University of Strathclyde

Industrial Control Centre
Department of Electronic and electrical Engineering

Development of Integrated Methods for Control of Networked Control Systems

Luis Felipe Recalde Camacho

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

2010

Copyright statement

Signed: *Date:*

Acknowledgements

This work has been supported by the Programme Alβan, The European Union Programme of High Level Scholarships for Latin America, scholarship No.E07D402078EC.

I would like to express my deep gratitude to my Supervisor Dr. Reza Katebi who continuously provided me with vision and wisdom. His patient guidance and experience were indispensable through the completion of this thesis.

Special thanks to my former co-supervisor and friend Dr. Leonardo Giovanini. His broad knowledge led me to give my ideas a theoretical background and showed me that the simplest idea is always the best one.

Finally, but most importantly, this thesis is dedicated to my family who always believed in me.

Luis Felipe Recalde Camacho

Glasgow, Uk

October 2010

Content

List of Figures

Abbreviations

ADS	**A**synchronous **D**ynamical **S**ystems
CAN	**C**ontroller **A**rea **N**etwork
CAS	**C**ontrol **A**ction **S**elector
CDF	**C**ontinuous **D**istribution **F**unction
DC	**D**irect **C**urrent
DCS	**D**istributed **C**ontrol **S**ystems
DMPC	**D**istributed **M**odel **P**redictive **C**ontrol
DMMPC	**D**istributed **M**ultirate **M**odel **P**redictive **C**ontrol
FDE	**F**unctional **D**ifferential **E**quations
GPC	**G**eneralized **P**redictive **C**ontrol
IMC	**I**nternal **M**odel **C**ontrol
IP	**I**nternet **P**rotocol
ITAE	**I**nternal **T**ime **A**bsolute **E**rror
LAN	**L**ocal **A**rea **N**etwork
LMI	**L**inear **M**atrix **I**nequality
LQG	**L**inear **Q**uadratic **G**aussian
LQR	**L**inear **Q**uadratic **R**egulator
LTI	**L**inear **T**ime **I**nvariant
MAC	**M**edium **A**ccess **C**ontrol
MADT	**M**aximum **A**llowable **D**elay **T**ime
MIMO	**M**ultiple-**I**nput **M**ultiple-**O**utput
MPC	**M**odel-based **P**redictive **C**ontrol
NCS	**N**etworked **C**ontrol **S**ystems
NPC	**N**etworked **P**redictive **C**ontrol
ODE	**O**rdinary **D**ifferential **E**quations
OSI	**O**pen **S**ystem **I**nterconnect

PID	**P**roportional **I**ntegral **D**erivative
PROFIBUS	**P**rocess **F**ield **B**us
QoS	**Q**uality **o**f **S**ervice
ZOH	**Z**ero **O**rder **H**old
RS232	**R**ecommended **S**tandard **232**
SCADA	**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SISO	**S**ingle-**I**nput **S**ingle-**O**utput
SNS	**S**ensor **N**etwork **S**ystems
TDS	**T**ime-**D**elay **S**ystems
TOD	**T**ry-**O**nce-**D**iscard
WAN	**W**ide **A**rea **N**etwork

Symbols

$\tau(t_k)$	Network-induced time delay
$\tau(k)$	Discrete network-induced delay
h or h_0	Sampling interval
h_{int}	Inter-sampling interval
$\tau_{\min}$	Minimum network delay
$\tau_{\max}$	Maximum network delay
$\tau_{SC}(t_k)$	Sensor-to-controller delay
$\tau_{CA}(t_k)$	Controller-to-actuator delay
*	Lost Packet
$\sum_1(t)$	Subsystem 1: Networked Controlled System model
$u(t)$	Subsystem 1 inputs
$x(t)$	Subsystem 1 states
$y(t)$	Subsystem 1 outputs
$q(y)$	Quantized outputs
$\sum_2(k)$	Subsystem 2: discrete controller model
$v(k)$	Controller input
$x_c(k)$	Controller states
$w(k)$	Controller output
K	Linear controller gain
$\sum_o(k)$	Observer system model
$\Gamma(s)$	Gamma distribution function

$\xi(k)$	Process noise
$\zeta(k)$	Measurement noise
δ_{ji}	Dirac delta
$\Delta(k-i)$	Kronecker delta
$\mu_x(0)$	Mean of the state
$\Pi(0)$	Covariance matrix of the state
$\phi(k-1)$	Packet delivery indicator
$\gamma(k-1)$	Packet dropping rate
γ	Sweeping factor
K_p	PID proportional gain
K_i	PID integral action
K_d	PID derivative action
$V(.)$	Lyapunov function
ϕ	System initial conditions
Ω	Convex set
α	Weighting factors in DMMPC

Abstract

Networked Control Systems (NCS) are feedback/feed forward control systems where control components (sensors, actuators and controllers) are distributed across a common communication network. With the introduction of the communication network in the control system, large-scale applications, multi-level distributed control applications and remote applications can be easily developed, but represent a challenge for control design.

Finite communication channels introduce network constraints or communication errors (packet dropouts, network delays and variable sampling time) that reduce the reliability of the system measurements as well as the stability and robustness of the overall system.

In this thesis, we investigate NCS under packet dropouts and network-induced delays that are bigger than the sampling time. To compensate network constraints two approaches are proposed. A non-recursive approach to compensate lumped network delays is developed by using the probability distribution function (pdf) of the delay in the design of PID controllers.

A recursive approach is also developed to compensate sensor-to-controller delays and packet dropouts. The methodology combines a time-driven Kalman filter with a bad data detector. The resulting estimated states are constraints-free and can be used to design the control system. The control law is based on Model Predictive Control (MPC) to compensate the controller-to-actuator delays recursively.

Both methodologies are initially applied to a single loop NCS. Multi-loop NCS are modelled as Distributed control systems (DCS) to decouple a multi-loop problem into

single loop problems and to solve a multiobjective control problem. The multiobjective formulation allows the implementation of the previous methodologies on multivariable NCS. The resulting Distributed control problem is formulated as multirate to decrease controller-to-actuator packet dropouts.

Chapter 1

Introduction

Any attempt of using a communication network to exchange information among control system components (sensors, actuators and controllers) can be classified as Networked or Networked-based Control System.

Wiener's ideas about the incorporation of information theory in feedback control theory revealed the critical aspects of the complex interconnected systems [1]. This complexity has been recently noted with the remarkable deployment of technologies in computer and communication networks for many parts of the industry.

The development of NCS came as both, the result of incorporating data networking technologies into large-scale applications [97]; and the limited capacity of the communication channel to transmit observations and control signals for remote applications [67]. To reduce the complexity of such systems with nominal economical investments, distributed control methodologies were employed based on network capabilities. Before the 1990's, the first formal attempt to describe these new systems appeared in the work of Halevi and Ray [32], however the fusion between communications and control systems as NCS appeared in the work of Walsh et al. [94].

Introduction to Networking Technologies for Distributed Applications

As mentioned by Almutairi et al. [2], data networking technologies have been widely applied to industrial and military control applications. Large-scale applications during the 1970's and 1980's pushed the development of control coordination schemes to share control tasks that were physically separated. The existing computational burden on centralized applications was mitigated with decentralized schemes. As reported in a paper in 1973, "on the Stabilitization of Decentralized Control Systems", stability and performance problems came together with the idea of decentralization [95]. Quasi-decentralized schemes became popular. They combined the good centralized performance with the distribution of decentralized schemes. Consequently, cross communications were necessary, as well as large bandwidth on the shared communication channels.

During the 1980's quasi-decentralized schemes eased the implementation of multilayer and hierarchical control for industrial processes. To implement hierarchical control, the system has to be decomposed into several sub-problems that represent a multilayer or multilevel architecture. Its implementation depends on the availability of the communications and real-time constraints.

Real-Time Constraints

Although there is not an established theory for real-time systems to manage timing constraints in real-time applications, the development of communication networks has allowed the achievement of accurate operations with logical results at the time at which they are produced.

A particular problem in large-scale industry is how to accommodate changing requirements to create minimal disruption to normal operations. Real-time systems implemented on large-scale applications offer correctness, timeliness and reliability for stable operating conditions. Nonetheless the more hardware used, the more tasks and communication problems appeared and required solution.

To satisfy timing requirements, effective resource allocation strategies were applied. However, to ensure correctness, timing constraints are assumed implicitly. These assumptions represent system components that are extremely difficult to integrate, as well as limited channel capabilities and processing time requirements.

Network-Based Applications

The necessity of real-time protocols pushed engineers to consider faster and more reliable communication networks to achieve the real-time systems capabilities. Most of the applications had proprietary software, but the communication media had been developed using agreed standards.

Standards of Industrial Networks appeared since 1969 with the well-known RS232 proposed by the Electronics Industry Association; however the major breakthrough was the standardization of the Open System Interconnection (OSI) model. The first commercial distributed control systems appeared in the 1970's. At the beginning, networks were only serial wired links; nonetheless with the development of communication protocols, networks could provide several increasing benefits of communication, reduction of wiring connections and ease of maintenance, among others.

The impact of these benefits influenced the development of industrial protocols such as PROFIBUS which was developed in 1987 followed by Fieldbus and DeviceNet. These protocols offered robustness for real-time control purposes and allowed the implementation of Process Control Systems. Practical applications may also require the modification of the protocols for Internet connectivity.

SCADA Applications

Networking Control Systems and Process Control Systems refer to systems able to monitor and measure hundreds of variables either locally or remotely.

Supervisory Control and Data Acquisition Systems (SCADA), is one of the most known Process Control Systems. It has been conceived as an ad-hoc technique based on functionality, self-configuration, failure resistance and security. SCADA technology started almost 30 years ago. Due to the major reduction of computers and networks cost, applications extended from central master stations to remote master stations. SCADA systems can easily implement multi-level distributed control schemes. These solutions offer redundancy, ease of maintenance, low cost, flexibility; upgrading and more flexible architectures to inter-operate with complementary systems such as Sensor Networks and Mesh Networks.

The advances in the implementation of SCADA systems have begun with the migration from monolithic architecture to networked architecture. Some of the most common applications of SCADA systems are in water and waste control, energy, oil and gas refining and transportation.

Originally SCADA systems employed proprietary software, but with the increase use of personal computers, office-base computer networking can be achieved. Networked SCADA systems offered new possible applications as well as Internet protocol (IP) for communications.

Nowadays, with the introduction of commercial communication networks in the industry, the functionalities of Networked SCADA systems can be applied in several large-scale applications, multi-level distributed control applications and remote applications as NCS.

1. 1 Motivation of the Research

NCS applications offer low cost, ease of maintenance, flexibility, upgrading, redundancy and scheduling. Today's networks offer high connectivity and medium to high data transmission rates. Transmission policies allow dynamic bandwidth allocation or scheduling methodologies to meet real-time constraints and furthermore

broadcasting data allows coordination of the control system and implementation of centralized schemes but with decentralized structures.

Connectivity of the control components is ad-hoc, only bounded by the NCS Information structure availability. This Information structure, that is connectivity and capacity of the communication network, as mentioned by Giovanini and Balderud [29] offers any type of coordination on the system scheme, leading to real-time implementation of distributed control systems with certain complexity. The complexity of NCS is not only in the process; but in the way the information is deployed through the communication network. Communication networks introduce inherent communication errors that carries overload of information over real-time systems.

To fully deploy the capabilities of the NCS, optimum performance of distributed control systems combined with Quality of service QoS-based networks are needed. In the study of NCS, network constraints and link failures have to be considered. Not all the communication channels are ideal data transmission medium (without errors). Missequencing, packet dropouts, network-induced delays and limited network bandwidths emerge when a control loop is closed across the network. These network constraints reduce performance of the NCS by affecting the observability of the process as well as the reliability of output measurements and control signals.

Almost all the literature in NCS is referred to stability. Regardless of the method to be used, sufficient conditions are formulated for the stabilization of the NCS either for packet dropouts, network delays, dynamic bandwidth allocation or quantization constraints.

Stability in the system will decrease as the delay grows. Finite communication channels introduce a maximum network delay before the system goes from closed loop to open loop. Nonetheless, reaching this value will depend on the network traffic. Network traffic is not always constant and consequently adds the time-varying nature to the delay. This characteristic makes the robustness of the system to vary in an intermittent or oscillatory way. Thus a less conservative stability analysis

can be useful using concepts such as practical stability [50]. This concept of stability is weak but applicable when combined with recursive state estimation and control methodologies, and is the main motivation of this research.

State Estimation and Control Methodologies in NCS are not that simple to implement. A trade-off between optimality and computational burden is needed for the methodology to be applicable. Conventional techniques for systems with network constraints require the constraints to be known and constant which contradicts variable and random nature of the network constraints. Recursive solutions can handle the variability and randomness of the delay as far as it is known at the beginning of the recursion. In some applications when uncertainty varies every sample time, updating states using measurements before each optimization step increases performance of the closed-loop.

1.2 Aims and Objectives

The aim of this thesis is to study the effects of distributing the control objective across the network to fully deploy networks capabilities. To achieve this aim, the following research topics are studied:

- *PID design for NCS:* The stability and robustness of PID controllers on NCS are studied. The controller design is subject to the following assumptions: The system is continuous with delayed inputs; sensors are time-driven; controller and actuators are also time-driven; packet dropouts as well as varying sampling intervals are not included. The controller is a discrete PID controller where its parameters are calculated by solving a Linear Matrix Inequalities (LMI) problem; and finally the system can be proved to be delay-dependent stable.

- *Kalman filtering design for delay and packet dropouts compensation:* For delays bigger than the sampling interval on the sensor-to-controller interface, a modified Kalman filter is used. The filter is time-driven. In the absence of information arrivals, the filter forwards the estimated states to compensate delays. The filter is combined with a bad data detector to differentiate between packet dropouts or delays increments. The proposed filter/detector is combined with a MPC to compensate delays on the controller-to-actuator interface. Its design is based on the method used for PID controllers analyzed in Chapter 3.

- *Model Predictive Control for Distributed NCS:* Computational issues can be addressed extending the NCS problem to a Distributed NCS problem. The Multivariable system is split into m agents or entities that can sense the states of the system and decide upon the values of its control variables. The process is observed by single channel modified Kalman filters and the control design is performed using a multiobjective optimization combined with model predictive control.

1.3 Outline of the Thesis

The structure of the thesis is described as follows:

1.3.1 Chapter 2 Networked Control Systems

This chapter describes a state of the art of the network constraints such as: network delays, packet dropouts, quantization and missequencing. General structures for NCS are also discussed. Network delays are defined to be finite and packet dropouts are defined as the absence of information arrivals. This chapter is structured such that in section 2.1 the NCS features are presented. The general structure of NCS is presented

in section 2.2. Network constraints are described in section 2.3. The way information can be updated on NCS is shown in section 2.4.

1.3.2 Chapter 3 Networked Control Systems modelling

This chapter provides a state of the art review of the different control methodologies employed for NCS in conjunction with the information structure. A functional differential equation is used to model NCS for linear systems. Some discretization methodologies are used to demonstrate the NCS existing models and the most used control methodologies are addressed based on the information structure. Section 3.1 shows different methodologies to model a linear time invariant NCS with network delays. Control strategies are presented in section 3.2

1.3.3 Chapter 4 PID Controller design for NCS: a pseudo-probabilistic approach

A non-recursive control methodology is applied to a single-loop NCS. The NCS is assumed to be affected by network delays only. PID controllers are designed using the pdf of the lumped delay. The incorporation of the pdf of the delay into the controller design leads to a set of LMI's. This set is bounded to a range of network delays that includes the most probable network delay. The optimization is expected to achieve good system performance. The structure of this chapter is as follows: In section 4.1 a discrete NCS model with delayed inputs is combined with a discrete PID controller. The pdf of the delay is used to model network delays. In section 4.2 the resulting closed loop system is described as a convex set based on the pdf of the delay. The convex set is reduced to a limited polytope bounded by a tuning parameter. In section 4.3 the PID controller design is solved by formulating a LMI problem. Performance and stability are included as constraints. A numerical example is presented in section 4.4 to test the design approach.

1.3.4 Chapter 5 Estimation Approach for Networked Control Systems

In this chapter a recursive control strategy is applied to single-loop NCS with packet dropouts in the sensor-to-controller interface and network delays longer than the sampling time in either the sensor-to-controller and controller-to-actuator interfaces. A Kalman filter is combined with a bad data detector to compensate network delays and packet dropouts at the sensor-to-controller interface. The resulting network constraints-free system states are used to determine a MPC controller. Controller-to-actuator delays are compensated by forwarding the control law using the calculated value of the network delay. The structure of this chapter is as follows: Section 5.1 concerns the formulation of the discrete Kalman filter for systems with neither delays nor packet dropouts. The resulting algorithm is modified to include sensor-to-actuator delays and a bad data detector is formulated to use the resulting filter to compensate for both delays and packet dropouts. In section 5.2, controller-to-actuator delays are incorporated in the system model and two control methodologies are proposed Linear Quadratic Regulator (LQR) with model transformation and closed loop MPC. A numerical example is presented in section 5.3. The resulting control methodology is a combination of the Kalman filter for estimating the system states affected by network constraints and closed-loop MPC.

1.3.5 Chapter 6 Distributed Multirate Approach for NCS

Multivariable NCS is modelled as a Distributed system and split into a known number of subsystems. Distributed MPC (DMPC) is used to control the multivariable NCS and Nash stability is used to achieve stability of the overall system. Single-channel constraints are compensated individually using the results of chapter 5. The resulting methodology is modelled as multirate systems in order to reduce the effects of packet dropouts in the controller-to-actuator interface. The resulting methodology is Distributed Multirate MPC (DMMPC) and compensates entirely the network constraints. The structure of this chapter is as follows: section 6.1 focuses on the disadvantages of centralized control from an estimator perspective. Section 6.2 offers a solution to the trade-off between scheduling policies and control design complexity by formulating the control problem within a distributed scheme, global optimality is

achieved as Nash optimal. Section 6.3 concerns the addition of a packet dropout mechanism which leads to a suboptimal stochastic controller design. Sub-optimality constraint is removed in section 6.4 by formulating the controller design problem as a DMMPC problem. A case study is presented in section 6.5 to test the resulting algorithm.

1.4 Contribution of the Thesis

1. A critical review of the state of the art of NCS is presented in chapters 2 and 3. Several survey papers on NCS show the most known control methodologies applied to NCS, but why these methodologies are chosen is explained briefly. This review emphasizes that information structure (connectivity and capacity) is a limiting factor and plays an important role when the control and estimation methodology is chosen.

2. PID controllers are well-known and widely used in industry, and consequently within NCS. Methodologies that incorporate the delay to tune a PID controller have been previously developed. Most of the resulting methodologies emphasize on stability and consequently new methodologies that consider performance are needed. In this thesis we present a PID control design for NCS that incorporates a range of delay values that maintains a good system performance and considers stability for NCS as practical stability.

3. In the literature of NCS, one of the simplest estimation strategy is the one presented by Larsen et al. [51]. This strategy uses a modified Kalman filter to avoid model augmentation and is aimed at compensating sensor-to-controller delays. In this thesis we use the modified Kalman filter developed by Larsen et al. [51] combined with a bad data detector. The filter-detector combination allows the compensation of not only network delays but also packet dropouts for NCS with time-driven components. The calculated network delay is also used to forward a MPC law and compensates network delays in the controller-to-actuator interface.

4. The resulting filter-detector-controller strategy developed for single channels is used for multivariable NCS by modelling the NCS as a distributed control system. It is shown that compensation of network constraints for individual channels is easier than multivariable NCS but stability can be reduced due to the resulting distributed system. To avoid instability, the DMPC developed by Giovanini and Balderud [29] is used. Furthermore, due to the information structure of the resulting distributed NCS (DNCS), a strategy to reduce the effects of packet dropouts in the controller-to-actuator interface is developed by modelling the resulting DNCS as multirate. The resulting DMMPC uses the inter-sampling control laws of every subsystem as acknowledgement messages to compensate in case of packet dropouts. Packet dropouts can not be entirely compensated but, the proposed methodology is able incorporate a measurement of the quality of the transmitted information as a controller design parameter.

1.5 Publications arisen from this Research

The contribution of this thesis has been presented in the following publications:

1. Recalde, L. F., Katebi, R. (2008): PID Control design for Networked Control Systems: A pseudo probabilistic Robust Approach. IFAC Control Conference, Soul, Korea.

2. Recalde, L. F., Katebi, R. (2010): State Estimation and Control Design of Networked Control Systems. IFAC Control Conference, Milan, Italy. Submitted for conference paper.

3. Recalde, L. F., Katebi, R. (2010): Model-based Predictive Control of Networked Control Systems. IFAC Control Conference. Milan, Italy. Submitted for conference paper.

4. Recalde, L. F., Katebi, R. (2010): Delay-dependent PID Control design for Networked Control Systems: a probabilistic Approach. Submitted for Journal publication. International Journal of Control.

5. Recalde, L. F., Katebi, R. (2010): Recursive compensation of packet dropouts and network delays in Networked Control Systems. Submitted for Journal publication. IEEE, Signal Processing.

Chapter 2

Networked Control Systems

Networked Control Systems (NCS) are feedback/feed forward control systems where control components (sensors, actuators and controllers) are distributed across a common communication network [107] and [87]. For control design NCS is suitable on multi-level distributed control as well as remote control [101]. Industrial applications include power plants, flight and automotive control systems, robotics, automated manufacturing, control over Internet and environmental monitoring among others.

The design of NCS depends to a particular design methodology in order to be feasible and reliable. A NCS solution offers redundancy, ease of maintenance, low cost, flexibility and upgrading [54].

To fully deploy the capabilities of the NCS, optimum performance of distributed control systems combined with Quality of service QoS-based networks are needed [54]. Today's networks offer high connectivity and medium to high data transmission rates. Transmission policies allow dynamic bandwidth allocation or scheduling methodologies to achieved real-time constraints. Furthermore broadcasting data allows coordination of the control objective and implementation of centralized schemes but with decentralized structures.

2.1 Networked Control System Features

When distributed control processes are sharing a common communication network, asynchronous operations as well as multirate sampling intervals are expected. Network utilization is affected by the increase of data traffic. Contention is also important for reliable transmissions. Contention depends on buffer size and collision detection/correction mechanisms. The way the Medium Access Control (MAC) sublayer protocol controls the transmission and resolves contention introduces time varying delays [54]. MAC describes the protocol for obtaining access to any network. Cyclic service networks, like Token Bus and Ring, add a deterministic behaviour to data traffic. On the other hand, random access network such as Ethernet makes data traffic probabilistic [90].

For control design, the features of a NCS involve distributed scenarios, asynchronized multirate sampling and data traffic either deterministic or probabilistic [107]. Thus, a suitable approximation of the NCS problem is as time-varying systems, asynchronized and distributed. The idea of distribution in NCS can be understood from several aspects:

2.1.1 NCS in geographically distributed systems

Firstly, the distributed nature of any NCS can be referred to as a physical distribution. In this context, the whole process is distributed over a geographical region and either centralized or decentralized control architectures may be implemented [101].

Large-scale systems based their control schemes on either decentralized or centralized control. The scheme used will mostly depend on communications availability. In some applications the complexity of a centralized scheme makes itself less attractive, however decentralized scheme implementations also cause performance degradation. A quasi-decentralized scheme offers significant improvement in the trade-off between

performance and distribution. A quasi-decentralized application increases implementation costs because some signals are transferred between local subsystems [101].

With the use of inexpensive communication networks within NCS schemes, control and process information are conveyed all over the system. Former communication limitations, due to individual wiring, can be completely removed on either a centralized or quasi-decentralized scheme. Hence NCS schemes allow distributed subsystems being integrated into a large system.

2.1.2 NCS with Sensor Networks

The use of Sensor Networks Systems (SNS) in a closed-loop system is considered as NCS where the sensors are physically distributed and the network has been enhanced for data traffic purposes [77]. Individual sensors allow limited capabilities. Nonetheless, sensor networks accomplish local cooperation, aggregation and data processing [99].

Sensors are easily added when needed (ad-hoc network). The information deployed by the sensors can be transmitted between them and fused before it reaches the controller [77]. SNS are a combined solution of the communications problem and the control problem. Effective congestion management, bandwidth allocation, queuing and network partitioning can be solved with the implementation of novel protocols for real-time systems and data processing; and consequently, any control methodology can be implemented.

Fig. 2.1, as presented in [77], represents a NCS with wireless ad-hoc sensor networks. S-blocks represent ad-hoc wireless sensors. A-blocks represent the actuators, C-blocks represent the controller and F-blocks are filters where both data fusion and information processing is made.

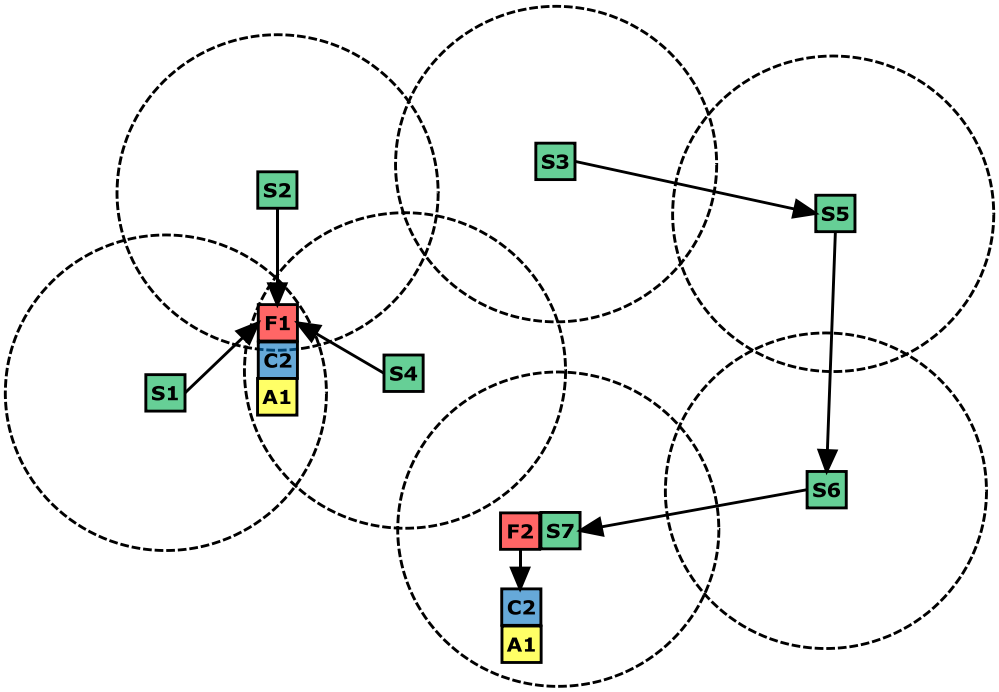


Fig. 2.1 Sensors, controllers and actuators in an ad-hoc wireless network

Even with reliable communication networks, network-delays are inevitable but can be modelled as time-varying delays. It is convenient to consider these delays into the system model by using either deterministic or stochastic models [23].

2.1.3 NCS in locally distributed systems

Networking Control Systems can easily be implemented using multi-level distributed control schemes. Common applications of locally distributed systems are vehicle control systems. The distribution is limited to a local environment where sensors, actuators and a main controller are connected to a Control Area Network CAN-type bus [101]. Multirate asynchronous sampling loops make the system multi-level and distributed. Nonetheless, a centralized control scheme is expected.

2.2 General Structure in NCS

NCS can be considered as the integration between Distributed Control, Sensor Networks and Communication Networks. NCS provides industrial solutions that fall into the features of both centralized and decentralized control schemes as presented in Fig. 2.2. The more information of the overall system, the more NCS resembles a centralized scheme and vice versa. Stability will depend on local stability (each loop) and global stability (overall system).

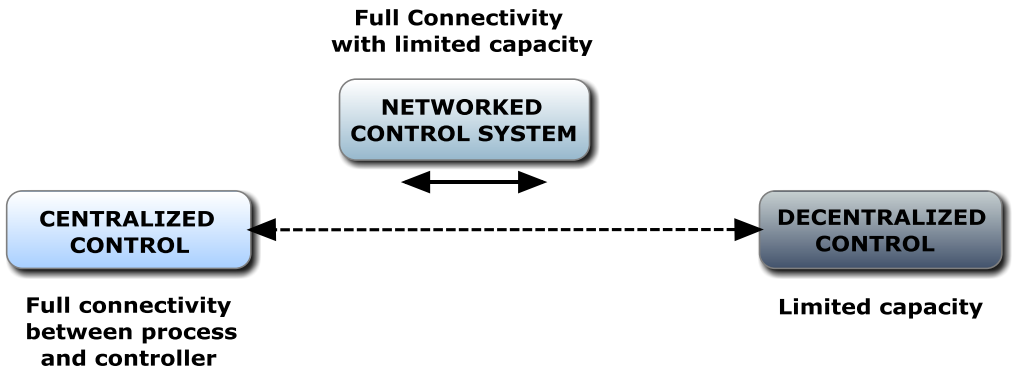


Fig. 2.2 NCS Features

Components distribution divides NCS into direct structures, hierarchical structures, and application-oriented structures (structures that depend on the application).

2.2.1 Direct structure

Any NCS expressed as a direct structure is composed of controllers remotely located from the plant, Fig. 2.3. The network is used to transmit both sensor measurements and control signals as independent packets. Receiver and transmitter queues appear due to buffer contentions. Single-input single-output (SISO) NCS can be easily implemented within this structure. A particular application of direct structures is the use of remote control on DC motors [101].

A variation of the direct structure is presented in Fig. 2.4 [77]. In this scheme the controllers are implemented locally (actuators and controllers are inside the process) and the sensors are physically distributed. This structure is suitable to implement wireless sensor networks for environmental monitoring and plant automation where wiring is almost impossible.

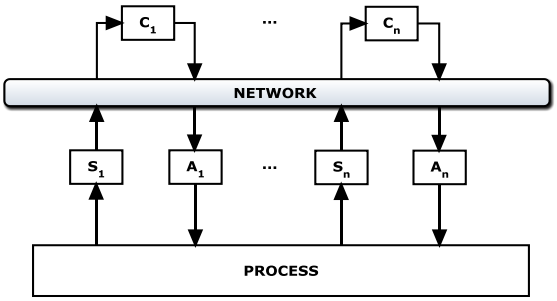


Fig. 2.3 NCS Direct Structure

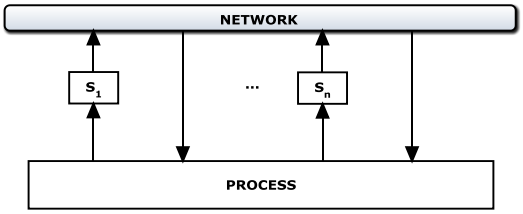


Fig. 2.4 NCS Direct Structure with local controller

2.2.2 Hierarchical structure

NCS in a hierarchical structure can be implemented as a two-layer scheme [101]. Local controllers guarantee the stability of simple processes and a main controller connected through a network can be used as a parameter optimization controller, adaptive controller or coordination manager controller. Further more Fault detection can be implemented. This structure is presented in Fig. 2.5

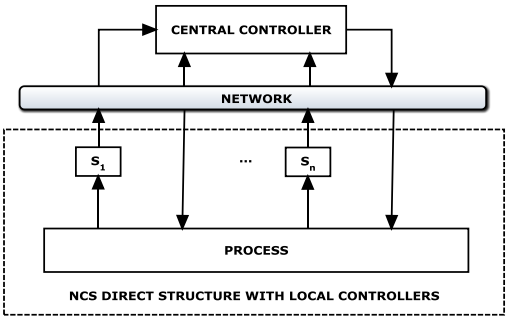


Fig. 2.5 NCS Hierarchical Structure

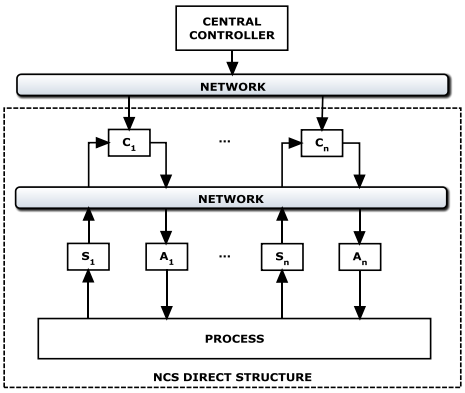


Fig. 2.6 NCS Hierarchical Structure with an industrial network

Most of the industrial applications can be considered as hierarchical including networked SCADA systems. On the other side, some authors use this structure for mobile robots and teleoperation [90].

Industrial applications incorporate local controllers connected through a field industrial network [101] in the low level. In the high level or supervisory level, computers coordinate low-level control actions. Supervisory levels can be implemented remotely using a Local Area Networks (LAN), Wide Area Networks (WAN) or the Internet [101]. The structure is presented in Fig. 2.6.

2.3 Network Constraints

In any network, there is an effective bandwidth for transmission purposes. This bandwidth is defined as the maximum amount of meaningful data of a certain size in bytes (packets) that can be transmitted per unit of time [53]. The utilization of this bandwidth depends on the packet size, the nodes requirements such as sampling times and synchronized operations, and the MAC sublayer protocol that controls the transmission of the information [53]. In control applications the number of nodes that share the network can vary. The network can be dedicated (only for control purposes) or shared (as in remote control applications over the Internet), and more importantly, the role of the protocols in the way the nodes are transmitting will affect the achievement of time-critical requirements. Here, nodes represent the control components as well as other unrelated nodes.

All these factors are responsible for transmission delays. Once the nodes are transmitting, the MAC protocol is responsible for resolving contention in the network. Not all the network traffic is due to successful transmissions; the type of data transmitted across the bus architecture is a countless number of small packets [53]. Thus, collisions are usual among nodes attempting to transmit; and the way the MAC solves these collisions adds either a random or deterministic time-varying delay.

2.3.1 Network-Induced delay

In NCS, there exists a transmission delay in each channel. This delay affects the performance of the system by pushing closed-loop stability conditions within the boundaries of the network stability.

For control purposes, the delay has to be bounded. A practical assumption is to define this delay as a scalar nonnegative time-varying function of time $\tau(t_k)$ on the sampling interval $t_k \in [kh,(k+1)h)$ [32], with h being a constant sampling time. $\tau(t_k)$ is commonly discrete and varying $\tau(k)$, which makes it different from the well-known plant pure delay and computational delay [101].

$\tau(k)$ appears in every single control component, attempting to transmit information (output measurements or control signals), independently. Thus $\tau(k)$ also represents the total time delay to transmit data from the source node to the destination node in a sampling interval and can be expressed as three components: the time at the source node, at the network channel and at the destination node [101].

At the source node, the delay time consists of computation time and waiting time. Waiting time is critical in network traffic. This consists of the time a message waits queuing in the buffer (queuing time T_{queue}) and the time a message waits once the node is ready to sent it (blocking time T_{block}).

At the network channel, the time delay is a combination of the transmission delay and the propagation delay. This delay depends on message size, data rate and length of the network cable [53]. Once the data has reached the destination node, there is a delay due to decoding and computation processes.

The delays at the network channel and at the destination channel can be calculated based on the network specifications and represent $\tau_{\min}$. However, the delay at the

source node, and more specifically the delay due to waiting time is difficult to analyze and is the responsible for the time varying nature of the network [53].

2.3.1.1 Varying nature of the delay

The time a message has to wait before it is sent across the communication network depends on collisions, contention and transmission mechanisms. It varies from network to network and is difficult to determine. Waiting time depends on the blocking time and periodicity of the messages. Blocking time is the time a packet must wait once a node is ready to send it, thus it is protocol-dependent [53] and depends the way protocols manage transmissions and collisions.
In some cases non-standard protocols can be used to discard old messages and set the queuing time to zero. Some authors have exploited this possibility by designing control-oriented protocols such as TOD (try-once-discard) [41] and other dynamic bandwidth allocation methods.

The way each node accesses the communication network can be random or prioritized. For Ethernet-based networks, the node that wants to transmit listens to the communication network and transmits once the network is idle. In case of collision, the transmitting node stops transmitting and waits T_j units of time to retransmit. This random time is determined by a Binary Exponential Back-Off algorithm where T_j is taken from the following distribution:

$$T_j \in \begin{cases} \left[0,\left(2^j-1\right)\right] & 0 \le j \le 10 \\ [0,1023] & 11 \le j \le 16 \end{cases} \quad (2.\ 1)$$

After 16 attempts, the node stops transmitting and a failure report is sent. Hence the blocking time has a probabilistic behaviour [39]. The collision mechanism is shown in Fig. 2.7 as presented in [40]:

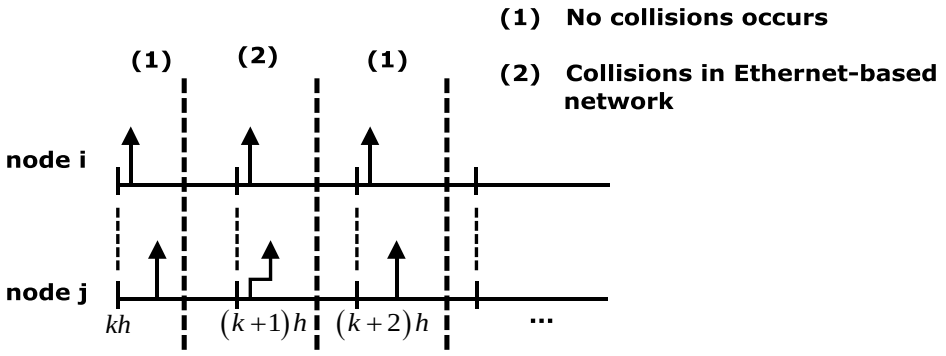


Fig. 2.7 Collision mechanisms in Ethernet Communication network

and can be calculated as follows:

$$E\{T_{block}\} = \sum_{i=1}^{j} E\{T_i\} + T_{resid} \tag{2. 2}$$

$E\{.\}$ is the expectated value of the time and T_{resid} denotes residual time that is seen by a node until the network is idle.

2.3.1.2 Delay distribution

Under heavy traffic load, the time-varying nature of the delay can hardly be deterministic [101]. Network traffic can be formulated as a probabilistic process where the distribution associates the probability that a particular delay happens. One statistical model of the delay was presented by Mukherjee [69] as a Weibull distribution. However, delay distribution implies delays between consecutive packets no correlated and also independent of the network utilization [77]. Correlation between packets was used by Nilsson [72] with a Markov chain of several delay distributions.

If the applications are extended to Internet, traffic load is Poisson-like [69]. Being $\tau(t_k)$ a random variable and equal to:

$$\tau(t_k) = \tau_{\min} + T_{block} \tag{2. 3}$$

Mukherjee [69] proposed that the delay resembles a gamma density function. If $\tau(t_k)$ is random variable and $\tau_{\min}$ be a constant, then $\tau(t_k)-\tau_{\min}$ is approximately gamma distributed or resembles a gamma distribution function.

Proof. Let T_j be a gamma distributed random variable with density function given by:

$$f_s(T_j) = \frac{(T_j)^{s-1} e^{-T_j}}{\Gamma(s)} \tag{2. 4}$$

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt \tag{2. 5}$$

with m being the scale and s the shape of the distribution. Using the Continuous Distribution Function (CDF) of T_j as follows:

$$\begin{aligned} \Pr\{T_j \le T_{block} \le T_j + \Delta T_j\} &= \frac{(T_q)^{s-1} e^{-T_q}}{\Gamma(s)} \Delta T_j \\ \Pr\left\{T_j \le \frac{\tau - c}{m} \le T_j + \Delta T_j\right\} &= \frac{(T_q)^{s-1} e^{-T_q}}{\Gamma(s)} \Delta T_j \end{aligned} \tag{2. 6}$$

and $\Pr\{mT_j + c \le \tau(t) \le mT_j + c + m\Delta T_j\} = \Pr\{\tau \le \tau(t) \le \tau + \Delta\tau\}$, then:

$$f_{s,m}(\tau(t_k)) = \frac{\left(\frac{\tau - c}{m}\right)^{s-1} e^{-\left(\frac{\tau-c}{m}\right)}}{m\Gamma(s)} \tag{2. 7}$$

where $f_{s,m,X(x)}$ is the pdf of the network induced-delay shifted by a minimum constant delay $\tau_{\min}$. $\Gamma(s)$ is the gamma function with s being the shape parameter and m representing the number of transition points from the source to the destination or hops. The peak in the pdf is the most probable delay and can be used for controller design.

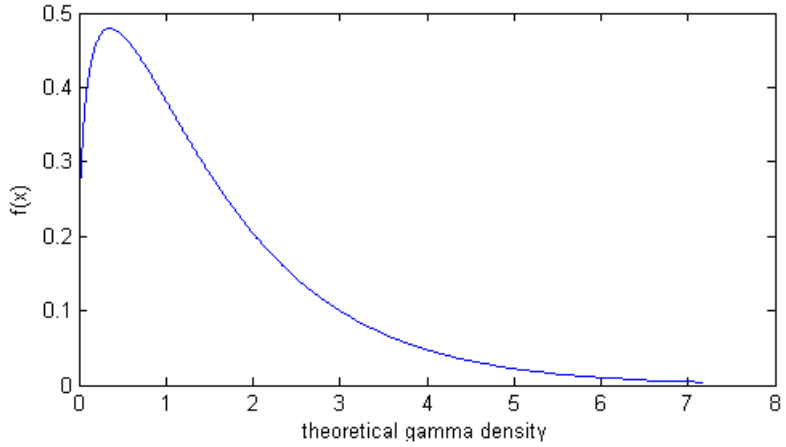


Fig. 2. 8 Distribution of the network delay for Ethernet

It is possible to use average delay, but for Ethernet the most probable delay may not always be the average.

2.3.2 Packet loss

As it is mentioned by Chang [19], packet loss is present in any communication network. The increase on network congestion due to limited buffer size, timeouts and collisions can be correlated to packet loss and vice versa. However, as stated by Mukherjee [69] a significant portion of packet losses can be associated to other transmission errors. Thus packet loss requires individual study. Some studies about the performance of NCS due to packet loss were focused in [59], [63], [83], [57] and [89]. Special attention of stability analysis is considered in the latter in either presence or absence of packet loss.

Packet loss can be defined as follows: $\tau(t_k) := \infty$. The effect of infinite delays can be seen as the lack of information arrivals. However, this definition does not state anything about the sampling time. Assuming that the delay is bounded by a known value $\tau_{\max}$ hence, it can be stated that a packet is being dropped if $\tau(t_k) \geq \tau_{\max}$.

Loss of information arrivals, for control purposes can be studied as minimum transmission rates to allow successful arrivals [107] or adding packet dropout

indicators $\phi(k)$ with known probability [59] and [86]. In other words, the stochastic process $\{\phi(k)\}$ can be stated as a Bernoulli process with $P[\phi(k)=0]=\gamma$ when the transmission fails and $P[\phi(k)=1]=1-\gamma$, otherwise. γ is a known packet dropping rate. Any of the above assumptions represent a system model that can be seen intermittently [59], and allows the system to be represented as Asynchronous Dynamical Systems (ADS) [107].

2.3.3 Additional Constraints

The increase in data traffic drives the network bandwidth to its capacity limits. At the same time, the way the information is packed depends on quantization and encoding/decoding techniques.

A channel with optimal encoding/decoding and maximum available bit rate is a common assumption for control purposes [36]. Nonetheless, when issues like channel capacity, quantization errors and encoding/decoding techniques are modelled, complicated nonlinear analysis and possible limited sampling times are expected [22].

In [71] and [88], the effects of having limited communication channels for control purposes are explored. From Shannon's theory, channel capacity is expressed as the maximum rate at which the channel can be used to transmit data with an arbitrary small probability of error.

This definition can be extended to maximum Shannon's capacity C or channel capacity with a small probability of error; and Zero error capacity C_0 or channel capacity without errors. For stability purposes C is too relaxed and C_0 is too conservative and unreal. A new notion of channel capacity is introduced in [83] as "*Any-time capacity* $C_{anytime}$". $C_{anytime}$ is a good indicator of the channel capacity to

maintain system stability with a probability of error that decays at least exponentially with delay at a given rate α.

$$C < C_{anytime(\alpha)} < C_0 \tag{2. 8}$$

The constraints of limited channels can be dealt with reducing the number of bits transmitted over the channel. Two approaches can be considered:

Tatikonda [88] stated that communication constraints are meant to limit the achievability of the control objective. This information theoretic approach depends on the information available in the encoder, decoder and controller. For low bandwidth channels the observation or information available is limited. Thus, a boundary for stability is directly related to the bit-rate and the communication scheme used to transmit the information.

In [15] the concept of "*attention*" is introduced in the control system to state that control laws $u(t)$ with small values of $\left\|\frac{\partial u}{\partial t}\right\|$ and $\left\|\frac{\partial u}{\partial x}\right\|$ require less frequent update and will be more robust due to small changes of data. $x(t)$ are the system states.

Quantized measurements $q(y)$ are measurement approximations based on partial or limited information of the states. As it is stated by Ye et al. [103] and Oppenheim et al. [74], once quantization is introduced, the system model requires complicated non-linear analysis and furthermore, limited information in closed-loop systems leads to instability.

In [22], stability of the system is dependent not only on the properties of the system but also on the properties of the quantizer. Thus, the selection of adequate control laws based on past quantized measurements is essential.

Quantizers are defined as a piecewise constant function $q:\mathbf{R}^l \to \Im$ with $\Im$ as a finite subset of $\mathbf{R}^l$.
In [21], it is assumed that the quantizer satisfies the following conditions:

1) If $\|y\| \le M$, then $\|q(y) - y\| \le \Delta$

2) If $\|y\| \ge M$, then $\|q(y)\| > M - \Delta$

where y represents the output of the quantizer, M and Δ are the quantization range and the quantization error of q, respectively.

To stabilize the system using quantized feedback measurements, the system input u is constrained to be a non anticipative function of the past quantized measurements of the state x. The resulting control strategy uses a quantized measurement of the form:

$$\mu_k q\left(\mu^{-1} y\right) \tag{2. 9}$$

where $\mu_k > 0$ is a function of time t_k and output y.

The method for choosing $\mu_k > 0$ defines the range of the quantizer. Using an event-based method as proposed in [57]. The quantized signal will be:

$\mu_k = g\left(\bar{\mu}_k\right)$ and $g(\cdot)$ is a logarithmic quantizer given by:

$$g(\beta) = \begin{cases} \mu_i, & \text{if} \quad \dfrac{1}{1+\delta_g}\mu_i < \beta \le \dfrac{1}{1-\delta_g}\mu_i \\ 0, & \text{if} \quad \beta = 0 \\ -g(-\beta) & \text{if} \quad \beta < 0 \end{cases} \tag{2. 10}$$

with $\delta_g = \dfrac{1-\rho_g}{1+\rho_g}$, $0 < \rho_g < 1$. The set of quantization levels is chosen as: $u_g = \left\{\pm u^{(i)} : u^{(i)} = \rho_g^i u_0, i = \pm 1, \pm 2, \ldots\right\} \cup \{\pm u_0\} \cup \{0\}$. For implementation purposes, finite number of levels has to be considered [21].

Quantized NCS was presented by Brockett [16] and Elia et al. [24]. Quantization is considered useful to design the closed loop system. The analysis is made for systems with countable number of fixed control laws.

In [16], asymptotic stability is achieved by keeping the quantizer values fixed, but changing quantization parameters as the system evolves. Quantization sensitivity $\Delta(t)$ can be increased for unknown initial states until the system can be adequately measured, as well as decreased to drive the state to zero.

Thus, the stability of unquantized states can be extended to quantized versions and the system will reach stability when $\Delta(t) \to 0$ as $t \to \infty$.

2. 4 Information Updates

In practical applications, the control components are distributed and have their own processing units and timing functions [53]. Traditional sampling techniques require the sampling interval to be constant, and furthermore, as stated in [5], it is possible to sample signals if they remain constant between sampling intervals. Assuming that the signal is piecewise constant after a sampler-and-hold device ZOH, the sampled signal will be: $f(t) = f(hk)$, with $t \in [kh, (k+1)h)$ and $k = 1, 2, \ldots$.

Sharing a common communication network causes control components to be inherently asynchronous or extremely difficult to synchronize [53]. The sampling time h between source node and destination node can become variable, leading to multirate sampling.

It is important to specify how control components update their transmissions and information arrivals. Control components can be either event-driven or time-driven. A time-driven component is a component, which updates its value every sampling time h. On the other hand, in an event-driven component, an event such as variation in the input, triggers the element to perform an update. Event-driven components are useful for actuators which use piecewise constant signals. For control components such as controllers and estimators, known sampling times facilitate the calculation of network delays as well as missing packets.

This framework was initially introduced by Low in [62] to study active queue management for feedback Internet congestion control; and lately was adapted by Yang in [101] to study NCS. Control components such as actuators and sensors are included into subsystem 1, whereas subsystem 2 contains controllers and possibly estimators. The interaction between communication network and control components can be seen in Fig. 2.9:

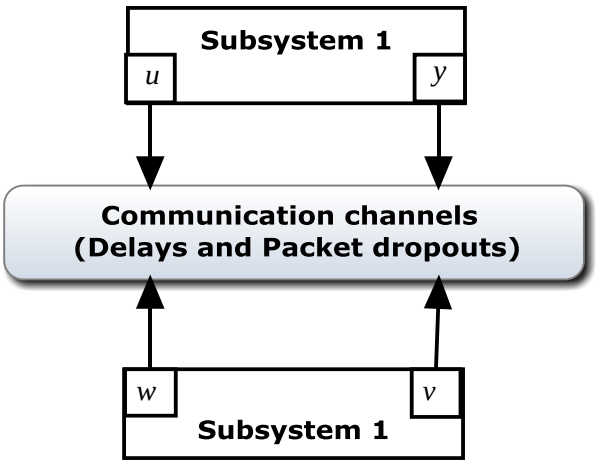


Fig. 2.9 NCS general framework

Lets define y and v as the control components outputs; and w and u as control component inputs respectively. Using h as time reference, the information transmitted by the sensor output y reaches the controller/estimator input v, $\tau_{SC}(t_k)$ times delayed or a packet is lost .*.

$$v(hk)=\begin{cases} y\left(hk-\tau_{SC}(t_k)\right) & \tau_{SC}(t_k)\le\tau_{SC\max} \\ * & \tau_{SC}(t_k)>\tau_{SC\max} \end{cases} \quad (2.11)$$

On the other hand, the information transmitted by the controller output w updates the actuator input u $\tau_{CA}(t_k)$ times delayed or can be lost.

$$u(hk)=\begin{cases} w\left(hk-\tau_{CA}(t_k)\right) & \tau_{CA}(t_k)\le\tau_{CA\max} \\ * & \tau_{CA}(t_k)>\tau_{CA\max} \end{cases} \quad (2.12)$$

$\tau_{CA}(t_k)$ is defined within the sampling interval $t_k\in\left[kh,(k+1)h\right)$. It is difficult to distinguish between packet dropouts and delayed information with long network

delays when time-driven components are used. Event driven components are better for that purpose. The differences on information deployment are shown in Fig. 2.10.

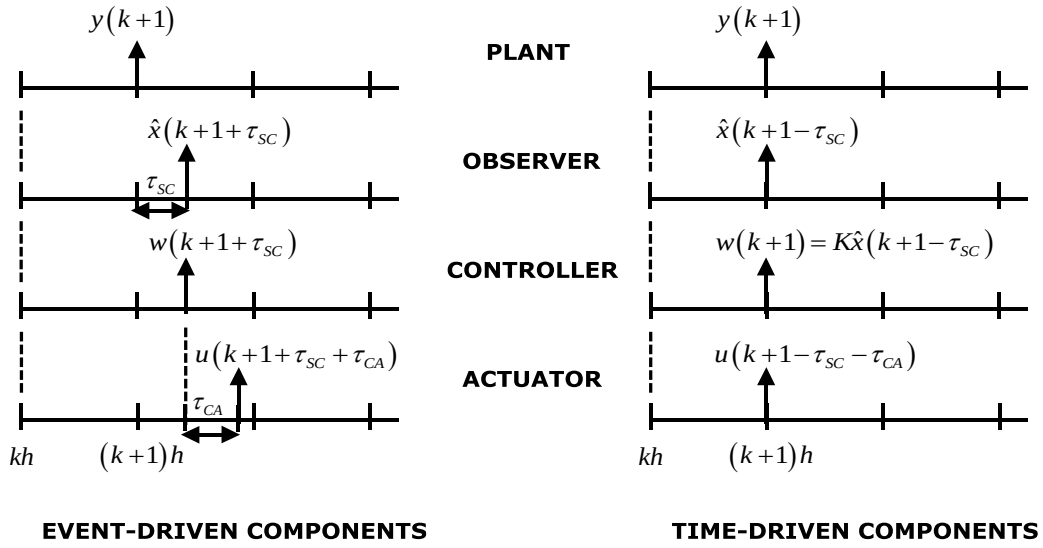


Fig. 2.10 Differences of information deployment between time-driven and event-driven components

In the above figure, an observer is also included in the information update an represented by the estimated state $\hat{x}$.

Based in Fig. 2.10, the NCS problem can be analyzed via two separated sides, the system side represented by subsystem 1 and control side represented by subsystem 2. Actuators and sensors are considered inside subsystem 1. Subsystem1 represents either a distributed plant to be controlled with sensors and actuators or a controlled plant with local controllers. For simplicity the plant is assumed to be linear and time invariant (LTI), however the type of system will depend on the application. A continuous LTI subsystem 1 can be expressed as

$$\dot{x}(t) = a_p x(t) + b_p u(t) \tag{2. 13}$$

$$y(t) = c_p x(t) \tag{2. 14}$$

$$t \in \mathbf{R}^+$$

here, $u(t) \in \mathbf{R}^{N_u}$ are the system input signals, $y(t) \in \mathbf{R}^{N_y}$ are the outputs and $x(t) \in \mathbf{R}^{N_x}$ are the states of the system. a_p, b_p and c_p are plant matrices of compatible dimensions. If subsystem 1 represents a plant to be controlled, a continuous model is suitable. For plants with local controllers, a discrete subsystem 1 that represent the plant and the controllers, may be more convenient to represent plant and control components. In this way the discrete nature of the network can be easily added.

Subsystem 1 expressed as discrete system is as follows:

$$x(k+1) = A_p x(k) + B_p u(k) \tag{2. 15}$$

$$y(k) = C_p x(k) \tag{2. 16}$$

with

$$A_p = \mathrm{e}^{a_p h}$$

and

$$B_{p1} = \mathrm{e}^{a_p h} \int_0^h \left(\mathrm{e}^{-a_p s} b_p \right) ds$$

where $y(k) \in \mathbf{R}^{N_y}$, $u(k) \in \mathbf{R}^{N_u}$, $x(k) \in \mathbf{R}^{N_x}$ and all the system matrices A_p, B_p, C_p are of compatible dimensions. The discrete form is obtained at discrete time instants $t \in \left[kh, (k+1)h \right)$, where h represents the sampling time and for convenience of notation is being omitted on the equations. It is assumed that h is constant, for all sensors to facilitate the implementation of SISO systems and Multiple-Inputs Multiple-Outputs MIMO systems.

$$h_i = h \tag{2. 17}$$

$$i = 1, 2, ..., N_y$$

This assumption does not include the varying sampling time of the scheduling methodology and furthermore, it is not suitable for distributed systems with multiple sampling intervals. A complete definition of h can be derived as follows:

$$h_{i,k} = g \cdot h \tag{2. 18}$$

$$g \in \mathbf{N}$$
$$i = 1, 2, \ldots, N_y$$
$$k = 1, 2, \ldots$$

and leads to asynchronous systems.

2. 5 Conclusions

NCS can be considered as the integration between Distributed Control, Sensor Networks and Communication Networks. A composite constraints model that includes physical dynamics, software oriented dynamics and transmission capacity leads to complex nonlinear models. Nonetheless, these network constraints can also be defined as a variable but bounded time delay. There is only empirical evidence of the correlation between packet loss and transmission delays, thus the study of packet loss still requires individual study. The definition of packet loss presented here can be sufficient for time-driven components. The resulting constrained signals (due to packet loss and network delays) can be fed into a system model to obtain a NCS model.

A general NCS model can be characterized by time-varying inputs/outputs, asynchronized transmissions and ad-hoc components distribution. If aspects as channel capacities and quantization are included, the NCS problem falls into a communications theory analysis. This type of analysis is beyond the limits of this thesis and will not be considered.

Chapter 3

Networked Control Systems Modelling

This chapter is a state of the art review of the different control methodologies employed for NCS in conjunction with the information structure (connectivity and capacity) of the network. Network constraints such as: network delays and packet dropouts are difficult to model. Longer delays than the sampling time, also introduce missequencing (packet arrivals out of order) and variable sampling intervals, making the modelling process more complex. These constraints can be presented within a general formulation of a continuous LTI system with delayed output measurements and delayed control inputs. Different model transformations and discretization methods allow the modelling of NCS to be expressed in different forms such as: continuous perturbation, hybrid, augmented discrete and discrete with parametric delays. These models are presented in section 3.1 to emphasize the need of a trade-off between computational burden and system dimensions. A NCS solution is network-dependent. The NCS problem can be divided into two categories: NCS for accurate communications and NCS for system analysis and design. Different ways to exploit network resources with existing control methodologies are presented in section 3.2 to show their adaptability against variable network constraints.

NCS can be modelled as a combination of system dynamics and network dynamics. Network dynamics are a direct consequence of the finite capabilities of information processing among various parts of the system [65]. A simple assumption as presented in equations (2. 13) and (2. 14) can be combined with LTI systems. Some of the network constraints such as bandwidth allocation are not addressed in this formulation

because they are protocol-dependent. Others such as Quantization and sampling devices models can be included in the discrete formulation of the problem.

Open loop NCS resembles time-delay systems (TDS) with undelayed states or retarded systems. Retarded systems can be modelled by a class of equations called Functional Differential Equations (FDE) [81]. These equations are different from the well-known Ordinary Differential Equations (ODE) because the state is function of a deviated time argument x_t corresponding to the past interval $[t-\tau, t)$. Delayed states can appear over closed loop NCS problems. Feedback delays are sometimes useful to stabilize control systems. A convenient model transformation can lead to perturbation models from the open loop NCS.

3.1 NCS modelling for network delays and LTI systems

In practice, many physical, industrial and engineering systems are interfaced through communication networks and consequently their control inputs are affected by the network constraints. LTI systems are not common in industry but facilitate the analysis of NCS as TDS. A general formulation of a TDS is presented by Richard [81] as follows:

$$\Sigma_{1(TDS)} = \left(a_i, b_i, c_i, g_j, h_j, n_j\right) \tag{3. 1}$$

$$\dot{x}(t) = \sum_{i=0}^{k}\left[a_i x(t-\tau_i) + b_i u(t-\tau_i)\right] + \sum_{j=1}^{r}\int_{t-\varsigma_j}^{t}\left[g_j(\theta)x(\theta) + h_j(\theta)u(\theta)\right]d\theta \tag{3. 2}$$

$$y(t) = \sum_{i=0}^{k} c_i x(t-\tau_i) + \sum_{j=1}^{r}\int_{t-\upsilon_j}^{t} n_j(\theta)x(\theta)\,d\theta \tag{3. 3}$$

where a_i, c_i, g_j and n_j corresponding to the system state dynamics; b_i and h_j the system input dynamics. $x(t) \in \mathbf{R}^{N_x}$ are the system states. The inputs $u(t) \in \mathbf{R}^{N_u}$ are delayed by the network and the sum of integrals represents distributed network

delays. Delayed states appear on closed loop NCS. The delay τ is assumed to be constant and is chosen as: $\tau = \max_{i,j,l}\left\{\tau_i, \varsigma_j, \upsilon_l\right\}$. For $\tau_0 = 0$ and a_0 represents open loop delay independent system dynamics. $y(t) \in \mathbf{R}^{N_y}$ are the system outputs.

In many real systems, the distributed effects can be replaced by a sum of discrete ones:

$$\int_{t-\varsigma_j}^{t} h_j(\theta)u(\theta)d\theta \approx \frac{\varsigma}{d}\sum_{i=1}^{d}\alpha_i h_j\left(\frac{i\varsigma}{d}\right)u\left(t-\frac{i\varsigma}{d}\right) \tag{3. 4}$$

with α_i and d as constants.

Due to this simplification, many authors investigate the particular case of discrete-delay systems [81]:

$$\Sigma_{1(TDS)} = (a_i, b_i, c_i) \tag{3. 5}$$

$$\dot{x}(t) = \sum_{i=0}^{k}\left[a_i x(t-\tau_i) + b_i u(t-\tau_i)\right] \tag{3. 6}$$

$$y(t) = \sum_{i=0}^{k} c_i x(t-\tau_i) \tag{3. 7}$$

$$\tau_0 < \tau_1 < \ldots < \tau_{k-1} < \tau_k$$

The above system matches almost all the models presented in the literature of NCS. Furthermore, the delays can also incorporate sensors and actuators delays.

3.1.1 Model transformation

Let's rewrite equation

(3. 6) as open loop NCS with no delayed states and single input delay as follows:

$$\dot{x}(t) = a_p x(t) + b_p u(t-\tau(t_k)) \tag{3. 8}$$

where sensor-to-controller $\tau_{SC,k}$ and controller-to-actuator $\tau_{CA,k}$ delays are lumped together as $\tau(t_k) = \tau_{SC,k} + \tau_{CA,k}$ and is time-varying. The resulting system with input delay can be transformed to a simpler model. Model transformation is usually associated with stability analysis; there exist several model transformations as presented in [81]. Two model transformations are particularly used in NCS, namely the Arstein model reduction and the disturbance model transformation. Both transformations make the implementation of classic state feedback control straightforward.

The so-called Arstein model reduction is a predictor-like technique that allows the formulation of delayed systems as delay-free systems with the delays on the system parameters. This formulation is used by Kim et al. [48] to formulate a dynamic controller. The transformation is defined as follows: given a system as expressed in equation (3. 8), a new variable can be defined such that:

$$z(t) = x(t) + \int_{t-\tau(t_k)}^{t} e^{a_p \cdot (t-s-\tau(t_k))} b_p u(s)\, ds \tag{3. 9}$$

with $z(t) \in \mathrm{R}^{N_x}$. Then $\left(x(t), u\left(t-\tau(t_k)\right)\right)$ is admissible for (3. 9)if and only if $\left(z(t), u(t)\right)$ is admissible for:

$$\dot{z}(t) = a_p z(t) + e^{-a_p \tau_k(t_k)} b_p u(t) \tag{3. 10}$$

Using the model from equation (3. 10), classical state feedback optimal control $u(t) = Kz(t)$ can be implemented provided that (a_p, b_p) are stabilizable, that so is $\left(a_p, e^{-a_p \tau_k(t_k)} b_p\right)$. Using equation (3. 10) the resulting control law contains a distributed component as follows:

$$u(t) = Kx(t) + \int_{t-\tau(t_k)}^{t} K e^{a_p \cdot (t-s-\tau(t_k))} b_p u(s)\, ds \tag{3. 11}$$

and uses previously controlled input stored data.

Disturbance models, on the other hand, are a plausible model transformation based on the fact that feeding back small delays can improve the stability of a delay-free system [44].

Rewriting equation (3. 8) in the following form

$$\dot{x}(t) = (a_p + b_p)x(t) + b_p\left(u(t - \tau(t_k)) - x(t)\right) \tag{3. 12}$$

the term $b_p\left(u(t - \tau(t_k)) - x(t)\right)$ can be considered as the disturbance of the system to be reduced such that equation (3. 12) is stable.

This transformation is proposed by Montestruque et al. [68] as a model-based method in Fig. 3.1 This method is used to treat network constraints as an error between actual plant states $x(t)$ and the estimated states $\hat{x}(t)$ presented as a reference model. a_m, b_m correspond to the reference model matrices of compatible dimensions.

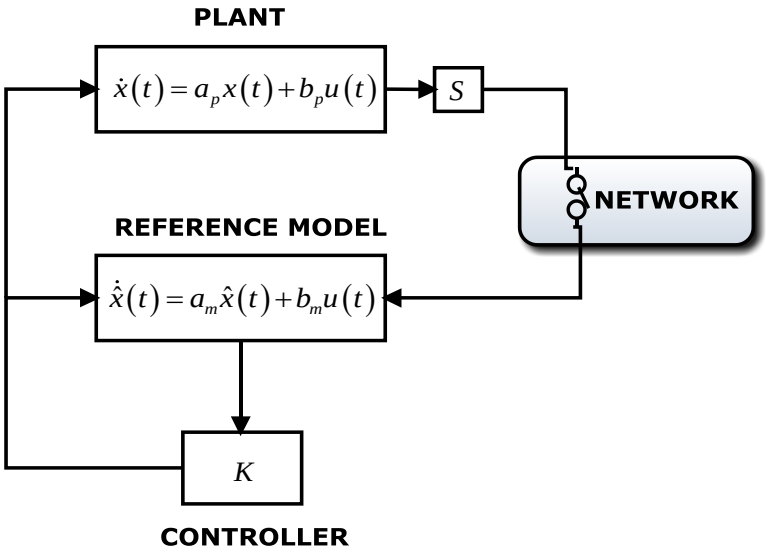


Fig. 3.1 Model-based NCS

K is the controller gain. State feedback control is implemented based on the reference model. The network is sampled at a fixed-rate, thus periodic transmissions are used to reduce communication bandwidth requirements. The disturbance is defined as the error between the real system and the reference model $e(t) = \hat{x}(t) - x(t)$.

The dynamics of the model from Fig. 3.1 are given by:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{e}(t)\end{bmatrix}=\begin{bmatrix}a_p-b_pK & b_pK\\ (a_p-a_m)+(b_m-b_p)K & a_m-(b_m-b_p)K\end{bmatrix}\begin{bmatrix}x(t)\\ e(t)\end{bmatrix} \tag{3. 13}$$

$$\begin{bmatrix}x(t_k)\\ e(t_k)\end{bmatrix}=\begin{bmatrix}\hat{x}(t_k)\\ 0\end{bmatrix} \tag{3. 14}$$

with $\begin{bmatrix}x(t_k)\\ e(t_k)\end{bmatrix}\in\mathbf{R}^{2N_x}$. The update time s_k in the model is defined as: $s_k=t_{kh+h}-t_{kh}$. This updating time is not constant and can acquire a certain value according to a probability distribution function of the network delay.

Beldiman et al. [11] addresses this modelling methodology for nonlinear systems. The dynamics with $x(t)\in\mathbf{R}^{N_x}$ and $u(t)\in\mathbf{R}^{N_u}$, are presented as follows:

$$\dot{x}(t)=f_p\left[x(t),u(t),t\right] \tag{3. 15}$$

$$y(t)=g_p\left[x(t),t\right] \tag{3. 16}$$

The controller is described as follows:

$$\dot{z}(t)=f_c\left[z(t),w(t),t\right] \tag{3. 17}$$

$$v(t)=g_c\left[z(t),t\right] \tag{3. 18}$$

here $w(t)$ is the input of the controller and represents the most recently transmitted value from the system output $y(t)$. In presence of delays, there is an error between both values. This error is defined as network induced delay $e(t)=y(t)-w(t)$. If $g_p(\cdot)$ is continuously differentiable [11] between too successive transmission times, a closed loop system can be obtained from equations (3. 15) to (3. 18) as follows:

$$\dot{x}_{cl}(t)=f_{cl}\left[x_{cl}(t),e(t),t\right] \tag{3. 19}$$

$$\dot{e}(t)=g_{cl}\left[x_{cl}(t),e(t),t\right] \tag{3. 20}$$

At each transition time, $e(t)$ has a discontinuous jump that needs to be set to zero. The perturbation can be defined as vanishing and is assumed to be a class C' function [11]. Two cases can be established:

1) Bounded vanishing perturbation. For a fast enough networks if the NCS state is bounded to E during p successive transmissions, then it will stay bounded afterwards. E is the worst case bounds. The perturbation is:

$$\|\dot{e}(t)\| < E \tag{3. 21}$$

2) Vanishing perturbation. The asymptotic stability of the perturbed NCS will depend of reaching certain conditions of a factor α affecting the state.

$$\|e(t)\| < \alpha \|x(t)\| \tag{3. 22}$$

The resulting model incorporates the network delay implicitly. When network constraints are incorporated into the system model explicitly, the network delay is discrete.

3.1.2 Model Discretization

As stated by Åström an Wittenmark [5], it is simple to sample systems with delays when the control signal remains constant between sampling intervals. The resulting sampled-data system is finite dimensional. If the control computation time is insignificant, sensor-to-controller delay as presented in equation (2. 14) can be lumped together with the controller-to-actuator delay as in equation (3. 8). Thus the control input is delayed and depending on the information arrivals the input value can be switched every time a packet has arrived. These events introduce discrete dynamics (from the control loop) into the continuous system [36]; and leading to a class of hybrid systems with fixed instants of impulse effects [107].

A hybrid NCS can be achieved under the following assumptions: plant outputs are sampled periodically with time-driven sensors; and controllers and actuators are event-driven. Equation (3. 8) can then be defined as follows**:**

$$\dot{x}(t) = a_p x(t) + b_p K \hat{x}(t) \tag{3. 23}$$

$$t \in \left[kh + \tau_k, (k+1)h + \tau_k \right)$$

$$\hat{x}(t^+) = x(t - \tau_k) \tag{3. 24}$$

$$t \in \{ kh + \tau_k, k = 0, 1, 2, \ldots \}$$

The state $x(t-\tau_k) \in \mathbf{R}^{N_x}$ in terms of $x(t)$ and $\hat{x}(t)$ is:

$$x(t-\tau_k) = e^{-a_p \tau_k} x(t) + e^{-a_p \tau_k} J(\tau_k) b_p K \hat{x}(t) \tag{3. 25}$$

where $\int_{t-\tau_k}^{t} e^{a_p \cdot (h-s)} b_p ds \equiv J(\tau_k) b_p$.

If $e^{-a_p \tau_k}$ and $e^{-a_p \tau_k} J(\tau_k) b_p K$, and also discretizing equation (3. 23) at sampling time h , the eigenvalues of the NCS hybrid system H will be:

$$H = \begin{bmatrix} e^{a_p h} & -J(h) b_p K \\ e^{a_p \cdot (h-\tau_k)} & -e^{-a_p \tau_k} \left[J(h) - J(\tau_k) \right] b_p K \end{bmatrix} \tag{3. 26}$$

Stability of this type of system is reduced to evaluating the Shur-ness (e.i. wether all the eigenvalues of the matrix H have magnitude less than one) [107]. Stability triangle can be used to explicitly calculate the relation between τ_k and h .

Hoyakem et al. [35] present a novel modelling method to obtain an LTI sampled-data model, where the variable network delay is introduced as a constant into the model [35]. The network is modelled as a variable-rate ideal sampler S_{hk} between the plant and the controller, and the zero-order hold H_{hk} between the controller and the plant as shown in Fig. 3.2

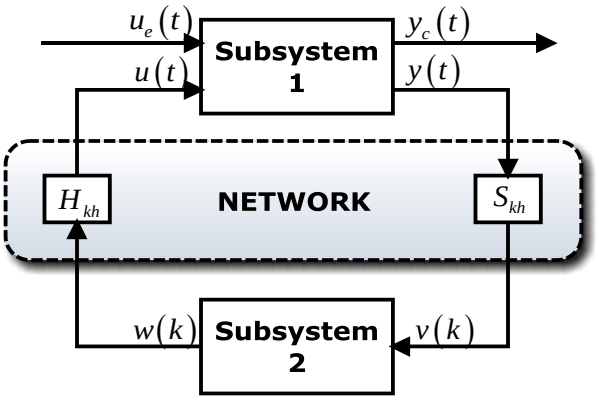


Fig. 3.2 Sampled-data model approach

If subsystem S_1 is a LTI system with exogenous inputs $u_e(t) \in \mathbf{R}^{N_{ue}}$ and controlled inputs $u(t) \in \mathbf{R}^{N_u}$, the state-space model is as follows:

$$\dot{x}(t) = a_p x(t) + b_{p1} u_e(t) + b_{p2} u(t) \tag{3. 27}$$

$$y_c(t) = c_{p1} x(t) + d_{p11} u_e(t) + d_{p12} u(t) \tag{3. 28}$$

$$y(t) = c_{p2} x(t) \tag{3. 29}$$

$x(t) \in \mathbf{R}^{N_x}$. $y_c \in \mathbf{R}^{N_{yc}}$ is the controlled outputs and $y(t) \in \mathbf{R}^{N_y}$ is the measurable outputs. a_p, b_{p1}, b_{p2}, c_{p1}, d_{p11}, d_{p12}, c_{p2} are matrices of compatible dimensions.

For convenience, the system can be expressed as follow:

$$S_1 = \begin{pmatrix} a_p & b_{p1} & b_{p2} \\ c_{p1} & d_{p11} & d_{p12} \\ c_{p2} & 0 & 0 \end{pmatrix} = \begin{pmatrix} S_{1(11)} & S_{1(12)} \\ S_{1(21)} & S_{1(22)} \end{pmatrix} \tag{3. 30}$$

d_{p21} and d_{p22} are set to zero to assure continuity in the measured output. Variable delays can be incorporated into an augmented state-space model. Thus the resulting model contains constant delays. This method is called lifting. The lifting operators L_{τ_k} accommodate the output measurements $y_c(t)$ into a new augmented variable $\tilde{y}_c$ equals to τ_k-times that of $y_c(t)$ [35] (hence the term lifting). This is presented in Fig. 3.3.

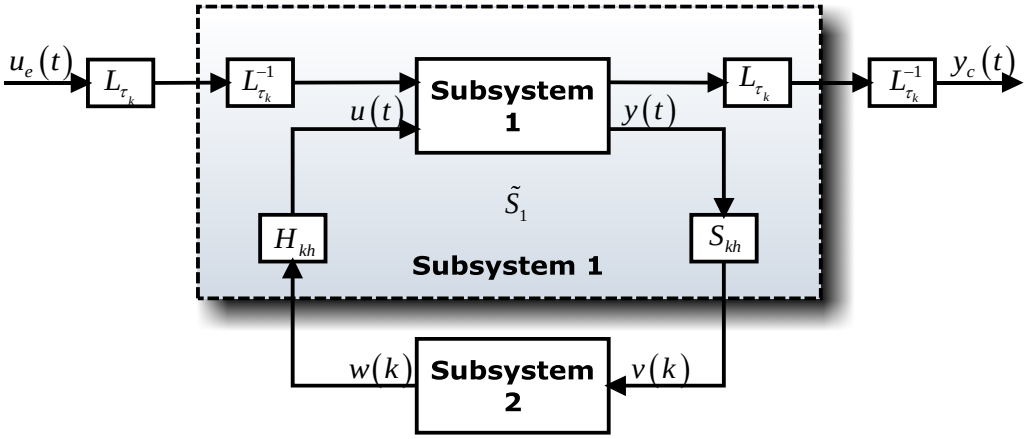


Fig. 3.3 Reconfigured sampled-data model

Lifted subsystem $\tilde{S}_1$ can be expressed as follows:

$$\tilde{S}_1 = \begin{pmatrix} L_{\tau_k} & 0 \\ 0 & S_{\tau_k} \end{pmatrix} S_1 \begin{pmatrix} L_{\tau_k}^{-1} & 0 \\ 0 & H_{\tau_k} \end{pmatrix} \tag{3. 31}$$

$$\tilde{S}_1 = \begin{pmatrix} L_{\tau_k} S_{1(11)} L_{\tau_k}^{-1} & L_{\tau_k} S_{1(12)} H_{\tau_k} \\ S_{\tau_k} S_{1(21)} L_{\tau_k}^{-1} & S_{\tau_k} S_{1(22)} H_{\tau_k} \end{pmatrix}$$

$$\tilde{S}_1 = \begin{pmatrix} \tilde{S}_{1(11)} & \tilde{S}_{1(12)} \\ \tilde{S}_{1(21)} & \tilde{S}_{1(22)} \end{pmatrix}$$

where $L(\tau_k)$ and $L^{-1}(\tau_k)$ are the lifting operators and the variable sampling time is: $\tau_k = t_k - t_{k-1}$. The mathematical transformations are as follows:

$S_{1(11)}$ relates the exogenous input $u_e(t)$ with the controlled output $y_c(t)$. To obtain $\tilde{S}_{1(11)}$, the linear operators are given by:

$$\tilde{A}_p = \mathrm{e}^{a_p \tau_k} \tag{3. 32}$$

$$\tilde{B}_{p1}\tilde{u}_e = \int_0^{\tau_k} \mathrm{e}^{a_p \cdot (\tau_k - s)} b_{p1} u_e(s)\, ds \tag{3. 33}$$

$$\left(\tilde{C}_{p1} x\right)(t) = c_{p1}\, \mathrm{e}^{a_p t}\, x \tag{3. 34}$$

$$\left(\tilde{D}_{p11}\tilde{u}_e\right)(t) = d_{p11} u_e(t) + c_{p1} \int_0^t \mathrm{e}^{a_p \cdot (t-s)} b_{p1} u_e(s)\, ds \tag{3. 35}$$

$S_{1(12)}$ relates the controlled input $u(t)$ with the controlled output $y_c(t)$. $\tilde{S}_{1(12)}$ is obtained using the following transformations:

$$\tilde{B}_{p2} = \int_0^{\tau_k} \mathrm{e}^{a_p s}\, ds \cdot b_{p2} \tag{3. 36}$$

$$\left(\tilde{D}_{p12}\tilde{u}\right)(t) = d_{p12}\tilde{u} + c_{p1}\int_0^t \mathrm{e}^{a_p s}\, ds \cdot b_{p2}\tilde{u} \tag{3. 37}$$

To obtain $\tilde{S}_{1(21)}$ and $\tilde{S}_{1(22)}$ the transformations presented on equations (3. 32)-(3. 37) are used respectively. Furthermore the system dimensions are extended to $\tilde{x}(k) \in \mathrm{R}^{N_x \cdot N_L}$, with N_L being the lifting operator dimensions.

To obtain a time reference, event-driven components can be replaced for time-driven components. Time-driven components offer a time reference that can be used to calculate the network delays if the network allows the transmission of time-stamped packets. Discretization can be done using this known delay value by augmenting the discrete state-space model as follows:

$$\Sigma_{d1} = \left[A_p(h), B_{p0}\left(h, \tau_k^{'}\right), B_{p1}\left(h, \tau_k^{'}\right), C_p \right] \tag{3. 38}$$

$$x(kh+h) = A_p(h)\,x(kh) + B_{p0}\left(h, \tau_k^{'}\right)u\left(kh - \alpha_k h\right) + B_{p1}\left(h, \tau_k^{'}\right)u\left(kh - \alpha_k h - h\right) \tag{3. 39}$$

where $\tau_k = \alpha_k h + \tau_k^{'}$, $0 < \tau_k^{'} \le h$, $\alpha_k > 0$ a positive integer; and:

$$A_p = \mathrm{e}^{a_p h} \tag{3. 40}$$

$$B_{p1} = \mathrm{e}^{a_p \cdot \left(h - \tau_k^{'}\right)} \int_0^{\tau_k^{'}} \mathrm{e}^{a_p s}\, dsb_p \tag{3. 41}$$

$$B_{p0} = \int_0^{h-\tau_k^{'}} \mathrm{e}^{a_p s}\, dsb_p \tag{3. 42}$$

The above sampled-data system is time varying in B_{pi}, $i = 0,1$. For constant sampling intervals the state-space augmented model is as follows:

$$\begin{bmatrix} x(k+1) \\ u(k-\alpha_k) \\ u(k-\alpha_k+1) \\ \vdots \\ u(k) \end{bmatrix} = \begin{bmatrix} A_p(h) & B_{p1}(h,\tau_k') & B_{p0}(h,\tau_k') & \cdots & 0 \\ 0 & 0 & \mathrm{I} & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \mathrm{I} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-\alpha_k-1) \\ u(k-\alpha_k) \\ \vdots \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \mathrm{I} \end{bmatrix} u(k) \tag{3. 43}$$

The states of equation (3. 43) are $\mathbf{R}^{N_x+(\alpha_k+1)N_u}$. Variable network delays can be incorporated but with a considerable increase in computational burden.

The transformations presented in equation (3. 9) and (3. 10) can be used to reduce the resulting system dimensions from equation (3. 43). The delay-free discrete model can be expressed as follows:

$$z(k+1) = A_p z(k) + A_p^{-\tau_k} B_p u(k) \tag{3. 44}$$

if and only if $z(k) \in \mathrm{R}^{N_x}$ is a solution of the equation (3. 43) and is defined as follows:

$$z(k) = x(k) + \sum_{i=-\tau_k}^{-1} A_p^{-(\tau_k+i+1)} B_p u(k+i) \tag{3. 45}$$

Discretization models presented in equations (3. 23) to (3. 45) assume periodic sampling times. Some works in non-identical sampling periods are reported by Liou et al. [58]. If network dynamics such as bandwidth allocation or scheduling protocols are included, the sampling time becomes variable [92]. Thus the resulting sampled-data time varying system depends on the network-induced delay τ_k as well as the varying sampling time h_k as follows.

$$\Sigma_{d1} = \left[A_p(h_k), B_{p0}(h_k,\tau_k'), B_{p1}(h_k,\tau_k'), C_p \right] \tag{3. 46}$$

Varying sampling intervals can be formulated as multirate systems for known values [17]. They can also be used to reduce the influence of packet lost onto the stability of the overall system. This approach will be exploited on chapter 6 for multi-loop NCS.

Some of the above models increase the system dimensions and consequently the computational burden on the control calculation. System dimensions can increase infinitely leading to complex solutions or no solutions. Network resources makes delays bounded and consequently the control methodologies have to take the network capacities into account.

3.2 Control Strategies for NCS

Provided the NCS system is controllable and observable, some specific control law can be used to regulate the system outputs and compensate the network constraints. These approaches are described by Yang [101] as analytically-oriented for system analysis and design.

3.2.1 NCS for system analysis and design

Almost all the literature in NCS is referred to stability analysis. Regardless of the method used, sufficient conditions are found to stabilize the NCS either for packet dropouts, network delays, and bandwidth allocation; or quantization constraints.

System stability will decrease as long as the delay grows. Finite communication channels introduce a maximum network delay before the packets are lost and the system goes from closed loop to open loop. This value will depend on the network traffic. Network traffic is not always constant and adds the time-varying nature to the delay. This characteristic makes the robustness of the system varying in an intermittent or oscillatory way.

Less conservative stability analysis such as practical stability can be a useful solution. In practical stability even instability may be good enough when the system oscillates

sufficiently near the state so that its performance is considered acceptable [50]. This concept of stability is weak but applicable because the worse delay is not always reachable. Brockett et al. [16] presents a containability concept derived from practical stability over memoryless feedback law.

More conservative stability analyses are presented as LMI. Here, the worse network delay is defined as an upper bound and sufficient conditions can be derived to stabilize the system [42]. Lyapunov stability can be extended to time delay systems. The analysis still requires the construction of a Lyapunov function to quantify the deviation of the state from the trivial solution [44]. The construction depends on the deviated time argument x_t that leads to a Lyapunov functional $V(t, x_t)$. This functional is called the Lyapunov-Krasovskii functional. This method is presented in [21] and [52] and can be used for delay independent stability or delay dependent stability for distributed delays.

When the model leads to jump systems, stochastic stability is analyzed as *almost sure stability*, [71] and [72]. A sufficient but not necessary stability test is used by Velasco [92] to avoid the complexity of LMIs. Using interval algebra, network delay and sampling time are transformed into intervals bounded by extreme real values. The resulting interval system is generic and wraps or contains the original NCS model. The test shows that the stability of a wrapping system implies the stability of any of the systems that it includes.

Controller design is based on either accurate models on any of the network constraints; or the network information availability. The delay dependency of the network is expected to be compensated with delay-dependent controllers of the form: $u = g(x_t)$. However, not only point-wise delayed control law: $u = g\left[x(t), x(t-\tau_k)\right]$ can be used, but also the classical state-feedback control law $u = g\left[x(t)\right]$ and structured controllers. Structured controllers, more especially PID controllers have been treated either as optimal [77] and [28] or adaptive with parameter configuration [26].

Network capabilities and information availability characterize the type of control methodology to be used. For control design, performance of the system depends on the network as well as the controllers. Additionally, not all networks allow packet time-stamping and acknowledgement messages, hence the development of better protocols to increase Quality of Service (QoS) is inevitable for systems with inefficient communications capabilities [36]. Yang [101] considers this strategy as a problem of network architecture, protocol formulation and scheduling strategy or NCS for accurate communication networks.

3.2.2 NCS for accurate communication networks

Many industrial protocols have been developed and tested within NCS architectures. They have been specifically developed for real-time applications and have a deterministic behaviour, assuring bounded delays. Some of the most known protocols are CAN for automotive and industrial automation, BACnet for building automation and Fieldbus for process control [101].

The trade-off between transmission rates and protocols constitutes a main limitation in NCS. Due to the popularity of commercial networks, the incorporation of Ethernet-based networks has to be addressed. Ethernet for industrial applications represents a disadvantage. Its non-deterministic behaviour is inadequate for real-time applications. The use of faster networks such as switching Ethernet represents an alternative solution for real-time control problems [96]. Networks like EtherCAT, EPL and Profinet presented in [41] and [3] reduce transmission problems by performing the communication functions in hardware, so that reaching high speed transmission rates.

To provide a more efficient use of network resources, Ethernet and wireless networks protocols are improved in [103] and [96]. Some applications are also extended over WAN networks and Internet applications [60.

On Internet applications the traffic is bursty [69]. Uneven pattern of data transmission makes queuing packets random. Delay compensation using hardware is possible by increasing the buffer size and determining the queue length for compensation purposes. Fig. 3.4 represents a NCS scheme with known network capabilities.

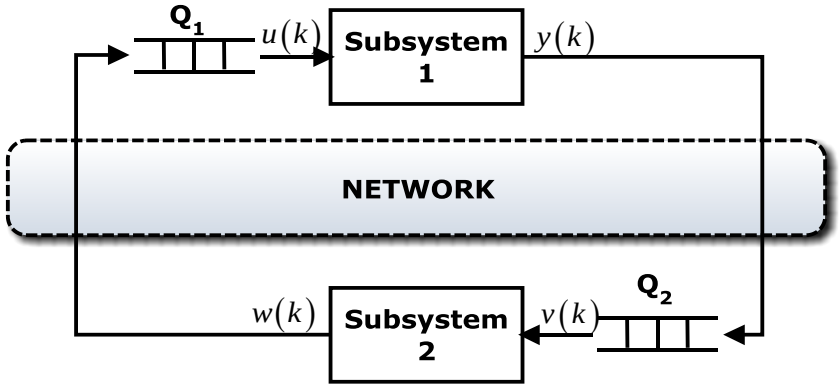


Fig. 3.4 NCS scheme with known network capabilities

Additional compensation approaches such as: network utilization improvement or QoS acknowledgement are also possible. The sampling time of the system can be incorporated into the protocol algorithm to improve network utilization. In [33] this methodology is called sampling scheduling methodology.

3.2.2.1 Sampling time Scheduling Methodology

The objective of this methodology, as presented by Hong [33], is to determine a data sampling time, which satisfies performance requirements of the control system. This methodology was originally developed for multiple NCS with periodic delays, but it can be modified for random network delays as demonstrated by Hong and Kim [34].

In [33], each control loop is assumed to be transmitting and receiving data. For M number of NCS, the sampling time of all NCS is calculated from the sampling time of the most sensitive NCS. The calculation is based on frequency domain analysis or worse case delay bound.

A network utilization algorithm is derived from the ratio of the sampling time. This is defined as the fraction of time during which the network medium remains busy for data transmissions. This algorithm is suitable for single-dimensional NCS. For multi-dimensional NCS, the extension of the algorithm is presented in [45] and [46].

Kim et al. [45], presents stability analysis over a Maximum Allowable Delay Time bound (MADT). Having MIMO systems, a scheduling algorithm that allocates the bandwidth of the network is presented. For a given node, the algorithm determines the maximum data sampling period under MADT consideration.

Bandwidth allocation or scheduling methodologies are also treated in [94]. The authors introduce an algorithm for network scheduling called Try-Once-Discard (TOD). TOD is aimed to solve scheduling problems in protocols with bitwise arbitration like CAN. In TOD, the node with the greatest weighted error from the last reported value will win the competition for the network resources. Once a packet losses the arbitration, it is discarded. Thus network utilization is reduced and data traffic depends only on new data.

This methodology can be used in conjunction with the perturbation model presented in equations (3. 13)-(3. 14). The error is determined within a bounded value defined as maximum allowable growth in error β over τ seconds. For any time t, β will be:

$$\beta > \left\| e_i\left(t+\tau\right) - e_i\left(t\right) \right\| \tag{3. 47}$$

The dynamic nature of the error appears because β depends on the internal state of the plant and the current nature of the errors. Furthermore, τ represents a deadline and depends on the p number of nodes. Thus the updating time t is expressed as: $t \geq t_0 + \tau p$.

Over improved protocols, delays can be assumed to be equal to transmission times and propagation times. If processing times on the controller are insignificant, round trip delays can be lumped together and Linear Quadratic Gaussian (LQG) controller design can be proposed using a sampled-data model defined as in equation (3. 43).

Nilsson [72] restricts the delay to be less than the sampling period so that missequencing is avoided. The resulting control law is point-wise delayed given by:

$$u(k) = -L(k,\tau)\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} \tag{3. 48}$$

where $L(k,\tau)$ is the result of minimizing a standard cost function of the form:

$$J(k) = E\left[X^T(N)Q(N)X(N)\right] + E\sum_{k=0}^{N-1}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{3. 49}$$

Padé approximations can also be used to introduce either constant or random delays in the system model. Using a modified Padé approximation method, such as the one presented by Göktas [30], the delays can be written as follows:

$$e^{-\tau\cdot\Delta s} \approx 1 - \frac{\tau s}{1+\frac{\tau s}{2}}\Delta = 1 + w(s)\Delta \tag{3. 50}$$

with $|\Delta| \le 1$. The resulting NCS structure is shown in Fig. 3.5:

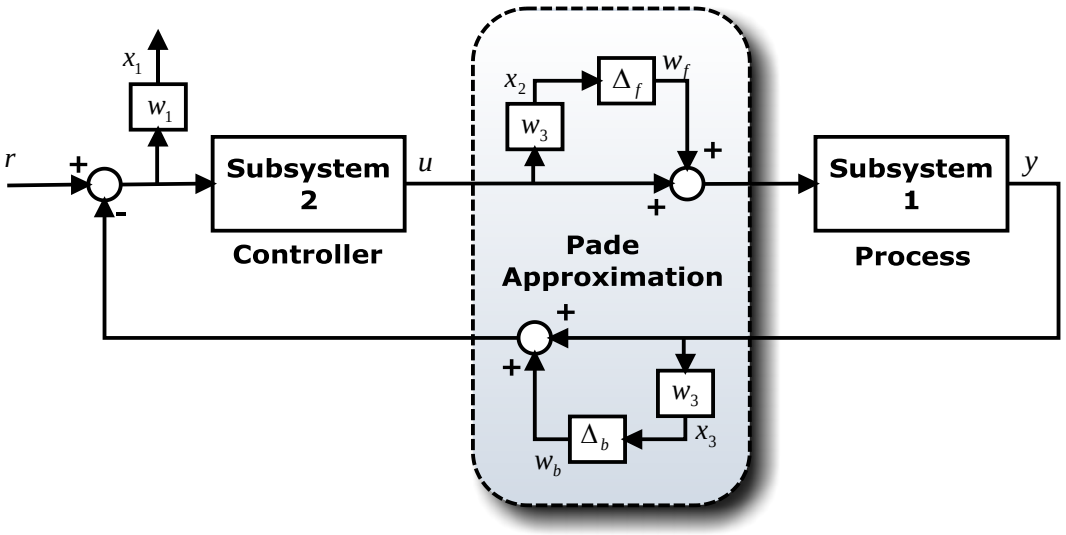


Fig. 3.5 NCS for Robust methodology

The control problem can be defined as a combination of H_∞ control and μ-synthesis. Thus system as presented in Fig. 3.5 can be transformed into Fig. 3.1 as follows:

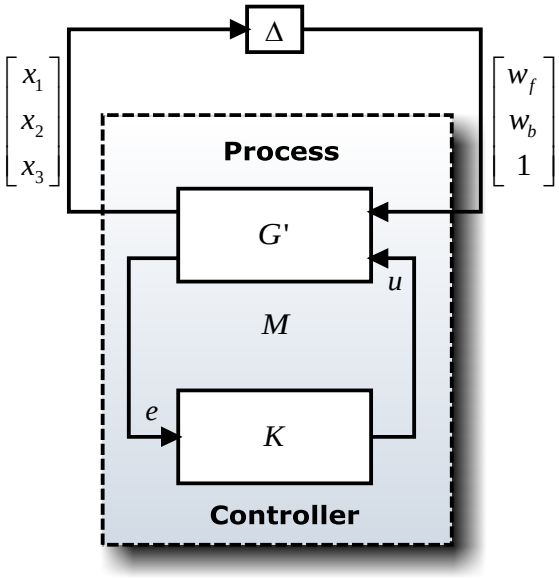


Fig. 3. 6 μ-synthesis for NCS

The introduction of μ-synthesis is possible by approximating the delays to uncertainties that yields to a proper rational interconnection matrix $M(s)$ and purely complex perturbation matrix $\Delta(s)$. Matrix $M(s)$ is given by $M = G_{11}^{'} + G_{12}^{'} K \left(I - G_{22}^{'} K\right)^{-1} G_{21}^{'}$ and:

$$G_{11}^{'} = \begin{bmatrix} 0 & 0 & 0 \\ W_3 G & 0 & 0 \\ -W_1 G & -W_1 & W_1 \end{bmatrix} \quad G_{12}^{'} = \begin{bmatrix} W_2 \\ W_3 G \\ -W_1 G \end{bmatrix}$$

$$G_{21}^{'} = \begin{bmatrix} -G & -1 & 1 \end{bmatrix} \qquad G_{22}^{'} = -G$$

Necessary conditions for robust performance are provided if $\left\| M(s) \right\|_{\infty} < 1$. If network performance can be monitored, network utilization can also be introduced in the controller design. This is called the QoS-based methodology.

3.2.2.2 QoS-based methodology

QoS variation during network utilization can be compensated by using an end-user control adaptation approach [89]. This methodology involves cooperation with real

time QoS negotiation scheme as proposed by Addelzaher et al. [1] and measurements through a middleware as proposed in [55]. The network can send at least two QoS measurements: point-to-point network throughput QoS_1 and point-to-point maximal delay bound of the largest packet QoS_2. QoS_2 can also be derived from QoS_1. These measurements limit the end-to-end network application requirements; and to be applicable, they are related with the sampling time as follows:

$$h > \tau + \tau_p \tag{3. 51}$$

The delays τ and τ_p are the transmission delay and processing time delay respectively. τ is function of QoS_1 and QoS_2, so that a sampling time-dependent structured controller such as PID controller can be formulated. PID controllers are widely used in industry. Their robustness combined with a parameter adaptability technique, make them suitable for NCS.

A cost function based on the control parameters can be calculated and stored. For real-time application, analytical techniques for cost function optimization are time consuming, thus in [55] a look up table of pre-calculated optimal control parameters is used.

Pahjola [77] uses optimization methods such as integral time absolute error (ITAE) optimization and gain scheduling to find the best PID controller parameters. Internal model control (IMC) and Ziegler-Nichols methods are presented for comparison purposes. In [28] a PID controller is optimized by using a gradient descent method.

Gain scheduling is used to adapt controllers to time varying parameters and it can also be used for time varying delay systems. In [89], the design of PI controllers with gain scheduling based on IP-based traffic is proposed by using network traffic statistical parameters such as: mean delay, delay variance, loss rate and other network QoS variables.

A simple controller given by $u(t) = g\left[y(t), \gamma \cdot p_u \right]$, with $p_u \in \mathbf{R}^{N_u}$ being the controller parameters and $\gamma \in \mathbf{R}^{N_u}$ being a variable gain. This control law can be

tuned externally using gain scheduling approach if a $\beta \in \mathbf{R}^{N_y}$ gain can be found such that:

$$\beta \cdot u(t) = \beta g\left[y(t), pu \right] \cong g\left[y(t), \gamma \cdot pu \right] \tag{3. 52}$$

In some cases the relation between β and γ is linear. An optimal β based on certain objective can be easily obtained. Nonetheless, for some systems like mobile robots, this relation is nonlinear and soft computing techniques are more suitable. Fuzzy logic methodologies are studied in [2], while in [19] external control-parameter configuration is based on Genetic algorithms.

In [100] the tuning of PID controllers is achieved using a predictor-based approach. This approach is a two layer structure implemented to control nonlinear time-varying systems. The lower layer consists of the system and the PID controller. The upper layer is used to identify systems parameters using a receding horizon window of the output measurements $\hat{y}(k+j)$. At the same time, controller parameters can be obtained at any instant. These calculations are achieved by minimizing a predictive control criterion with no constraints. Additionally if the buffer size is known, random delays can be bounded.

3.2.2.3 Queuing methodology

Reshaping random delays has been studied in [63] and [18]. Luk and Ray [63] store control signals and output measurements in a First-In-First-Out (FIFO) queue and a shift register. The size of the queue is therefore used to construct an augmented model of the system based on estimation methodologies.

To compensate network delays on the controller-to-actuator side, predictor-like techniques can be applied. Predictive control for NCS attempts to compensate network-induced delays by using N-step ahead bounded control actions [63]. This approach is Ethernet-oriented and sends a set of control actions into a single packet. Ethernet networks allow long size packets. Whether a bit or a hundred of bits are

sent, the packet uses the same amount of network resources [106]. Thus, the future control actions can be sent to the actuator queue for storage. For known queue size, Fig. 3.4 can be modified as in Fig. 3.7.

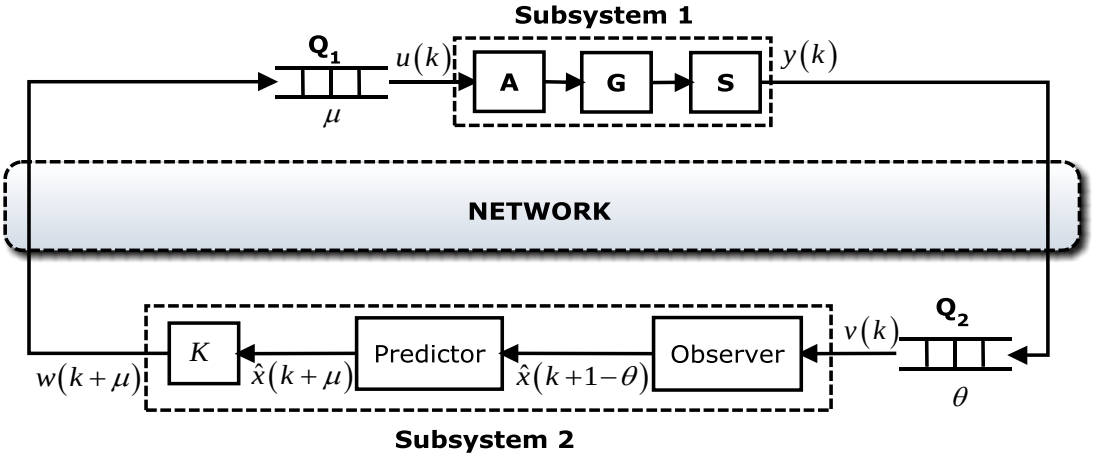


Fig. 3.7 Estimation and Control scheme for NCS with known queue sizes

The control signal is forwarded $u(k+\mu)$ times, μ is the size of the controller-to-actuator buffer. Based on the size θ, sensor-to-controller delay can also be compensated by using an observer. The resulting states are delayed $\hat{x}(k-\theta+1)$ times.

N-step ahead predictions are mostly suitable for NCS applications with delays greater than the sampling period such as: remote control or control over Internet. Liu et al.. [63] uses predictive control for NCS with time-driven components. In [60] this approach is extended for event-driven controllers. To avoid synchronization, the network sends time-stamped measurements to the controller, which generates a set of all possible future control predictions. At this point, network delay has not been entirely compensated. To entirely compensate the delay, the authors propose the use of a network delay compensator on the plant side. This scheme is presented in Fig. 3.8:

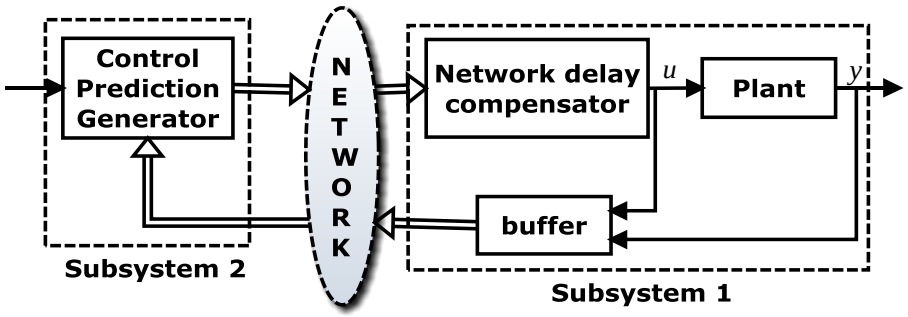


Fig. 3.8 NCS with predictive network delay compensation

The network delay compensator calculates the round trip time delay based on the time-stamped received packets. This round time delay represents the total network delay and is the base for the selection of the future control prediction. Packet dropouts are considered if the delay exceeds the N-steps boundary. The network delay compensator will use the latest control value from the control prediction sequences available on the plant side. These sequences are as follows:

$$
\begin{aligned}
&[u^T_{k-j1|k-j1}, u^T_{k-j1+1|k-j1}, ..., u^T_{k|k-j1}, ..., u^T_{k+N-j1|k-j1}]^T \\
&[u^T_{k-j2|k-j2}, u^T_{k-j2+1|k-j2}, ..., u^T_{k|k-j2}, ..., u^T_{k+N-j2|k-j2}]^T \\
&\vdots \\
&[u^T_{k-jt|k-jt}, u^T_{k-jt+1|k-jt}, ..., u^T_{k|k-jt}, ..., u^T_{k+N-jt|k-jt}]^T
\end{aligned}
$$

For u_{k-j_i}, $i = 1, 2, ..., t$ and j beign the control horizon, these values are available to be used as the control input of the plant at time k. Thus the output of the network delay compensator will be:

$$u_k = u_{k|k-\min\{j1, j2, .., jt\}} \tag{3. 53}$$

In [105], a more detailed networked predictive control system NPC is presented as an Internet-based control strategy.

To reduce processing time, a network delay compensator can be added on the controller side as in Fig. 3.9.

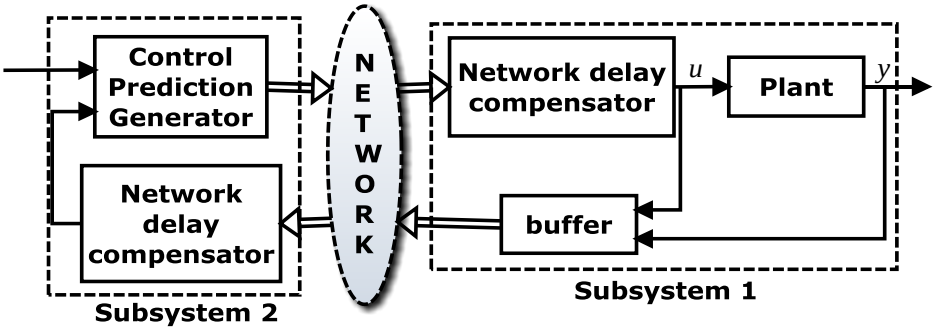


Fig. 3.9 Networked Predictive Control

Again the network delay is compensated by choosing the latest transmitted prediction sequences available on the actuator or the controller.

A similar control methodology is derived using this network characteristic is called Packet-based control. It was firstly introduced in [105]. Zao [106] extends this approach to compensate network delays, packet dropouts and missequencing. The scheme used resembles Fig. 3.8 with the implementation of state feedback control laws. The network delay compensator is replaced for a more simple structure called Control Action Selector (CAS). Synchronized operation is expected and the delays are considered separately. The resulting control action is given by:

$$u(t) = K\left(\tau_{SC,k}, \tau_{AC,k}\right) \cdot x\left(k - \tau_{SC,k} - \tau_{CA,k}\right) \tag{3. 54}$$

The controller sends a packet with a set of predicted control actions; the CAS saves the latest time stamp sequence and compares times with a sequence already stored. If the calculated time of the latest sequence is bigger than the previous one, the sequence is applied; otherwise missequencing is assumed and the sequence is discarded.

Packet-based methodology is an accurate method for communication network constraints, nonetheless, it is restricted to Ethernet-based and Internet applications.
Predictor-like and packed-based methodologies are usually combined with estimation techniques. Provided synchronized communications, the network delay in the sensor-to-controller side can also be compensated [107]. As presented in [20], predictors can be used to estimate the plant outputs between two successive transmission times.

Beldiman et al. [10] present sufficient conditions for NCS stability with static scheduling algorithms when a predictor is introduced in the loop.

The structure of the predictor is chosen based on the speed of the network. For slow networks an open loop predictor is proposed. The control law to be implemented is: $u(t) = -Ky(t)$. Thus the open loop predictor will be defined as follows:

$$\dot{\tilde{x}}(t) = a_p \tilde{x}(t) + b_p u(t) \tag{3. 55}$$

$$\tilde{y}(t) = c_p \tilde{x}(t) \tag{3. 56}$$

with $\tilde{x}(t) \in \mathbf{R}^{N_x}$ the estimated state. In addition, it is assumed that c_p is invertible to define an updating law given by:

$$\tilde{x}(t_i +) = \tilde{x}(t_i) + c_p^{-1} v_i^T v_i c_p (x - \tilde{x}) \tag{3. 57}$$

where i represents the channel transmitted at transmission time t_i and $v_i = [0,...,0,1,0,...,0] \in \mathbf{R}^{N_y}$ is a row vector with 1 on the ith position.

For fast networks, all the states can be estimated and closed loop predictors can be proposed. The resulting predictor will be:

$$\dot{\tilde{x}}(t) = a_p \tilde{x}(t) + b_p u(t) + L[y(t) - w(t)] \tag{3. 58}$$

$$w(t) = c_p \tilde{x}(t) \tag{3. 59}$$

Both structures assure asymptotic stability for a given Maximum Allowable Transfer Interval (MATI) that depends on the scheduling algorithm.

Zhang et al. [107] extend these techniques to estimate the state $x(t + \tau_{SC,k})$ with $\tau_{SC} < h$ and to calculate a forward control action. The resulting control action is given by:

$$u(k + \tau_{SC,k}) = -K\bar{x}(k + \tau_{SC,k}) \tag{3. 60}$$

with $\bar{x}\left(k+\tau_{SC,k}\right)=e^{a\cdot\tau_{SC,k}}x(k)+\int_{k}^{k+\tau_{SC,k}}e^{a\cdot\left(k+\tau_{SC,k}-s\right)}bu(s)ds$.

Partial information can also be compensated with estimation techniques. The calculation of the estimated state is made in two steps. Firstly the past estimated state is projected forward one step and then it is compensated using the receiving plant output. The estimator is built as follows:

$$\bar{x}(k)=\hat{x}(k)+L_e\left[y(k)-C_p\hat{x}(k)\right] \tag{3. 61}$$

This correction is based on $y(k)$. $\bar{x}(k)$ is the estimated state and $\hat{x}(k)$ is the projected state. At time $k+1$ the resulting projected state will be:

$$\hat{x}(k+1)=e^{a\cdot\left(h-\tau_{SC,k}\right)}\bar{x}\left(k+\tau_{SC,k}\right)+\int_{k+\tau_{SC,k}}^{k+1}e^{a\cdot(k+1-s)}bu(s)ds \tag{3. 62}$$

Consequently, it is shown that the separation principle holds for the estimator design without loss of stability.

This model can also be used for asynchronized communications. However the observer design may be extended to intermittent methodologies such as intermittent Kalman Filtering [86] or estimation techniques with uncertain observation [70]. Intermittent methodologies are rather information-dependent.

3.3 Conclusion

The use of different control methodologies and their applicability to NCS was presented in this chapter. The methodologies for control and estimation show the necessity of real-time adaptation to network traffic variation. These variations appear on the controller as unreliable information arrivals, thus information-dependent methodologies constitute general solutions for NCS. It can also be stated that network constraints can be approximated into variable but bounded delays. These delays

reduce the observability of the system and the reliability of the information transmitted, making estimation necessary. The estimation of time-delay systems leads to models with increased dimensions. To make a given methodology applicable, model transformations are needed to reduce the computational burden. Finally due to the discrete nature of the network delays, the control and estimation methodologies to be applied have to be discrete combined with time-driven components. For event driven components, structured controllers can be implemented to improve the system performance. These characteristics are exploited on the forthcoming chapters for control and estimation of NCS.

Chapter 4

PID Controller design for NCS: a pseudo-probabilistic approach

This chapter presents a PID controller design methodology for NCS. The methodology incorporates the variability of the network delay using the pdf of the delay. In section 4.1, the closed loop system of LTI-NCS and PID controller is determined for systems with delayed inputs. Section 4.2 shows delay-dependent Stabilization of time delayed systems with implicit transformations. In section 4.3 the pdf of the delay is bounded for controller design purposes. A polytope is determined based on a range of delay values that assure good performance. Worst delay value is used just for stability analysis. A numerical example is presented to test the design approach.

4.1 Introduction

Most of the literature on PID controllers for NCS has reported the necessity of either using parameter optimization [28], [77] and [107]; or adaptive parameter adjustment [26] and [56] to achieve desired specifications. In an optimal fashion, the controller design is jointly feasible if the closed loop system meets the design specifications

[35]. This feasibility in NCS can lead to a search for controllers with higher dimensions or more sophisticated type such as LQR or LQG. Additionally, sufficient conditions for stability have to be derived from maximum allowable delays ($\tau_{\max}$).

The controller design depends on information availability. Assuming that the plant is controllable and observable, a typical NCS can be thought as two subsystems, the networked controlled system $\sum_1(t)$ and the controller $\sum_2(k)$; interacting across a common communication network.

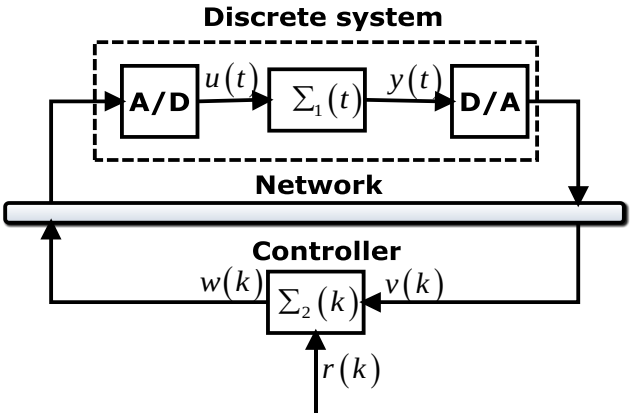


Fig. 4. 1 A general Framework for NCS with structured controllers

$\Sigma_1(t)=(a_p,b_p,c_p)$ is described as in Chapter 3 in equation (3. 8). The discrete form is obtained at discrete time instants $t_k=kh$, where h represents the sampling time. Using h as time reference, the information transmitted by the sensor output $y(k)$ reaches the controller input $v(k)$ after $\tau_{SC}(t_k)$. The information transmitted by the controller output $w(k)$ is sent immediately and updates the actuator input $u(k)$ after $\tau_{CA}(t_k)$. Consequently, as defined in Chapter 2 equation (2. 14) and with both delays lumped together as in Chapter 3 equation (3. 8); the delayed system input can be written as:

$$u(hk)=w(hk-\tau(k)) \tag{4. 1}$$

with $\tau(k)$ as the round trip delay presented as follows:

$$\tau(k) = \tau_{SC}(k) + \tau_{CA}(k) \tag{4. 2}$$

$\Sigma_2(k) = (A_c, B_c, C_c, D_c)$ represents a discrete controller. The controller is dependent on the quality of available transmitted information. Without an accurate dynamic model of the communication network dynamics or synchronization in the control loops; information on the controller side is limited. The easiest way to design a controller for NCS is to use known nominal values of the delay ($\tau_{\min}$). This solution is too conservative because delays can be random and variable. Moreover, although the controller may be accurately tuned, the randomness and variability of the communication network delays reduce the system performance.

As stated by Silva [85], PID controllers are stable when the controller parameters depend of the delays. The control signal of a PID controller is based on the past (I), present (P) and future (D) input errors.

$$w(t) = w_P(t) + w_I(t) + w_D(t) \tag{4. 3}$$

The standard structure of the PID controller in time domain is as follows:

$$w(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(\theta)\,d\theta + T_d\frac{de(t)}{dt}\right] \tag{4. 4}$$

To obtain a discrete form of the controller, the error $e(k)$ is computed at discrete time instants $t_k = kh$. The resulting discrete actions are:

Proportional Action:

$$w_p(t_k) = K_p e(k) \tag{4. 5}$$

where K_p is the proportional gain.

Integral Action:

$$w_I(t) = \frac{K_p}{T_i}\int_0^t e(\theta)\,d\theta$$

$$w_I(kh) = K_i \sum_{n=0}^{k-1} he(n) \tag{4. 6}$$

Using first backward difference, the discrete integral (4. 6) will become:

$$\begin{aligned} \Delta w_I(k) &= K_i h \sum_{n=0}^{k-1} \Delta e(n) \\ w_I(k) - w_I(k-1) &= K_i h\big(e(k) - e(0)\big) \\ \big(1 - q^{-1}\big) w_I(k) &= K_i h e(k) \\ w_I(k) &= \frac{K_i h}{\big(1 - q^{-1}\big)} e(k) \end{aligned} \tag{4. 7}$$

here, $K_i = \dfrac{K_p}{T_i}$ is the integral gain and q^{-1} is the back shift operator.

Derivative Action:

The derivative action is implemented by adding a first order filter to the derivative term to reduce the effects of high frequency disturbances. The derivative term is determined by using the first backward difference discretization as follows:

$$\begin{aligned} w_D(t) &= -\frac{T_d}{N} \frac{dw_D(t)}{dt} + k_p T_d \frac{de(t)}{dt} \\ w_D(k) &= -\frac{T_d}{N}\left(\frac{w_D(k) - w_D(k-1)}{h}\right) + K_d\left(\frac{e(k) - e(k-1)}{h}\right) \\ w_D(k) &= \left(\frac{T_d}{T_d + Nh}\right) w_D(k-1) + \left(\frac{K_d N}{T_d + Nh}\right)\big(e(k) - e(k-1)\big) \\ \big(1 - a_d(h) q^{-1}\big) w_D(k) &= b_d(h)\big(1 - q^{-1}\big) e(k) \\ w_D(k) &= \frac{b_d(h)\big(1 - q^{-1}\big)}{\big(1 - a_d(h) q^{-1}\big)} e(k) \end{aligned} \tag{4. 8, 4. 9}$$

with N defined as a filtering factor in a range of 2-10 [74]; and $K_d = k_p T_d$ being the derivative gain.

The discrete control signal $w(k)$ can now be written as follows:

$$w(k) = w_P(k) + w_I(k) + w_D(k) \tag{4. 10}$$

$$w(k) = K_p e(k) + \frac{K_i h}{\left(1 - q^{-1}\right)} e(k) + \frac{b_d\left(1 - q^{-1}\right)}{\left(1 - a_d q^{-1}\right)} e(k)$$

$$w(k) = \frac{\left(d_0 + d_1 q^{-1} + d_2 q^{-2}\right)}{\left(1 + c_1 q^{-1} + c_2 q^{-2}\right)} e(k) \tag{4. 11}$$

where:

$$c_0 = 1$$
$$c_1 = -\left(1 + a_d(h)\right)$$
$$c_2 = a_d(h)$$

$$d_0 = K_p + K_i h + b_d(h)$$
$$d_1 = -\left(2b_d(h) + c_2 K_i h - c_1 K_p\right)$$
$$d_2 = c_2 K_p + b_d$$

$$a_d(h) = \frac{T_d}{T_d + Nh}$$
$$a_d(h) = \frac{K_d N}{T_d + Nh}$$

The controller parameters are sample-rate-dependent $(c_i, d_i)(h)$, $i = 0,1,2$. The sampling time can be set as $h > \tau(k)$, so that the controller can be re-tuned every time the delay changes. This solution can also incorporate network utilization and leads to scheduling methodologies. The main disadvantage of this methodology is that delays may be less than the sampling time. For delays bigger than the sampling time such as Ethernet delays in Internet applications, models transformations like in Chapter 3, equation (3. 44) can be useful, but as shown in equation (3. 45), this solution leads to controllers with higher dimensions. A more simple approach is to include the delays using delay dependent Stabilizability.

4.1.1 Model transformation

From Fig. 4. 1 and delays as defined in equation (4. 2), it can be stated that subsystem $\Sigma_1(t)$ is delay-free, but the closed loop $\Sigma_{cl}(k)=(\Sigma_1(k),\Sigma_2(k))$ is $\tau(k)$ times delayed.

To determine the closed loop system of Fig. 4. 1, the controller of equation (4. 10) is used in state-space representation. The resulting discrete PID controller will be:

$$x_c(k+1)=A_c x_c(k)+B_c e(k) \tag{4. 12}$$

$$w(k)=C_c x_c(k)+D_c e(k) \tag{4. 13}$$

$$e(k)=r(k)-v(k) \tag{4. 14}$$

where the matrix coefficients are:

$$A_c=\begin{pmatrix}0 & 1\\ -c_2 & -c_1\end{pmatrix}$$

$$B_c=\begin{pmatrix}0\\ 1\end{pmatrix}$$

$$C_c=\begin{pmatrix}d_2-c_2d_0 & d_1-c_1d_0\end{pmatrix}$$

$$D_c=d_0$$

The above state-space representation of the PID discrete controller is stable if $c_2<1$. The state-space formulation is presented in appendix A.1.

Combining the discrete PID controller of equations (4. 12) to (4. 14) with the equation (3. 8), the resulting closed loop system is given by:

$$x_{cl}(k+1)=(A+B_0KC)x_{cl}(k)+B_1KCx_{cl}(k-\tau(k))-B_0KRr(k)-B_1KRr(k-\tau(k)) \tag{4. 15}$$

$$\tau_{\min}\le\tau(k)<\tau_{\max}$$

with:

$$A + B_0 KC = \begin{pmatrix} A_p & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} -D_c & C_c \\ -B_c & A_c \end{pmatrix} \begin{pmatrix} C_p & 0 \\ 0 & I \end{pmatrix}$$

$$B_1 KC = \begin{pmatrix} B_p & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -D_c & C_c \\ -B_c & A_c \end{pmatrix} \begin{pmatrix} C_p & 0 \\ 0 & I \end{pmatrix}$$

$$R = \begin{pmatrix} I \\ 0 \end{pmatrix}$$

and

$$x_{cl}(k) = \begin{pmatrix} x(k) \\ x_c(k) \end{pmatrix}$$

A_p and B_p are calculated as in chapter 2. Equation (4. 15) contains a delayed state $x_{cl}(k - \tau(k))$ and consequently $\Sigma_{cl}(k, \tau(k))$ is delayed. In order to control and stabilize the closed loop system $\Sigma_{cl}(k, \tau(k))$, a delay-dependent stabilizability approach is employed.

4.2 Delay-dependent Stabilizability

Stability analysis does not necessarily require the incorporation of the delays. Delay-independent Stability and Stabilizability are too restrictive because it does not depend on the delay [13] and stability can be achieved if the eigenvalues of $\Sigma_1(k)$ are lesser than 1, $|\lambda(A)| < 1$. This stability test does not take into account the effects of the delays, so that the controller should only be designed based on the worst case delays. If time delays are taken into account, the controller design becomes delay-dependent and the resulting stability test can be expressed as follows:

$$|\lambda(A + BK)| < 1 \qquad (4.\ 16)$$

where $\lambda(A + BK)$ are the eigenvalues of the closed-loop system.

Lyapunov-Krasovskii functional can be used to prove stability and Stabilizability. Equation (4. 15) can be transformed to include the delays explicitly in the resulting LMI. A explicit model transformation leads to terms within the LMI that contain quadratic controller gains, making the resulting LMI, unsolvable. A more convenient approach can be based on that during the bounding in deriving stability criteria, information $x_{cl}(k+\theta)$, $-\tau \le \theta \le 0$ is used. Thus the constraint:

$$x_{cl}(k+1+\theta) = (A + B_0 KC)x_{cl}(k+\theta) + B_1 KCx_{cl}(k-\tau(k)+\theta) - B_0 KRr(k+\theta) - B_1 KRr(k-\tau(k)+\theta)$$

is partially but not fully accounted for as described in [44]. This method is called implicit transformation. The derivation is presented in Appendix A.2. Using implicit transformation over a Lyapunov-Krasovskii functional, stabilizability can be achieved as follows:

Theorem 4.1

Given that all state variables of the system in equation (4. 15) are measurable and available for feedback, the discrete state-space system $\Sigma_1(k, \tau(k))$ is stabilizable. The closed loop system is stable if and only if, given values $P = P^T > 0, X^T = X, Y, Z^T = Z$ and τ_j, $j = 1,2,3$; the following set of LMI's are feasible:

$$\tilde{N} = \begin{bmatrix} \tilde{N}_{11} & -Y & -\bar{A}^T Y^T \\ -Y^T & -S + \bar{B}^T P \bar{B} & -\bar{B}^T Y^T \\ Y\bar{A} & Y\bar{B} & \frac{1}{\tau_j} X \end{bmatrix} < 0 \qquad (4.\ 17)$$

Proof. The formulation requires a Lyapunov-Krasovkii functional of the form:

$$V(x_{cl}, k) = x_{cl}(k)^T P x_{cl}(k) + V_1(x_{cl}, k) + \sum_{j=-\tau_k}^{-1} x_{cl}(k+j)^T S x_{cl}(k+j) \qquad (4.\ 18)$$

More explicitly, the functional can be written with the initial condition ϕ as follows:

$$V(\phi) = V_1(\phi) + V_2(\phi) + V_3(\phi) \qquad (4.\ 19)$$

with:

$$V_1(\phi) \equiv \phi(0)^T P \phi(0) \tag{4. 20}$$

$$V_2(\phi) \equiv \sum_{j=-\tau_k}^{-1} \sum_{i=j}^{-1} f^T(\phi(i)) Z f(\phi(i)) \tag{4. 21}$$

$$V_3(\phi) \equiv \sum_{j=-\tau_k}^{-1} \phi(j)^T S \phi(j) \tag{4. 22}$$

And

$$f(\phi(0)) = (A + B_0 KC)\phi(0) + B_1 KC\phi(-\tau_k) \tag{4. 23}$$

The first forward difference of $V(\phi)$ will be as follows:

$$\Delta V_1(\phi) = \Delta\phi(0)^T P \Delta\phi(0)$$

$$\Delta V_1(\phi) = \phi(1)^T P \phi(1) - \phi(0)^T P \phi(0) \tag{4. 24}$$

$$V_2(\phi) = \sum_{j=-\tau_k}^{-1} \left[f^T(\phi(0)) Z f(\phi(0)) - f^T(\phi(j)) Z f(\phi(j)) \right]$$

$$V_2(\phi) = \tau_k f^T(\phi(0)) Z f(\phi(0)) - \sum_{j=-\tau_k}^{-1} f^T(\phi(j)) Z f(\phi(j)) \tag{4. 25}$$

For $V_3(\phi)$

$$\sum_{j=-\tau_k}^{-1} \Delta\phi(j)^T S \Delta\phi(j) = \phi(0)^T S \phi(0) - \phi(\tau_k)^T S \phi(\tau_k) \tag{4. 26}$$

Assuming $\phi(1) = f(\phi(0))$, equation (4. 24) can be written as follows:

$$\begin{aligned} \Delta V_1(\phi) = \phi(0)^T \left[(A + B_0 KC)^T P (A + B_0 KC) - P \right] \phi(0) \\ + \phi(-\tau_k)^T (B_1 KC)^T P (B_1 KC) \phi(-\tau_k) \\ + 2\phi(0)^T (A + B_0 KC)^T P (B_1 KC) \phi(-\tau_k) \end{aligned}$$

Considering implicit transformation for $V_1(\phi)$, $\Delta V_1(\phi)$ can also be written as follows:

$$\begin{aligned}\Delta V_1(\phi) \le \phi(0)^T \left[(A+B_0KC)^T P (A+B_0KC) - P \right] \phi(0) \\ +\phi(-\tau_k)^T (B_1KC)^T P (B_1KC) \phi(-\tau_k) \\ +2\phi(0)^T (A+B_0KC)^T P (B_1KC) \phi(-\tau_k) \\ +\sum_{j=-\tau_k}^{-1} \begin{bmatrix} \phi(0)^T & \Delta\phi(j)^T \end{bmatrix} \begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} \begin{bmatrix} \phi(0) \\ \Delta\phi(j) \end{bmatrix}\end{aligned}$$

Consequently $\Delta V(\phi)$ can be written as follows:

$$\Delta V(\phi) \le \phi_{0\tau}{}^T \begin{bmatrix} N_{11} & N_{12} \\ N_{12}^T & N_{22} \end{bmatrix} \phi_{0\tau} \tag{4. 27}$$

$$\phi_{0\tau} = \begin{bmatrix} \phi(0) \\ \phi(-\tau_k) \end{bmatrix}$$

$$\begin{aligned} N_{11} &= \bar{A}^T P \bar{A} - P + S + \tau_k \bar{A}^T Z \bar{A} + \bar{A}^T P \bar{B} + X + Y + Y^T \\ N_{12} &= \bar{A}^T P \bar{B} + \tau_k \bar{A}^T Z \bar{B} - Y \\ N_{22} &= -S + \tau_k \bar{B}^T Z \bar{B} + \bar{B}^T P \bar{B} \\ \bar{A} &= A + B_0 KC \\ \bar{B} &= B_1 KC \end{aligned}$$

Additionally considering that $Z + Y^T X^{-1} Y = 0$, then $Z = -Y^T X^{-1} Y$, and N can be expressed without Z as in equation (4. 17) and this proves the theorem.

Theorem 4.1 considers that τ_j, $j = 1,2,3$ is known and constant during the control design. To obtain the values of τ_j the pdf of the delay can be used and bounded to a closed interval $[\tau_1, ..., \tau_3]$, where good performance is guaranteed. Adding a known interval, so that a set of LMI's can be defined and leads as to a probabilistic control design approach.

4.3 Controller Probabilistic Design Approach

The resulting LMI from equation (4. 17) $\tilde{N}\left(\tau\left(k\right)\right)$ is subject to $\tau\left(k\right)$ and represents a convex set:

$$\Omega = Co\left\{\tilde{N}\left(\tau_{\min}\right),\cdots,\tilde{N}\left(\tau_{\max}\right)\right\} \tag{4. 28}$$

with Co referring to the convex hull and defined by linear models. The number of linear models in the above set is not finite since $\tau\left(k\right)$ can take any value. Using the information of the pdf of the delay, Ω may be approximated to a finite number of linear models.

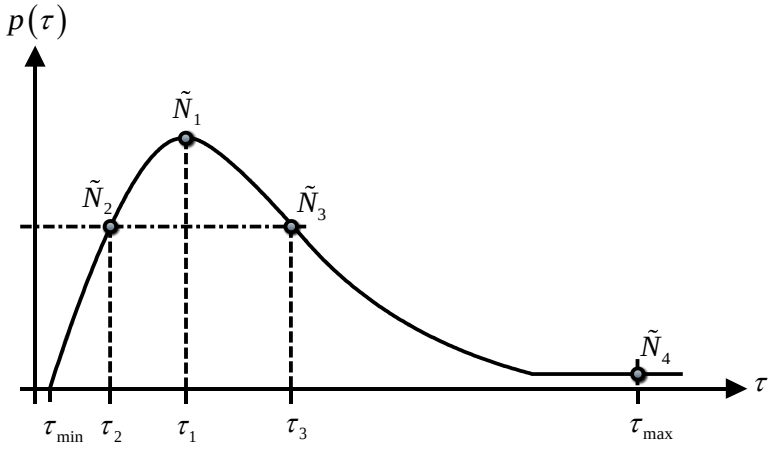


Fig. 4. 2 pdf. of the delay with known interval delay.

For stability analysis the worse delay value can be used $\tilde{N}\left(\tau_4\right)$. However, to achieve good performance, it is sufficient to include a set of models corresponding to the most probable value and other values defined by a tuning parameter γ.

$$\Omega = Co\left\{\tilde{N}_1\left(\tau_1\right),\tilde{N}_2\left(\tau_2\right),\tilde{N}_3\left(\tau_3\right)\right\} \tag{4. 29}$$

These models represent a polytope to be used in the controller design. γ is a sweeping factor in the pdf of the delay and allows the selection of a set of possible models with good performance for a given controller.

Design specifications can be achieved for particular system models. An optimum solution will determine all possible controller parameters for each model, making performance and stability model-dependent. The trade-off between good performance and stability can be stated as a design constraint. It is logical that the controller design should be based on the most probable delay. Using the tuning parameter, a finite set of models can be included in the design. However, the bigger the range of these models involve, a lesser performance is achieved. A convenient approach is to design the controller based on good performance. It is expected that this approach presents performance degradation when the delay is out of the designing parameters range.

Assuming that the polytope is known $\tilde{N}_j(\tau_j)$, $j=1,2,3$, at least two steps are necessary in the controller design. Firstly the LMI that represents the most probable delay $\tilde{N}_1(\tau_1)$ must be feasible. If the problem is feasible for this LMI, it can be stated that there exists a controller that satisfies all the set, otherwise less conservative specifications have to be considered.

The other models depend on the tuning parameter γ. By sweeping the pdf a range of possible models can be bounded by the set $\{\tilde{N}_2(\tau_2),...,\tilde{N}_3(\tau_3)\}$.

Given $\tilde{N}_4(\tau_4)$, a conservative stability test can be made. If the LMI $\tilde{N}_4(\tau_4)$is feasible, then stability is assured for all the set. These results are shown via a numerical example in the following section.

4.4 Numerical example

To illustrate the above design method, the step response of a SISO NCS is simulated and system stability is observed. The NCS is connected to a 10 node netwok. $\tau(k)$ is the lumped delay and represents the round trip delay for Ethernet-based.

Communication network parameters and system are taken from [39] to match with real values. The packet size is 8 bytes. Each node represents a system as follows:

$$G(s)=\frac{1}{10s^3+5s^2+10.4s+1}$$

The total time to send 10 messages over the communication networks is [53]: 576 μs for Ethernet. If the period is shorter than the total transmission time the traffic load increases and the communication network can become unstable. In the simulations the sampling period will be equal to $\tau_{(\min)}$, thus the communication network induced-delay is zero or $\tau(k)$.

Assuming a tuning factor γ corresponding to the 70% of confidence of the probability of the delay, the following delays are obtained: $\tau=\begin{bmatrix}789e-6 & 853e-6 & 901e-6\end{bmatrix}$
The polytope will be:

$$x(k+1)=\begin{pmatrix} .99 & -.0006 & -.0001 \\ .0006 & 1 & 0 \\ & .0006 & 1 \end{pmatrix}x(k)+.213u\left(k-\tau(k)\right)$$
$$h=572e-6$$

Solving a feasibility problem for the resulting polytope, the following gain values were obtained:

$$B_c=\begin{bmatrix}-.0007 \\ .0012\end{bmatrix} \qquad D_c=-.0026$$

Simulations are based on a Simulink model using TrueTime1.5. This simulator allows co-simulation of controller-task execution in real-time kernels, network transmission and continuous plant dynamics [73]. Models 1 and 3 from Fig. 4. 2 are tested. Two controllers are implemented as nodes in an 10Mbps Ethernet network. The remaining nodes are simulated as network traffic.

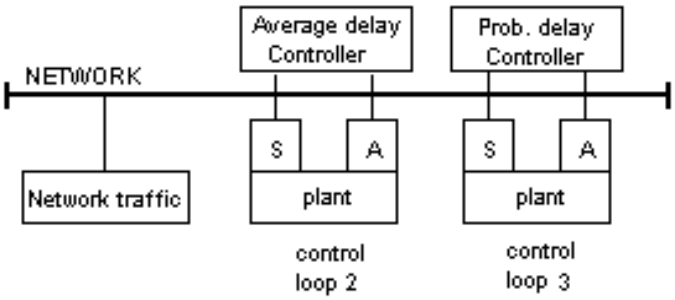


Fig. 4.3 Control Loops block diagram

The delays used to obtain the models are simulated as random delay values $\tau_1 \le \tau_k \le \tau_3$. The following results were obtained:

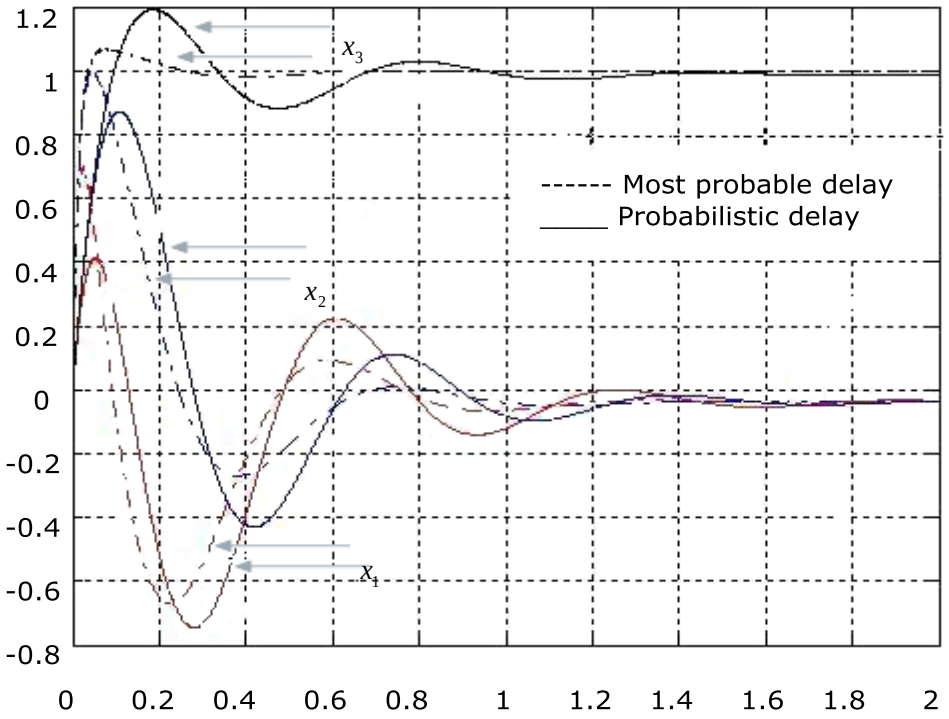


Fig. 4.4 System states for τ_1. Dotted lines represent system states with the controller designed using the most probable delay. Continuous lines represent system states with controller designed with the probabilistic delays.

The simulation results show better performance for model 1 $\tilde{N}_1(\tau_1)$ with the controller designed for the most probable delay. The simulation results for the controller designed using the range of delays shows small performance degradation.

The controller designed only for the most probable delay is the best controller for model 1, thus this degradation is expected.

The simulation results of model 3 using the designed controllers are presented in Fig. 4. 5. The system is still controllable with the probabilistic controller, although with a slightly higher performance degradation. The system can no longer be controlled by the most probable delay controller. Thus the resulting controller design offers relative good performance for the range of bounded delays as expected. The performance can be increased if the controller design includes more delay values within the range.

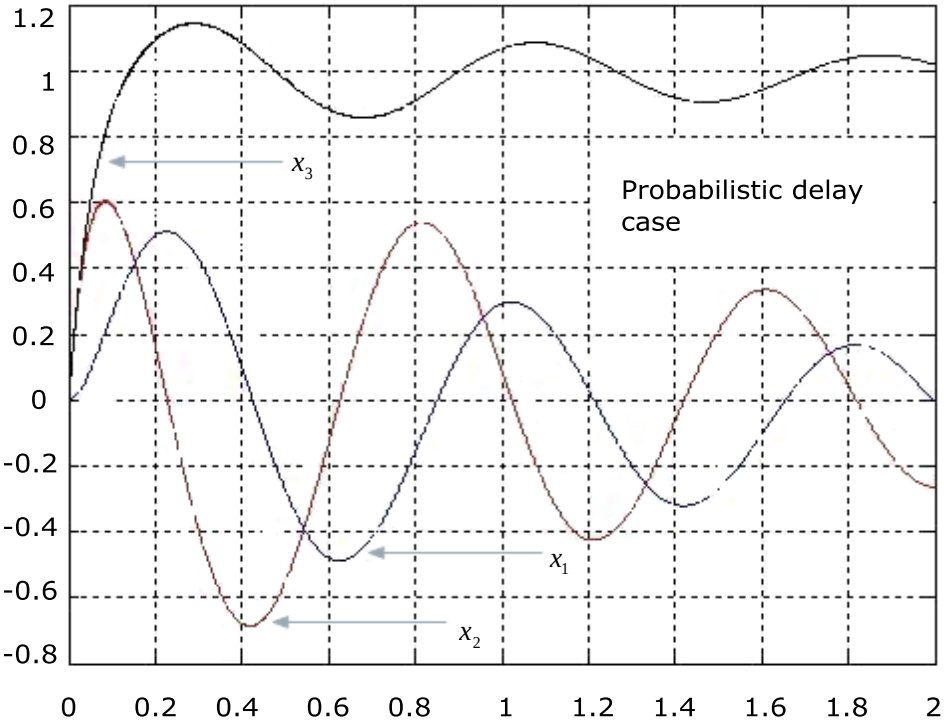


Fig. 4. 5 System states for model $\tilde{N}_1(\tau_3)$. Continuous lines represent system states with controller designed with the probabilistic delays. Dotted lines are not included because simulation results are unstable.

4. 5 Conclusions

It was argued that to achieve good performance, reliable networks and system performance are needed. Furthermore, instability of the NCS is merely instability of the communication network due to heavy traffic.

Communication network induced-delays have been lumped together for design purposes. The PID controller design is based on the expectation of the most probable delay accompanied by a set of models that depends on a tuning parameter γ.

Packet dropouts as well as quantization have not been considered. Additionally it has been stated that dynamic bandwidth allocation is a varying communication network parameter that need to be included. Stability analysis should be extended by using time-delay systems theory.

Chapter 5

Estimation approach for Networked Control Systems

This chapter presents a modified discrete Kalman filter combined with a bad data detector for sensor-to-controller delay and packet dropouts compensation. This methodology is applied on time-driven components to use the filter updating time as reference for the compensation. The structure of this chapter is as follows: Section 5.2 demonstrates the formulation of the discrete Kalman filter for systems with neither delays nor packet dropouts. This standard algorithm is modified to include sensor-to-actuator delays and a bad data detector is formulated to use the resulting filter to compensate both delays and packet dropouts. In section 5.3, controller-to-actuator delays are incorporated in the system model and two control methodologies are proposed LQR and MPC, both on NCS with Arstein model transformation. A numerical example is presented in section 5.4. The resulting control methodology is a combination of the Kalman filter for estimating the system states affected by network constraints and a MPC controller.

5.1 Introduction

The complexity of NCS is not in the process, it is rather in the way the information is deployed through the communication network. The distributed nature of NCS carries overload of information in real-time systems. Time-critical applications require constant information flow as well as real-time computation but not all the communication channels are ideal data transmission medium, they can be contaminated with noise and errors. Missequencing, packet dropouts, network-induced delays and limited network bandwidth emerge when a control loop is closed across a network. These network constraints reduce performance of the NCS by affecting the observability of the process as well as the reliability of output measurements and control signals.

Lack of observability can be compensated by using observers to estimate necessary signals. Conventionally, Estimation techniques can be defined as the problem of estimating the pdf of the states of the process that are not directly observable, using the past measurements. Thus unreliable measurements may reduce the optimality of the applied observer.

Observers in NCS are difficult to design and implement. A trade-off between optimality and computational burden is needed for the observer to be applicable and useful [51]. Observing the states of the process through a communication network adds additional dynamics to the problem that the control techniques have to address. In an observer for NCS, network constraints are sample-dependent. They can change every time step, making tracking almost impossible [78]. Furthermore, not all the network constraints can be treated together. It is possible to assume packet dropouts as infinite delays, as demonstrated in chapter 2, and reduce the NCS problem to delayed systems with finite delays.

When only network delays are present, the model of the observer can be augmented to accommodate delayed measurements. This approach is acceptable for small delays. Model augmentation increases the computational burden based on the size of the

delay and the sample time, thus the model dimensions can become undesirably high [43].

Conventional techniques for systems with delays require the delay to be known and constant which contradicts variable and random nature of network-induced delays. On the other side, recursive solutions can handle the variability and randomness of the delay as far as it is known at the beginning of the recursion. Consequently, it is reasonable to include a pre-estimation stage to estimate the delay value.

Estimation techniques based on noisy measurements are very common in industry. The most known technique is the discrete Kalman filter. The implementation of Kalman filtering algorithms offers an optimal estimation of the systems states when the process is linear and the noise is Gaussian. The following section discusses the design of a Kalman filter for NCS systems.

5.2 Standard Kalman Filtering

Kalman Filtering is a plausible solution for NCS with time-driven components. Time-driven components have their sampling times set. The correct reception of information arrivals is known at the beginning of every sample time. Thus the information arrivals received at time kh as presented in Fig. 2.2 (See chapter 2) for time-driven components, can be $\left(kh-\tau_{SC}\left(kh\right)\right)-$delayed on the observer and $\left(kh-\tau_{CA}\left(kh\right)\right)-$delayed on the actuator. $\tau_{SC}\left(kh\right)$ and $\tau_{SC}\left(kh\right)$ are the sensor-to-controller and controller-to-actuator delays respectively.

The structure of an NCS for control and estimation is presented in Fig. 5. 1. For output feedback synthesis, subsystem $\sum_1\left(t\right)$ represents a LTI stochastic system with Gaussian process noise and measurement noise, $\sum_2\left(k\right)$ is the controller and $\sum_o\left(k\right)$ is the observer. The system is assumed to be controllable and observable.

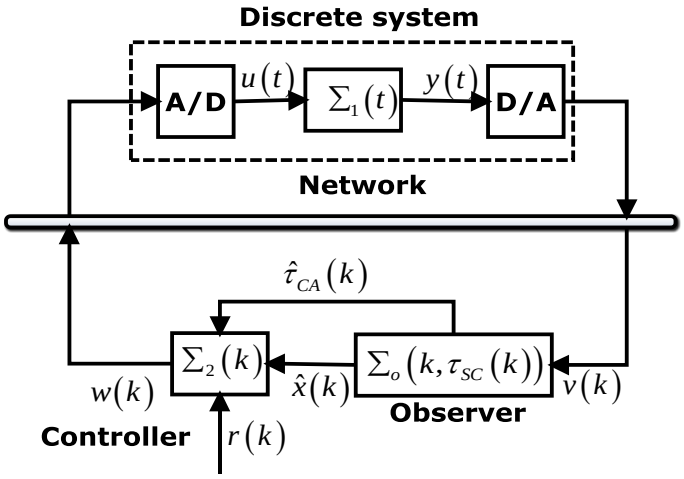


Fig. 5. 1 Feedback output synthesis for NCS

Network constraints do not affect the distribution of the process or measurement noises, therefore these noises will remain Gaussian. Kalman filter algorithms are simple to implement. Having a LTI system of the form:

$$x(k+1) = A_p x(k) + B_p u(k) + \xi(k) \tag{5. 1}$$

$$y(k) = C_p x(k) + \zeta(k) \tag{5. 2}$$

where $y(k) \in \mathbf{R}^{N_y}$, $u(k) \in \mathbf{R}^{N_u}$, $x(k) \in \mathbf{R}^{N_x}$ and all the system matrices A_p, B_p, C_p are of compatible dimensions. $\xi(k) \in \mathbf{R}^{N_x}$ is zero mean value process noise, $\zeta(k) \in \mathbf{R}^{N_y}$ is zero mean value measurements noise where:

$$E\left\{\begin{bmatrix}\xi(k)\\ \zeta(k)\end{bmatrix}\begin{bmatrix}\xi^T(i) & \zeta^T(i)\end{bmatrix}\right\} = \begin{bmatrix}Q & 0\\ 0 & R\end{bmatrix}\Delta(k-i) \tag{5. 3}$$

represents the covariance matrix of the noise with $\Delta(k-i)$ being the Kronecker delta. The initial state is $x(0)$ with mean $E\{x(0)\}$ and covariance matrix $\Pi(0)$, i.e.:

$$\begin{aligned} E\{x(0)\} &= \mu_x(0)\\ \Pi(0) &= E\left\{[x(0)-\mu_x(0)][x(0)-\mu_x(0)]^T\right\}\end{aligned} \tag{5. 4}$$

Assuming that process and measurements vectors are jointly Gaussian, the filter gives the minimum variance estimate of the state, that is, it evaluates the conditional mean of $x(k)$ given a minimal $\sigma - algebra$ that makes $\{x(s), s \le k\}$ measurable and its denoted by the observations $F_k = \sigma\{y(k),...,y(0)\}$ [43]. The $\sigma - algebra$ F_k satisfies $F_{k-1} \subset F_k, k-1 \le k$.

Consequently, if $\hat{x}(k+1)$ denotes the conditional mean of $x(k+1)$, given the observations of F_k up to and including k, then $\hat{x}(k+1)$ satisfies the following recursion:

$$\hat{x}(k+1|k) = A_p \hat{x}(k) + B_p u(k) + K(k) e(k) \qquad (5.5)$$
$$\hat{x}(0) = \mu_x(0)$$

$K(k)$ represents the filter gain and is given by:

$$K(k) = A_p P(k) C_p{}^T \left[C_p P(k) C_p{}^T + R \right]^{-1} \qquad (5.6)$$

$P(k)$ is the state error covariance and is defined as follows:

$$P(k) \equiv E\left\{ \left[x(k) - E\left[x(k)|F_{k-1} \right] \right] \left[x(k) - E\left[x(k)|F_{k-1} \right] \right]^T \right\}$$

and satisfies a Riccati Difference Equation:

$$P(k+1|k) = A_p P(k) A_p{}^T + Q - K(k)\left[C_p P(k) C_p{}^T + R \right] K(k)^T \qquad (5.7)$$
$$P(0) = \Pi(0)$$

For Gaussian processes, the first two moments describe the complete probability distribution. Thus the conditional distribution of $\hat{x}(k+1)$ given F_{k-1} has mean $\mu_x(k+1)$ and covariance $P(k+1)$ determined as follows [31]:

$$E\{x(k+1)|F_{k-1}\} = \hat{x}(k+1) = \mu_x(k) + P_{12} P_{22}^{-1}\left[y(k) - \mu_y(k) \right] \qquad (5.8)$$

$$E\left\{\left[x(k+1)-\hat{x}(k+1)\right]\left[x(k+1)-\hat{x}(k+1)\right]^T|F_{k-1}\right\}=P(k+1|k)=P_{11}-P_{12}P_{22}^{-1}P_{21} \tag{5. 9}$$

with:

$$\begin{bmatrix}\mu_x(k+1)\\ \mu_y(k)\end{bmatrix}=E\left\{\begin{bmatrix}x(k+1)|F_{k-1}\\ y(k)|F_{k-1}\end{bmatrix}\right\}$$
$$=E\left\{\begin{pmatrix}A_p & I & 0\\ C_p & 0 & I\end{pmatrix}\begin{bmatrix}x(k)\\ \xi(k)\\ \zeta(k)\end{bmatrix}+\begin{pmatrix}B_p\\ 0\end{pmatrix}u(k)\right\}$$
$$=\begin{pmatrix}A_p\\ C_p\end{pmatrix}\hat{x}(k)+\begin{pmatrix}B_p\\ 0\end{pmatrix}u(k)$$

and:

$$\begin{pmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{pmatrix}=E\left\{\begin{bmatrix}x(k+1)-E\left[x(k+1)|F_{k-1}\right]\\ y(k)-E\left[y(k)|F_{k-1}\right]\end{bmatrix}\begin{bmatrix}x(k+1)-E\left[x(k+1)|F_{k-1}\right]\\ y(k)-E\left[y(k)|F_{k-1}\right]\end{bmatrix}^T\right\}$$
$$=\begin{bmatrix}A_pP(k)A_p^T+Q & A_pP(k)C_p^T\\ C_pP(k)A_p^T & C_pP(k)C_p^T+R\end{bmatrix}$$

The presence of external inputs in Kalman filtering does not affect the estimation problem when the inputs are $F_k-measurable$. These inputs are functions of the output measurements and consequently they are admissible.

In order to keep the process Gaussian, these inputs should be formed by a linear output feedback process such as:

$$w(k)=L(k)v(k) \tag{5. 10}$$

where $L(k)$ is the controller gain. For linear systems, separation principle holds and the state estimation can be used for feedback, i.e. $w(k)=L(k)\hat{x}(k)$

$e(k)$ represents the estimation error and is defined as follows:

$$e(k) \equiv y(k) - C_p \hat{x}(k) \tag{5. 11}$$

Rearranging the above expression gives:

$$y(k) = C_p \hat{x}(k) + e(k)$$

It is clear that $e(k)$ also called as the innovations sequence, represents the new information contained in $y(k)$, which is not contained in $\{y(k-1),..., y(0)\}$.

5.2. 1 Sensor-to-controller delay compensation

In NCS, the observation process $v(k)$ on the observer is delayed. This delay $\tau_{SC} = 1, 2, ...$ is assumed to be a multiple integer of the sampling time. Smaller delay values than the sampling time can be rounded up in the subsequent computation cycle [43].

At time k the resulting observation process is:

$$v^*(k) = C_p x\left(k - \tau_{SC}(k)\right) + \zeta^*(k) \tag{5. 12}$$

where $v^*(k)$, as presented in Fig. 5. 2, is the delayed output measurement arrived at the filter at time k.

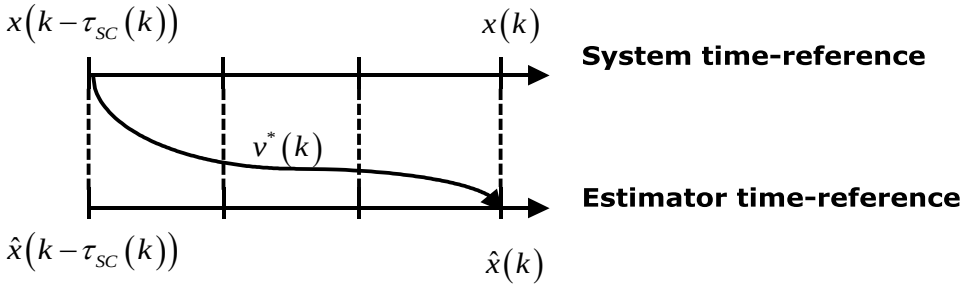


Fig. 5. 2 Delayed Output Measurements arrivals

Optimality cannot be achieved with the Kalman filter algorithm for NCS. The estimation of the delay is no longer optimal and consequently applying Kalman filtering only gives a linear minimum variance estimate of the delay that is not generally the conditional mean. Instead other methods of delay estimation have to be applied i.e. by using Recursive Bayesian estimation [75].
Recursive Bayesian estimation is used by Pahjola et al. [78] to estimate the delay. The most probable delay value is used in an extended model of the system and the information is fused at time k. Kalman filter is used for the estimation of the process. The delay is assumed to be bounded. Furthermore it is stated the use of Kalman filtering, based on the reliability of the estimation of the delay, leads to suboptimal estimation.

Suboptimal estimation affects the covariance matrix and hence the Kalman gain in a complex manner [43]. As stated by Larsen et al. [51], using equation (5. 12), it is possible to perform the fusion by using extrapolated states. Extrapolation can also be useful for packet dropouts. When there is no correlation between packet dropouts and delays, it is impossible to distinguish which constraint produced the trigger event.

The extrapolation causes a correction in the state estimate and consequently a decrease in the covariance matrix. This solution is parametric-based because the correction can be included in the covariance matrix as a correction factor M_* using the system and filter parameters.

5.2.2 Suboptimal measurement fusion

It is possible to sub-optimally fuse the measurements $v^*(k)$ at time k by extrapolating the measurements [43]. Due to the linearity of the innovation process, a definition for an extrapolated innovation process $e^{ex}(k)$ can be stated as follows:

$$e^{ex}(k) \equiv e\left(k-\tau_{SC}(k)\right) = v^*(k) - C_p \hat{x}^*\left(k-\tau_{SC}(k)\right) \tag{5. 13}$$

For time-driven components, the absence of information arrivals triggers an event that represents either delays or packet dropouts. When no information arrivals are present, the delay is increased one time-step and an extrapolated measurement $v^{ex}(k)$ at time k is used, otherwise the delay is kept constant and $v^{*}(k)$ is extrapolated to a present measurement defined as $v^{ex}(k)$ so that:

$$\begin{aligned} v^{ex}(t) &= e\left(k-\tau_{SC}(k)\right)+C_p\hat{x}(k) \\ &= v^{*}(k)-C_p\hat{x}\left(k-\tau_{SC}(k)\right)+C_p\hat{x}(k) \\ &= C_p x\left(k-\tau_{SC}(k)\right)+\zeta^{*}(k)-C_p\hat{x}\left(k-\tau_{SC}(k)\right)+C_p\hat{x}(k) \\ &= C_p x(k)+\zeta^{*}(k)+C_p\tilde{x}\left(k-\tau_{SC}(k)\right)-C_p\tilde{x}(k) \\ v^{ex}(k) &= C_p\hat{x}(k)+\zeta^{ex}(k) \end{aligned}$$

with $\zeta^{ex}(k)=\zeta^{*}(k)+C_p\tilde{x}\left(k-\tau_{SC}(k)\right)$. Equation (5. 12) is similar to equation (5. 2) but now; there exist a correlation between the extrapolated noise process $\zeta^{ex}(k)$ and the state $x(k)$ [51].

The resulting observation process can now be written as:

$$v^{ex}(k)=C\hat{x}(k)+\zeta^{ex}(k) \tag{5. 14}$$

The modified Kalman filter, as given by Larsen et al. [51], is given by the following recursion:

$$\hat{x}(k+1|k)=A_p\hat{x}(k)+B_p w(k)+K\left(k,\tau_{SC}(k)\right)\left[v^{ex}(k)-C_p\hat{x}(k)\right] \tag{5. 15}$$

$$\hat{x}(0)=\mu_x(0)$$

where the Kalman gain and error covariance are functions of the correction factor $M(k)$.

$$K\left(k,\tau_{SC}(k)\right)=A_p M(k)C_p{}^{T}\left[C_p P\left(k-\tau_{SC}(k)\right)C_p{}^{T}+R\right]^{-1} \tag{5. 16}$$

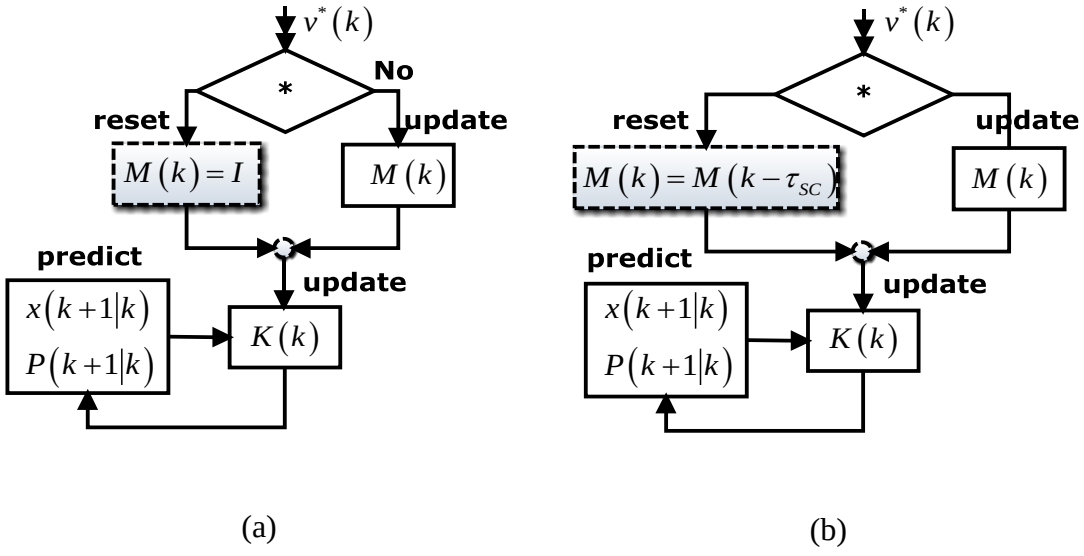
$$P\left(k+1|k,\tau_{SC}\left(k\right)\right)=-K\left(k,\tau_{SC}\left(k\right)\right)\left[C_pP\left(k-\tau_{SC}\left(k\right)\right)C_p{}^T+R\right]K\left(k,\tau_{SC}\left(k\right)\right)^T +Q+A_pP\left(k\right)A_p{}^T \tag{5. 17}$$

$$P\left(0\right)=\Pi\left(0\right)$$

The filter algorithm proof is given in B. 1 and keeps the original initial conditions because assumptions of initial delays can be too conservative.

5. 3 Bad Data Detector

In absence of data arrivals, the modified Kalman filter increases the correction factor. For time-driven components, the absence of data arrivals can be due to either packet dropouts or network delays. Thus the difference is rather in the predefined value of the correction factor after it is reset. For packet dropouts the correction factor is reset to a unity value and the filter performs as a standard Kalman filter [51]. On the other hand, for network delays the correction factor is reset to the last or predefined delayed value.



(a) (b)

Fig. 5. 3 (a) Modified Kalman Filter for packet dropouts; (b) Modified Kalman Filter for network delays

It can be seen that the trigger event in both cases is the lack of data arrivals. It is possible to add a detection mechanism to distinguish between both constraints. This detection mechanism can be defined as a bad data detector of the estimated states $\hat{x}(k)$ and compares the square of the estimation error with a decision threshold. The decision threshold will be the square of the optimal estimation error [32]. Optimality on Kalman filter is proved when the innovations process is unbiased and the covariance of the estimation error is minimum.

To design a bad data detector the observation process is used. This contains the estimation error as well as the innovations process. The detector is implemented to as follows:

Using equation (5. 11) innovations process for packet dropouts is given by:

$$e_{loss}(k) = v^{*}(k) - C_p \hat{x}(k)$$

On the other hand, using equation (5. 13) the innovations process for network delays will be:

$$e_{delay}(k) = v^{*}(k) - C_p \hat{x}\left(k - \tau_{SC}(k)\right)$$

Comparing both innovations process and defining one of them as a decision threshold $e_{th}(k)$, thus:

An increase of network delays has been detected if:

$$e_{loss}^2(k) > e_{th}^2(k) \qquad (5.\ 18)$$

$$e_{th}(k) \equiv e_{delay}(k)$$

Otherwise packet dropouts have been compensated in the previous iteration.

Proof of optimality: if the detector is designed using equation (5. 18) and the inequality is true, then the mean of $e_{delay}(k)$ is equal to the mean of the $\tilde{x}\left(k - \tau_{SC}(k)\right)$ and consequently zero, as follows:

$$
\begin{aligned}
E\left\{e_{delay}\left(k\right)\right\} &= E\left\{v^{*}\left(k\right)-C_{p}\hat{x}\left(k-\tau_{SC}\left(k\right)\right)\right\} \\
&= E\left\{C_{p}\tilde{x}\left(k-\tau_{SC}\left(k\right)\right)+\zeta\left(k\right)\right\} \\
&= CE\left\{\tilde{x}\left(k-\tau_{SC}\left(k\right)\right)\right\} \\
E\left\{e_{delay}\left(k\right)\right\} &= 0
\end{aligned}
$$

$e_{loss}\left(k\right)$ has mean:

$$
\begin{aligned}
E\left\{e_{loss}\left(k\right)\right\} &= E\left\{v^{*}\left(k\right)-C_{p}\hat{x}\left(k\right)\right\} \\
&= E\left\{C_{p}x\left(k-\tau_{SC}\left(k\right)\right)+\zeta\left(k\right)-C_{p}\hat{x}\left(k\right)\right\} \\
&= C\left[E\left\{x\left(k-\tau_{SC}\left(k\right)\right)\right\}-\hat{x}\left(k\right)\right] \\
E\left\{e_{delay}\left(k\right)\right\} &\neq 0
\end{aligned}
$$

Thus the estimation is unbiased and the filter-detector is optimal.

5.4 Controller-to-actuator delay compensation

Network delays are difficult to handle in the controller-to-actuator side during the actual time step. Certain networks retrieve an acknowledgement message once the packets have been received at the destination point. However applications of this type are network-dependent. Without information about the controller-to-actuator delay on the controller, it is difficult to compensate delays and almost impossible to compensate packet dropouts.

A plausible approach is to assume a known average value of the delay before the controller design. Thus, a LQR can be designed for a LTI system with delayed inputs. The resulting control law, although it is optimal for the average delay value, it can not be considered optimal when the delay changes [9]. Furthermore, no packet loss compensation can be implemented.

5.4. 1 LQR design

After the filter, the system is sensor-to-controller delayed-free. Using equation (2. 14), the input delayed system is given by:

$$x(k+1) = A_p x(k) + B_p w\left(k - \tau_{CA}(k)\right) \tag{5. 19}$$

$\tau_{CA}(k)$ can change every sampling time. To remain system dimensionality, the Arstein transformation as presented in equations (3. 44) and (3. 45) can be used. The resulting transformed model is:

$$z(k+1) = A_p z(k) + \left(A_p^{\tau_{CA}(k)} B_p\right) w(k) \tag{5. 20}$$

if and only if $z(k) \in \mathbf{R}^{N_x}$ is a solution of the equation (5. 20) and is defined as follows:

$$z(k) = x(k) + \sum_{j=-\tau_{CA}(k)}^{-1} A_p^{-(\tau_{CA}(k)+j+1)} B_p w(k+j) \tag{5. 21}$$

In accordance to the dynamic programming principle that claims that part of the optimal control function in any subinterval must be the optimal control itself in this subinterval; the optimal control for systems with variable input delays must be designed as much times as delays changes [9].

A LQR can be designed assuming that the delay $\tau_{CA}(k)$ is known and constant $\tau_{CA}(k) = \bar{\tau}_{CA}$, $k = 1, 2, \ldots$. This assumption is restrictive but with no information about the delay, it offers a simple solution with possible system performance degradation.

Optimal Control Problem Solution

The optimal control law for the system given in equation (5. 20) and minimizing a cost function defined as follows:

$$J_c = \frac{1}{2}\langle z(N), Sz(N)\rangle + \frac{1}{2}\sum_{k=0}^{N-1}\left[\langle z(k), Qz(k)\rangle + \langle w(k), Rw(k)\rangle\right]$$

$$s.t.$$

$$z(k+1) = A_p z(k) + \left(A_p^{\bar{\tau}_{AC}} B_p\right) w(k)$$

is given by:

$$w(\mathrm{k}) = -L(k) z(k) \tag{5. 22}$$

with:

$$L(k) = R^{-1}\left(A_p^{\bar{\tau}_{CA}} B_p\right)^T \left(I + \left(A_p^{\tau_{CA}} B_p\right) R^{-1} \left(A_p^{\tau_{CA}} B_p\right)^T K(k+1)\right)^{-1} A_p \tag{5. 23}$$

$$K(k) = Q + A_p^T K(k+1)\left(I + \left(A_p^{\tau_{CA}} B_p\right) R^{-1} \left(A_p^{\tau_{CA}} B_p\right)^T K(k+1)\right)^{-1} A_p \tag{5. 24}$$

Proof. The solution of the optimal problem is given in appendix

Equation (5. 24) is the discrete Riccati Equation where $K(k)$ is known as the Riccati gain. S, Q and R are chosen as follows [14]:

a. Make, for lack of better knowledge, the off diagonal elements zero,
b. Pick the diagonal elements as follows:

$$S_{ii} = \frac{1}{z_i^2(N)_{\max}}$$

$$Q_{ii} = \frac{1}{z_i^2(k)_{\max}}$$

$$R_{ii} = \frac{1}{w_i^2(k)_{\max}}$$

It is also possible to design a LQ regulator based on a polytope that include a range of the most probable delay values, so that, performance can be achieved for certain delay values [79]. Nonetheless, more accurate approaches require a recursive calculation of the controller gain based on the delay variation.

The delayed system states can be forwarded $\hat{\tau}_{CA} - steps$ ahead [61]. The resulting design represents a predictive controller design approach for NCS. If an estimated value of the controller-to-actuator delay can be known every time step, the controller gain calculation can be updated iteratively.

5.4. 2 Predictive Controller design

Bollot [12] states that there exist a correlation between sampled delays and observed packet loss. This correlation can be used to create a Markov model of either delays or packet dropouts. The resulting observations sequences can be used to implement a Bayesian estimator triggered by the result of the bad data detector.

It can also be inferred that data traffic affects the entire network and consequently a change in the constraints on the sensor-to-controller interface evidences a change in the constraints on the controller-to actuator interface. Thus the actuator-to-controller delay value, without considering the network dynamics, can be approximated to the value of the calculated sensor-to-actuator delay:

$$\tau_{CA}^{ap}(k) \equiv \tau_{SC}(k)$$

Obviously packet dropouts can not be compensated, but the control signal $w(k)$ can be forwarded $n - steps$ ahead. The resulting control law is a predictive controller.

Generalized Predicitve Control (GPC) is presented in [61] to compensate bounded sensor-to-controller and controller-to-actuator delays over Internet-based networks. Open loop Predictive Control is used to derive a set of control inputs. The set is then transmitted to the plant. A buffer detector mechanism is implemented to calculate the delay based on time stamps and consequently choose the corresponding delayed control signal.

MPC for state-space models can be reformulated as a LQR framework. The resulting optimization problem over a variable horizon makes it suitable for applications such as time-delayed systems [49].

Theorem 5.1

For the system given in equation (5. 20) and variable controller-to-actuator delay $\tau_{CA}^{ap}(k)$, the optimal predictive control law that minimizes the following predictive objective function:

$$J_{\left(H_P,H_C,\tau_{CA}^{ap}(k)\right)} = \sum_{j=1}^{H_P} e_{k+j}^T Q_1 e_{k+j} + \sum_{j=0}^{H_C-1} \Delta w_{k+j}^T Q_2 \Delta w_{k+j} \tag{5. 25}$$

is given as follows:

$$\Delta w_k = -L_{\left(k,\tau_{CA}^{ap}(k)\right)} Z_k \tag{5. 26}$$

with:

$$L_{\left(k,\tau_{CA}^{ap}(k)\right)} = \left[Q_w + B^T \Phi_{k+1} B\right]^{-1} B^T \Phi_{k+1} A \tag{5. 27}$$

$$\Phi_k = Q_Z + L_{\left(k,\tau_{CA}^{ap}(k)\right)}^T Q_w L_{\left(k,\tau_{CA}^{ap}(k)\right)} + \left[A - BL_{\left(k,\tau_{CA}^{ap}(k)\right)}\right]^T \Phi_{k+1} \left[A - BL_{\left(k,\tau_{CA}^{ap}(k)\right)}\right] \tag{5. 28}$$

where $\Delta w_{k+j} = w_{k+j} - w_{k+j-1}$ is the control deviation and e_{k+j} is the predicted error between the output of the estimated model v_{k+j} and a reference signal r_{k+j}.

For the remaining of this chapter , $e(k+j)$ and $\Delta w\,(k+j)$ are written as e_{k+j} and w_{k+j} respectively. H_P is the prediction horizon, H_C is the control horizon and $H_P \geq H_C$. After the control horizon, the control signal will remain constant and $\Delta w\,(k+j)$ will be zero.

From equation (5. 20) the mean of $\hat{z}(k+1)$ is given as follows:

$$E\left\{z_{k+1|k} \middle| F_{k-1}\right\} = A_p z_k + B_p\left(\Delta w_k + w_{k-1}\right)$$

And its extended model will be:

$$\begin{pmatrix} z_{k+1} \\ w_k \end{pmatrix} = Z_{k+1} = AZ_k + B\Delta w_k$$

thus $A = \begin{pmatrix} A_p & B_p \\ 0 & I \end{pmatrix}$ and $B = \begin{pmatrix} B_p \\ I \end{pmatrix}$

Proof. The proof is given in appendix B. 3.

Lemma 5.1

The predictive objective function of equation (5. 25) can be expressed as a finite cost function as follows:

$$J_{N_p,N_w} = x^T_{k+N_w} S_x x_{k+N_w} + \sum_{j=1}^{N_w-1} \left\{ x^T_{k+j} Q_x x_{k+j} + \Delta w^T_{k+j} Q_u \Delta w_{k+j} \right\} + r^2 Q_1 \tag{5. 29}$$

Proof. Let's assume that the reference signal remains constant during the prediction horizon, thus the predicted error will be:

$$e_{k+j} = v_{k+j} - r$$

Since the stochastic part of the original NCS problem is being tackled in the filter stage e_{k+j} can be written as function of z_{k+j} as follows:

$$z_{k+j} = C\left(z_{k+j} - r_c\right)$$
$$r_c = Lr$$
$$CL = I$$
$$C = \begin{pmatrix} C_p & I \end{pmatrix}$$

thus the sum $\sum_{j=1}^{H_p} e^T_{k+j} Q_1 e_{k+j}$ will be:

$$\sum_{j=N_U}^{H_p} e^T_{k+j} Q_1 e_{k+j} = \sum_{j=N_U}^{N_p} \left\{ Z^T_{k+j} Q_z Z_{k+j} + r^2 Q_1 \right\}$$
$$Q_z = C^T Q_1 C$$

In the time interval $k \in [k+H_c, k+H_P]$, the predictor $\hat{x}_{k+H_P}$ is given as follows:

$$Z(k+H_P|k) = A^{H_P-H_C}\hat{z}(k+H_C) + \sum_{i=0}^{H_P-H_C-1} A^i B w(k+H_C+i) \tag{5. 30}$$

and writing the control signal as a sum of deviations:

$$w(k+H_C) = \sum_{i=1}^{H_C-1} \Delta w(k+i)$$

$\sum_{j=H_C}^{H_P} z_{k+j}^T Q_1 z_{k+j}$ will be:

$$\sum_{H_C}^{H_P} Z_{k+j}^T Q_1 Z_{k+j} = Z_{k+H_c}^T S_Z Z_{k+H_C} + \sum_{j=1}^{H_C-1} w_{k+j}^T S_w w_{k+j} + r^2 Q_1$$

with:

$$S_Z = \sum_{j=0}^{H_P-H_C} A^{jT} Q_x A^j$$

$$S_w = \sum_{i=0}^{H_P-H_C} B^T \left(\sum_{j=0}^{i} A^{jT}\right) Q_x \left(\sum_{j=0}^{i} A^j\right) B$$

and $Q_w = S_w + Q_2$ completes the proof.

Lemma 5.2

let equation (5. 13) be forwarded in the time interval $k \in [k+H_C, k+H_P]$, the optimal $N_p - steps$ ahead predictor $\hat{z}_{k+N_P}$ of $\hat{z}_k$ is the conditional mean $E\{z(k+H_p)|F_k\}$ [31] given as follows.

$$\hat{Z}(k+H_p|k) = E\{Z(k+H_p)|F_{k+H_p-1}\} = E\{e\{Z(k+H_p)|F_{k+H_p-1}\}|F_k\}$$

Applying the smoothing property of expectations, equation (5. 30) is determined. The resulting predictor is delay-dependent.

The controller is updated by using the bad-data detector. H_p is chosen to be equal to the inverse of the rise time, Q_1 and Q_2 are chosen as in [14] respectively. The resulting control law can be extended to compensate packet dropouts, but its

application requires event-driven actuators. When a packet dropout is detected, the controller updates the control law for $\tau_{CA}^{ap}(k) = \tau_{SC}(k) + P_d$, where P_d is the number of packets dropped. P_d is reset to zero when a delay is detected. The controller will not transmit until $P_d = 0$, thus the compensation can be too conservative.

Another approach to reduce packet dropouts can be based on including the traffic load in the controller design. An increase of traffic load will make the sampling time of the system variable. Applying the resulting control synthesis on Multirate Systems introduces the variability of the sampling time into the controller design and consequently can decrease the uncertainty of the data traffic.

5.5 Numerical Example

To illustrate the above design methodology, the following example is studied. The NCS is defined as a one-node system connected across an Ethernet communication network. $\tau_{SC}(k)$ and $\tau_{CA}(k)$ are the sensor-to-controller delay and controller-to-actuator delay respectively. Communication network parameters are taken from [53] to match with real values. Packet size is 8 bytes. The open loop transfer function of the system is as follows:

$$G(s) = \frac{1}{10s^3 + 5s^2 + 10.4s + 1}$$

Using a sampling time $h = 0.001$, its discrete form will be:

$$x(k+1) = \begin{pmatrix} 0.99 & -0.0104 & -0.001 \\ 0.01 & 0.99 & 0 \\ 0 & 0.01 & 1 \end{pmatrix} x(k) + \begin{bmatrix} 0.01 \\ 0 \\ 0 \end{bmatrix} u\left(k - \tau_{CA}(k)\right) + \xi(k)$$

$$y(k) = \begin{bmatrix} 0 & 0 & 0.1 \end{bmatrix} x(k) + \zeta(k)$$

where $\xi(k)$ and $\zeta(k)$ are process and measurements Gaussian noise respectively, with zero mean and covariance:

$$E\left\{\begin{bmatrix}\xi(k)\\ \zeta(k)\end{bmatrix}\begin{bmatrix}\xi(i) & \zeta(i)\end{bmatrix}\right\}=\begin{bmatrix}Q & 0\\ 0 & R\end{bmatrix}\Delta(k-i)$$

$$Q=0.0001$$

$$R=0.01$$

The above state-space model represents a stochastic system that contains controller-to-actuator delays only. The communication network is implemented in Simulink using TrueTime 1.5 simulator [73]. This simulator allows co-simulation of controller-task execution in real-time kernels, network transmission and continuous plant dynamics.

Network transmission delays are simulated as random positive values. Packets are dropped by software in the sensor. The proposed modified Kalman filter allows packet dropouts compensation as well as time delay compensation every sampling time. Packet dropouts in the sensor-to-controller interface are implemented by software by only transmitting a percentage of the available packets. This percentage is generated randomly every sampling time and compared with a predefined value limit. In the simulations this value was set from 10% to 90% of packet lost to test the filter robustness.

Calculation of the sensor-to-controller delay is as stated in the filter design procedure. On the other hand controller-to-actuator delay is based in the assumption presented for the controller design. This delay can freely be lesser, equal or bigger than the sensor-to-controller delay. Packet dropouts in the controller-to-actuator interface cannot be compensated with the proposed controller, thus they are not implemented.

The system is represented by a continuous state space model. Sensor and actuator are discretized using TrueTime kernels. These kernels contain A/D and D/A channels, set by software. Packets are transmitted through the network according to a predefined scheduling policy. In this case Fixed Priority scheduling was chosen. The simulation diagram is given in Fig. 5. 4.

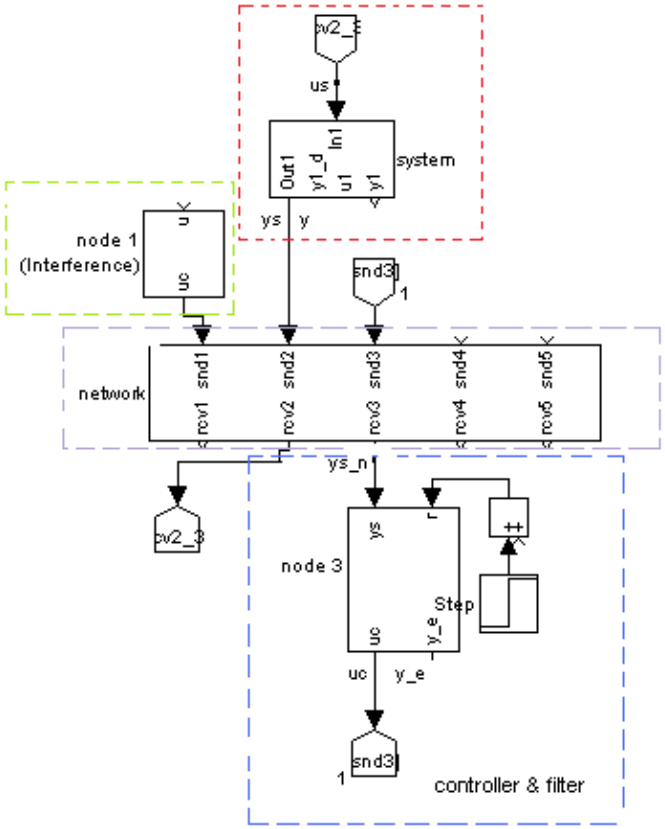


Fig. 5. 4 Simulation Diagram

Filter and MPC controller are implemented by software. Prediction horizon is chosen to be equal to the settling time of the system. The resulting algorithm is presented in Fig. 5. 5 in the following flow chart.

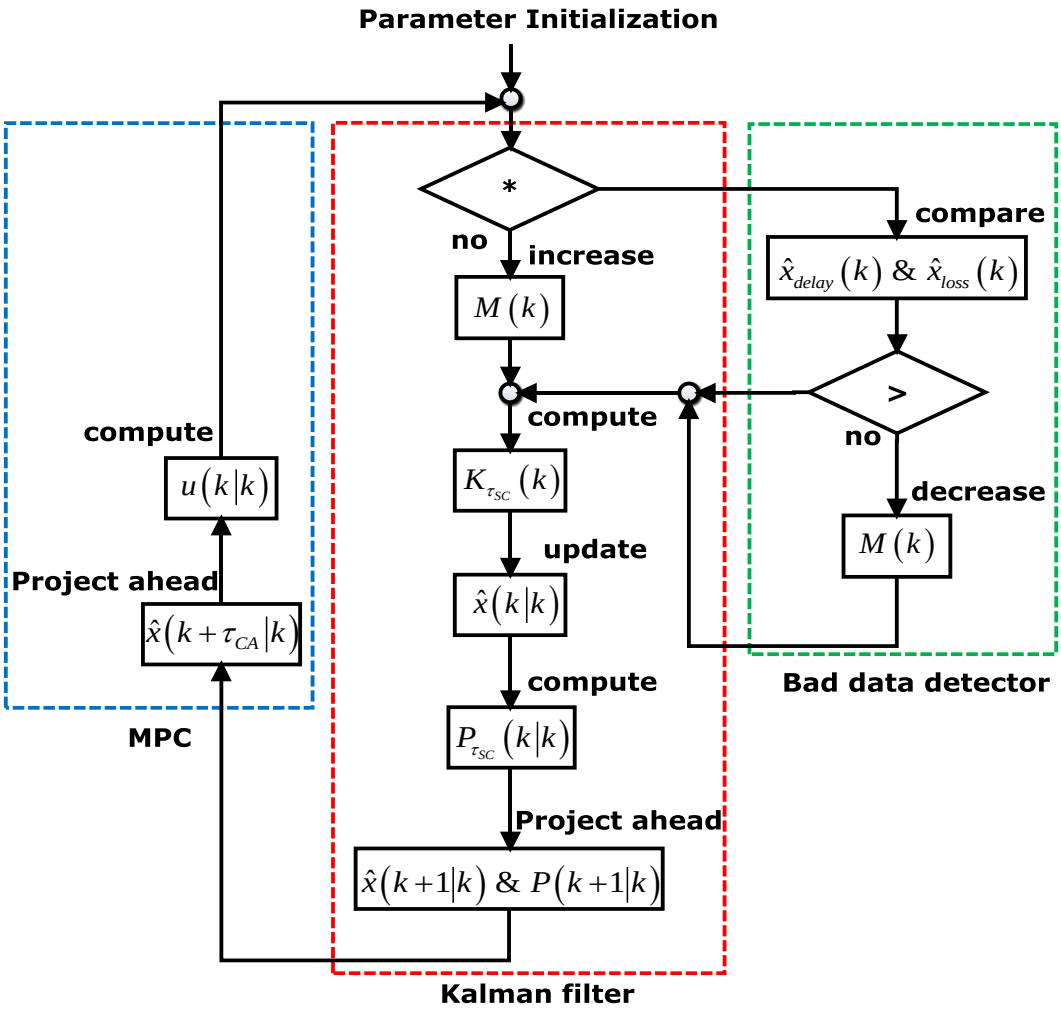


Fig. 5. 5 MPC Algorithm Flow Chart

Simulation results

LQR and modified Kalman filter for $\tau_{CA} = \tau_{SC}$:

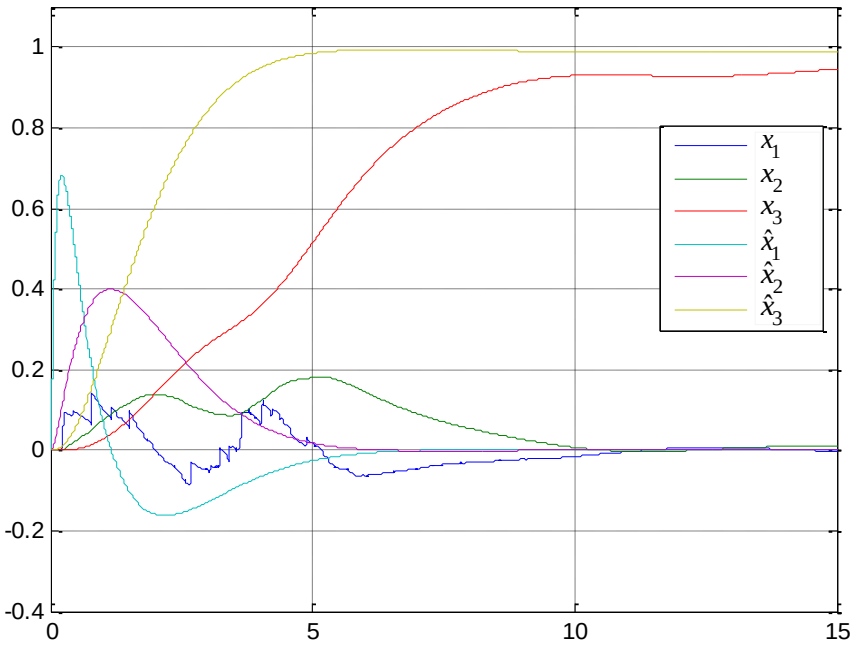


Fig. 5. 6 Comparison between Estimated states and real states of the proposed NCS system using a LQR controller with Artstein transformation for $\tau_{CA} = \tau_{SC}$. The real system states appears at $t = \tau_{CA}$

First the LQR controller is simulated. τ_{CA} is set to be equal to τ_{SC} and known. It is clear that for $\tau_{CA} < \tau_{SC}$ the controller will perform better, thus the τ_{CA} is changed to be $\tau_{CA} \geq \tau_{SC}$ only. Simulations for both cases are presented on Fig. 5. 6 and Fig. 5. 7. Simulation results show that by increasing τ_{CA} the non-observed states become oscillatory and the systems tends to instability.

LQR and modified Kalman filter for $\tau_{CA} > \tau_{SC}$:

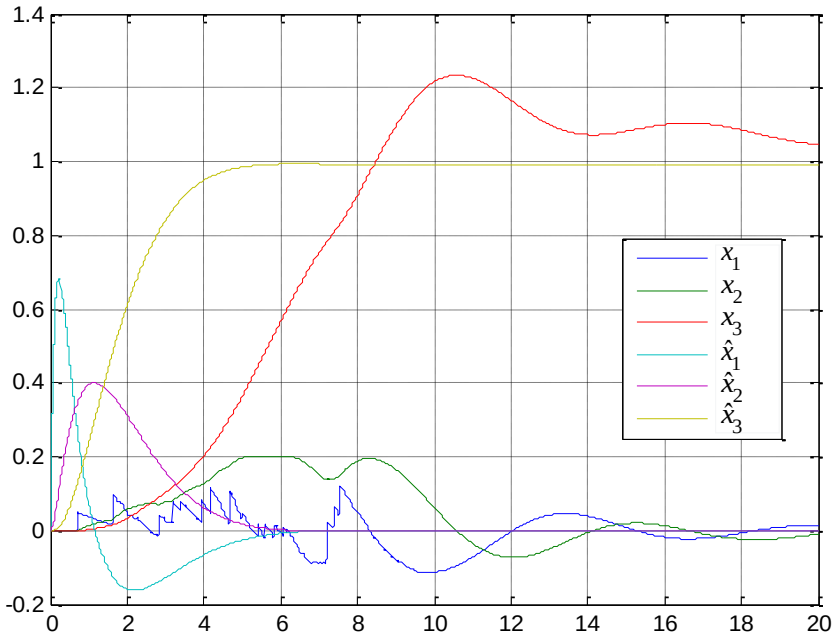


Fig. 5. 7 Comparison between Estimated states and real states of the proposed NCS system using a LQR controller with Artstein transformation for $\tau_{CA} > \tau_{SC}$. The real system states appears at $t = \tau_{CA}$

MPC controller is implemented under the same simulation conditions. The MPC implementation gave better results that the LQR controller as expected. Due to its variable control horizon H_c, the controller can be tested for any value of τ_{CA}^{ap}. Simulations for $\tau_{CA} < \tau_{SC}$ and $\tau_{CA} \geq \tau_{SC}$ are presented in Fig. 5. 8 to Fig. 5. 13.

MPC and extended Kalman filter simulation for $\tau_{CA} < \tau_{SC}$:

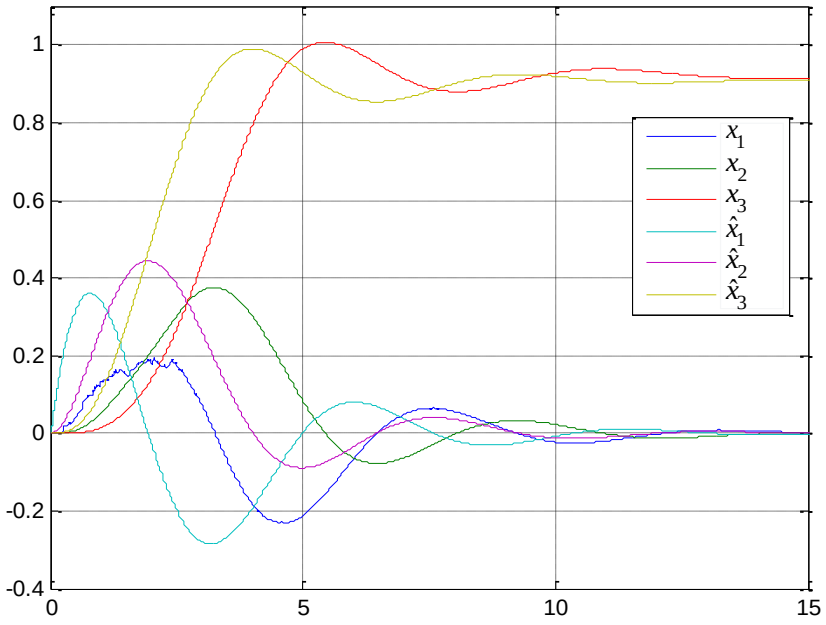


Fig. 5. 8 Comparison between Estimated states and real states of the proposed NCS system using a MPC controller with variable control horizon. The real system states appears at $t = \tau_{CA}$ and $\tau_{CA} < \tau_{SC}$.

Fig. 5.8 and Fig. 5.9 represent the step response of the system when $\tau_{CA} < \tau_{SC}$. Fig. 5.8 represents a comparison between real states and estimated states. Real states appear after τ_{CA} . As expected, real states al almost similar than the estimated states when $\tau_{CA} < \tau_{SC}$. The effects of forwarding the control signal to compensate delays can be seen in a slightly better attenuation of x_1 and x_2 .

Fig. 5.9 represents the real and estimated output. The real containts noise and has an initial negative peak due to the delays. It can be seen that the peak is inmitially compensated by the controller and thus the real output can track the reference signal.

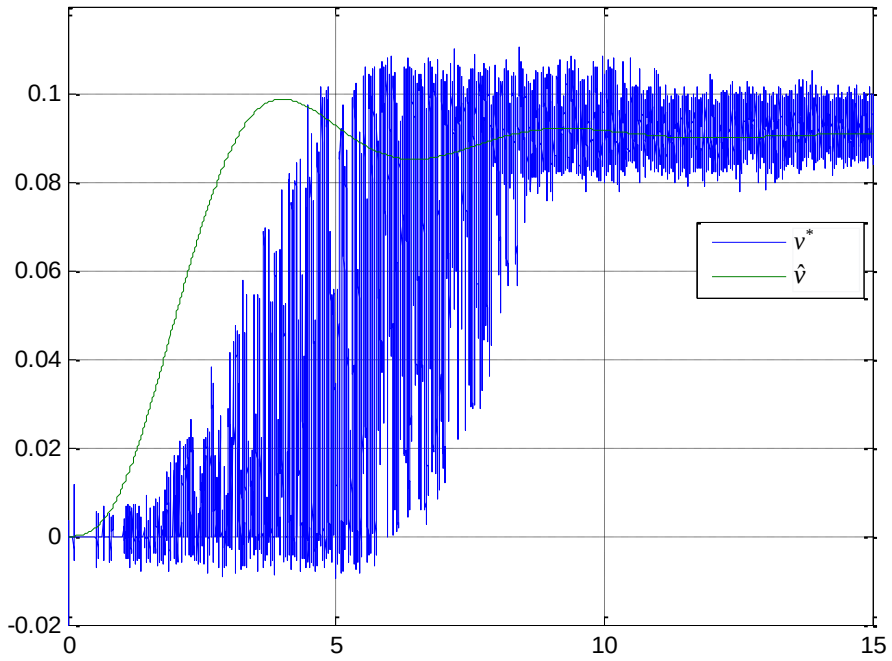


Fig. 5. 9 Comparison between Estimated states and real states of the proposed NCS system using a MPC controller with variable control horizon. The real system states appears at $t = \tau_{CA}$ and contains noise.

MPC and extended Kalman filter simulation for $\tau_{CA} = \tau_{SC}$:

Fig. 5.10 and 5.11 represent the system step response when $\tau_{CA} = \tau_{SC}$. It can be seen in Fig. 5.10 that the real states are more attenuated than in the previous case because Of the filter delay compensation.

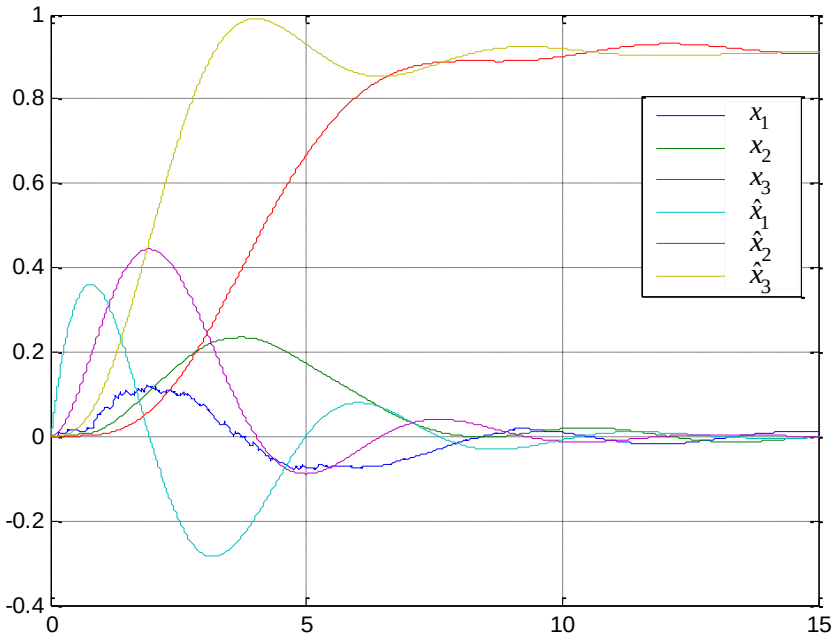


Fig. 5. 10 Comparison between Estimated states and real states of the proposed NCS system using a MPC controller with variable control horizon. The real system states appears at $t = \tau_{CA}$ and $\tau_{CA} = \tau_{SC}$.

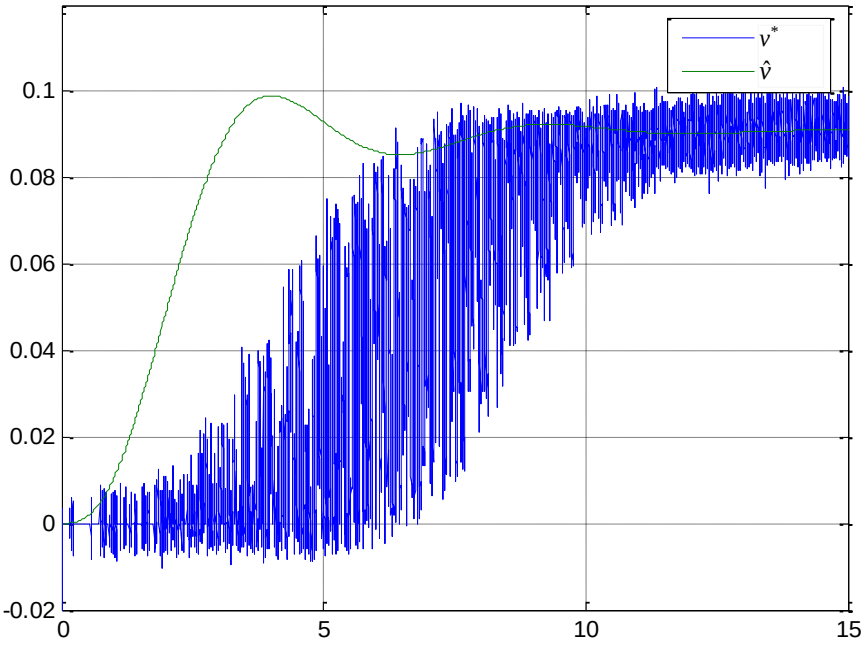


Fig. 5. 11 Comparison between Estimated states and real states of the proposed NCS system using a MPC controller with variable control horizon. The real system states appears at $t = \tau_{CA}$ and contains noise.

MPC and extended Kalman filter simulation for $\tau_{CA} > \tau_{SC}$:

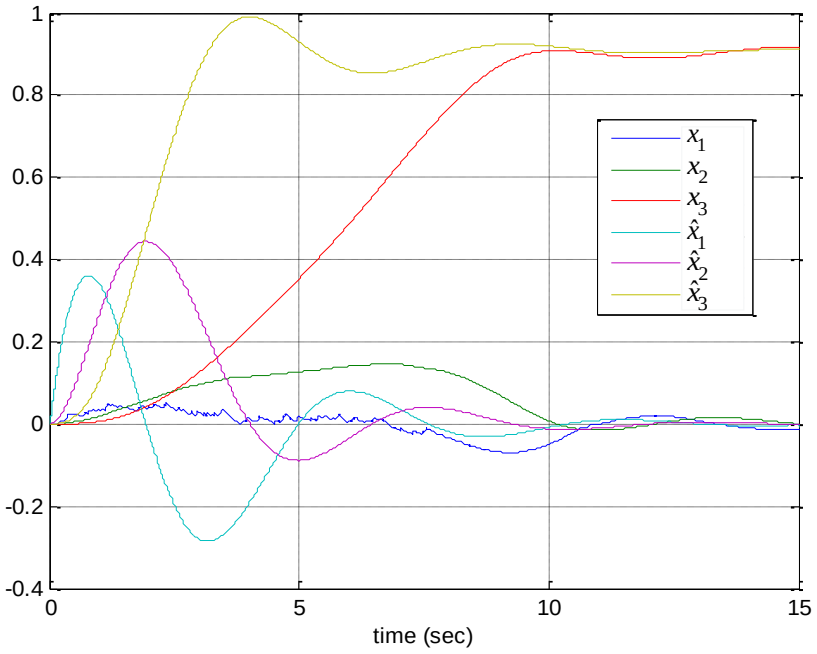


Fig. 5. 12 Comparison between Estimated states and real states of the proposed NCS system using a MPC controller with variable control horizon. The real system states appears at $t = \tau_{CA}$ and $\tau_{CA} > \tau_{SC}$.

Fig. 5.12 and Fig. 5.13 represent the states and outputs when $\tau_{CA} > \tau_{SC}$. Higher attenuation was expected due to the filter.

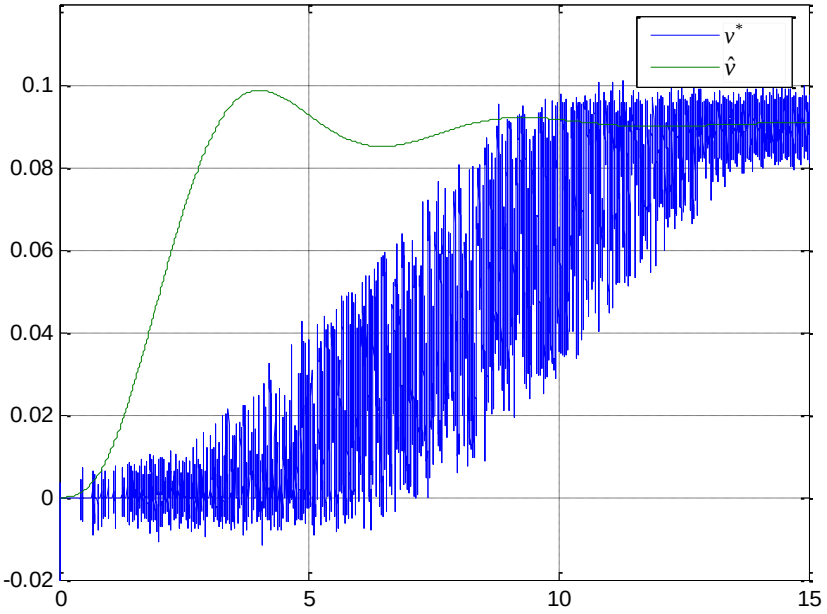


Fig. 5. 13 Comparison between Estimated output and real output of the proposed NCS system using a LQR controller with variable control horizon. The real system output appears at $t = \tau_{CA}$ and contains noise

.

5.6 Conclusions

This chapter presents a simple, applicable and reliable filter-controller formulation. Network delays and packet dropouts in the sensor-to-controller interface have been successfully compensated by modifying a standard Kalman filter as well as by adding a bad data detector to distinguish between trigger events. A trigger event is any constraint represented by the lack of arrivals on the filter. Bad data detection is possible due to the time-driven characteristic of the filter.

Variable controller-to-actuator delay compensation has been achieved by assuming that the appearance of constraints in one of the channels of the network will affect the dynamics of the other channels.

Combining the modified Kalman filter with a MPC controller, a controller design was proposed where its implementation is straight forward. However controller-to-actuator packet dropouts have not been compensated. A plausible solution is given in the next chapter.

Chapter 6

Distributed Multirate control design for Multivariable NCS

This chapter presents a control solution for multi-loop NCS. The applied control synthesis is a Distributed Multirate Model Predictive Control (DMMPC) for network delays and packet dropouts minimization. The structure of this chapter is as follows: An introduction is given in section 6.1. Section 6.2 focuses on the disadvantages of centralized control from an estimator perspective. Section 6.3 offers a solution to the trade-off between scheduling policies and control design complexity by formulating the control problem within a distributed scheme. Global optimality is achieved as Nash optimal. Section 6.4 concerns with the addition of a packet dropout mechanism which leads to a suboptimal stochastic controller design. Sub-optimality constraint is removed in section 6.5 by formulating the controller design problem as a multirate model predictive control problem. A case study is presented in section 6.6 to demonstrate the performance of the proposed technique.

6.1 Introduction

Connectivity of the control components in a NCS is ad-hoc, only bounded by the Information structure availability. This Information structure, that is connectivity and capacity of the communication network [29], offers many types of coordination on the system scheme; and leading to real-time implementation of distributed control systems. Distributed Control Systems are control systems composed by several subsystems that communicate, coordinate and negotiate with each other to accomplish a centralized control objective [29]. This interactions lead to dynamic games where each subsystem deploys, through the network, information about their decision variables. In this way global optimal solutions are achieved, but at a cost of increasing network traffic load.

Computational issues can be addressed formulating the NCS problem as a Distributed control problem. The Multivariable system is decomposed into m agents or entities that can sense the states of the system and decide upon the values of its control variables [29]. The process is observed by single channel modified Kalman filters as in [80] and the control objective is formulated as multi-objective optimization problem.

Individual decision variables can be forwarded n-steps ahead for network delay compensation, leading to Distributed Model Predictive Control approach. Each agent will solve a local and simpler MPC optimization problem, so that the distributed system reaches optimality in the sense of Nash optimality at every sampling time [29].

Inherent Network Constraints present on single loops such as: network delays, packet dropouts and packet missequencing are more complex for multi-loop NCS. Communication channels cannot be compensated independently and more importantly, it is unfeasible to make any correlation assumptions among common communication links such as network constraints propagation [80]. Lost packets in the controller-to-actuator interface can be taken into account by adding packet delivery indicators [7]. Their introduction makes the controller design stochastic and

furthermore since they are not observable, the separation principle for controller/estimator design is not applicable. Approximated suboptimal solutions can be implemented as in [7]. Due to information structure, a packet-dropping rate may be known at the system estimation stage if the control signals can be deployed several times during the sensor's sampling time. This approach allows the application of the principle of optimality but leads to a multirate control problem.

In practical applications, sharing a common communication network causes control components to be inherently asynchronous or extremely difficult to synchronize [54]. The implementation of monolithic structures is very common [54] but exhibits heavy computational burden in the fusion processes and controllers. Predictive control over centralized structures can also be susceptible to poor fault tolerance [29]. Furthermore, Predictive Control design for NCS is network-dependent.

6.2 Multi-loop NCS with Centralized Control Scheme

Output observations on multi-loop NCS are asynchronized and unreliable. Control components on multi-loop NCS are distributed across a common communication network as presented in Chapter 5 in Fig. 2.5. Fig. 6. 1 represents a more detailed multi-loop centralized NCS, where each control component will inherently be affected by its network constraints link. Assuming that the system can be represented by an LTI stochastic multivariable state space model $\sum_1(t)$ as follows:

$$\dot{x}(t) = a_p x(t) + b_p u(t) + \xi(t) \tag{6. 1}$$

$$y(t) = c_p x(t) + \zeta(t) \tag{6. 2}$$

where $y(t) \in \mathbf{R}^{N_y}$, $u(t) \in \mathbf{R}^{N_u}$, $x(t) \in \mathbf{R}^{N_x}$ and all the system matrices a_p, b_p, c_p are of compatible dimensions. $\xi(t) \in \mathbf{R}^{N_x}$ is the process noise, $\zeta(t) \in \mathbf{R}^{N_y}$ is the measurements noise where:

$$E\left\{\begin{bmatrix}\xi(j)\\ \zeta(j)\end{bmatrix}\begin{bmatrix}\xi^T(i) & \zeta^T(i)\end{bmatrix}\right\}=\begin{bmatrix}q & 0\\ 0 & r\end{bmatrix}\delta_{ji} \tag{6. 3}$$

represents the covariance matrix of the noise with δ_{ji} being the Dirac delta. Assuming also that system outputs are sampled at the same sampling time $t=kh^i=kh_0$, $i=1,...,N_y$ and h_0 is constant, network traffic makes the information arrivals asynchronized and network constraints reduce the observability of the system outputs. Sampling equation (6. 1) at h_0, the expected discrete state space model will be:

$$x(k+1)=A_p x(k)+B_p u(k)+\xi(k) \tag{6. 4}$$

$$y(k)=\mathrm{H}_p x(k)+\zeta(k) \tag{6. 5}$$

where $y(k)\in\mathbf{R}^{N_y}$, $u(k)\in\mathbf{R}^{N_u}$, $x(k)\in\mathbf{R}^{N_x}$ and all the system matrices A_p, B_p, H_p are of compatible dimensions. $\xi(k)\in\mathbf{R}^{N_x}$ is the process noise, $\zeta(k)\in\mathbf{R}^{N_y}$ is the measurements noise where:

$$E\left\{\begin{bmatrix}\xi(k)\\ \zeta(k)\end{bmatrix}\begin{bmatrix}\xi^T(i) & \zeta^T(i)\end{bmatrix}\right\}=\begin{bmatrix}Q & 0\\ 0 & R\end{bmatrix}\Delta(k-i) \tag{6. 6}$$

represents the covariance matrix of the noise with $\Delta(k-i)$ being the Kronecker delta. The state initial condition $x(0)$ with mean $\mu_x(0)$ and covariance matrix $\Pi(0)$ is:

$$\begin{aligned}E\{x(0)\}&=\mu_x(0)\\ \Pi(0)&=E\left\{\left[x(0)-\mu_x(0)\right]\left[x(0)-\mu_x(0)\right]^T\right\}\end{aligned} \tag{6. 7}$$

By adding the network induced delays to the individual outputs, the resulting networked outputs contain variable sampling intervals as follows:

$$\begin{aligned}v_i(h_i k)&=y_i\left(h_i k-\tau_{SC}^i(h_i k)\right)\\ i&=1,...,N_y\end{aligned} \tag{6. 8}$$

To facilitate the fusion process for the filter $\sum_o(k)$ as in Chapter 5, it is necessary to implement it as time-driven. Communication links are assumed to be not correlated so the innovation signal can be decomposed and equation (6. 8) is valid. For time-driven components, the trigger event that represents either delays or packet dropouts is the absence of information arrivals and more importantly the synchronization dependence is being removed. The resulting NCS is present on Fig. 6. 1.

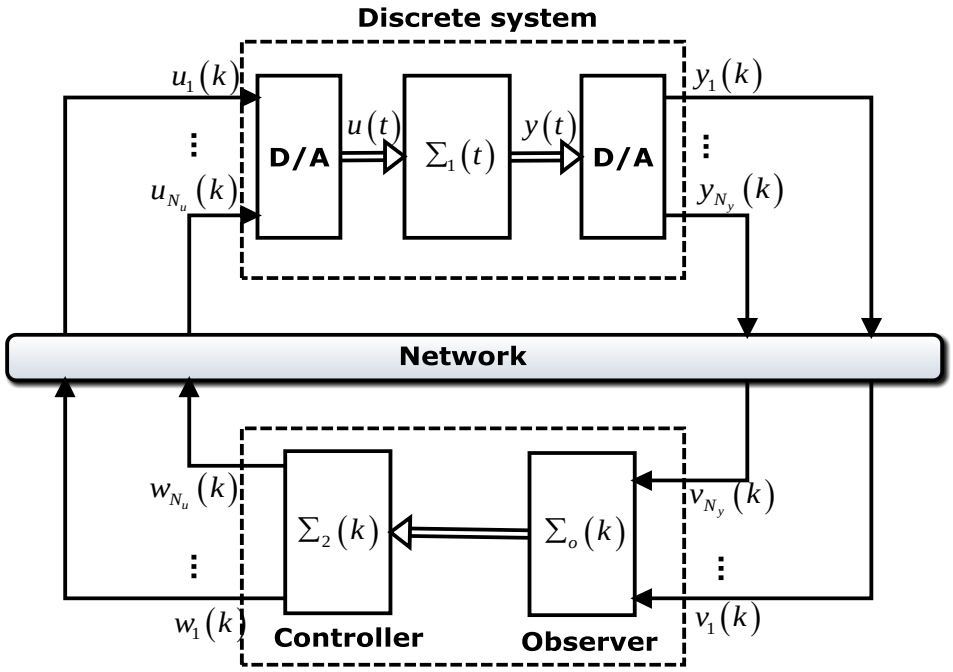


Fig. 6. 1 NCS Centralized Control Scheme

Larsen's modified Kalman filter presented in appendix B. 1 can be applied to the multi-loop NCS directly. Nonetheless, there is a restriction to the application of the filter, which is that due to time driven components, the correction factor $\mathrm{M}(k)$ cannot be updated until all the observations $\{V^{ex}(k)\}$ with $V^{ex}(k)=\left[v_1^{ex}(k)...v_{N_y}^{ex}(k)\right]$, are available in the filter.

The innovation process as presented in Chapter 5, equation (5. 13) can be defined as follows:

$$e_i^{ex}(k) \equiv e_i\left(k-\tau_{SC}^i(k)\right)$$

$$\begin{aligned} E^{ex}(k) &= \left[e_1^{ex}(k), ..., e_{N_y}^{ex}(k) \right]^T \\ &= V^*(k) - H_p \hat{x}\left(k - \tau_{SC}^{\max}(k)\right) \end{aligned} \tag{6. 9}$$

$V^*(k)$ is the observation process in the filter and $\tau_{SC}^{\max}(k)$ is given by:

$$\tau_{SC}^{\max}(k) = \max\left\{ \tau_{SC}^1(k), ..., \tau_{SC}^{N_y}(k) \right\} \tag{6. 10}$$

Three main aspects can be observed from the above equations: firstly the filter cannot tackle single delays, the innovation process for the multi-loop NCS can be computed separately; computational burden is as many times higher as the number of outputs; finally the controller to be implemented will be monolithic and furthermore, information about delays from single compensated loops cannot be forwarded ahead to implement model predictive control.

Traditionally, centralized implementations arise from the need to operate the system in an optimal fashion [29], however controller design can be formulated sub-optimally by reducing the number of decision variables. To reduce even more computational burden, sensors scheduling policies can be implemented, which extends the filter design problem to a sensors network design problem [67].

It is also possible to formulate the controller optimization problem as a multiobjective formulation. The main control problem is transformed to a dynamic game of local optimization problems and optimality is reached as Nash optimal [29]. This approach leads to distributed NCS formulation as described in the following sections.

6.3 NCS Distributed Control Scheme

Multivariable NCS can be decomposed into $m \leq N_y$ subsystems or single-loop agents interconnected to each other as shown in Fig. 6. 2. The continuous distributed control system can be defined as follows:

$$x_i(k+1) = A_{p_{ii}} x_i(k) + B_{p_i} u_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^{m} B_{p_j} u_j(k) + \xi_i(k) \tag{6. 11}$$

$$y_i(k) = C_{p_{ii}} x_i(k) + \zeta_i(k) \tag{6. 12}$$

$$x_i(0) = x_{i0}$$

where $y_i(t) \in \mathbf{R}^1$, $u_i(t) \in \mathbf{R}^{N_{u_i}}$, $x_i(t) \in \mathbf{R}^{N_{x_i}}$. $A_{p_{ii}}$, B_{p_i}, B_{p_j} and $C_{p_{ii}}$ are matrices of compatible dimensions and $i = 1, 2, ..., m$.

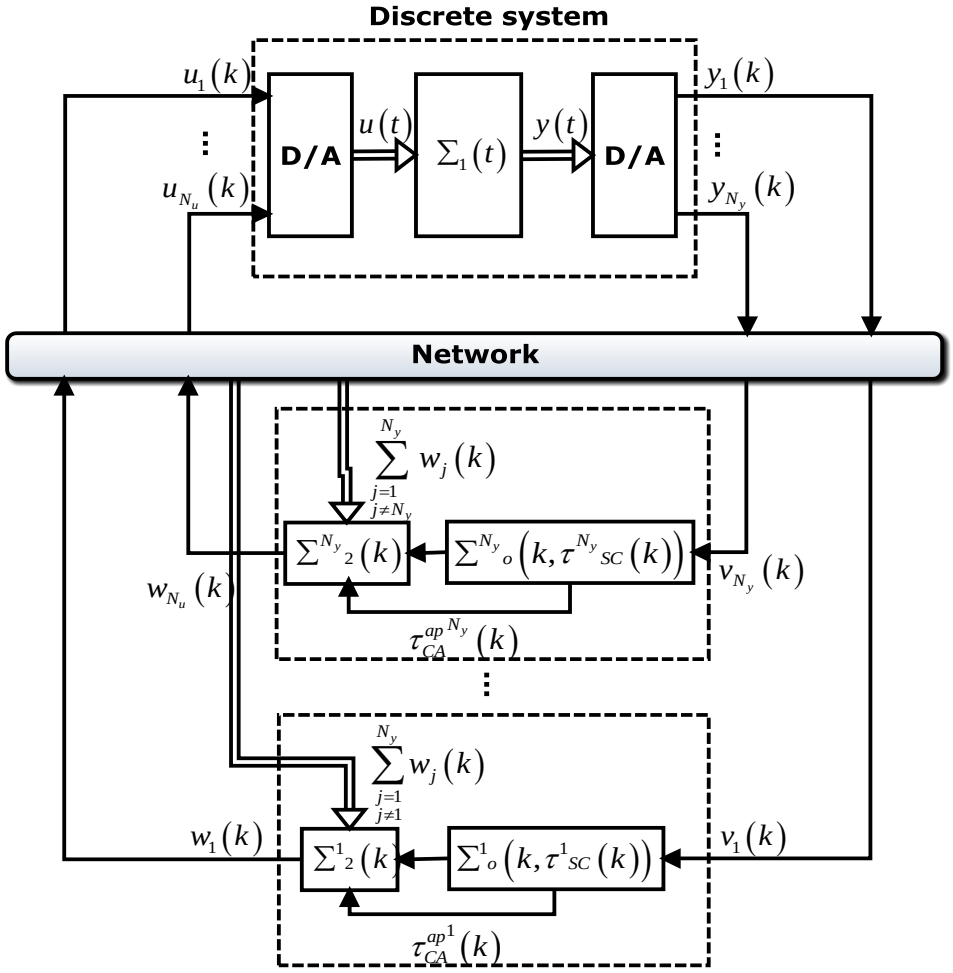


Fig. 6. 2 NCS Distributed Control Scheme for N_y subsystems

If the capacity of communication and connectivity of the network is not a constraint, single loop agents can be estimated as follows:

Corolary 6.1

Let $\hat{x}_i(k+1)$, $i=1,...,m$ represent the conditional mean of $x_i(k+1)$ given observations $\{v_i^{ex}(k)\}$ and the remaining decision variables $\sum_{\substack{j=1 \\ j\neq i}}^{m} B_{Pj} w_j(k)$ up to and including k. Then $\hat{x}_i(k+1)$ is modified by a correction factor $M_i(k)$ and satisfies the following recursion:

$$\hat{x}_i(k+1|k) = A_{p_{ii}}\hat{x}_i(k) + B_{p_i} w_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} B_{p_j} w_j(k) + K_{ii}\left(k, \tau_{SCi}(k)\right)\left[v_i^{ex}(k) - C_{p_{ii}}\hat{x}_i(k)\right]$$

(6. 13)

$$\hat{x}_i(0) = \mu_{x_i}(0)$$

where the Kalman gain and error covariance are functions of the correction factor $M_i(k)$.

$$K_{ii}\left(k, \tau_{SCi}(k)\right) = A_{p_{ii}} M_i(k) C_{p_{ii}}{}^T \left[C_{p_{ii}} P_{ii}\left(k - \tau_{SCi}(k)\right) C_{p_{ii}}{}^T + R_i\right]^{-1} \quad (6.\ 14)$$

$$P_{ii}\left(k+1|k, \tau_{SCi}(k)\right) = -K_{ii}\left(k, \tau_{SCi}(k)\right)\left[C_{p_{ii}} P_{ii}\left(k - \tau_{SCi}(k)\right) C_{p_{ii}}{}^T + R_i\right] K_{ii}\left(k, \tau_{SCi}(k)\right)^T + Q_{ii} + A_{p_{ii}} P_{ii}(k) A_{p_{ii}}{}^T$$

(6. 15)

$$P_{ii}(0) = \Pi_{ii}(0)$$

where $v_i(k) \in \mathbf{R}^1$, $w_i(k) \in \mathbf{R}^{N_{u_i}}$, $x_i(k) \in \mathbf{R}^{N_{x_i}}$ and:

$$v_i(k+1) = C_{p_{ii}}\hat{x}_i(k+1|k) \quad (6.\ 16)$$

The proof of corollary 6.1 can be derived from appendix B. 1. The estimated states of each agent can now be forwarded ahead to implement local model predictive controllers as described in chapter 5, but the cost function should be modified to accommodate the information about the decision variables of the other agents.

6.3.1 Distributed Model Predictive Control

Distributed Model Predictive Control is aimed to solve a centralized MPC optimization problem from a multiobjective formulation. The original cost function is decomposed into N_y agents that represent single components of the multiobjective formulation.

The resulting cost function is as follows:

$$\min_{w_i(k)} J_i\left(\hat{x}(k), W(k), \Lambda\right) \tag{6. 17}$$

s.t.

$$\hat{x}_i\left(k+H_c\right) = \mathrm{A}_p^{H_c-1}{}_{ii}\hat{x}_i\left(k\middle|k\right) + \sum_{l=0}^{H_C-1} A_{p_{ii}}^l \left(B_{p_i} w(k+l) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} B_{p_j} w_j\left(k+l\right) \right)$$

where:

$$J_i\left(\hat{x}(k), W(k), \Lambda\right) = \sum_{l=1}^{m} \alpha_l J_l\left(\hat{x}_l(k), w_l(k), w_{j\neq l}(k)\right)$$

$$\alpha_l \geq 0$$

$$\sum_{l=1}^{m} \alpha_l = 1$$

Λ is a vector of the weighting factors α_l that are included to define the influence of each local performance index J_l and $\hat{x}_i\left(k+H_c\right)$ is the i-th H_c-step ahead predictor. If $\alpha_{\substack{l \\ l=i}} = 1$ and $\alpha_{\substack{l \\ l\neq i}} = 0$, the control problem represents a centralized control problem. Therefore the condition $\alpha_l > 0$ assures that the control problem is an organizational problem of $m \leq N_y$ agents and each individual optimal decision depends on the information of the decisions of the others [29].

Global optimality will depend on the interconnection between the sub-problems which can be solved by means of Nash optimality concept [29]; and is defined as follows:

Definition 6.1: a group of control decisions $W^*(k)=\left[w_1^*(k)...w_m^*(k)\right]$ is said to be Nash optimal if:

$$J_i\left(\hat{x}_i(k),w_i^q(k),w_{j\neq i}^q(k)\right)\leq J_i\left(\hat{x}_i(k),w_i^{q-1}(k),w_{j\neq i}^{q-1}(k)\right) \quad (6.\ 18)$$

Definition 6.1 leads to a non-cooperative dynamic game. Every sub-problem is optimized recursively based of known information about the optimal remaining decision variables from equation (6. 17). Recursion is kept until all individual decisions reach an equilibrium point (attractor) [29].

An attractor has to rise every sampling time for the optimization problem to be optimal. Information structure of the network is crucial for each agent to solve its optimization problem. Additionally, deploying inter-sampling decision variables offers a monitoring mechanism on the controller-to-actuator links. It can be seen that, local optimal inter-sampling decision variables cannot be transmitted to the plant because they have not reached global optimality. To make this transmission possible, the original local optimization problems can be reformulated using a known inter-sampling known rate. This approach leads to a multirate control design problem.

6.4 Controller-to-actuator Packet Dropouts Minimization

Packet delivery indicators can be implemented to model packet dropouts on the controller to actuator interface. A packet delivery indicator can be defined as follows:

$$\phi_1(k-1)=\begin{cases}1 & \text{if packet was recieved}\\ 0 & \text{if packet was lost}\end{cases} \quad (6.\ 19)$$

with $\Phi(k-1)=\left[\phi_1(k-1),...,\phi_{N_y}(k-1)\right]$ being the vector of packet indicators $\phi_i(k-1)$ at the i-th subsystem.

The resulting estimated model would require not only observations about the process $\left\{v_i^{ex}(k)\right\}$ and the decision variables $\left\{W^*(k)\right\}$, but also information about these indicators $\left\{\Phi(k-1)\right\}$. Since decision variables are not acknowledged to the controllers, these indicators are not necessarily observable at time k [7]. Additionally, this lack of information makes the separation principle inapplicable for separate controller and estimator design. As stated by Babak [7], approximated solutions can be obtained by assuming that separation principle is possible. This approach leads to a stochastic control design problem.

A finite horizon approximated control policy for each agent can be expected to be as follows:

$$w_k^i\left(E\left\{x_i\middle|I_k\right\}\right)=L_i\left(k,\gamma_i(k-1)\right)E\left\{x_i\middle|I_k\right\} \tag{6. 20}$$

with $i=1,...,N_y$ and $I_k=\left\{v_i^{ex}(k),W^*(k),\Phi(k-1)\right\}$. $L_i\left(k,\gamma_i(k-1)\right)$ is the controller gain that can easily be the solution of a Receding Horizon Dynamic Game Control problem, and $\gamma_i(k-1)$ is the dropping packet rate such that:

$$P(\phi_{k-1}=1)=1-\gamma_i(k-1)$$

$$P(\phi_{k-1}=0)=\gamma_i(k-1)$$

Since corollary 6.1 shows a recursion algorithm that compensates packet dropouts and delays in the sensor-to-actuator interface, packet delivery indicators are only necessary in the control inputs. For Kalman filter with exogenous inputs, if the control inputs are $I_k-measurables$, so that they depend on the observations of the process; and also by using the linear output feedback controller from equation (6. 20), then the process is still Gaussian. More importantly, the filter covariance is not affected by the input packet delivery rates $\gamma(k-1)$. The resulting estimated states will be:

Theorem 6.1

Let $\hat{x}_i(k+1)$, $i=1,...,m$ represent the conditional mean of $x_i(k+1)$ given information $\left\{v_i^{ex}(k), w_i(k), \Phi(k-1)\right\}$ and the remaining decision variables $\sum_{\substack{j=1 \\ j\neq i}}^{m} B_{Pj} w_j(k)$ up to and including k. Then $\hat{x}_i(k+1)$ is modified by a correction factor $M_i(k)$ and satisfies the following recursion:

$$\begin{aligned}\hat{x}_i(k+1|k) &= A_{p_{ii}}\hat{x}_i(k) + \gamma_i(k-1)B_{p_i}w_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y}\gamma_j(k-1)B_{p_j}w_j(k) \\ &+K_{ii}\left(k,\tau_{SCi}(k)\right)\left[v_i^{ex}(k) - C_{p_{ii}}\hat{x}_i(k)\right]\end{aligned} \tag{6. 21}$$

with Kalman gain and covariance error determined as in equations (6. 14) and (6. 15) respectively.

Proof: let's suppose that the solution of $x_i\left(k-\tau_{SC}^i(k)\right)$ can be expressed as follows:

$$x_i\left(k-\tau^i(k)\right) = x_i^w(k) + x_i^\xi\left(k-\tau_{SC}^i(k)\right) \tag{6. 22}$$

where $x_i^w(k)$ and $x_i^\xi\left(k-\tau_{SC}^i(k)\right)$ are control and process noise delayed inputs states respectively. So far $\tau_{CA}^i(k)$ is still unknown at the filter and is not included in equation (6. 22). The observation process from equation (6. 12) can be reformulated using equation (6. 22) for every agent as follows:

$$v_i(k) = C_{p_{ii}}x_i^w(k) + C_{p_{ii}}x_i^\xi\left(k-\tau_{SC}^i(k)\right) + \zeta_i(k)$$

Rearranging terms such that:

$$v_i(k) - C_{p_{ii}}x_i^w(k) = C_{p_{ii}}x_i^\xi\left(k-\tau_{SC}^i(k)\right) + \zeta_i(k)$$

the observation process due to x_i^ξ can be defined as follows:

$$v_i^\xi(k) \equiv v_i(k) - C_{p_{ii}}x_i^w(k)$$

$$v_i^{\xi}(k) = C_{p_{ii}} x_i^{\xi}\left(k - \tau_{SC}^{i}(k)\right) + \zeta_i(k) \tag{6. 23}$$

Applying standard Kalman filter and defining an extrapolated innovation process $e_i^{ex}(k)$ for equation (6. 22) as follows:

$$\begin{aligned} e_i^{ex\xi}(k) &\equiv e_i^{\xi}\left(k - \tau_{SC}^{i}(k)\right) \\ &= v_i^{\xi^*}(k) - C_{p_{ii}} \hat{x}_{\xi}^{\ *}\left(k - \tau_{SC}^{i}(k)\right) \end{aligned}$$

Using the definition of $v_i^{\xi}(k)$ for extrapolated measurements, then the innovation process will be:

$$e_i^{ex\xi}(k) = v_i^{*}(k) - C_{p_{ii}} x_i^{w}(k) - C_{p_{ii}} \hat{x}_{\xi}^{\ *}\left(k - \tau_{SC}^{i}(k)\right)$$

$$e_i^{ex\xi}(k) = v_i^{*}(k) - C_{p_{ii}} \hat{x}^{*}\left(k - \tau_{SC}^{i}(k)\right)$$

$$e_i^{ex\xi}(k) = e_i^{ex}(k)$$

This result shows that theorem 6.1 can be applied for distributed systems with input packet arrival rates and demonstrates equation (6. 21).

6.4.1 Elimination of packet dropouts constraint

So far it is shown that: information about packet arrival indicators $\{\Phi(k-1)\}$ are not necessarily observable at time k; the filter recursion is independent of the inputs and consequently of a packet dropping rate γ; and more importantly, the separation principle cannot be directly applied.

The compensation of packet dropouts on the controller-to-actuator interface, as stated by Babak [7] becomes a constraint for controller design. Nonetheless, the distributed control approach for control design requires the controller-to-actuator interface to be

able to transmit the control objectives to the other controllers several times within a sensor-sampling time h_0.

This decision variables deployment can be used as inter-sampling packet arrival indicators $\{\Phi(kh-h)\}$, with $h_{\text{int}} = \frac{h_0}{N}$, so that it is possible to define a measurable packet dropping index β_i at $t = kh_0$, $k = 1,2,...$; and reduce the effects of packet dropouts on the actuator by transmitting the locally optimized decision variables to the plant.

Updating the decision variables $w_i(k|k)$ at different sampling intervals of the system outputs $y_i(k)$, inevitably leads to multirate sampled-data systems. The input multiplicity N, so far depends on the necessary number of iterations for the dynamic game to reach the attractor. N can also be assumed known, so that the control inputs can be sampled regularly. The resulting control law is assumed to be piecewise constant in the sampling interval $t \in [jh_{\text{int}}, (j+1)h_{\text{int}})$ and given by:

$$w_i(kh_0 + qh_{\text{int}}) = L_i\left(q, \gamma_i\left[(k-1)h_0\right]\right) E\left\{x_i \middle| I_{kh_0}\right\} \tag{6. 24}$$

with $q = 0,...,N-1$. This type of controller is called Multirate constant output feedback controller, and its gain $L_i(.)$ can be formulated as the solution of a continuous MPC problem sampled at time h_{int} [17].

6.5 Multirate MPC Design for NCS

Implementation of distributed schemes on multivariable NCS, for packet dropouts minimization, inevitably introduces inter-sampling behaviour on the loops. This inter-sampling behaviour of the process can be addressed by using sampled-data methods. Sampled-data models present a more realistic model of a control system

which contains a continuous plant controlled by a digital control or controllers. If data in such systems are sampled in more than one rate, such systems become multirate [17].

Let's define a continuous distributed control system as follows:

$$\dot{x}_i(t) = A_{ii}x_i(t) + B_i u_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} B_j u_j(t) \tag{6. 25}$$

$$y_i(t) = C_{ii}x_i(t) \tag{6. 26}$$

$$x_i(0) = x_{i0}$$

where $y_i(t) \in \mathbf{R}^1$, $u_i(t) \in \mathbf{R}^{N_{u_i}}$, $x_i(t) \in \mathbf{R}^{N_{x_i}}$. A_{ii}, B_i, B_j and C_{ii} are matrices of compatible dimensions.

To formulate a Multirate MPC for the Distributed NCS sampled-data model, let's assume that state measurements are available and sampled at $t_k = kh_0$. Let's also define a control horizon $H_C^i h_0$, such that $u_i(kh_0 + t|kh_0) = 0$, $\forall t > H_C^i h_0$. Thus the single rate MPC problem [17] is to design the control input $u_i(kh_0 + t|kh_0)$, $0 \le t < H_C^i h_0$, which minimize the following objective function:

$$\min_{u_i(kh_0+t|kh_0),0\le t<H_c^i h_0} J_i = \alpha_i \int_{kh}^{\infty} \left[x_i^T(t) Q_{ii} x_i(t) + u_i^T(t) R_i u_i(t) \right] dt + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} \alpha_j J_j$$

$Q_{ii} > 0$, $R_i > 0$ and $R_j > 0$.

Using zero order hold, the objective function can be rewritten as follows:

$$\min_{w_i(k+j|k),i=0,\ldots,H_c^i-1} J_k = \alpha_i \int_{kh}^{\infty} \left[x_i^T(t) Q_{ii} x_i(t) + u_i^T(t) R_i u_i(t) \right] dt + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} \alpha_j J_{k\,j}$$

The resulting function is a hybrid optimization problem. Sampling the objective function within the interval $\left[hh_0, (k+1)h_0 \right)$ leads to a N-periodic discrete Residing

Horizon problem [17]. The performance index J_k^i is a function of h_0 and the estimates states. J_k^i can be written as follows:

$$J_k^i = \tilde{J}_k^i + \hat{x}_i^T\left(k+H_c\left|k\right.\right)Q_{H_c}^{ii}\hat{x}_i\left(k+H_c\left|k\right.\right) \tag{6. 27}$$

where

$$Q_{H_c}^{ii} = \int_{H_c h_0}^{\infty}\left[\exp\left(A_{ii}^T\cdot\left(t-H_c h_0\right)\right)Q_{ii}\exp\left(A_{ii}\cdot\left(t-H_c h_0\right)\right)\right]dt$$

$$Q_{H_c}^{ii} = \int_0^{\infty}\left[\exp\left(A_{ii}^T t\right)Q_{ii}\exp\left(A_{ii}t\right)\right]dt \tag{6. 28}$$

Equation (6. 28) resembles a LQR problem where only the states are considered. Considering that A_{ii} is invertible, $Q_{H_c}^{ii}$ can be found by solving a continuous LMI problem:

$$A_{ii}^T Q_{H_c}^{ii} + Q_{H_c}^{ii}A_{ii} + Q_{ii} < 0 \tag{6. 29}$$

The solution of $Q_{H_c}^{ii}$ is independent of the sampling time.

The term J_k^i can be written as:

$$\tilde{J}_k^i = \int_{kh_0}^{(k+H_c)h_0}\left[x_i^T\left(t\right)Q_{ii}x_i\left(t\right)+u_i^T\left(t\right)R_i u_i\left(t\right)\right]dt$$

Sampling $\tilde{J}_k^i$ at interval sampling time h_{int}, leads to the following performance index:

$$\tilde{J}_k^i = \sum_{g=1}^{NH_c-1}\begin{bmatrix}x_i\left(g\right)\\ w_i\left(g\right)\end{bmatrix}^T\begin{bmatrix}Q_g^{ii} & S_g^i\\ S_g^{i\,T} & R_g^i\end{bmatrix}\begin{bmatrix}x_i\left(g\right)\\ w_i\left(g\right)\end{bmatrix} \tag{6. 30}$$

with

$$Q_g^{ii} = \int_0^{h_{\text{int}}}\left[\exp\left(A_{ii}^T t\right)Q_{ii}\exp\left(A_{ii}t\right)\right]dt$$

$$S_g^i = \int_0^{h_{\text{int}}}\left[\exp\left(A_{ii}^T t\right)Q_{ii}\bar{H}_i\left(t\right)\right]dt$$

$$\bar{H}_i\left(t\right) = \int_0^t\exp\left(A_{ii}^T v\right)B_i dv$$

$$R_g^i = \int_0^{h_{\text{int}}} \left[\bar{H}_i^T(t) Q_{ii} \bar{H}_i(t) \right] dt$$

The upper limit of $\tilde{J}_k^i$ is N-periodic and consequently time varying. To remove this periodicity a time-invariant discrete time predictor can be formulated. Sampling equation (6. 25) at time h_0 and considering also inter-sampling time qh_{int}, $q = 0,...,N-1$, the discrete system will be:

$$x_i(kN+1+q) = A_q^{ii} x_i(kN+q) + B_q^i w_i(kN+q) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} B_q^j w_j(kN+q)$$

forwarding the above equation N-steps ahead gives:

$$x_i(kN+N) = A_q^{ii\,N} x_i(kN) + \sum_{g=0}^{N-1} A_q^{ii\,g} \left[B_q^i w_i(kN+N-1-g) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} B_q^j w_j(kN+N-1-g) \right]$$

Adding output states to equation (6. 26) leads to:

$$\hat{v}_i(kN+1+q) = v_i(kN)$$
$$\breve{v}_i(kN+q) = v_i(kN)$$

forwarding again N-steps ahead:

$$\hat{v}_i(kN+N) = v_i(kN)$$
$$\breve{v}_i(kN+N-1) = v_i(kN)$$

Then, an augmented model can be formulated using the above results as follows:

$$\hat{x}_N^i(k+1) = \tilde{A}^{ii} \hat{x}_N^i(k) + \gamma_i(k-1) \tilde{B}^i w_N^i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N_y} \gamma_j(k-1) B^j w_N^j(k) \tag{6. 31}$$

$$\breve{v}_N^i(k) = \tilde{C}^{ii} \hat{x}_N^i(k) \tag{6. 32}$$

with:

$$\hat{x}_N^i(k) = \begin{bmatrix} x_N^i(k) \\ \hat{v}_N^i(k) \end{bmatrix},\ \tilde{A}^{ii} = \begin{bmatrix} A_q^{ii\,N} & 0 \\ C_{ii} & 0 \end{bmatrix},\ \Psi_{ii} = \begin{bmatrix} A_q^{ii} & 0 \\ 0 & I \end{bmatrix},$$

$$\tilde{B}^i = \begin{bmatrix} \Psi_{ii}{}^{N-1}\hat{B}_i & \cdots & \Psi_{ii}\hat{B}_i & \hat{B}_i \end{bmatrix},\ \tilde{B}^j = \begin{bmatrix} \Psi_{ii}{}^{N-1}\hat{B}_j & \cdots & \Psi_{ii}\hat{B}_j & \hat{B}_j \end{bmatrix},$$

$$\hat{B}_i = \begin{bmatrix} B_q^i \\ 0 \end{bmatrix},\ \hat{C}_{ii} = \begin{bmatrix} C_q^{ii} & 0 \end{bmatrix},\ \tilde{C}^{ii} = \begin{bmatrix} \hat{C}_{ii} & \cdots & \hat{C}_{ii} \end{bmatrix}$$

The presence of $\gamma_i(k-1)$ and $\gamma_j(k-1),\ j=1,...,N_y,\ j\neq i$ shows that equation (6. 31) is an augmented time invariant model equivalent to the original model sampled at time h_0.

Using equation (6. 31), the performance index is also modified as follows:

$$\tilde{J}_k^i = \sum_{g=1}^{H_c-1} \begin{bmatrix} \hat{x}_N^i(k) \\ w_N^i(k) \end{bmatrix}^T \tilde{\Psi} \begin{bmatrix} \hat{x}_N^i(k) \\ w_N^i(k) \end{bmatrix} \qquad (6.\ 33)$$

$$\tilde{\Psi} = \begin{bmatrix} \tilde{Q}^{ii} & \tilde{S}^i \\ \tilde{S}^{iT} & \tilde{R}^i \end{bmatrix}$$

with :

$$w_N^i(k) = \begin{bmatrix} w^i(kN),...,w^i(kN+N-1) \end{bmatrix},\ \breve{v}_N^i(k) = \begin{bmatrix} \breve{v}_N^i(kN),...,\breve{v}_N^i(kN+N-1) \end{bmatrix}^T$$

and:

$$\tilde{Q} = \sum_{j=0}^{N-1} A_N^{jT} Q_g A_N^j$$

$$\tilde{R} = \sum_{j=0}^{N-1} B_{1N}{}^T Q_g B_{1N} + \sum_{j=0}^{N-1} \left(B_{1N}{}^T S_g + S_g{}^T B_{1N} \right) + diag\left[R(0),...,R(N-1) \right]$$

$$\tilde{S} = \sum_{j=0}^{N-1} A_N^{jT} S_g + \sum_{j=0}^{N-1} A_N^{jT} Q_g B_{1N}$$

Equation (6. 24) can now be reformulated as follows:

$$w_N^i(k) = \tilde{K}_i\left(\gamma_i(k-1)\right) v_N^i(k) \qquad (6.\ 34)$$

where: $\tilde{K}_i\left(\gamma_i(k-1)\right) = \left[\tilde{K}_0^i,...,\tilde{K}_{N-1}^i \middle| \gamma_i(k-1) \right]$.

Finally substituting equation (6. 27) into (6. 17), the distributed multirate MPC problem will be:

$$\min_{w_N(k)} J_k\left(\hat{x}_N(k), w_N(k), \Lambda\right) = \sum_{i=1}^{N_y} \alpha_i J_k^i \left(\hat{x}_N^i(k), w_N^i(k), \sum_{\substack{j=1 \\ j\neq i}}^{N_y} w_N^j(k) \right) \tag{6. 35}$$

The above performance index can be solved by applying dynamic programming in a recursive way as follows:

$$J_k\left(\hat{x}_N(k), w_N(k), \Lambda\right) = \min_{w_N(k)} \left(\tilde{J}_k(\Lambda) + J_{k+1}\left(\hat{x}_N(k+1), w_N(k+1), \Lambda\right)\right)$$

where the terminal condition is:

$$J_{k+H_c}^{\ *}\left(\hat{x}_N(k+H_c), \Lambda\right) = \hat{x}_i^T(k+H_c|k) Q_{H_c}^{ii} \hat{x}_i(k+H_c|k)$$

Proof. For a deterministic case and assuming that $J_{k+H_c}^{\ *}\left(\hat{x}_N(k+H_c), \Lambda\right)$ is the terminal condition at time $k+H_c$ with $Q_{H_c}^{ii} = N^{ii}(H_c)$.

For time $k+H_c-1$, $J_{k+Hc-1}\left(\hat{x}_N(k+Hc-1), w_N(k+Hc-1), \Lambda\right)$ will be:

$$J_{k+H_c-1}(\Lambda) = \min_{w_{1N}(k+H_c-1)} \left\{ \begin{bmatrix} x_N(k+H_c-1)^T & w_{1N}(k+H_c-1)^T \end{bmatrix} \tilde{\Psi} \begin{bmatrix} x_N(k+H_c-1) \\ w_{1N}(k+H_c-1) \end{bmatrix} + J_{k+H_c}(\Lambda) \right\}$$

Forwarding the predictor developed in (6. 31) $k+H_c$-steps ahead and replacing it into the above equation, $J_{k+H_c-1}(\Lambda)$ can be written as follows:

$$J_{k+H_c-1}(\Lambda) = \min_{w_{1N}(k+H_c-1)} \left\{ \begin{bmatrix} x_N(k+H_c-1)^T & w_{1N}(k+H_c-1)^T \end{bmatrix} \Psi \begin{bmatrix} x_N(k+H_c-1) \\ w_{1N}(k+H_c-1) \end{bmatrix} \right\}$$

where that Ψ is:

$$\Psi = \begin{bmatrix} \tilde{Q} + A_N^T N(H_c) A_N & \tilde{S} + A_N^T N(H_c) B_{1N} & A_N^T N(H_c) \sum_{j=1}^{N_y} B_{jN} \\ \left(\tilde{S} + A_N^T N(H_c) B_{1N}\right)^T & \tilde{R} + B_{1N}^T N(H_c) B_{1N} & B_{1N}^T N(H_c) \sum_{j=1}^{N_y} B_{jN} \\ \left(A_N^T N(H_c) \sum_{j=1}^{N_y} B_{jN}\right)^T & \left(B_{1N}^T N(H_c) \sum_{j=1}^{N_y} B_{jN}\right)^T & \left(\sum_{j=1}^{N_y} B_{jN}\right)^T N(H_c) \sum_{j=1}^{N_y} B_{jN} \end{bmatrix}$$

$$\Psi = \begin{bmatrix} \psi_1 & \psi_{12} & \psi_{13} \\ \psi_{21} & \psi_{22} & \psi_{23} \\ \psi_{31} & \psi_{32} & \psi_{33} \end{bmatrix}$$

In order to complete the squares in the above matrix, define: $\psi_{22} L = \psi_{12}^T$, such that:

$$x_N^T \left(\psi_{11} - L^T \psi_{22} L\right) x_N + \left(w_{1N} + L x_N + C\right)^T \psi_{22} \left(w_{1N} + L x_N + C\right)$$

then:

$$w_{1N} = -\left(L x_N + C\right)$$

with:

$$L = \left(\tilde{R} + B_{1N}^T N(H_c) B_{1N}\right)^{-1} \left(\tilde{S} + A_N^T N(H_c) B_{1N}\right)$$

$$C = \left(\tilde{R} + B_{1N}^T N(H_c) B_{1N}\right)^{-1} \left[B_{1N}^T N(H_c) \sum_{j=1}^{N_y} B_{jN}\right]^T$$

This verifies the solution of the controller using dynamic programming.

The resulting MDMPC algorithm is:

Step 1 Each agent sets its inter-sampling time h_{int} using input multiplicity N equals to the last iterative index q that achieved global optimality, communicates its decision variables to the others and set $q = 0$.

Step 2 Each agent calculates the SD weighting factors Ψ.

Step 3 Each agent solves its optimization problem (equation(6. 35)).

Step 4 Each agent computes its instant control law (equation (6. 34) and communicates to the plant and each other.

Step 5 Each agent checks its terminal iteration condition (equation (6. 18). If all the conditions are satisfied, go to step 4; otherwise go to step 3.

Step 6 Each agent repeats step 5 until $q = N$. If global optimality was not achieved, N is doubled and go to step 2; otherwise go to step 3.

6.6 Numerical Example

To test the above algorithm a MIMO LTI will be used. The system matrix transfer function is described as follows:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{10s+1} & \frac{-1}{7.5s+1} \\ \frac{1.5}{11s+1} & \frac{1.5}{8s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

As described in [80], if the above system is decomposed into two subsystems, they show a strong interaction with a non-cooperative behaviour between them because one of the gains is negative.

This example can be used to simulate either decentralized or distributed schemes by choosing the value of $\alpha_i, i = 1,2$. Simulations will only be run for the distributed example as the algorithm was developed for multivariable NCS rather than Predictive Control. α_i is chosen to 0.5. The resulting subsystems will be:

$$\text{Subsystem 1: } y_1 = \frac{u_1}{10s+1} - \frac{u_2}{7.5s+1}$$

$$\text{Subsystem 2: } y_2 = \frac{1.5u_1}{11s+1} + \frac{1.5u_2}{8s+1}$$

The simulation is done using Matlab. A continuous model of the subsystems is built in Simulink as shown in Fig. 6. 3:

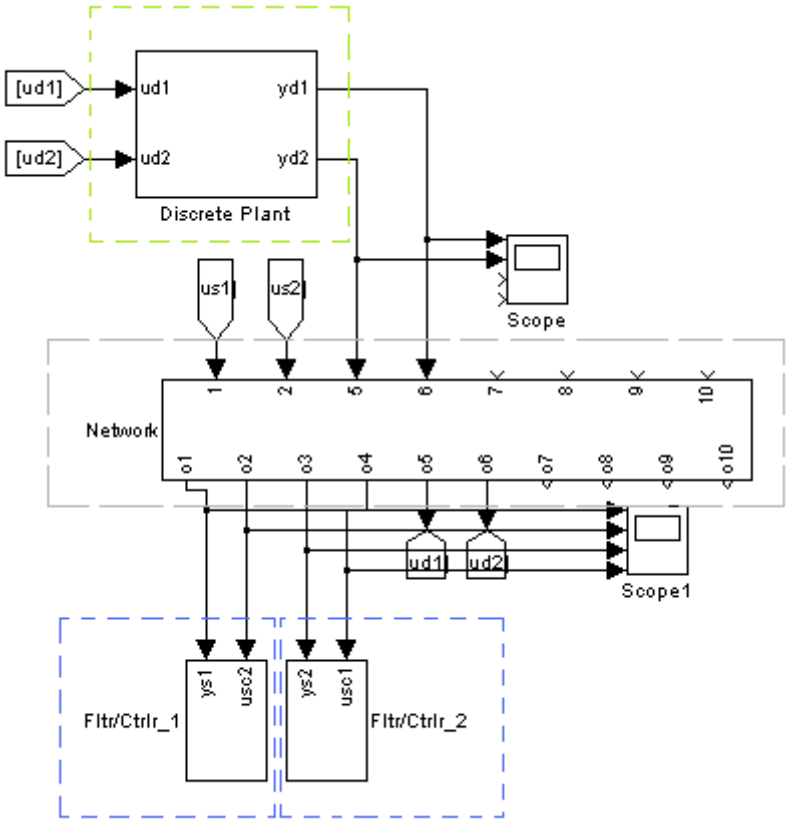


Fig. 6. 3 Block diagram of the Distributed NCS

The network is an Ethernet network implemented by using TrueTime 1.5 simulator. Sensors, actuators and controllers are built by using real-time kernels. Both subsystems are sampled at $h_0 = 0.1$. Actuators update their signals periodically at $h_{\text{int}} = \frac{h_0}{N}$, with N chosen to initially be equal to 10. Individual filters are updated at h_0, whereas controllers at h_{int}. The network specifications are:

TYPE	CSMA/CD Ethernet
Speed	10Mbps
Packet size	80 bits
Packet Loss probability	Variable

For the simulation, packet loss probability is chosen to be 40%, however due to data traffic, its value can be bigger. Reference signals are chosen to be equal to 1 with a

change in reference at t = 30 seconds. Simulation results are presented in Fig. 6. 4. and Fig. 6. 5

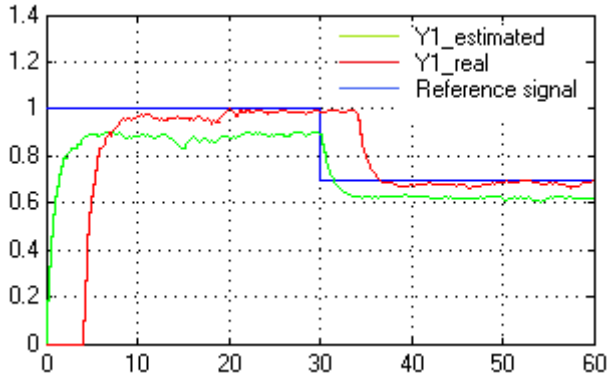


Fig. 6. 4 Subsystem 1: output response due to change in reference signal. Comparison between estimated output and real output

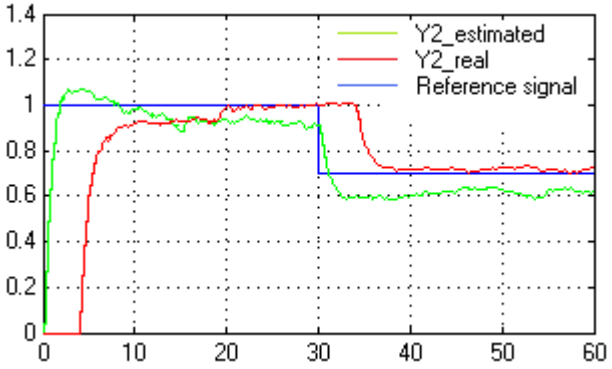


Fig. 6. 5 Subsystem 2: output response due to change in reference signal. Comparison between estimated output and real output

It can be seen that due to the time driven nature of the filter, the estimated outputs are delay-free responses. On the other hand, real outputs are delayed $t = \left\lfloor th_0 + \tau_{SC}(k) + \tau_{CA}(k) \right\rfloor$ times. Both outputs show good tracking response. Process and measurement noise have been filtered before the control law calculation, thus what it seems to be noisy signals are in fact the effects of compensating controller-to-actuator packet dropouts. Output response of subsystem 2 is also nonnegative due to its forwarded control law. There exists a difference between the estimated responses and the real responses because the packet drops indicator is also considered in the

estimated system althought the estimated system is not affected by packet dropouts. Controller outputs are shown in Fig. 6. 6 and Fig. 6. 7.

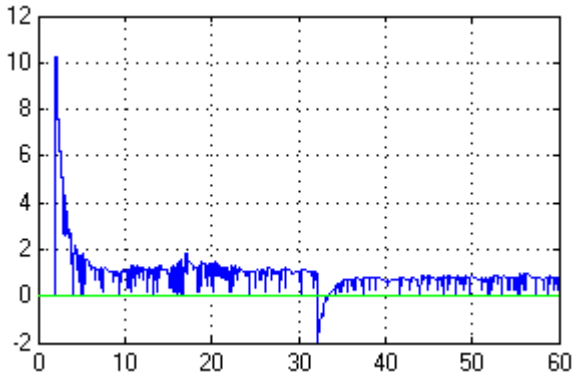


Fig. 6. 6 Subsystem 1: control signal 1 applied to the real subsystem after the network delays

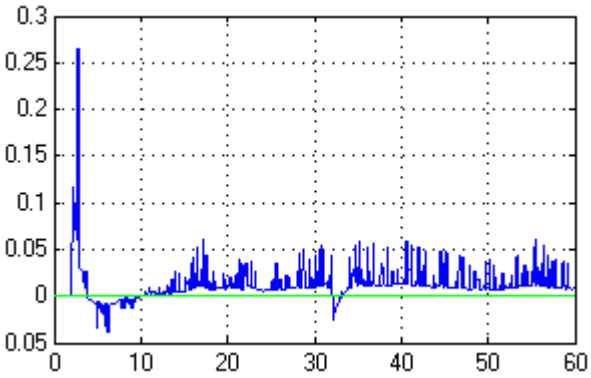


Fig. 6. 7 Subsystem 2: control signal 2 applied to the real subsystem after the network delays

Due to the 40% packet loss probability, it can be seen that control signals are not smooth. Peaks represent the controller compensation due to packet loss. These signals reach the time-driven actuators at $t = \left[th_0 + \tau_{CA}(k) \right)$. Control signal 1 shows a considerably high initial peak due to the negative gain present in the subsystem. Control signal 2 initial peak shows an initial increment followed by a sudden decrease immediately after control signal 1 increase drastically. This is the effect of the Nash optimal to keep stability and provide cooperation between both subsystems.

6.7 Conclusions

The proposed methodology decomposes a multivariable NCS into $m \leq N_y$ subsystems in order to compensate single-loop network delays. Distributed control leads to multiobjective optimization problems that reduce the computational burden of the filter and controller calculation. Individual subsystems can forward the calculated i-th sensor-to-controller delay to implement Predictive control methodologies and compensates i-th controller-to-actuator delays. The proposed methodology is formulated as multirate to reduce the effects of packet dropouts in the controller-to-actuator interface. During inter-sampling intervals every agent reaches local optimality and controls its subsystem. Global optimality is achieved in the sense of Nash and therefore the overall multivariable NCS can be controlled.

Chapter 7

Final Conclusion and Future work

Integrated Methods for Control of NCS were proposed in this thesis. Suitable approximations of the NCS problem as time-varying, asynchronized and distributed systems are used to develop non-recursive and recursive Control and Estimation methodologies. Control system components are mostly considered as time-driven components. The resulting methodologies are applied to single-loop and multi-loop NCSs. Reasons that motivated these assumptions and results as well as suggestions for future work are addressed in the following sections.

7.1 Main Conclusions

Any attempt of using a communication network to exchange information among control components (sensors, actuators and controllers) can be classified as Networked or Networked-based Control Systems. NCS applications usually present poor stability and performance degradation. Performance degradation has been the main concern in this thesis.

The first contribution of the research work addressed a delay-dependent stabilizability approach to design structured controller for NCS using implicit model transformation of a Lyapunov-Krasovskii functional and the pdf of the network delay. To achieve

good performance, it was necessary to incorporate only a portion of the delay distribution. The incorporation of the reduced region of the pdfof the delay led the research to areas of probabilistic design for control.

The second contribution of the thesis is the implementation of a bad data detector and its combination with Larsen's modified Kalman filter. On time-driven components the resulting estimation methodology can calculate and compensate network delays and packet dropouts in the sensor-to-controller interface. These calculations also facilitated the implementation of Optimal control with predictor-like model transformation and Predictive control methodologies with forwarded control signals to fully compensate the round trip network delay.

Without information about the transmitted control signal, it is almost impossible to compensate controller-to-actuator packet dropouts during the control calculation. The third contribution of this thesis is the use of packet delivery indicators over multirate controllers to count and compensate packet dropouts in the controller-to-actuator interface. The resulting methodology has to be implemented on multivariable NCS after decoupling the overall system into single-loop subsystems leading to a distributed structure for NCS. The proposed methodology incorporates DMPC together with multirate systems control design and Nash optimality to achieve a DMMPC methodology that stabilizes multivariable NCS and fully compensates network constraints.

Finally to fully deploy the network capabilities, NCS can be considered as the integration between Distributed Control, Sensor Networks and Communication Networks, where the trade-off between communication errors, transmission rates and protocols constitutes a main limitation for control and estimation. A final contribution of this research work is a critical review of the state of the art of NCS that addresses the motivations behind the use of each of the control methodologies within the NCS control problem.

7.2 Future work

The research work done in this thesis showed the applicability of recursive methodologies to address the computational issues involved in NCS. Research areas involving recursive methodologies are worth to be investigated in the near future. Additionally aspects such as nonlinearities and constraints correlation can be incorporated in the controller design as follows:

1. Missequencing in NCS. It was stated that network delays longer than the sampling time makes the delivered packets to be out of order. Coherent data analysis can be exploited to improve the robustness of the control methodologies proposed in this thesis.

2. Adaptive Control methodologies for NCS. For fast NCS applications can offer reliable parameter adjustment as well as controller-to-actuator delay compensation. Model Predictive Control speed depends on the control horizon, thus adaptive control is a good candidate to replace the optimal controllers developed in this thesis.

3. Bayesian estimators for nonlinear NCS. It was stated that for linear stochastic systems the delays do not affect the distribution of the disturbances. If network constraints such as channel capacity and quantization are considered, more general estimation methodologies should be considerd.

4. Markov jump systems. The use of the bad data detector combined with the filter revealed an adaptation mechanism for modelling controller-to-actuator network delays and packet dropouts. Furthermore if it is possible to prove a direct correlation between network constraints, a composite model can be determined as Markov Jump system. Controller design for jump linear NCS have been studied but none of the studies uses a composite model of the constraints.

5. Dynamic bandwidth allocation. Real-time systems depend on the network allocation. Incorporating dynamic bandwidth allocation in the controller design can allow the development of better control methodologies for multivariable NCS and a plausible control of the network.

References

[1] Addelzaher, T.F., Atkins, E.M., Shin, K.G.: "QoS negotiation in real-time systems and its application to flight control", IEEE Transactions on Computers, 2000.

[2] Almutairi, N.B., Chow, M.Y., Tipsuwan, Y.: "Network-based controlled DC motor with Fuzzy compensation", The 27th Annual Conference of the IEEE Industrial electronics society (IECON 01), Denver, CO., 2001.

[3] Anon: "Scale communicates via Profinet", Prof. Eng., 2004.

[4] Artstein, Z.: "Linear Systems with Delayed Controls: A reduction", Auomatic Control, IEEE, 1982.

[5] Åstrom K. J., Wittenmark B.: "Computer controlled systems, Theory and Design". Prentice Hall information and System Science Serires. 1990.

[6] Aswin, N., Venkat, J., Rawlings, B., Wright, S.: "Stability and optimality of distributed model predictive control", Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, December 2005.

[7] Babak, A.: "Stability of Networked Control Systems in the Presence of Packet Losses". 42nd IEEE conference on Decision and Control Proceedings, 2003

[8] Balderud, J., Giovanini, L., and Katebi, R.: "Distributed Control Design for Underwater Vehicles". ImechE Proceedings, Part M: Journal of Engineering of the Maritime Environment, 2008.

[9] Basin, M., "New trends in Optimal Filtering and Control for Polynomial and Time-delay Systems". Lecture Notes in Control and Information Science. 2008.

[10] Beldiman O., Bushnell L., Walsh G.: "Predictors for Networked Control Systems", Proceedings of the American Control Conference, Chicago, Ill. June 2002.

[11] Beldiman O., Bushnell L., Walsh G., Wang H., Hong Y.: "Perturbation in Network Control Systems", Proceedings of ASME-IMECE'01, New York, USA November 2001.

[12] Bollot J. C.: "Characterizing End-to-End Packet Delay and Loss in the Internet Journal of High-Speed Networks", pp. 289 – 298, Sigcomm 1993.

[13] Boukas, E.: "Deterministic and Stochastic Time-Delay Systems", Springer-Verlag New York, Inc, 2002.

[14] Bryson, A., Ho, Y.: "Applied Optimal Control: Optimization, Estimation and Control". Taylor & Francis, 1975.

[15] Brockett R. W., Liberzon D.: "Quantized Feedback Stabilization of Linear Systems", IEEE Transactions on Automatic Control, Vol. 45, No 7, July 2000.

[16] Brockett R. W.: "Minimum attention control", Proceedings of the IEEE Conference on Decision and Control, San Diego, California, USA, December 1997.

[17] Cao, Y., Hu, L., Frank, P.: "Model Predictive Control via Piecewise Constant Output Feedback for Multirate Sampled-data Systems", Proceedings of the 39th IEEE, Conference on Decision and Control, Sydney, Australia, December, 2000.

[18] Chang H., Özgüner Ü.: "Closed loop control of systems over a communication network with queues", International journal of Control, 1995

[19] Chang, K., Lee, S.: "Remote Controller Design of Networked Control Systems using Genetic Algorithms", ISIE, Korea, 2001.

[20] Comer D. E.: "Computer Networks and Internet", International Edition Prentice Hall, 1999

[21] Dai, J., Cui, B.: "A new delay systems approach to quantized networked control systems", Manuscript draft, Jiangnan University, 2008

[22] Delchamps D. F.: "Stabilizing a Linear system with quantized state feedback", IEEE Transactions in Automatic Control, Vol. 35, No 8, August 1990.

[23] Ekanayake M.M., Premaratne K., Douligeris C., Bauer P.H.: "Stability of Discrete-Time Systems with Time-Varying Delays", In proc. American Control Conference, Arlington, VA, 2001.

[24] Elia N., Mitter S. K.: "Stabilization of Linear Systems with Limited Information", IEEE Transactions on Automatic Control, Vol. 46, No 9, September 2001.

[25] Fang, Y.: "A New General Sufficient Condition for Almost Sure Stability of Jump Linear Systems", IEEE Transactions on Automatic Control, Vol. 42, No. 3, March 1997.

[26] Fang, L., Wu, Z.: "Fuzzy immune self regulating PID control for Communication networked Control Systems", International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06). 2006.

[27] Franklin G. F., J. D. Powell: "Digital control of Dynamic systems". Ellis-Kagle Press, 1998.

[28] Ghude, S.: "Design a PID Controller with Missing Packets in a Networked Servo-System", Master dissertation, University of Southern Queensland, Faculty of Engineering and Surveying. March, 2007.

[29] Giovanini, L., Balderud, J.: "Game approach to distributed Model Predictive control", UKACC group. Control Conference, 2006.

[30] Göktas, F.: "μ-Synthesis for Distributed control of systems with Network Induced Delays", Proceedings of the 35th Conference on Decision and Control, Japan, 1996.

[31] Goodwin, G., Sin, K., "Adaptive Filtering Prediction and Control". Prentice Hall Information and System Sciences Series. 2009.

[32] Halevi Y., Ray A: "Integrated communication and control systems, Part I-Analysis", Journal of Dynamic Systems, Measurements and Control, Vol. 110, pp. 367-373, December 1988.

[33] Hong S.: "Scheduling algorithm of data sampling time in the integrated communication and control systems", IEEE transactions on Control Systems Technology, 1995.

[34] Hong S., Kim W. : "Bandwidth Allocation scheme in CAN protocol", IEE proceedings- Control Theory and Application, 2000.

[35] Hoyakem, P., Abdallah, F., Chaouki, T.: "Networked Control Systems: A sampled-data Approach", IEEE International Symposium on Intelligent Control, Houston , Texas, USA, October 2003.

[36] Hokayem Peter F., Addallah Chaouki T.: "Inherent Issues in Networked Control Systems: A survey", Proceedings in the American Control Conference, Boston, Massachusetts, July 2004.

[37] Hörjel A.: "Bluetooth in control", MS thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 2001.

[38] Hristu D., Morgansen K.: "Limited Communication Control", Systems and Control Letter, 1999.

[39] Hu, Z.S., Zhu, Q.X.: "Stochastic Optimal Control and analysis of stability of networked control systems with long delay", Automatica, 2003.

[40] Jain R., Simsek T., Varaiya P.: "Control under Communication Constraints", Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, pp.3209-3216, December 2002.

[41] Jansen D., Buttner H.: "Real-time Ethernet – the EtherCAT solution", Computing & Control Engineering Journal, 2004.

[42] Ji, K., Kim, W.: "Robust Control of Networked Control Systems with Admissible Parameter Uncertainties", International Journal of Control, Automation and Systems, vol. 5, August 2007.

[43] Katayama, T.: "Subspace Methods for System Identification", Springer-Verlag, January 2005, Japan.

[44] Kharitonov, V., Gu, K., Chen, J.: "Stability of Time-Delay Systems". Birkhäuser, Technology and Engineering, 2003. Pg. 10-15.

[45] Kim Y., Kwon W., Park H.: "Stability and scheduling method for network-based control systems", the 22dn Annual Conference of the IEEE industrial electronics society, Taiwan, 1996.

[46] Kim Y., Kwon W., Park H.: "A scheduling method for Network-based Control systems", Proceedings of the American Control conference, 1998.

[47] Krtolica R., Özgüner Ü., Chan H., Göktas H., Winkelman J., Liubakka M.: "Stability of linear Feedback Systems with random communication delays", International Journal of Control, Vol. 59, No 4, 1994.

[48] Kun, J., Kim, W.: "Robust Control for Networked Control Systems with Admissible Parameter Uncertainties", International Journal of Control, Automation and Systems, Vol. 5, No. 4, pp.373-378, August 2007.

[49] Kwon, W., Han, S., "Receding Horizon Control: Model Predictive Control for state models". Springer, Technology and Engineering, 2005

[50] Lakshmikantham, V., Leela,S., Martynyuk, A.: "Practical Stability of Nonlinear Systems", Syngapore, World Scientific, 1990.

[51] Larsen, T., Andersen, N., Ravn, O., Poulsen, N.: "Incorporation of Time Delayed Measurements in Discrete-time Kalman Filter". Proceedings of the 37th IEEE Conference on Decision and Control, 1998.

[52] Leite, V., Tarbouriech, S., Peres, P.: "A convex approach for robust state feedback control of discrete-time systems with state delay", Proceedings of the 2004 American Control Conference, Boston, Massachusetts, July 2004.

[53] Lian F., Moyne J., Tilbury D.: "Performance Evaluation Control Networks: Ehternet, ControlNet and DeviceNet", Technical report UM-MEAN-99-02, Febrary 1999.

[54] Lian F., Moyne J., Tilbury D.: "Analysis and Modeling of Network Control Systems: MIMO case with multiple delays", Proceedings of American Control Conference, Arlington, Virginia, June 2001.

[55] Li, B., Nahrstedt, K.: "A control-based middleware framework for quality-of-service adaptations", IEEE Journal on selected Areas in Communications, 1999.

[56] Li, S., Wang, Z., Sun, Y.: "A novel Auto-tuning Robust PID controller for Communication networked Control Systems". The 29th Annual Conference of the IEEE, Industrial Electronics Society, IECON, 2003.

[57] Liberzon, D.: "On Stability of linear systems with limited information", IEEE Transactions in Automatic Control, Vol. 50, 2005.

[58] Liou, L.W, Ray, A.: "Integrated communication and control systems: Part III-nonidentical sensor and controller sampling", Journal of Dynamic Systems, Measurements and Control, 1990.

[59] Liu X., Goldsmith A.: "Wireless communication tradeoffs in Distributed Control", Submitted to the First International workshop on Sensor Network Protocols and Applications (SNPA 2003).

[60] Liu G. P., Rees D., Chai S. C., X. Nie Y.: "Design, Simulation and Implementation of the Networked Predictive Control Systems". IEEE Proceedings on Networking, Sensing and Control, 2005.

[61] Liu, G. P., Mu, J. X., Rees, D., Chai, S. C., "Design and stability of Networked Control Systems with random communication time delay using modified MPC", International Journal of Control, April 2006.

[62] Low,S., Paganini, F., Doyle, J.: "Internet congestion control", IEEE Control systems Magazine, February 2002.

[63] Luck R., Ray A.: "An observer-based compensator for distributed delays", Automatica, 1990.

[64] Luck R., Ray A.: "Experimental verification of a delay compensation algorithm for integrated communication and control systems", International journal of control, 1994.

[65] Mahmoud, M.: "Robust Control and Filtering for Time-Delay Systems". Marcel Dekker, Technology & Engineering. New York, 2000.

[66] Matveev A. S., Savkin A. V.: "Optimal LQG Control via Limited Capacity Communication Networks", Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, pp.4047-4052, December 2002.

[67] Matveev and Savkin: "Estimation and control over communication networks". Book, Birkhäuser, 2009.

[68] Montestruque L. A., Antsaklis P. J.: "Stochastic Stability for model-based Networked control Systems", Proceedings of American Control Conference, Denver, CO. 2003.

[69] Mukherjee, A.: "On the Dynamics and Significance of Low Frequency Components of Internet Load". Techincal Report, CIS. University of Pensylvania, December, 1992.

[70] Nahi, N.: "Optimal Recursive Estimation with Uncertain Observation", IEEE Transactions on Information Theory, July, 1969.

[71] Nair G. N., Evans R. J.: "Mean square stability of stochastic linear systems with data rate constraints" Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, pp. 1632-1637, December 2002.

[72] Nilsson, J.: "Real-time control systems with delays", Ph.D. dissertation, Lund Institute of Technology, Department of Automatic Control, 1998.

[73] Ohlin, M., Henriksson, D., Cervin, A.: "True Time 1.5–Reference Manual", Department of Automatic Control, Lund University, January 2007.

[74] Oppenheim A. V., R. Schafer W., Buck J.R.: "Discrete-Time Signal Processing". Prentice-Hall Signal Processing Series, 1999.

[75] Orderud, F., "Comparisson of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements". Proceedings of Scandinavian Conference on Simulation and Modelling, SIMS 2005.

[76] Orham Imer Ç, Yuksel Serdar, Başar Tamer: "Optimal control of LTI systems over unreliable communication links". Automatica, Journal of IFAC, 2006.

[77] Pahjola Michael: "PID Controller Design for Networked Control Systems", Master's thesis for the degree of Master of Science in Technology, Espoo, 9. January, 2006.

[78] Pahjola, M., Koivo, H.: "Measurement Delay Estimation for Kalman Filter in Networked Control Systems", Proceedings of the 17th World Congress IFAC, 2008.

[79] Recalde, L., Katebi, R., "Networked PID control Design: A Pseudo-Probabilistic Robust Approach", IFAC, 2009.

[80] Recalde, L., Katebi, R.: “On Estimation Approach of Networked Control Systems”, IFAC, 2010.

[81] Richard, J.: “Time-delay Systems: an overview of some recent advances and open problems”, Automatica, Vol. 39, April 2003.

[82] Ryu, S., Cho, C.: “PI-PD-controller for robust and adaptive queue management for supporting TCP congestion control”, Proceedings of the 37th Annual Simulation Symposium, IEEE, 2004.

[83] Sahai A.: "Evaluating Channels for Control: Capacity Reconsidered", Proceedings of the American Control Conference, Vol. 4, pp. 2358-2362, 2000.

[84] Seiler P., Sengupta R. "Analysis of Communication Losses in Vehicle control problems", Proceedings of the American Control Conference, Arlington, VA, pp. 1491-1496, 2001.

[85] Silva, G.: “PID controllers for Time-delay systems”. Birkhäuser, Technology & Engineering, 2004

[86] Sinopoli, B.: “Kalman Filtering with Intermittent Observations”, Proceedings of the 43rd IEEE Conference on Decision and Control, Maui, Hawaii, 2003.

[87] Stankovic, J.: “A Serious Problem for Next-Generation Systems”, IEEE, Computers, 1988.

[88] Tatikonda S. C.: "Control under communication constraints", IEEE Transactions on Automatic Control, 2004.

[89] Tipsuwan, Y., Chow, M.-Y.: "Network-based controller adaptation based on QoS negotiation and deterioration", The 28th Annual Conference of the IEEE Industrial electronics society (IECON 02), 2001.

[90] Tipsuwan, Y., Chow, M.: "Control Methodologies in networked control systems", Control Engineering practice 11, February 2003

[91] Tipsuwan, Y., Chow, M.Y.: "Gain scheduler middleware: a methodology to enable existing controllers for network control and teleoperation – Part I: networked control", IEEE Trans. Ind. Electron., 2004.

[92] Velasco, M., Marti, P., Villa, R., Fuertes, J.: “Stability of Networked Control Systems with Bounded Sampling Rates and Time Delays”, 31st Annual Conference of IEEE, Industrial Electronics Society, IECON 2005.

[93] Verriest E., Egerstedt M.: "Control with Delay and Limited Information", Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, pp.1231-1236, December 2002.

[94] Walsh G. C., H. Ye, Bushnell L. G.: "Stability analysis of networked control systems", IEEE Trans. Control Syst. Technol., pp. 438-446, 2002.

[95] Wang S.H., Davisson E.J.: “On the Stabilization of Decentralized Control Systems”, IEEE Trans. Automatic Control (18), 1973.

[96] Wang J. G., Ravindran B.: "Time-utility function-driven switched Ethernet: packet scheduling algorithm, implementation and feasibility analysis", IEEE Trans. Parallel Distrb. Syst., pp. 119-133, 2004.

[97] Wang fei-yue, Liu derong: “Networked control systems: theory and applications”. Springer-Verlag, 2008.

[98] Wing, S., Brockett, R.: “Systems with Finite Communication Bandwidth Constraints-II: Stabilization with Limited information feedback”, IEEE Transactions on automatic Control, Vol. 44, No. 5, May 1999.

[99] Wood Anthony D., Stankovic John A.: "Denial of Service in Sensor Networks", IEEE proceedings in Computers, October 2002.

[100] Xu, M., Li, S., Qi, C., Cai, W,: "Auto-tunning of PID controller parameters with supervised reciding horizon optimization", ISA transactions, 2005.

[101] Yang T. C.: "Networked Control Systems: a brief survey", IEE proceedings in Control Theory and Application, Vol. 153, July 2006.

[102] Yang, F., Wang, Z., Hung, S., Gani, M.: "H∞ Control for Networked Systems with Random Communication Delays", IEEE transactions on Automatic Control, March 2006.

[103] Ye H., Walsh G., Bushnell L.: "Wireless local area networks in the manufacturing industry", Proc. Amer. Control Conf., Chicago, IL, 2000

[104] Yook J.K., Tilbury D. M., Soparkar N.R.: "A design methodology for distributed control systems to optimize performance in presence of time delays", Proceedings of the 2000 American Control Conference.

[105] Yuanqing Xia, Chen, J., Liu, G.P., Rees, D.: "Stability Analysis of Networked Predictive Control Systems with random Network Delay", Proceedings of the IEEE International Conference on Networking, Sensing and Control, London, UK, 2007.

[106] Zhao, Y.: “Packet-Based Control for Networked Control Systems”, PhD thesis, University of Glamorgan, Prifysgol Morgannwg, pp. 22-29, May 2008.

[107] Zhang W., Branicky M., Phillips M.: "Stability of Networked Control Systems", IEEE Control Systems Magazine, 2001.

Appendix A

A.1 Controller State-Space formulation

Having the following transfer function:

$$w(k)=\frac{\left(d_0q^2+d_1q^1+d_2\right)}{\left(q^2+c_1q^1+c_2\right)}e(k) \tag{A. 1}$$

define the following states: $\begin{bmatrix} w(k) \\ w(k+1) \end{bmatrix}=\begin{bmatrix} x_{c1}(k) \\ x_{c2}(k) \end{bmatrix}$

assuming that linear relation holds so that $w(k)\propto e(k)$, then equation (A.1) can be expressed in state- space for as follows:

$$\begin{bmatrix} x_{c1}(k+1) \\ x_{c2}(k+1) \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ -c_2 & -c_1 \end{bmatrix}\begin{bmatrix} x_{c1}(k) \\ x_{c2}(k) \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}e(k) \tag{A. 2}$$

And

$$w(k)=d_2x_{c1}(k)+d_1x_{c2}(k)+d_0\left\{\begin{pmatrix} -c_2 & -c_1 \end{pmatrix}\begin{bmatrix} x_{c1}(k) \\ x_{c2}(k) \end{bmatrix}+e(w)\right\}$$

$$w(k)=\begin{pmatrix} d_2-c_2d_0 & d_1-c_1d_0 \end{pmatrix}\begin{bmatrix} x_{c1}(k) \\ x_{c2}(k) \end{bmatrix}+d_0e(w) \tag{A. 3}$$

A.2 Implicit Transformation of a Lyapunov-Krasovskii functional

Lyapunov-Krasovskii functional for time delayed systems can be defined as follows:

$$V\left(\phi(-\tau)\right):=\phi(-\tau)^T\,P\phi(-\tau) \tag{A. 4}$$

Knowing that the difference of a series is:

$$\sum_{j=-\tau}^{-1}\Delta V\left(\phi(j)\right)=V\left(\phi(0)\right)-V\left(\phi(-\tau)\right) \tag{A. 5}$$

then the standard Lyapunov-Krasovskii functional can be expressed as follows:

$$V\left(\phi(0)\right)=\sum_{j=-\tau}^{-1}\Delta V\left(\phi(j)\right)+V\left(\phi(-\tau)\right) \tag{A. 6}$$

using the same approximation as in (A.5) into (A.4), equation (A.6) can be given by:

$$V\left(\phi(0)\right)=V_1\left(\phi(0)\right)+V_2\left(\phi(0)\right)+V_3\left(\phi(0)\right) \tag{A. 7}$$

$$V_1\left(\phi(0)\right)=\phi(0)^T P\phi(0)-2\phi(0)^T P\sum_{j=-\tau}^{-1}\Delta\phi(j)$$

$$V_2\left(\phi(0)\right)=\sum_{j=-\tau}^{-1}\Delta V\left(\phi(j)\right)$$

$$V_3\left(\phi(0)\right)=\sum_{j=-\tau}^{-1}\Delta\phi(j)^T P\Delta\phi(j)$$

Additionally, it can be considered that the following inequality:

$$2\sum_{\alpha}a(\alpha)^T b(\alpha)=\sum_{\alpha}\begin{bmatrix}a(\alpha)^T & b(\alpha)^T\end{bmatrix}\begin{bmatrix}X & Y\\ Y^T & Z\end{bmatrix}\begin{bmatrix}a(\alpha)\\ b(\alpha)\end{bmatrix}$$

can be used in $V_1\left(\phi(0)\right)$ such that:

$$V_1\left(\phi(0)\right)\le\phi(0)^T P\phi(0)+\begin{bmatrix}\phi(0)^T & \sum_{j=-\tau}^{-1}\Delta\phi(j)^T\end{bmatrix}\begin{bmatrix}X & Y\\ Y^T & Z\end{bmatrix}\begin{bmatrix}\phi(0)\\ \sum_{j=-\tau}^{-1}\Delta\phi(j)\end{bmatrix}$$

then:

$$V\left(\phi(0)\right)\le V_1\left(\phi(0)\right)+V_2\left(\phi(0)\right)+V_3\left(\phi(0)\right) \tag{A. 8}$$

A. 3 Linear Matrix Inequalities (LMI)

A Linear Matrix Inequality (LMI) is a set of n polynomial inequalities in a variable $x\in R^m$. This set has the form:

$$F(x):=F(0)+\sum_{i=1}^{m}x_i F_i>0$$

with $F_i = F_i^T \in R^n \times R^n$ being positive definite such as $u^T F(x) u > 0$. This is a strict LMI and represents a convex constraint on x. Depending of the nature of these constraints the form can vary from linear inequalities, quadratic inequalities, matrix norm inequalities; to particular inequalities that arise in control theory such as Lyapunov inequalities and convex quadratic matrix inequalities. Nonlinear inequalities are also possible. Nonlinear as well as quadratic inequalities can be converted to a Linear LMI form by using Schur complements. The transformation is as follows:

Having an LMI of the form:

$$\begin{bmatrix} Q(x) & S(x)^T \\ S(x) & R(x) \end{bmatrix} > 0$$

with $Q(x) = Q(x)^T$, $R(x) = R(x)^T$ and $S(x)$ depending affinely on x, the following representation is possible:

$$Q(x) > 0,\ Q(x) - S(x)^T R(x)^{-1} S(x) > 0.$$

Schur complement is useful because the LMI problems arising in Control are sometimes quadratic. The variable x is a random variable chosen to demonstrate the theory of LMIs and has no connection with the variables used to describe the NCS problem.

There are some standard convex problems that arise in LMI theory such as:

- Feasibility problem, where given a LMI, the problem is to find a feasible x^{feas}, such as $F(x^{feas}) > 0$ or probe infeasibility. This problem arises either as stability or Stabilizability test.

- Eigenvalue problem, where the problem is to minimize the maximum Eigenvalue of a matrix associated to the variable x, subject to a LMI constraint, or to determine that the constraint is infeasible. This problem is stated as follows:

$$\begin{aligned} &\min \quad \lambda \\ &s.t. \quad \lambda I - A(x) > 0 \qquad B(x) > 0 \end{aligned}$$

$A(x)$ and $B(x)$ are symmetric matrices that depend affinely on x. In control theory such problems arise at L_2 and *RMS* gains.

- Generalized Eigenvalue problem, where the problem involves the minimization of the maximum generalized Eigenvalue of a pair of matrices. This problem appears on decay rate for bounded stability and is defined as follows:

$$\begin{aligned} &\min \quad \lambda \\ &s.t. \quad \lambda B(x) - A(x) > 0 \quad B(x) > 0 \quad C(x) > 0 \end{aligned}$$

The above LMI problems constitute convex optimization problems for strict LMIs. For nonstrict LMIs $\left(F(x) \geq 0\right)$, it can be stated that the feasible set of the nonstrict case is the closure of the strict LMI, so that it is sufficient to solve the strict LMI problem.

A way to describe a LMI is through differential inclusions (DI). A differential inclusion is described as follows:

$$\dot{x} \in F\left(x(t), t\right), \; x(0) = x_0$$

where $F(\cdot)$ is a set-valued function on $R^n \times R_+$ and any $x : R_+ \to R^n$ that satisfies the differential inclusion is its solution. Using a standard result called the Relaxation theorem, it can be stated that $F(x,t)$ is a convex set for every x and t. So that the DI can be defined by:

$$\dot{x} \in \mathbf{Co}F\left(x(t), t\right)$$

Differential inclusions can also be described by a family of linear systems $\dot{x} \in \Omega x$, where $\Omega \in R^n \times R^n$ can be time-invariant, time-varying or polytopic. Thus, a LDI can be generalized as a Linear State-Space System with inputs and outputs.

$$\begin{aligned} \dot{x} &= A(t)x + B(t)u, \quad A(t) \in \Omega \\ y &= C(t)x \end{aligned} \tag{A. 9}$$

State properties as well as input-output properties of the resulting LDI can be described as LMI problems.

a. Feedback Synthesis for LDIs

Feedback Controller design can be stated by using either state or input-output properties. The simplest design method is Quadratic Stabilizability. In terms of linear systems theory, Stabilizability is equivalent to the condition that every unstable mode can be controllable with a controller of the form $u(t) = Kx(t)$. Here $u(t)$ is the control signal, $x(t)$ are the states of the system and K is the state-feedback gain that can be either static or dynamic.

The system described in (A. 9)is said to be quadratically stabilizable, if there exists a state-feedback gain K such that the closed loop system derived as follows:

$$\dot{x}(t) = (A + BK)x(t)$$

and having a quadratic function $V(\zeta) = \zeta^T P \zeta$, $P > 0$; is quadratically stable and consequently stable.

Using $V(\zeta) = V(x)$, where $V(x)$ is a quadratic Lyapunov function, the function decreases along every nonzero trajectory of (A. 9) and is derivate is negative such that:

$$(A + BK)^T P + P(A + BK) < 0$$

this condition is not jointly convex in K and P, but changing variables as follows: $Q := P^{-1} \quad Y := KQ$, and using the new variables, the above condition becomes:

$$AQ + QA^T + BY + Y^T B^T < 0$$

This condition is LMI in Q and Y and is feasible if there exist $Q > 0$ and Y. The control signal will be: $u(t) = YQ^{-1}x(t)$.

The above condition can be reduced in number of variables by using the following elimination matrix method:

Consider the following inequality:

$$G(z) + U(z)XV(z)^T + V(z)X^T U(z)^T > 0$$

by using the Finsler's lemma the above condition can expressed as the following form:

$$G(z) + \sigma U(z)U(z)^T > 0$$

Using this result, the condition for Stabilizability can be expressed as follows:

$$AQ + QA^T - \sigma BB^T < 0$$

and without loss of generality, it is possible to choose $\sigma = 1$. The resulting gain will be:

$$2BY = -\sigma BB^T$$
$$K = -\left(\frac{\sigma}{2}\right)B^T Q^{-1}$$

For time delayed systems, stability and stabilizabity requires the construction of a Lyapunov-Krasovskii functional $V(t, x_t)$. This functional will include the delay-free states as well as the delayed states. The simplest functional is of the form:

$$V(x_t) = x(t)^T Px(t) + \int_{t-\tau}^{t} x(\theta)^T Px(\theta)\,d\theta \qquad \text{(A. 10)}$$

Hence, Stabilizability is achieve if the following condition is feasible:

$$\begin{pmatrix} AQ + QA^T + Q & BY \\ Y^T B^T & -Q \end{pmatrix} < 0 \qquad \text{(A. 11)}$$

For discrete time systems the same ideas can be applied.

Appendix B

B. 1 Larsen's Modified Kalman filter

Consider the system and observations process model given in (5. 1) and (5. 14) respectively, with initial conditions as given in equation (5. 4) and jointly Gaussian initial state and noise sequences (process and measurements). Let $\hat{x}(k+1)$ represent the conditional mean of $x(k+1)$ given observations $\{v^{ex}(k)\}$ up to and including k. Then $\hat{x}(k+1)$ is modified by a correction factor $M(k)$ and is given by calculating the first two moments as follows:

$$\begin{bmatrix} \mu_x(k+1) \\ \mu_{v^*}(k) \end{bmatrix} = E\left\{ \begin{bmatrix} x(k+1)|F_{k-1} \\ v^*(k)|F_{k-1} \end{bmatrix} \right\}$$

$$= E\left\{ \begin{pmatrix} A_p & 0 & I & 0 \\ 0 & C_p & 0 & I \end{pmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \\ \xi(k) \\ \zeta^*(k) \end{bmatrix} + \begin{pmatrix} B_p \\ 0 \end{pmatrix} w(k) \right\}$$

$$= \begin{pmatrix} A_p \\ C_p \end{pmatrix} \hat{x}(k) + \begin{pmatrix} B_p \\ 0 \end{pmatrix} w(k)$$

$$\begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = E\left\{ \begin{bmatrix} x(k+1) - E\left[x(k+1)|F_{k-1}\right] \\ v^*(k) - E\left[v^*(k)|F_{k-1}\right] \end{bmatrix} \begin{bmatrix} x(k+1) - E\left[x(k+1)|F_{k-1}\right] \\ v^*(k) - E\left[v^*(k)|F_{k-1}\right] \end{bmatrix}^T \right\}$$

$$= \begin{bmatrix} A_p P(k) A_p^T + Q & A_p M(k) C_p^T \\ C_p M(k) A_p^T & C_p P\left(k - \tau_{SC}(k)\right) C_p^T + R \end{bmatrix}$$

There is a correlation between the $\tau_{SC}-$delayed estimated state error and the estimated error at time k. This correlation is represented by a correction factor $M(k)$. Then the recursion given by (5. 16) and (5. 17) immediately follows from the application of equations (5. 8) and (5. 9).

The correction factor $M(k)$ depends on the estimation error dynamics. Its calculation is presented as follows:

Let the correction factor be defined as follows: $M \equiv E\left\{\tilde{x}(k)\tilde{x}^T\left(k-\tau_{SC}(k)\right)\right\}$ then $\tau_{SC}-$*delayed* correlation is given by:

$$M(k) = M_*(k)P\left(k-\tau_{SC}(k)\right) \tag{B. 1}$$

$$M_*(k) = \prod_{i=0}^{\tau_{SC}(k)-1}\left[A_p - K(k-i)C_p\right] \tag{B. 2}$$

The proof is given using equation (5. 5) and knowing that the state error at time k is $\tilde{x}(k) \equiv x(k) - \hat{x}(k)$, state error $\tilde{x}(k+1)$ at time $k+1$ will be:

$$\tilde{x}(k+1|k) = \left[A_p - K(k)C_p\right]\tilde{x}(k) + K(k)\zeta(k) + \xi(k)$$

then from $k-\tau_{SC}(k)$ to k, through τ_{SC} successive time steps, the estimation error becomes [51]:

$$\tilde{x}(k) = \prod_{i=0}^{\tau_{SC}(k)-1}\left[A_p - K(k-i)C_p\right]\tilde{x}\left(k-\tau_{SC}(k)\right) + \sum_{i=0}^{\tau_{SC}(k)-1}\left[\xi(k-i) - K(k-i)\zeta(k-i)\right] \tag{B. 3}$$

The noise sequences $\xi(k)$ and $\zeta(k)$ are not correlated with the state $x(k)$ and consequently the correction factor can be calculated as follows:

$$M(k) \equiv E\left\{\tilde{x}(k)\tilde{x}^T\left(k-\tau_{SC}(k)\right)\right\} = E\left\{\left(\prod_{i=0}^{\tau_{SC}(k)-1}\left[A_p - K(k-i)C_p\right]\tilde{x}\left(k-\tau_{SC}(k)\right)\right)\tilde{x}\left(k-\tau_{SC}(k)\right)^T\right\}$$

This is the $\tau_{SC}-$*delayed* correlation.

B. 2 Proof of solution of the optimal control problem

by applying the extremun principle and its conditions of optimality. First define the Hamiltonian as follows:

$$H(k)=\frac{1}{2}\left\{\left\langle z(k),Qz(k)\right\rangle+\left\langle w(k),Rw(k)\right\rangle\right\}+\left\langle p(k+1),A_p z(k)+\left(A_p^{\bar{\tau}_{CA}}B_p\right)w(k)\right\rangle \tag{B. 4}$$

Determine the necessary conditions for optimality:

$$\frac{\partial H(k)}{\partial z(k)}=p(k)=Qz(k)+A^T p(k+1) \tag{B. 5}$$

$$\frac{\partial H(k)}{\partial p(k+1)}=z(k+1)=A_p z(k)+\left(A_p^{\tau_{CA}}B_p\right)w(k) \tag{B. 6}$$

Determine the transversality condition:

$$\frac{\partial G(N)}{\partial z(N)}=p(N)=Sz(N) \tag{B. 7}$$

Determine the optimal control law:

$$\frac{\partial H(k)}{\partial w(k)}=0=Rw(k)+\left(A_p^{\bar{\tau}_{CA}}B_p\right)^T p(k+1)$$

$$w(k)=-R^{-1}\left(A_p^{\bar{\tau}_{CA}}B_p\right)^T p(k+1) \tag{B. 8}$$

Substituting equation (B. 8) into (B. 6) and(B. 5), and assuming an inductive solution as follows:

$$p(k)=K(k)z(k) \tag{B. 9}$$

equations (5.25) to (5.27) can be proved.

B. 3 **Solution of the predictive optimal control problem**

By the principle of optimality, there is an optimal control that satisfies the following functional equation [31]:

$$V_k = \min_{\Delta w_k} E\left\{ l_k + V_{k+1} \middle| F_{k-1} \right\} \tag{B. 10}$$

Expanding equation (B.7) over the interval $k \in \left[k, k+N_C\right]$, leads to a dynamic programming problem of the form:

$$V_{k+1} = \min_{\Delta w_{k+1}} E\left\{ \min_{\Delta w_{k+2}} E\left\{ \ldots l_{k+H_C} + \sum_{j=1}^{H_C-1} l_{k+j} \ldots \middle| F_{k+1} \right\} \middle| F_k \right\} \tag{B. 11}$$

with:

$$l_{k+H_C} = Z_{k+H_C}^T S Z_{k+H_C}$$

$$l_{k+j} = Z_{k+j}^T Q_z Z_{k+j} + \Delta w_{k+j}^T Q_w \Delta w_{k+j}$$

Boundary conditions for equation (B.8) can be found at time $k+H_C$ as follows:

$$\begin{aligned} V_{k+H_C} &= \min_{w_{k+H_C}} E\left\{ Z_{k+H_C}^T S_Z Z_{k+H_C} \middle| F_{k+H_C-1} \right\} \\ &= Z_{k+H_C}^T S_x Z_{k+H_C} + trace\left(S_z P_{k+H_C} \right) \end{aligned}$$

By induction, the solution of V_{k+j} will be given by:

$$V_{k+j} = Z_{k+H_C}^T \Phi_{k+j} Z_{k+H_C} + \phi_{k+j}$$

with Φ_{k+j} symmetric definite, consequently at time k :

$$V_k = \min_{\Delta w_k} E\left\{ \left(Z_k^T Q_z Z_k + \Delta w_k^T Q_w \Delta w_k + Z_{k+1}^T \Phi_{k+1} Z_{k+1} + \phi_{k+1} \right) \middle| F_{k-1} \right\}$$

$$V_k = \min_{\Delta w_k} \left\{ \left(Z_k{}^T Q_Z Z_k + \text{trace}\left(A P_k \right) + \Delta w_k{}^T Q_u \Delta w_k + \left(A Z_k + B \Delta w_k \right)^T \Phi_{k+1} \left(A Z_k + B \Delta w_k \right) + \phi_{k+1} \right) \right\} \tag{B. 12}$$

Differentiating equation (B. 12) with respect to Δw_k gives equations (5.29) to (5.31) and completes the proof.