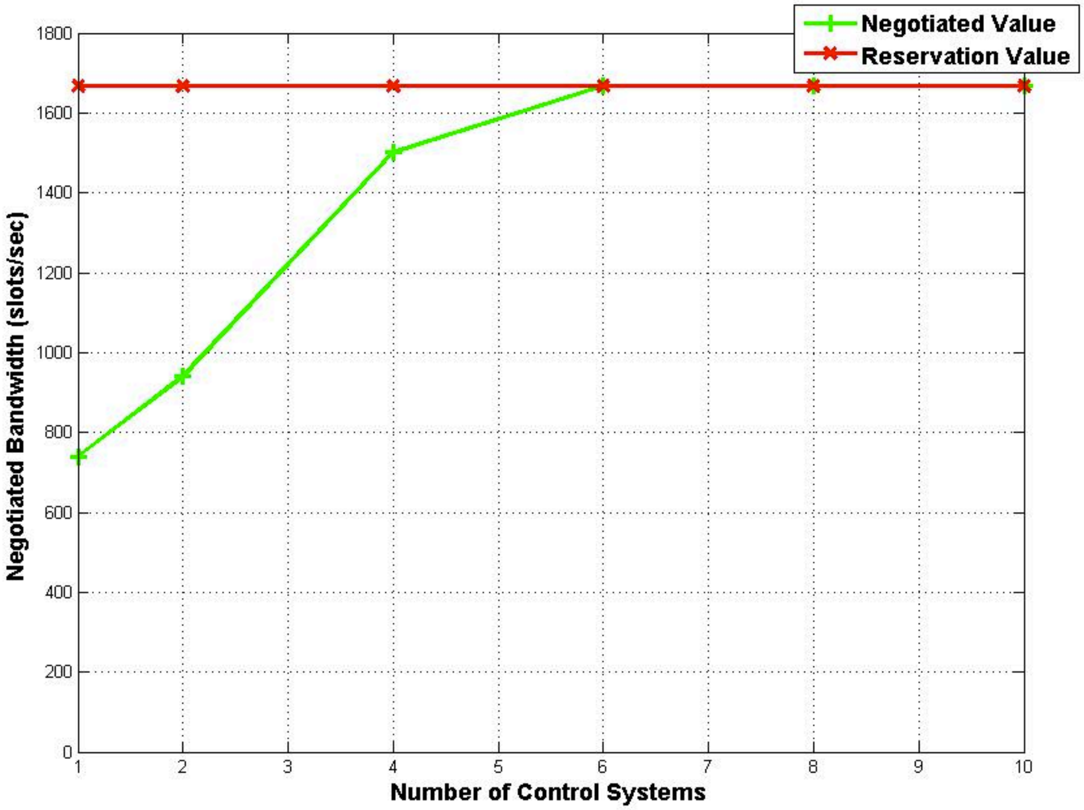

Figure 40. Negotiated Value, Reservation Value show $bw_{negot}(k), bw_{\max}$ respectively versus number of control systems.

The computation time of INtERCEDE does not increase much by increasing the number of control systems when the network configuration is fixed. INtERCEDE computation time basically takes more time in network player optimization than control player optimization. Added control systems incur low computation time to the INtERCEDE.

The optimality of the INtERCEDE depends on the NCS designer choices. INtERCEDE outputs the results that satisfy both control system designer and network system designer. While there are many feasible solutions, designer adjusts bargaining parameters to reach the best outcome of the INtERCEDE for himself.

CHAPTER 9

CONCLUSION

Designing an NCS is a problem involving many other disciplines. The main two disciplines that are inevitable are control research area and networking research area. However, often many other disciplines such as computing, telecommunications, security are also included. So, when a problem occurs in an NCS design, the designer may not have the required information to investigate the problem. For example, when a controller parameter determination is needed in design, a computing oriented professional may be stuck at that point of the design. So, a main question arises: Should the designer know everything? Or, should everybody do his own job? It is obvious that it is not possible to know everything. So, the division of the design job into various tasks that will be done by not only one designer but several designers seems the clever way. By accepting this implication, another question arises: Who will do the integration job? Or, what is the procedure to integrate these all different tasks from different disciplines? At this part of the problem, we propose INtERCEDE that is a unique way to solve this integration problem and proposes several solutions to complete the design problem. Our main idea is straightforward: all the tasks are evaluated with some criteria. INtERCEDE proposes Quality-of-Service (QoS) for network performance measure and Quality-of-Control (QoC) for control system performance measure. These are stimulated by a bargaining approach and the bargaining parameters that are tied to these measures. For QoC, the parameter that is tied is the Negotiated Maximum Network Delay (NMND) and for QoS, the parameter that is tied is the negotiated bandwidth. The unique way of the bargaining proposed by INtERCEDE proceeds the negotiation by adapting these parameters by fuzzy rules.

In this study, we deploy INtERCEDE by assuming control systems have strict deadlines as the most of the control systems have this kind of importance. We can give several examples to the control systems which have safety requirements: automobiles, military air, land and marine vehicles, nuclear power plants, industrial automation systems, etc. These are all control systems that have to be operated safely to prevent economic or public damages. The main idea for this consideration is to guarantee worst-case operation conditions. The critical component for establishing this guarantee is the deployed network technology. The network technologies that provide deterministic guarantees are time–slotted communication protocols. There are other similar technologies, but the best of these is time-slotted communication protocols. So, in order to build INtERCEDE we develop our method by relying on time-slotted communication protocols. We propose Necessary Condition Evaluation Algorithms to measure QoS for the time-slotted communication protocol. These algorithms work on worst-case scenario, so we obtain results that are consistent with our previous assumptions. In the control side, there are existing methods to measure plant performance and we use step response measure since it is the conventional way of performance measure for control systems.

The critical part of INtERCEDE is the decoupling of control tasks and networking tasks from each other. Since NCSs are dynamic systems, each component of the NCS may be replaced by some other technology. So, integrated controller and network designs eliminate this positive flexibility completely, since they are designed by global assumptions that are imposed on all the components on the NCS. This is a very important aspect such that the main idea of inventing NCSs is this idea. The NCSs are invented to bring this reality to life: establishing flexibility and extensibility as much as it can be. By integrating controller to the NCS, the advantage is eliminated. So, we assume the controller technology for the physical system exists a priori as the network technology exists a priori. We base all our design on this reality.

When the performance and success of INtERCEDE is evaluated, we see that we provide not only stability to the control system but also minimized error at the output of the control system compared to the ideal communication case. Our scalability results show that even if the bandwidth in the network is not sufficient, INtERCEDE provides nonzero QoC to the control systems.

INtERCEDE is tested at various case studies and the results show that the bargaining approach provides both flexibility, extensibility to the NCS by conserving bandwidth, and provides high QoC to the control systems by minimizing output error.

We compared INtERCEDE to some studies in the literature that gained attention. Finding studies for comparison with INtERCEDE is a difficult task, since INtERCEDE is the only complete solution. We observed that INtERCEDE converges better than the proposed solutions, and also more fair than in distributing bandwidth to the control systems. While the control systems obtain close QoC values by INtERCEDE, other methods favor some control systems to others.

Scalability simulations show that network can not service unlimited number of control systems. We saw that after some additional control nodes, average QoC in the NCS decreases drastically after a specified number of control nodes existing in the system. So, the designer must consider choosing right networking technology for the existing control system profile.

In this thesis study, we propose a new algorithmic design approach for Networked Control Systems. For given control system parameters and network bandwidth, our INtERCEDE approach computes the control system message parameters and network service parameters together using a bargaining game. The bargaining game formulation enables the parameterization of the NCS design and emphasizing the performance of the control system or sparing the network bandwidth as desired. INtERCEDE outputs a message schedule which can be directly used in any time-slotted communication network. Our future work includes adopting INtERCEDE to event-triggered industrial communication networks where the schedule is determined by assigning priorities to the messages rather than assigning time slots.

APPENDIX A

ALGORITHMIC COMPUTATION OF SCHEDULE SERVICE PARAMETERS

Algorithm 1:

This algorithm computes the lowest transmission rate without sample loss for the given control system message set. The idea depends on finding the largest time difference between two messages that are not lost given a message schedule.
N: Number of slots assigned to control system I
N_{Total}: Total number of slots in the periodic cycle

```
Inputs: $frequency_i$ , $Loss_{final,i}$, $S_i$, $T_{slot}$
Outputs: $Rate_i$, $Loss_{final,i}$
1) StartSlot=1, CurrentSlot=1
2) $T_S$ =Low_Value, $MSD=\frac{T_S}{T_{slot}}$,  sampleNum=HighValue
3) PreviousSlot = CurrentSlot, CurrentSlot=(StartSlot+1) % N,
LastTxSlot=StartSlot
4) maxSampleNum=0
5) If $\left( \Delta_s = \left( \frac{s_{c,k}(CurrentSlot)-s_{c,k}(PreviousSlot)-jitter)}{MSD} \right) \right) > maxSampleNum$
$maxSampleNum = \left( \Delta = \left( \frac{s_i(CurrentSlot)-s_i(PreviousSlot)-MSD)}{MSD} \right) \right)$
   End
6) If LastTxSlot equals one of the previous LastTxSlot's,
        Goto step 7
   Else
        PreviousSlot = CurrentSlot
        increment CurrentSlot
        Goto step 5
   End
7) If sampleNum<maxSampleNum
      sampleNum=maxSampleNum
      Increment StartSlot
```

```
    End
8) If (sampleNum ==1)
        Goto Step 9
    End
    If No First Loss Occurs
        LowerPrevious$T_S$=$T_S$
        $T_S$=2*$T_S$
    Elseif Loss Occurs
        UpperPreviousTs=$T_S$
        $T_S = \frac{LowerPreviousTs+T_S}{2}$
    Else
        LowerPrevious$T_S$=$T_S$
        $T_S = \frac{UpperPreviousTs+T_S}{2}$
    End
    Goto step 4
9) Stop.  Rate$_i$=1/$T_S$, $L_{final,i}=\frac{(Loss_{final,i}+1)}{(T_S*frequency_i)}-1$ where i denotes the ith control system
```

Example: N=1, S_i={2}, $frequency_i$ =2/3, T_{slot}=1, N_{Total}=6:
Rate and Loss Calculation Algorithm Run

```
1) StartSlot=1
2) $T_S$ =1, MSD=$\frac{T_S}{T_{slot}}$=1,  sampleNum=0
3) PreviousSlot = 1 , CurrentSlot=1, LastTxSlot=2
4) maxSampleNum=0
5) $\Delta$=5, maxSampleNum=5, LastTxSlot=2
6) Goto step 7
7) sampleNum=5
8) LowerPrevious$T_S$=1, Ts =2*Ts=2, MSD=2,Goto step 4
4) maxSampleNum=0
5) $\Delta$=2, maxSampleNum=2, LastTxSlot=2
6) Goto step 7
7) sampleNum=2
8) LowerPrevious$T_S$=2,Ts =Ts*2 = 4, MSD=4, Goto step 4
4) maxSampleNum=0
5) $\Delta$=0.5, maxSampleNum=0.5, LastTxSlot=2
6) Goto step 7
7) sampleNum=0.5
8) UpperPrevious$T_S$=4, Ts = (Ts+ LowerPrevious$T_S$)/2 = 3, MSD=3,  Goto step 4
4) maxSampleNum=0
5) $\Delta$=1, maxSampleNum=1, LastTxSlot=2
6) Goto step 7
```

```
7) sampleNum=1
8) Goto step 9
9) $Rate_i=0.33$, $Loss_{final,i}=T_S*frequency_i-1=1$
```

Algorithm 2:

This algorithm calculates the worst case maximum message delay (max(Delay(.))) given the schedule s_i. The idea depends on finding the largest delay that a message is incurred in the worst case given a message schedule.

```
Inputs: $frequency_i$ , $S_i$, $T_{slot}$
Outputs: $D_{,i}$
1) StartSlot=1, MSD=$\frac{T_s}{T_{slot}}$,  DelayArray is array of length N
2) DelayArray(StartSlot)=0
3)  CurrentSlot= StartSlot +1, LastTxSlot= StartSlot, LastDelay=0, k=0,
4) While ( $\left(\Delta = \left\lfloor \frac{s_i(CurrentSlot) - s_i(StartSlot)}{MSD} \right\rfloor - k\right) == 0$)
        Increment CurrentSlot
   End
5) If $\Delta_s$=1
        LastDelay=$s_{c,k}$(CurrentSlot- StartSlot-1)%MSD
        k= k+$\Delta$
   End
   If DelayArray(CurrentSlot)<LastDelay
        DelayArray(CurrentSlot)=LastDelay
        LastTxSlot=CurrentSlot
        Increment CurrentSlot
   End
6) If LastTxSlot equals one of the previous LastTxSlot's
        Goto step 7
   Else
        Goto step 4
   End
7) Increment StartSlot
   If All Slots Used
        Goto step 8
   Else
        DelayArray(StartSlot)=0
        Goto step 3
   End
8) $D_i = \max_m$ DelayArray(m)where i denotes the ith control system .
9)Stop.
```

Example: $S_{,i}$={2}, $frequency_i$ =2/3, T_{slot}=1, N_{Total}=6:

Delay Calculation Algorithm Run:

1) *StartSlot=1*
2) *DelayArray(StartSlot)=1, k=0*
3) *CurrentSlot=1, LastTxSlot=1*

4) *Δ=1 at the first iteration*
5) *LastDelay=2, k=1*
6) *DelayArray(CurrentSlot)=2, LastTxSlot=1, CurrentSlot=1*
7) *Goto step 8*
8) $D_i = 2$ *(as expected)*
9) *Stop*

APPENDIX B

BARGAINING GAME DETAILS

In the bargaining step of the INtERCEDE, the update of the negotiated parameters are performed as follows:

$$\eta_{negot,i}(k+1) = \eta_{negot,i}(k) + \delta\eta_{negot,i}(k+1)$$

$$bw_{negot}(k+1) = bw_{negot}(k) + \delta bw_{negot}(k+1)$$

where $\delta\eta_{negot,i}(k+1)$ and $\delta bw_{negot}(k+1)$ are the update values for the next step of bargaining game for control player and network player, respectively. The computation of these update values is presented below.

$$\delta\eta_{negot,i}(k+1) = \alpha_\eta * n_i(k+1) * p_{0,i} + \left(1-\alpha_\eta\right) * |n_i(k+1)| * t_{0,i}$$

$$n_i(k+1) = \beta_\eta * n_i(k) + \left(1-\beta_\eta\right) * s_{0,i}(k+1)$$

$$t_{0,i} = t_{\eta,i} - \eta_{negot,i}(k)$$

$$\delta bw_{negot}(k+1) = \alpha_{bw} * b(k+1) * p_1 + (1-\alpha_{bw}) * |b(k+1)| * t_1$$

$$b(k+1) = \beta_{bw} * b(k) + (1-\beta_{bw}) * s_1(k+1)$$

$$t_1 = t_{bw} - bw_{negot}(k)$$

$p_{0,i}$ and p_1 are constants depending on the reservation values of the bargaining game for control player for *ith* control system and network player, respectively. $t_{\eta,i}$ and t_{bw} are target values of the bargaining game for control player for *ith* control system and network player, respectively. α_η and α_{bw} are the constants at which rate the past values affect the current negotiated parameters. $n_i(k)$ and $b(k)$ are the step sizes for determining the next update values for the negotiated parameters for control player for *ith* control system and network player, respectively. β_η and β_{bw} are the constants at which rate the past values affect the current step sizes for the control player and the network player. $p_{0,i}$, p_1, $t_{\eta,i}$, t_{bw}, α_η, α_{bw}, β_η and β_{bw} are parameters chosen as reflecting the user desired design.

The bargaining is proceeded with fuzzy inference techniques due to the fact that fuzzy logic handles systems with less understood dynamics better and handles nonlinear systems.

$s_{0,i}(k)$ and $s_1(k)$ values are step update values obtained via fuzzy inference rules depending on the first derivative and second derivative of QoC and QoS Fitness values defined as below:

$$\frac{\partial}{\partial k} QoC_{i,k} = QoC_{i,k-1} - QoC_{i,k-2}$$

$$\frac{\partial}{\partial^2 k} QoC_{i,k} = \frac{\partial}{\partial k} QoC_{i,k-1} - \frac{\partial}{\partial k} QoC_{i,k-2}$$

where $QoC_i(k)$ is the QoC value for the i^{th} control system at the k^{th} step of the bargaining game, and

$$\frac{\partial}{\partial k} QoS_k = QoS_{k-1} - QoS_{k-2}$$

$$\frac{\partial}{\partial^2 k} QoS_k = \frac{\partial}{\partial k} QoS_{k-1} - \frac{\partial}{\partial k} QoS_{k-2}$$

where QoS_k is the QoS fitness value for the given *schedule* at the k^{th} step of the bargaining game.

The assumptions and criteria for the update of the lower bounds of the worst case overall delay during game procedure are stated below:

1) $n_i(0) = \gamma$, $b(0) = \delta$ where $-2 \leq \gamma \leq 2$, $-2 \leq \delta \leq 2$
2) $-2 \leq s_{0,i}(k) \leq 2$, $-2 \leq s_1(k) \leq 2$
3) The number of rounds of the bargaining game that will end up with an agreement should be as small as possible.
4) Each player knows other player's moves.
5) Control player increases its $s_{0,i}(k)$ with the decreasing value of $\frac{\partial}{\partial k} QoC_{i,k}$.
6) Control player increases its $s_{0,i}(k)$ with the decreasing value of $\frac{\partial}{\partial^2 k} QoC_{i,k}$.
7) Network player increases its $s_1(k)$ with the decreasing value of $\frac{\partial}{\partial k} QoS_k$.
8) Network player increases its $s_1(k)$ with the decreasing value of $\frac{\partial}{\partial^2 k} QoS_k$.
9) If network cannot find a feasible schedule, then $s_{0,i}(k)$ and $s_1(k)$ will be increased with a fixed value.

The rule values are given symbolically with the names negative_high, negative_medium, negative_low, zero, positive_low, positive_medium, positive_high for both inputs and outputs where

$$negative_{high} < negative_{medium} < negative_{low} < zero = 0 < positive_{low} < positive_{medium} < positive_{high}$$

The defuzzification method for the fuzzy inference rules conforms to Mamdani type defuzzification. The membership function for each value is a trapezoidal membership function. The rules for the defuzzification of negotiated worst case overall delay and negotiated bandwidth are given in Table 13 and Table 14, respectively. These rules only take $\frac{\partial}{\partial k} QoC_{i,k}, \frac{\partial}{\partial^2 k} QoC_{i,k}, \frac{\partial}{\partial k} QoS_k, \frac{\partial}{\partial^2 k} QoS_k$ as inputs. The short hand notation for membership functions $negative_{high}$, $negative_{medium}$, $negative_{low}$, $zero$, $positive_{low}$, $positive_{medium}$, $positive_{high}$ are NH, NM, NL, Z, PL, PM, PH, respectively. The rules are chosen according to the following criteria:

1) If player gains some benefit with the increase of input parameters, it wants more.
2) If player loses some benefit with the decrease of input parameters, it sacrifices more to reach at least an agreement.
3) Network player is more willing to agree than the control player, because control player has real-time requirements which make it more conservative.

Table 13. Rules for $\eta_{negot,i}$. Inputs are $\frac{\partial}{\partial k} QoC_{i,k}$ and $\frac{\partial}{\partial^2 k} QoC_{i,k}$.

$\eta_{negot,i}$ Step Size		$\frac{\partial}{\partial^2 k} QoC_{i,k}$						
		NH	NM	NL	Z	PL	PM	PH
$\frac{\partial}{\partial k} QoC_{i,k}$	NH	PH	PM	PM	PM	PL	PL	PL
	NM	PH	PM	PM	PL	PL	PL	Z
	NL	PM	PM	PL	PL	Z	Z	Z
	Z	PL	PL	PL	Z	NL	NL	NL
	PL	Z	Z	Z	NL	NL	NL	NM
	PM	NL	NL	NL	NM	NM	NM	NH
	PH	NM	NM	NM	NH	NH	NH	NH

Table 14. Rules for bw_{negot}. Inputs are $\frac{\partial}{\partial k} QoS_k$ and $\frac{\partial}{\partial^2 k} QoS_k$.

bw_{negot} Step Size		$\frac{\partial}{\partial^2 k} QoS_k$						
		NH	NM	NL	Z	PL	PM	PH
$\frac{\partial}{\partial k} QoS_k$	NH	PH	PH	PM	PM	PL	PL	PL
	NM	PH	PH	PM	PM	PL	PL	PL
	NL	PH	PH	PM	PL	PL	Z	Z
	Z	PL	PL	PL	Z	Z	Z	NL
	PL	Z	Z	Z	NL	NL	NL	NM
	PM	Z	Z	Z	NL	NL	NM	NM
	PH	Z	NL	NL	NM	NM	NM	NH

APPENDIX C

SIMULATION RESULTS RELATED TO BARGAINING

The values of the parameters that are employed in the bargaining as explained in Appendix B are listed in Table 15. Choice of T_{slot} leads to a system that has no feasible schedule in the beginning of the bargaining between control player and network player. Schedulability is achieved in the further iterations of the bargaining. Initial step sizes of the bargaining are chosen as some small positive values, which indicate the willingness of the control player and network player to reach an agreement. β_η and β_{bw} are the bargaining parameters that determine the emphasis on either bargaining tradeoff or the speed to reach an agreement. β_η and β_{bw} are set to 0.8 at the lack of a feasible schedule to increase the bargaining tradeoff for both players. β_η and β_{bw} are set to 0.5 in the presence of a feasible schedule to increase the speed to reach an agreement and to decrease tradeoff. Reservation values of NMND values are chosen according to the stability margins. Reservation value of the negotiated bandwidth is the total available bandwidth in the network. Reservation value of the negotiated bandwidth may be chosen lower to statically conserve bandwidth. Target values of the negotiated parameters must be chosen according to the user demands. We choose symbolic target values significantly lower than reservation values. Fuzzy inference model of step sizes in the bargaining for both control player and network player is Mamdani type that is one of the known fuzzy inference techniques while other choices are also possible.

Table 15. Specific bargaining parameters.

Component	Parameters
Communication Network	T_{slot} =0.0006 sec/slot
Control Player Bargaining	$n_i(0)$=0.3 α_η =0.95 β_η=0.8, if not scheduled β_η=0.5, if scheduled $t_{\eta,i}$=η_{max}/4 for all control systems $p_{0,i}$=η_{max} for all control systems
Network Player Bargaining	$b(0)$=0.3 α_{bw} =0.95 β_{bw}=0.8, if not scheduled β_{bw}=0.5, if scheduled t_{bw}=1/(T_{slot} * 1.7) p_1=$1/T_{slot}$ for all control systems
QoC Fitness Parameters	*cr=0.1*
Control Player Optimization	$P_{crossover}$=0.9 $P_{mutation}$=0.25
Network Player Optimization	$P_{crossover}$=0.9 $P_{mutation}$=0.25

The step size evolutions for control systems and network are sketched over the iterations below:

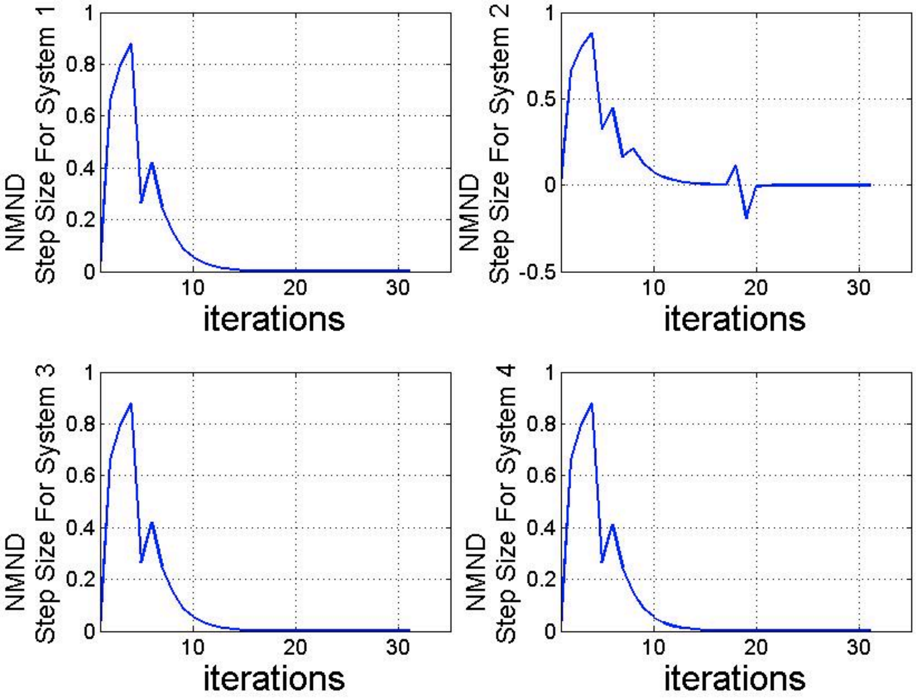

Figure 41. Convergence of Negotiated Maximum Network Delay Step Sizes $n_i(k)$

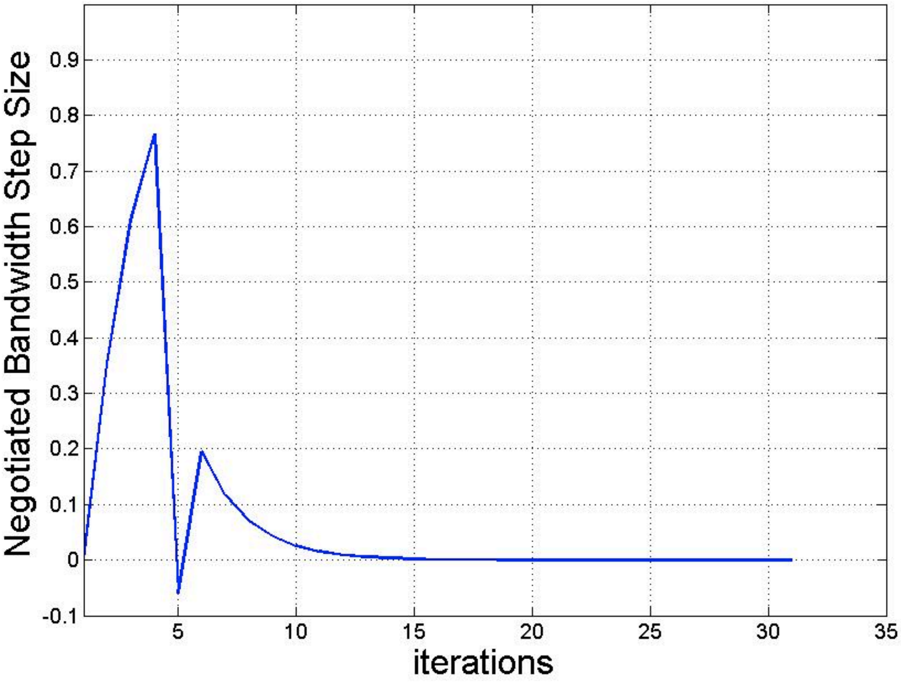

Figure 42. Convergence of Negotiated Bandwidth Step Sizes $b(k)$

The simulation results show that the step sizes of the negotiated maximum network delay and negotiated bandwidth update values initiate with a high positive increase, since initially both control player and network player’s demands are not satisfied. After sufficient self-sacrifices are made from the negotiated values in the subsequent bargaining iterations, the step sizes attenuate to zero. The difference in system dynamics between control player and network player is evident in the negotiated bandwidth step size values with the higher fluctuations. The bargaining’s effect on step sizes show that as in the negotiated values, control systems sacrifice more resources than communication network as the peak values of step sizes of the control systems are higher. Moreover, the communication network exhibit negative step sizes before convergence due to plenty of resources.

REFERENCES

[1] T. C. Yang, "Networked control system: a brief survey," *IEE Proceedings-Control Theory and Applications,* vol. 153, pp. 403-412, Jul 2006.

[2] J. P. Hespanha, P. Naghshtabrizi, and Y. G. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE,* vol. 95, pp. 138-162, Jan 2007.

[3] M. P. Thomesse, "Fieldbus technology in industrial automation," *Proceedings of the IEEE,* vol. 93, pp. 1073-1101, Jun 2005.

[4] N. Navet, Y. Q. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE,* vol. 93, pp. 1204-1223, Jun 2005.

[5] M. Felser, "Real-time ethernet - Industry prospective," *Proceedings of the IEEE,* vol. 93, pp. 1118-1129, Jun 2005.

[6] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE,* vol. 93, pp. 1130-1151, Jun 2005.

[7] T. C. GmbH., "Time-Triggered Protocol TTP/C, High-Level Specification Document, Protocol Version 1.1," ed. [Online]: http://www.tttech.com, 2003. Last accessed on 18/06/12.

[8] L. Almeida, P. Pedreiras, and J. A. G. Fonseca, "The FTT-CAN protocol: Why and how," *IEEE Transactions on Industrial Electronics,* vol. 49, pp. 1189-1201, Dec 2002.

[9] J. J. Scarlett and R. W. Brennan, "Re-evaluating event-triggered and time-triggered systems," *2006 IEEE Conference on Emerging Technologies & Factory Automation, Vols 1 -3,* pp. 348-354, 1320, 2006.

[10] Echelon, "Introduction to the LonWorks® Platform," 2009.

[11] G. Leen and D. Heffernan. (2002, Formal Verification of the TTCAN Protocol. *[Online]*.

[12] F. Consortium, "FlexRay Communication System, Protocol Specification, Version 2.0," ed. [Online]: http://www.flexray.com, 2004. Last accessed on 18/06/12.

[13] B. Wang, Y. S. Zhang, X. Xiong, and S. A. Wang, "Time-triggered architecture (TTA) in industry hard R-T embedded control software design," *Proceedings of the 26th Chinese Control Conference, Vol 4,* pp. 736-739, 814, 2007.

[14] A. Albert, "Comparison of event-triggered and time-triggered concepts with regard to distributed control systems," *Proc. Embedded World,* pp. 235–252, 2004.

[15] I. Erturk, "A new method for transferring CAN messages using wireless ATM," *Journal of Network and Computer Applications,* vol. 28, pp. 45-56, Jan 2005.

[16] V. M. Sempere and J. Silvestre, "Multimedia applications in industrial networks: Integration of image processing in profibus," *IEEE Transactions on Industrial Electronics,* vol. 50, pp. 440-448, Jun 2003.

[17] X. L. Xu, J. P. Wen, Y. M. Xu, and Y. C. Shi, "Remote monitor & measurement instrument system faced to Internet interoperable service of manufacturing," *Proceedings of the Third International Symposium on Instrumentation Science and Technology, Vol 3,* pp. 1112-1116, 1239, 2004.

[18] S. Senol, K. Leblebicioglu, and E. G. Schmidt, "INtERCEDE: An algorithmic approach to networked control system design," *Journal of Network and Computer Applications,* vol. 34, pp. 1326-1341, Jul 2011.

[19] M. E. Ben 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,* vol. 14, pp. 776-787, Jul 2006.

[20] Z. X. Li, W. L. Wang, and Y. L. Jiang, "Managing Quality-of-Control and Requirement-of-Bandwidth in Networked Control Systems via Fuzzy Bandwidth Scheduling," *International Journal of Control Automation and Systems,* vol. 7, pp. 289-296, Apr 2009.

[21] L. Chen, J. M. Zhang, and S. Q. Wang, "Scheduling and control co-design for delay compensation in the networked control system," *Asian Journal of Control,* vol. 8, pp. 124-134, Jun 2006.

[22] L. J. Deng and H. S. Wang, "Codesigen of control and network scheduling in networked control systems," *2008 Proceedings of Information Technology and Environmental System Sciences: Itess 2008, Vol 4,* pp. 370-374, 2008.

[23] Y. B. Zhao, G. P. Liu, and D. Rees, "Integrated predictive control and scheduling co-design for networked control system," *IET Control Theory and Applications,* vol. 2, pp. 7-15, Jan 2008.

[24] S. Şenol, K. Leblebicioğlu, and E. G. Schmidt, "FlexRay Simulator Development For Networked Control System," presented at the IEEE SIU, 2007.

[25] X. F. Jiang, Q. L. Han, S. R. Liu, and A. K. Xue, "A new H-infinity stabilization criterion for networked control systems," *IEEE Transactions on Automatic Control,* vol. 53, pp. 1025-1032, May 2008.

[26] P. Marti, J. M. Fuertes, G. Fohler, and K. Ramamritham, "Improving quality-of-control using flexible timing constraints: Metric and

scheduling issues," *23rd IEEE Real-Time Systems Symposium, Proceedings,* pp. 91-100, 2002.

[27] K. Ogata, *Modern control engineering*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[28] P. Marti, J. Yepez, M. Velesco, R. Villa, and J. M. Fuertes, "Managing quality-of-control in network-based control systems by controller and message scheduling co-design," *IEEE Transactions on Industrial Electronics,* vol. 51, pp. 1159-1167, Dec 2004.

[29] R. A. Gupta and M. Y. Chow, "Networked Control System: Overview and Research Trends," *IEEE Transactions on Industrial Electronics,* vol. 57, pp. 2527-2535, Jul 2010.

[30] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems Magazine,* vol. 21, pp. 57-65, Feb 2001.

[31] K. Gu, V. L. Kharitonov, and J. Chen, Eds., *Stability of Time-Delay Systems*. Birkhauser Boston, 2003, p.^pp. Pages.

[32] D. Yue, Q. L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Transactions on Circuits and Systems Ii-Express Briefs,* vol. 51, pp. 640-644, Nov 2004.

[33] W. Zhang and M. S. Branicky, "Stability of Networked Control Systems with Time-Varying Transmission Period," presented at the Allerton Conf. Communication , Contr. And Computing, Urbana, IL, 2001.

[34] H. Lin, G. S. Zhai, and P. J. Antsaklis, "Robust stability and disturbance attenuation analysis of a class of networked control systems," *42nd IEEE Conference on Decision and Control, Vols 1-6, Proceedings,* pp. 1182-1187, 2003.

[35] H. Lin and P. J. Antsaklis, "Stability and persistent disturbance attenuation properties for a class of networked control systems: switched system approach," *International Journal of Control,* vol. 78, pp. 1447-1458, Dec 15 2005.

[36] P. H. Bauer, K. Premaratne, and J. Duran, "A Necessary and Sufficient Condition for Robust Asymptotic Stability of Time-Variant Discrete-Systems," *IEEE Transactions on Automatic Control,* vol. 38, pp. 1427-1430, Sep 1993.

[37] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine,* vol. 21, pp. 84-99, Feb 2001.

[38] W. A. Zhang and L. Yu, "Output feedback stabilization of networked control systems with packet dropouts," *IEEE Transactions on Automatic Control,* vol. 52, pp. 1705-1710, Sep 2007.

[39] C. H. Chen, C. L. Lin, and T. S. Hwang, "Stability of networked control systems with time-varying delays," *IEEE Communications Letters,* vol. 11, pp. 270-272, Mar 2007.

[40] D. S. Kim, Y. S. Lee, W. H. Kwon, and H. S. Park, "Maximum allowable delay bounds of networked control systems," *Control Engineering Practice,* vol. 11, pp. 1301-1313, Nov 2003.

[41] H. S. Park, Y. H. Kim, D. S. Kim, and W. H. Kwon, "A scheduling method for network-based control systems," *IEEE Transactions on Control Systems Technology,* vol. 10, pp. 318-330, May 2002.

[42] D. Yue, Q. L. Han, and J. Lam, "Network-based robust H-infinity control of systems with uncertainty," *Automatica,* vol. 41, pp. 999-1007, Jun 2005.

[43] Y. L. Sun and N. H. El-Farra, "Resource aware quasi-decentralized control of networked process systems over wireless sensor networks," *Chemical Engineering Science,* vol. 69, pp. 93-106, Feb 13 2012.

[44] H. Yu and P. J. Antsaklis, "Event-Triggered Real-Time Scheduling For Stabilization Of Passive and Output Feedback Passive Systems," *2011 American Control Conference,* 2011.

[45] H. Y. Wu, L. Lou, C. C. Chen, K. Kuhnlenz, and S. Hirche, "Distributed Computation and Data Scheduling for Networked Visual Servo Control Systems," *IEEE/Rsj 2010 International Conference on Intelligent Robots and Systems (Iros 2010),* pp. 6216-6221, 2010.

[46] 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,* vol. 56, pp. 2305-2318, Oct 2011.

[47] L. M. Yang, G. L. Liu, and Q. Zheng, "Bandwidth Scheduling Strategy with Deterministic Time-constraint for Networked Control Systems," *2011 6th IEEE Conference on Industrial Electronics and Applications (Iciea),* pp. 2011-2016, 2011.

[48] Z. L. Wang, Q. Wang, M. M. Wang, and C. Y. Dong, "Intelligent Scheduler Design for Networked Control Systems of Guided Weapon," *2011 6th IEEE Conference on Industrial Electronics and Applications (Iciea),* pp. 2454-2459, 2011.

[49] L. J. Xu, M. R. Fei, and T. C. Yang, "Non-cooperative Game Model Based Bandwidth Scheduling and the Optimization of Quantum-Inspired Weight Adaptive PSO in a Networked Learning Control System," *Life System Modeling and Intelligent Computing, Pt Ii,* vol. 98, pp. 8-15, 2010.

[50] H. Voit and A. Annaswamy, "Adaptive Control of a Networked Control System with Hierarchical Scheduling," *2011 American Control Conference,* pp. 4189-4194, 2011.

[51] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology,* vol. 10, pp. 438-446, May 2002.

[52] W. A. Zhang and L. Yu. (2007, 9). *Output feedback stabilization of networked control systems with packet dropouts*. Available: <Go to ISI>://000249665900017

[53] Q. M. Wu, Y. S. Li, and Y. Qin, "A scheduling method based on deadline for CAN-based networked control systems," *IEEE ICMA 2006: Proceeding of the 2006 IEEE International Conference on Mechatronics and Automation, Vols 1-3, Proceedings,* pp. 345-350, 2006.

[54] S. H. Hong and S. M. Song, "Transmission of a scheduled message using a foundation fieldbus protocol," *IEEE Transactions on Instrumentation and Measurement,* vol. 57, pp. 268-275, Feb 2008.

[55] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings, "Applying New Scheduling Theory to Static Priority Preemptive Scheduling," *Software Engineering Journal,* vol. 8, pp. 284-292, Sep 1993.

[56] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems,* vol. 35, pp. 239-272, Apr 2007.

[57] E. G. Schmidt and K. Schmidt, "Schedulability Analysis and Message Schedule Computation for the Dynamic Segment of FlexRay," *IEEE VTC* vol. Fall, 2010.

[58] K. Schmidt and E. G. Schmidt, "Optimal Message Scheduling for the Static Segment of FlexRay," *IEEE VTC,* vol. Fall, 2010.

[59] C. Peng, D. Yue, Z. Gu, and F. Xia, "Sampling period scheduling of networked control systems with multiple-control loops," *Mathematics and Computers in Simulation,* vol. 79, pp. 1502-1511, Jan 2009.

[60] K. Ji and W. J. Kim, "Optimal bandwidth allocation and QoS-adaptive control co-design for networked control systems," *International Journal of Control Automation and Systems,* vol. 6, pp. 596-606, Aug 2008.

[61] F. Xia, Y. X. Sun, and Y. C. Tian, "Feedback scheduling of priority-driven control networks," *Computer Standards & Interfaces,* vol. 31, pp. 539-547, Mar 2009.

[62] Y. Q. Wang, S. X. Ding, H. Ye, L. Wei, P. Zhang, and G. Z. Wang, "Fault detection of networked control systems with packet based periodic communication," *International Journal of Adaptive Control and Signal Processing,* vol. 23, pp. 682-698, Aug 2009.

[63] D. S. Kim, D. H. Choi, and P. Mohapatra, "Real-time scheduling method for networked discrete control systems," *Control Engineering Practice,* vol. 17, pp. 564-570, May 2009.

[64] D. Nesic and D. Liberzon, "A Unified Framework for Design and Analysis of Networked and Quantized Control Systems," *IEEE Transactions on Automatic Control,* vol. 54, pp. 732-747, Apr 2009.

[65] F. Göktas, "Distributed control of systems over communication networks," Thesis (Ph D in Computer and Information Science), University of Pennsylvania, 2000., 2000.

[66] N. Xi and T. J. Tarn, "Planning and control of Internet-based teleoperation," *Telemanipulator and Telepresence Technologies V,* vol. 3524, pp. 189-195, 240, 1998.

[67] L. A. Zadeh, "Outline of a New Approach to Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems Man and Cybernetics,* vol. Smc3, pp. 28-44, 1973.

[68] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice,* vol. 11, pp. 1099-1111, Oct 2003.

[69] J. W. Overstreet and A. Tzes, "An Internet-based real-time control engineering laboratory," *IEEE Control Systems Magazine,* vol. 19, pp. 19-34, Oct 1999.

[70] M. Y. Chow and Y. Tipsuwan, "Gain adaptation of networked DC motor controllers based on QOS variations," *IEEE Transactions on Industrial Electronics,* vol. 50, pp. 936-943, Oct 2003.

[71] F. L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Systems Magazine,* vol. 21, pp. 66-83, Feb 2001.

[72] Y. Halevi and A. Ray, "Integrated Communication and Control-Systems .1. Analysis," *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme,* vol. 110, pp. 367-373, Dec 1988.

[73] B. C. Kuo, *Automatic control systems*, 7th ed. Englewood Cliffs, NJ: Prentice Hall, 1995.

[74] R. Luck and A. Ray, "Experimental-Verification of a Delay Compensation Algorithm for Integrated Communication and Control-Systems," *International Journal of Control,* vol. 59, pp. 1357-1372, Jun 1994.

[75] H. Chan and U. Ozguner, "Closed-Loop Control of Systems over a Communications Network with Queues," *International Journal of Control,* vol. 62, pp. 493-510, Sep 1995.

[76] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays (vol 34, pg 57, 1998)," *Automatica,* vol. 34, pp. 1154-1154, Sep 1998.

[77] S. H. Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control-Systems," *IEEE Transactions on Control Systems Technology,* vol. 3, pp. 225-230, Jun 1995.

[78] Y. Tipsuwan and M. Y. Chow, "Network-based controller adaptation based on QoS negotiation and deterioration," *IECON'01: 27th Annual Conference of the IEEE Industrial Electronics Society, Vols 1-3,* pp. 1794-1799, 2207, 2001.

[79] L. Zhang and H. J. Fang, "A novel controller design and evaluation for networked control systems with time-variant delays," *Journal of the Franklin Institute-Engineering and Applied Mathematics,* vol. 343, pp. 161-167, Mar 2006.

[80] G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh, "A moving horizon approach to networked control system design," *IEEE Transactions on Automatic Control,* vol. 49, pp. 1427-1445, Sep 2004.

[81] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica,* vol. 42, pp. 953-958, Jun 2006.

[82] O. C. Imer, S. Yuksel, and T. Basar, "Optimal control of LTI systems over unreliable communication links," *Automatica,* vol. 42, pp. 1429-1439, Sep 2006.

[83] L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control,* vol. 49, pp. 1562-1572, Sep 2004.

[84] L. A. Montestruque and P. J. Antsaklis, "Static and dynamic quantization in model-based networked control systems," *International Journal of Control,* vol. 80, pp. 87-101, Jan 2007.

[85] G. P. Liu, J. X. Mu, D. Rees, and S. C. Chai, "Design and stability analysis of networked control systems with random communication time delay using the modified MPC," *International Journal of Control,* vol. 79, pp. 288-297, Apr 2006.

[86] A. S. Matveev and A. V. Savkin, "Optimal state estimation in networked systems with asynchronous communication channels and switched sensors," *Journal of Optimization Theory and Applications,* vol. 128, pp. 139-165, Jan 2006.

[87] A. S. Matveev and A. V. Savkin, "Optimal design of networked control systems: computer control via asynchronous communication channels," *International Journal of Control,* vol. 77, pp. 1426-1437, Nov 10 2004.

[88] A. V. Savkin, "Analysis and synthesis of networked control systems: Topological entropy, observability, robustness and optimal control," *Automatica,* vol. 42, pp. 51-62, Jan 2006.

[89] V. Malyavej and A. V. Savkin, "The problem of optimal robust Kalman state estimation via limited capacity digital communication channels," *Systems & Control Letters,* vol. 54, pp. 283-292, Mar 2005.

[90] M. Caccamo, T. Baker, A. Burns, G. Buttazzo, and L. Sha, "Real-Time Scheduling for Embedded Systems," in *Handbook of Networked and Embedded Control Systems*, D. Hristu-Varsakelis and W. S. Levine, Eds., ed Boston: Birkhauser, 2005.

[91] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *Journal of the Acm,* vol. 20, pp. 46-61, 1973.

[92] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," *Proceedings of the 41st IEEE Conference on Decision and Control, Vols 1-4,* pp. 1211-1217, 4789, 2002.

[93] Q. Ling and M. D. Lemmon, "Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain," *Proceedings of the 2003 American Control Conference, Vols 1-6,* pp. 4845-4850, 5426, 2003.

[94] K. Tindell, A. Burns, and A. J. Wellings, "Calculating Controller Area Network (Can) Message Response-Times," *Control Engineering Practice,* vol. 3, pp. 1163-1169, Aug 1995.

[95] J. Greifeneder and G. Frey, "Optimizing quality of control in networked automation systems using probabilistic models," *2006 IEEE Conference on Emerging Technologies & Factory Automation, Vols 1 -3,* pp. 615-622, 2006.

[96] B. C. Li and K. Nahrstedt, "A control-based middleware framework for quality-of-service adaptations," *IEEE Journal on Selected Areas in Communications,* vol. 17, pp. 1632-1650, Sep 1999.

[97] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*, 2nd ed. Philadelphia: Society for Industrial & Applied Mathematics, 1999.

[98] M. J. Osborne and A. Rubinstein, *A course in game theory*. Cambridge, Mass ; London: MIT Press, 1994.

[99] J. Y. Le Boudec, "Application of network calculus to guaranteed service networks," *IEEE Transactions on Information Theory,* vol. 44, pp. 1087-1096, May 1998.

[100] V. Firoiu, J. Y. Le Boudec, D. Towsley, and Z. L. Zhang, "Theories and models for Internet quality of service," *Proceedings of the IEEE,* vol. 90, pp. 1565-1591, Sep 2002.

[101] M. Velasco, P. Marti, and M. Frigola, "Bandwidth management for distributed control of highly articulated robots," *2005 IEEE International Conference on Robotics and Automation (ICRA), Vols 1-4,* pp. 265-270, 4756, 2005.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Şenol, Sinan
Nationality: Turkish (TC)
Date and Place of Birth: 30 April 1976, Ankara
Marital Status: Single
Phone: +90 312 2153107
email: e102374@metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	METU Electr.-Electro. Eng.	**2004**
BS	METU Electr.-Electro. Eng.	**2000**
High School	Atatürk Anad. H. S., Ankara	**1994**

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2002-Present	Aselsan A.S.	Senior Expert Engineer

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1) **S. Şenol, K. Leblebicioğlu, and E. G. Schmidt, " INtERCEDE: An algorithmic approach to networked control system design", Journal of Network and Computer Applications, Volume 34 Issue 4, July, 2011, Pages 1326-1341.**

2) **S. Şenol, K. Leblebicioğlu, and E. G. Schmidt, "FlexRay Simulator Development For Networked Control System," presented at the IEEE SIU, 2007.**

3) **S. Şenol, K. Leblebicioğlu, and E. G. Schmidt, "Message Scheduling Algorithms for Networked Control Systems with Time Slotted Communication Protocols," presented at the IEEE SIU, 2008.**

4) **S. Şenol, K. Leblebicioğlu, and E. G. Schmidt, "Ağ Tabanlı Kontrol Sistemleri için Kontrol Kalitesi Tanımı ve Normalizasyonu," presented at the SAVTEK, METU, 2010.**

HOBBIES

Books, Various Technologies, Movies, Music, Games