DYNAMICS ANALYSIS AND INTEGRATED DESIGN OF REAL-TIME CONTROL SYSTEMS

A THESIS SUBMITTED TO

THE SCHOOL OF ELECTRICAL AND INFORMATION ENGINEERING

OF THE UNIVERSITY OF SYDNEY

IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Yu-Chu Tian

School of Electrical and Information Engineering

The University of Sydney

December 2008

DYNAMICS ANALYSIS AND INTEGRATED DESIGN OF REAL-TIME CONTROL SYSTEMS

A THESIS SUBMITTED TO
THE SCHOOL OF ELECTRICAL AND INFORMATION ENGINEERING
OF THE UNIVERSITY OF SYDNEY
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Yu-Chu Tian

Computer and Software Engineering
School of Electrical and Information Engineering
The University of Sydney

December 2008

DYNAMICS ANALYSIS AND INTEGRATED DESIGN OF REAL-TIME CONTROL SYSTEMS

Yu-Chu Tian

Email: yctian@ieee.org

Computer and Software Engineering

School of Electrical and Information Engineering

The University of Sydney

Copyright in Relation to This Thesis

Author's Statement of Originality

This thesis is my original work and has not been submitted, in whole or in part, for a degree at this or any other university. Nor does it contain, to the best of my knowledge and belief, any material published or written by another person, except as acknowledged in the text.

Abstract

Real-time control systems are widely deployed in many applications. Theory and practice for the design and deployment of real-time control systems have evolved significantly. From the design perspective, control strategy development has been the focus of the research in the control community. In order to develop good control strategies, process modelling and analysis have been investigated for decades, and stability analysis and model-based control have been heavily studied in the literature. From the implementation perspective, real-time control systems require timeliness and predictable timing behaviour in addition to logical correctness, and a real-time control system may behave very differently with different software implementations of the control strategies on a digital controller, which typically has limited computing resources. Most current research activities on software implementations concentrate on various scheduling methodologies to ensure the schedulability of multiple control tasks in constrained environments. Recently, more and more real-time control systems are implemented over data networks, leading to increasing interest worldwide in the design and implementation of networked control systems (NCS). Major research activities in NCS include control-oriented and scheduling-oriented investigations.

In spite of significant progress in the research and development of real-time control systems, major difficulties exist in the state of the art. A key issue is the lack of integrated design for control development and its software implementation. For control design, the model-based control technique, the current focus of control research, does not work when a good process model is not available or is too complicated for control design. For control implementation on digital controllers running multiple tasks, the system schedulability is essential but is not enough; the ultimate objective of satisfactory quality-of-control (QoC) performance has not been addressed directly. For networked control, the majority of the control-oriented investigations are based on two unrealistic assumptions about the network induced delay. The scheduling-oriented research

focuses on schedulability and does not directly link to the overall QoC of the system. General solutions with direct QoC consideration from the network perspective to the challenging problems of network delay and packet dropout in NCS have not been found in the literature.

This thesis addresses the design and implementation of real-time control systems with regard to dynamics analysis and integrated design. Three related areas have been investigated, namely control development for controllers, control implementation and scheduling on controllers, and real-time control in networked environments. Seven research problems are identified from these areas for investigation in this thesis, and accordingly seven major contributions have been claimed. Timing behaviour, quality of control, and integrated design for real-time control systems are highlighted throughout this thesis.

In control design, a model-free control technique, pattern predictive control, is developed for complex reactive distillation processes. Alleviating the requirement of accurate process models, the developed control technique integrates pattern recognition, fuzzy logic, non-linear transformation, and predictive control into a unified framework to solve complex problems.

Characterising the QoC indirectly with control latency and jitter, scheduling strategies for multiple control tasks are proposed to minimise the latency and/or jitter. Also, a hierarchical, QoC driven, and event-triggering feedback scheduling architecture is developed with plug-ins of either the earliest-deadline-first or fixed priority scheduling. Linking to the QoC directly, the architecture minimises the use of computing resources without sacrifice of the system QoC. It considers the control requirements, but does not rely on the control design.

For real-time NCS, the dynamics of the network delay are analysed first, and the non-uniform distribution and multi-fractal nature of the delay are revealed. These results do not support two fundamental assumptions used in existing NCS literature. Then, considering the control requirements, solutions are provided to the challenging NCS problems from the network perspective. To compensate for the network delay, a real-time queuing protocol is developed to smooth out the time-varying delay and thus to achieve more predictable behaviour of packet transmissions. For control packet dropout, simple yet effective compensators are proposed. Finally, combining the queuing protocol, the packet loss compensation, the configuration of the worst-case communication delay, and the control design, an integrated design framework is developed for real-time NCS. With this framework, the network delay is limited to within a single control period, leading to simplified system analysis and improved QoC.

Acknowledgements

I would like to thank my supervisor, Associate Professor David Levy, for his continuous support and generosity at all times. Discussions with David are always interesting and constructive.

With its international reputation in research excellence and higher education leadership, the School of Electrical and Information Engineering at the University of Sydney has a creative and flexible research environment. This is invaluable in my research and is very much appreciated.

Most of the developments presented in this thesis have been published in peer reviewed journal papers. The contribution of numerous and anonymous reviewers is significant; and their detailed comments, compliments, and criticisms have added considerably to the completed work.

I am extremely grateful to my family for the understanding and continuous support, without which this work would have not become possible.

Preface

This thesis is submitted to the School of Electrical and Information Engineering of the University of Sydney in fulfilment of the requirements for the degree of Doctor of Philosophy. It was completed through part-time study starting from 2002 in the Computer and Software Engineering Group, School of Electrical and Information Engineering, the University of Sydney.

During this period of time, I conducted considerable research in broad areas of complex systems engineering, real-time computing, embedded systems, networked applications over wired and wireless networks, complex industrial computing, and control theory and engineering. Around sixty refereed papers have emanated from these research activities. Some of these publications have been highly cited by other researchers. All those have provided the evidence of my creative, productive, and quality research in a wide range of research areas.

As indicated in the title, I have chosen to present in this thesis my research outcomes related to Real-Time Control Systems with respect to Dynamics Analysis and Integrated Design in multitasking and networked environments. In this area, nine papers have been written in support of this thesis. These publications are summarised in Chapter 1 Subsection 1.7.1 (My Publications in Support of This Thesis).

For complete information about my recent research activities and outcomes, all other refereed publications during my PhD study which are not listed in Chapter 1 Subsection 1.7.1 (My Publications in Support of This Thesis) are summarised in Appendix, including four first-authored papers and other forty-seven publications. This indicates that the thesis presented here reflects only a small fraction of my recent research activities.

The information presented in this thesis is current as at 08 September 2008. New reports on the advances in dynamics analysis and integrated design of real-time systems after this cut-off date have not been reviewed or referenced in this thesis.

This thesis is typeset using LaTeX 2_ε with the document class 'report'. Some formatting specifications are described below:

- The typesetting options chosen for this thesis include 12pt (for font size), twoside (for double sided printing), a4paper, and openright.
- LaTeX package 'setspace' is used to control the spacing of the text, and the spacing is set to be 1.5\baselineskip (i.e., one and half spacing).
- The images included in this thesis are all in eps format and are processed using LaTeX package 'graphicx'.
- The Table of Contents, List of Tables, and List of Figures are automatically generated by LaTeX system. In the body text, all chapter and section headings are in title style, whereas all table and figure captions are in sentence style. Accordingly, all entries in the Table of Contents are in title style, while all entries in the List of Tables and List of Figures are in sentence style.
- References are typeset using BibTeX style file 'named', which automatically generates a citation index for each bibliography entry in one of the following author-year styles: [author-last-name, year], or [author-last-name and author-last-name, year], or [author-last-name *et al.*, year]. If several bibliography entries have the same author index and are in the same year, they are differentiated in alphabets a, b, c, etc., following the year.
- A package is created to define the layout of this thesis. It renews a number of LaTeX commands and environments, including \maketitle, \chapter, \section, \subsection, etc. It also sets the page margins, text height, text width, and a few other parameters.

The completion of this PhD study is a significant milestone in my career. It is also a new start to tackle new challenging problems. Look ever forward!

Table of Contents

Nomenclature

Abbreviations

CAN	Controller Area Network
DM	Deadline-Monotonic (Scheduling)
EDF	Earliest-Deadline-First (Scheduling)
ETBE	Ethyl *Tert*-Buty Ether
FP	Fixed Priority
IAE	Integral of Absolute Error
IP	Internet Protocol
ITAE	Integral of Time Absolute Error
LCT	Least-Compute-Time (Scheduling)
QoC	Quality of Control
QoS	Quality of Service
PD	Proportional plus derivative (Control/Controller)
PD2	Proportional plus up to the Second-Order Derivative (Compensation)
PD3	Proportional plus up to the Third-Order Derivative (Compensation)
PI	Proportional-Integral (Control/Controller)
PID	Proportional-Integral-Derivative (Control/Controller)
PP	Pattern Predictor/Predictive
PPC	Pattern Predictive Control
RD	Reactive Distillation
RM	Rate-Monotonic (Scheduling)

NCS	Networked Control System/s
SSE	Sum of Square Error
TCP	Transport Control Protocol
WCCD	Worst-Case Communication Delay

Symbols		**Chapter**
A1 $\sim$ Am	Smart actuators	Ch8,9
$A_x^i \in \{B, M, S\}$	Linguistic value of FS_x for the *i*th rule, $x \in \{\frac{T}{\theta}, S_4, r_3\}$	Ch3,4
B	Bottoms flow rate (L/min), or the linguistic term “Big”	Ch3,4
b	Constant in the non-linear transformation	Ch3,4
C1 $\sim$ C5	Control computers	Ch8,9
C_q	Specific heat	Ch8
c	Constant in the non-linear transformation	Ch3,4
	Worst case computation time	Ch5,6,7
D	Distillate flow rate (L/min)	Ch3,4
D_q	Fractal dimension	Ch8
d	Number of prediction steps	Ch3,4
	Deadline	Ch5,6,7
e	Control error, $e = r - y_d$	Ch1
F_c	Feed composition (stoichiometric ratio)	Ch3,4
F_f	Feed flow rate (L/min)	Ch3,4
FS_x	Fuzzy set of the subscript variable $x \in \{T/\theta, S_4, r_3\}$	Ch3,4
F_t	Frequency of time series T_t	Ch8
f	Non-linear transformation within the PPC system	Ch3,4
$f(\cdot)$	Function	Ch10
G	Controller or its transfer function	Ch6,7
G_c	Controller or its transfer function	Ch3,4,10

G_L	Process transfer function in disturbance path	Ch3,4
G_p	Process transfer function in manipulation path	Ch3,4,10
g	Intermediate function, at steady state $y = g(u)$	Ch3,4
H	Hurst exponent	Ch8
h	Updating factor for process prediction	Ch3,4
K	Gain	Ch6,7
K_c	Controller gain	Ch3,4,10
$\bar{K}_c$	Equivalent controller gain	Ch10
K_p	Process gain	Ch3,4,10
K_{u1}, K_{u2}, K_{u3}	Proportional coefficients	Ch10
k	Present sampling instant	Ch3,4
	Integer	Ch8
	Integer representing the kth control period	Ch10
L	Reflux flow rate (L/min) or Load	Ch3,4
lr	Loss rate	Ch10
M	The linguistic term “Medium”	Ch3,4
M1 $\sim$ M5	Management computers	Ch8,9
m	Integer variable, or the number of smart actuators	Ch8,9
N	Number of requests for scaling down periods	Ch6,7
	Length of sequence	Ch8
n	Total number of tasks	Ch5,6,7
	The number of data, or the number of smart sensors	Ch8,9
	Integer	Ch10
O	Offset	Ch5
p	Period	Ch5,6,7
$p_1 \sim p_5$	Parameters for non-linear transformation f	Ch3,4
Q_1	Queue to store one control packet	Ch9,10,11

Q_2	First-in-first-out (FIFO) queue of past control packets	Ch9,10
Q_c	Condenser duty (kW)	Ch3,4
Q_r	Reboiler duty (kW)	Ch3,4
q	Real number	Ch8
R	One-dimensional Euclidean space	Ch8
R	Set-point	Ch3,4
$R(n)$	$R(n) = \max_{1 \le i \le n} X(i,n) - \min_{1 \le i \le n} X(i,n)$	Ch8
r	Set-point of the plant output	Ch1
$r_1 \sim r_3$	Parameters for prediction of the controlled variable	Ch3,4
S	The linguistic term “Small”	Ch3,4
$S_1 \sim S_4$	Process feature patterns	Ch3,4
S1 $\sim$ Sn	Smart sensors	Ch8,9
$S(n)$	$S(n) = \left[\frac{1}{n}\sum_{i=1}^{n}(x_i - < x >_n)^2\right]^{1/2}$	Ch8
s	Laplace transform operator	6,7,10
T	Process time constant	Ch3,4
	Task	Ch5,6,7
	Control period	Ch9,10,11
T^I	Subtask set $T^I = \cup_{i=1}^{n} T_{i1}$	Ch6,7
T^{II}	Subtask set $T^{II} = \cup_{i=1}^{n} T_{i2}$	Ch6,7
T_7	Stage 7 temperature (°C)	Ch3,4
T_C	Time instant to deal with control packet dropout	Ch9,11
T_c	Control integral time	Ch6,7
T_D	Time instant to dequeue Q_1	Ch9,10,11
$\bar{T}_d$	Equivalent derivative time	Ch10
T_E	Earliest time instant to receive control packet	Ch9
T_i	Controller integral time (min)	Ch3,4,10
$\bar{T}_i$	Equivalent integral time	Ch10

T_O	Time instant at which a control period commences	Ch9
T_L	Latest time instant to accept control packet	Ch9,10
T_p	Time constant for plant	Ch6,7,10
T_s	Sampling period (min)	Ch3,4
T_t	Time series	Ch8
t	Integer, $t = 1, 2, \cdots, N$	Ch8
	Time variable	Ch10
t_{ca}	Controller-to-actuator delay	Ch9
t_{ctr}	Control computation time	Ch9
t_j	Sum of sensor-to-controller-to-actuator jitters	Ch9
t_{sc}	Sensor-to-controller delay	Ch9
U	CPU utilization	Ch5,6,7
U^I	Workload of the subtask T^I	Ch6,7
U^{II}	workload of the subtask T^{II}	Ch6,7
u	Manipulated variable (e.g., reboiler duty)	Ch1,3,4,10
	Integer	Ch8
$\hat{u}$	The estimate of the control signal u	Ch10
u_d	The digitised form of u	Ch1
V	Boilup flow rate (L/min)	Ch3,4
v	Transformed manipulated variable in PPC	Ch3,4
$V\left(A^i_{r_3}\right)$	Central numerical values of $A^i_{r_3} \in \{B, M, S\}$	Ch3,4
V_T	Column top vapour flow rate (L/min)	Ch3,4
wrt	Worst response time	Ch5
$X(i,n)$	$X(i,n) = \sum_{u=1}^{i} [x_u - < x >_n]$	Ch8
x	Set of a sequence of data	Ch8
x_i, x_k, x_u	The i-, k- and kth data in a sequence of data	Ch8
$< x >_n$	Mathematical expectation for n sequence data	Ch8

$Y(x)$	Used for defining μ, $d\mu(x) = Y(x)dx$	Ch8
y	Controlled variable (e.g., stage 7 temperature)	Ch1,3,4,6,7,10
$\hat{y}$	d steps ahead prediction of y	Ch3,4
$\tilde{y}$	Control error	Ch6,7
y_m	Measurement of the plant output y	Ch1
y_d	The digitised form of the measurement y_m	Ch1
y_r	Set-point of the controlled variable y	Ch10
Z	Partition sum	Ch8

Greek Letters

β_i	Degree of fulfilment for the *i*th rule	Ch3,4
Δ_H	QoC upper bound	Ch6,7
Δ_L	Dead zone around $\delta\tilde{y}$. $\Delta_L < \Delta_H$	Ch6,7
Δy_j	Control error $(= y_r - y_j)$	Ch10
$\Delta\hat{y}$	d steps ahead prediction of the y deviation	Ch3,4
$\delta\tilde{y}$	One-step difference of $tildey$	Ch6,7
ϵ	Small and positive threshold in S_2	Ch3,4
	Forgetting factor	Ch6,7
	Side length of a non-empty box	Ch8
μ	Membership function	Ch3,4
	A measure on interval $[0, 1]$	Ch8
τ	Time delay	Ch6,7
	Scaling exponent	Ch8
τ_c	Sum of network delay and control computation delay	Ch10
τ_p	Process time delay	Ch10
θ	Process time delay (min)	Ch3,4

Superscripts

(i)	The first-, second, or third-order derivative ($i = 1, 2$, or 3)	Ch10
max	Maximum value or upper bound (for p, p_i and N)	Ch6,7
min	Minimum value or lower bound (for p and p_i)	Ch6,7

Subscripts

c	Controller	Ch6,7
i	The first or sole subscript to some variables (c, d, O, p, T, wrt)	
	to indicate the ith task	Ch5,6,7
j	The second subscript to some variables (c, d, O, T, wrt)	
	to indicate the jth subtask decomposed from a task	Ch5,6,7
	Integers representing the jth control period	Ch10
k	Integers representing the kth control period	Ch10
p	Plant	Ch6,7
rq	Requests (for N)	Ch6,7
sp	Set-point for U and y	Ch6,7

List of Figures

List of Tables

Chapter 1

Introduction

This thesis focuses on dynamics analysis and integrated design of real-time control systems. While real-time control systems is a very broad topic, three related areas will be addressed for investigation into various aspects of dynamics analysis and integrated design of a control system: control development, control task scheduling, and networked control.

We start with a brief introduction into the background of the research in Section 1.1 followed by discussions of the motivation in Section 1.2. Section 1.3 identifies seven research problems for investigation. The significance of the research is described in Section 1.4. The main contributions are highlighted in Section 1.5. Section 1.6 outlines the structure of the thesis. Finally, a list of refereed publications is shown in Section 1.7 in support of this thesis.

1.1 Background

Real-time control systems are widely deployed in both military and civilian applications. Typical examples are modern manufacturing, oil refinery, mineral processing, chemical process control, pharmaceutical plant operation, transport management and scheduling, aircraft control, satellite and missile guidance, to just mention a few. Looking around in our homes, we also find that real-time control systems are being increasingly integrated into our daily life, in items such as DVD player, microwave oven, mobile phone, fridge, washing machine, and many others.

A simple control system typically consists of a plant to be controlled, a sensor to measure the plant variable, a digital controller to implement control strategies, and an actuator to impose

the control to the plant. It is shown graphically in Figure 1.1, in which y and y_m are plant output and its measurement, respectively; r is the set-point of y; u stands for control signal; $e = r - y_d$ is control error; and y_d and u_d are digitised forms of y_m and u, respectively. Such a simple control system is a building block for more complex control systems. In modern networked control, bus-type communication networks are employed to interconnect various control system components, and typically the links for feedback (to the controller) and control output (from the controller) become part of the networks, as depicted in Figure 1.2.

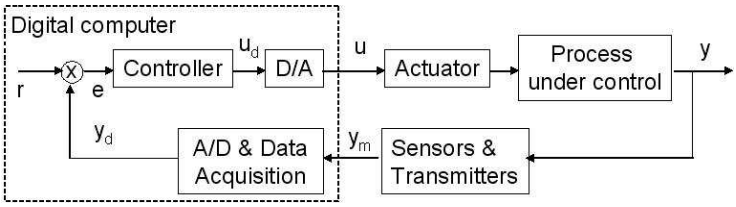

Figure 1.1: A general digital computer control system.

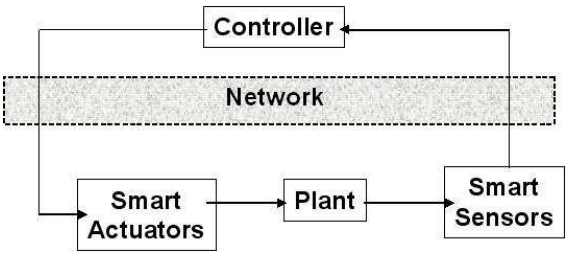

Figure 1.2: Block diagram of networked control systems.

Real-time control systems are a type of real-time systems for control applications. They require timeliness of the computational results in addition to the logical correctness of the computation. There are basically two categories of real-time tasks: soft and hard:

- Soft real-time tasks are those which if completed after their respective deadlines are less important than those whose deadlines have not yet expired.
- Hard real-time tasks are ones which must meet their hard deadlines. Tasks completed after their deadline are at best valueless and at worst harmful.

Some systems contain only soft real-time tasks. Many systems, however, contain a mixture of hard and soft real-time tasks [Sprunt *et al.*, 1989].

Building a real-time control system requires two phases in general: controller design and its digital implementation. Those two phases have been traditionally addressed separately by

different groups of engineers/scientists. The controller design is carried out from the control theory point of view, while the digital implementation is conducted from the real-time computing perspective. Both the fields of real-time computing and control theory are vast and have attracted large groups of researchers and engineering practitioners.

However, recent research and engineering practice have revealed that the separation of these two design phases will lead to some major difficulties in improving the performance of modern large-scale real-time control systems. This becomes more evident with the integration of networks into control systems. Therefore, much effort is being made to fit these two phases into a unified framework for integrated design. New theories and methodologies are being developed in this direction. This is an indication of a new trend of real-time control systems.

Following this trend, this thesis focuses on dynamics analysis and integrated design of real-time control systems. Dynamics analysis gives insight into, and a good understanding of, the system behaviour, requirements, and constraints; and integrated design makes the best use of system resources for improvement of the performance of the overall system.

Three related areas are addressed in this thesis for investigation into dynamics analysis and integrated design, i.e.,

- Control strategy development – This is about control design for controllers;
- Control task scheduling – This relates to implementation of controllers; and
- Networked control development – This is about control design and implementation in networked environments.

The overall aim of the research is to provide effective solutions to some challenging problems in real-time control systems.

1.2 Motivation of the Research

1.2.1 Control Strategy Development

Our discussions about development of control strategies are carried out in different categories depending on how complex the control processes are.

Most industrial processes can be well controlled using the proportional-integral-derivative (PID) control algorithm. For those processes, it does not really make sense to design more advanced and sophisticated control strategies from the application point of view although some theoretical investigations are still worthwhile. This thesis does not address the control design for this category of industrial processes.

Some processes have complex dynamics and/or highly coupled variables. Simple control strategies such as PID are ineffective. However, the variables to be controlled can be measured easily and reliably, and a good process model can be built. Thus, advanced and model-based control can be developed for the processes. Developing model-based control still attracts significant interest; however, this thesis does not address the development of model-based control.

There are some processes which can be directly controlled by neither PID nor advanced control. A typical example is the product composition in reactive distillation (RD). The main reasons are the lack of reliable and cost-effective sensors to measure the variable, the lack of a good understanding of the process dynamics, and the lack of good process models on which model-based control relies. Even if a reasonably good process model is available, the model may be too complicated for either online computation or controller design. Therefore, major difficulties still exist for the control of these processes. This thesis will address this category of industrial processes through integrated design of model-free control strategies.

For the plant variables that cannot be directly controlled due to the lack of reliable measurements, indirect control using alternative variables is usually deployed in the process operation to simplify the complexity of the control problems. Inferential control is one of such indirect control methodologies. For example, the product composition of RD processes can be indirectly controlled by maintaining some temperature and pressure variables at their respect desired values [Tian and Tadé, 2000]. Inferential control and soft sensing have been developed using various methods, e.g., the genetic algorithm [Kordon *et al.*, 2003].

To facilitate our research on the difficult control problems, the complex and industrially significant RD processes for synthesis of ethyl *tert*-buty ether (ETBE) is investigated in this study. One of the main tasks in control of an ETBE RD process is to maintain the ETBE purity. As the ETBE purity cannot be reliably measured online and in real-time, it is controlled indirectly by regulating the column temperature. Because the desired ETBE purity (purity set-point) may be changed during the plant operation, the control system should have fast set-point

tracking ability (tracking problem). On the other hand, RD and other industrial processes always have various disturbances and uncertainties, which cannot be well modelled and predicted, resulting in a requirement of effective disturbance rejection (regulatory problem). Control design should consider both tracking and regulatory problems for RD processes.

There are basically two directions to deal with the complexity of RD control strategy design:

- One is to develop better understanding of the processes so that more accurate yet simple models can be obtained for model-based control design. However, the available models are still far away from practical applications. Robust control that deals with uncertainties and poor modelling can be classified into this direction.
- The other is to develop model-free control to alleviate the requirement of accurate process models. Progress is being made in this direction for practical operation of RD processes.

This thesis develops model-free control strategies for complex RD processes through dynamics analysis and integrated design of various aspects of pattern recognition, fuzzy logic, non-linear transformation, and predictive control.

1.2.2 Control Task Scheduling

Conventional control theory has inherently assumed that the digital controller does not consume any time to compute the control signals. This means that an instant control signal is assumed to be available from the controller once a process measurement is fed back to the controller. This is an unrealistic assumption because a control task not only needs time to compute the control but also shares the processor with other tasks.

Control computation does consume time, and the time consumption is even significant in some control systems. For example, in a large-scale control system that we developed [Tian and Tadé, 2002], the control computation takes about 1s in a 4s control period, implying that a quarter of the period is spent for control computation. This type of time delay is relatively easy to handle because the control computation time does not fluctuate much under normal conditions (a big change in the computation time may be an indication of control mode switch or abnormal conditions and thus needs to be considered separately). Therefore, a fixed time delay with small variations may be used to represent the control computation. Then, this delay

can be compensated in the control design and implementation. This thesis addresses the control computation delay indirectly by lumping it and the process delay together.

A control task also needs to compete for time slices with other tasks running on the same processor. Most real-time control applications have the nature of embedded systems. They are typically deployed in hardware platforms with limited resources, e.g., uni-processor, limited memory, constrained input/output, and restricted file operations. Also typically, a controller is responsible for multiple control tasks, in addition to many non-control tasks. This leads to various constraints to computing and scheduling of real-time control tasks. One of the constraints is multi-tasking that shares the uni-processor for multiple control loops. Multi-tasking of the controller makes the timing behaviour of a specific task more unpredictable.

Real-time scheduling theory has advanced significantly with the focus on the schedulability. Major real-time scheduling paradigms include Rate-Monotonic (RM) and Deadline-Monotonic (DM) scheduling for static environments, Earliest-Deadline-First (EDF) scheduling for dynamic environments with sufficient resources, Spring scheduling for dynamic environments with insufficient resources, and feedback scheduling for dynamic environments with uncertainties.

Schedulability is an essential requirement for all real-time systems including real-time control systems. However, schedulability analysis in existing scheduling strategies has not been well linked with the characteristics of real-time control systems. Schedulability is not enough for real-time control systems; we also expect good control performance!

The control performance can be addressed in control task scheduling from the following two perspectives, both of which will be investigated in this thesis through integrated design:

- Use the time delay and jitter resulting from multi-tasking scheduling to indirectly characterise the control performance, and then minimise the delay and jitter in the scheduling.
- Link the scheduling directly with the Quality of Control (QoC), and then schedule system resources dynamically through feedback to fulfil the overall QoC requirement.

It is also worth mentioning that an optimal scheduling method may be invalid in a specific application. This is because the operating system or programming environment of the application may not directly support the scheduling method. For example, the V+ programming environment for robot control supports priority-driven scheduling at the application level, but

does not directly support the EDF policy, which carries more overhead that increases its instability in overload conditions. The real-time and object-oriented programming language Ada also supports priority-based multi-tasking scheduling at the application level.

1.2.3 Networked Control Development

Recent advances in networking pervasive devices and systems have enabled a huge number of networked applications [Estrin *et al.*, 2002]. As an important class of such networked applications, networked control systems (NCS) implement control functionality over data communication networks, which are used to transmit measurement, control, and management signals.

Due to their scalability, flexibility and cost-effectiveness, bus-type network technologies have been promoted for real-time control applications. There has been an increasing demand for real-time networked control in manufacturing automation, industrial process control, robotics, and many other applications. This requires highly reliable, flexible, simple, and cost-effective network technologies to replace traditional peer-to-peer interconnection techniques.

While networked systems and applications are not a new concept, systematic investigations into the interactions among network components and the complex dynamics of networked systems did not occur until recently. With much effort in this area, some basic concepts and approaches of complex networks have been developed to describe the connectivity, structure, and dynamics of complex systems. It has been realised that networked environments introduce challenges to system analysis and development, and the distributed nature of many system components and services requires new technologies to guide the system design.

Introducing networks into real-time control poses challenging problems: time-varying network induced delay and packet dropout. These challenges become more evident and severer when wireless networks are employed and/or scheduling of computing and network resources is considered in the networked control. Because of these challenging problems, many existing control technologies may become infeasible for specific networked control applications.

Research is undertaken to address those challenges in two extremes. Control engineers have been focusing on the controller design, and expect to solve the problems through improved design of sophisticated control strategies. Quite often, they have to use complex algorithms to deal with a simple network problem which has been considered as a constraint. Overall,

approaches developed from the perspective of the controller design provide stability conditions, but do not directly deal with the real-time requirement of the NCS network communications. Thus, conservative control is usually designed to guarantee the stability of an NCS.

The other extreme is from computer and networking engineers. Conventionally, "throwing bandwidth" was an effective method to resolve many problems in networking. Thus, some computer/networking engineers are optimistic that over-provisioning of network capacity can resolve the challenging problems of both network delay and packet dropout. It has not been well recognised that a high bandwidth does not necessarily mean predictable communication behaviour, and real-time control does not necessarily require fast data transmission. Over-provisioning may work for some applications, but is not a general solution.

It is our belief that any issues arising from a system should be first addressed locally. For networked control, network issues such as network induced delay and packet dropout should be first addressed in network design, rather than in controller design. In the same token, problems that can be easily solved in the controller design, such as compensation for a fixed network delay, should not be brought to the network area. Such a design philosophy requires a good understanding of control, scheduling, and networking for NCS design.

From the perspective of systems engineering, it is also our belief that a good system architecture design with simple algorithms is preferable to sophisticated algorithm design for poor system architecture. In networked control, an integrated design of network, scheduling, and control will provide a solution that can maximise the overall QoC of the NCS.

1.3 Research Problems

Following the above discussions, seven research problems of dynamics analysis and integrated design are identified from the three areas of control development, task scheduling, and networked control for further investigation in this thesis. These problems are summarised in the following:

Control Strategy Development:

Problem 1. Pattern predictive control (PPC) of complex RD processes. This relates to analysis of process dynamics, and integrated design of model-free control.

Control Task Scheduling:

Problem 2. Reducing control latency and jitter in real-time control. This relates to real-time scheduling of control tasks in static environments. The control latency and jitter are used to indirectly characterise the QoC of the control systems.

Problem 3. Hierarchical feedback scheduling of real-time control tasks. This relates to real-time scheduling of control tasks in dynamic and resource-insufficient environments. The scheduling is directly linked with the QoC of the control systems.

Networked Control Development:

Problem 4. Analysis of the dynamics of NCS network induced delay. This aims at a better understanding of the network delay dynamics for improved system design and stability criteria. The dimensional and fractal features of the dynamics of the network induced delay in real-time networked control will be investigated.

Problem 5. Real-time queuing protocols for predictable NCS network timing behaviour. This relates to co-design of networks and control, and aims to smooth out time-varying and unpredictable network induced delay in networked control.

Problem 6. Compensating for packet dropout in networked control. This also relates to co-design of data networks and control systems. The aim is to predict lost control packets over the NCS data networks.

Problem 7. Integrated design of networked control systems. This integrates the results from the above Problems 4 to 6 into a unified framework. It aims to provide more predictable network timing behaviour, which is favourable to real-time systems.

1.4 Significance of the Research

1.4.1 Pattern Predictive Control of RD Processes

Reactive distillation is becoming increasingly significant in industry [Taylor and Krishna, 2000; Malone and Doherty, 2000]. It has demonstrated potential improvements in capital productivity and selectivity, and reduction in solvents, energy consumption, and capital investments. Some

improvements are dramatic, as described in [Taylor and Krishna, 2000] for a methyl acetate RD process: 5 times lower investment and 5 times lower in energy consumption by using a hybrid RD process to replace an entire flowsheet with 11 units and related processes!

RD is advantageous for many chemical syntheses, including fuel ether production. As fuel additives, fuel ethers are widely used for improvement of fuel quality, and thus for reduction of greenhouse gas emissions. The significance of reducing greenhouse gas emissions is highlighted by the fact that 182 parties have ratified the Kyoto Protocol [Wikipedia, 2008].

RD is feasible for production of ethyl *tert*-butyl ether (ETBE), a high-performance fuel additive. However, the RD of ETBE is still limited worldwide and thus needs further research to make it technically reliable and economically viable. Moreover, RD involves considerable uncertainties and displays complex behaviour such as high non-linearity, strong interactions, bifurcation and multiplicity, and time delay. Due to the functional integration of reaction and separation, and the dynamics complexity, RD is exceptionally difficult to operate and control. This calls for systematic research on RD process dynamics and control aspects.

The difficulties in RD operation and control can be largely reduced by developing innovative control schemes that do not require accurate process models. Pattern predictive control is one of such control schemes. Our investigations show that it is not only simple and easy to design, but also effective for complex RD processes. The basic principle and design procedures of the pattern predictive control are also applicable in other complex industrial processes.

1.4.2 Real-Time Task Scheduling

When multiple real-time control tasks run on a uni-processor controller, the control systems need to meet the following two requirements in addition to logical correctness in each of the control tasks: schedulability and satisfactory timing behaviour of the control tasks.

The schedulability issue can be handled using various scheduling strategies, e.g., RM, DM, and EDF. The timing behaviour of the control tasks depends on what scheduling strategies are used, how they are implemented, and what task models are adopted. Nevertheless, both the schedulability and timing behaviour issues relate to time consumption and time delay in control computation and scheduling. This has not been well addressed in control software design.

Reducing Control Latency and Jitter in Real-Time Control

Time delay, which is also called latency in the computer area, usually results in rich dynamics, and is a source of system instability. It is one of the key problems that multi-tasking on a controller brings into the control systems. Especially, due to the limited resources of the controller, control latency becomes more complex and control jitter appears. While time delay existing in industrial processes has been largely addressed in process modelling and control design, the issue of latency and jitter resulting from control computation and multi-tasking scheduling has not been well understood in the control community [Kim and Park, 2003]. It remains unclear how the performance of the control software affect the performance of the overall control systems [Sanz and Alonso, 2001]. We contend that without a systematic performance analysis of the digital controller as part of the overall control system, how the performance of the overall real-time control system can be analysed and guaranteed [Tian and Sun, 2005].

This thesis will address this issue through developing effective scheduling methods and control task models to reduce control latency and jitter resulting from multi-tasking of real-time systems. This highlights the importance of the integration of control design and its software implementation explicitly, and provides solutions for the integration.

Hierarchical Feedback Scheduling of Real-Time Control Tasks

In many existing feedback scheduling methods for control systems, a scheduler is normally implemented as a separate task. The scheduler is triggered periodically to evaluate the QoC and to re-schedule control periods if the processor utilisation deviates from the desired one. Scheduling the control periods is feasible to most control systems although in some cases it may not be possible to increase the periods of critical tasks as their periods may be determined by external hardware such as hardware timers.

However, in fixed priority (FP) scheduling, tasks with lower priorities may be blocked or significantly delayed when the processor is heavily loaded or overloaded. Using variable priority scheduling may give all control tasks a chance to run. It has been shown [Cervin, 2003] that using EDF scheduling, in overload conditions, all periodic tasks are executed as if they were running with larger periods, which are automatically scaled. However, this method is schedulability driven, and does not directly link to the QoC of the system. Also, EDF is an

optimal dynamic scheduling algorithm in resource-sufficient environments; but its performance degrades rapidly in resource-insufficient environments. Moreover, EDF carries a much larger processor overhead than priority-based scheduling, increasing its instability in overload conditions. Therefore, in order to ensure that the processor is not overloaded when using the method of [Cervin, 2003], a frequent evaluation of the processor utilisation has to be carried out.

On the other hand, when a control loop has reached its steady state, scaling down its period in existing scheduling methods when actual processor utilisation is lower than the desired one does not really make sense but wastes computing resources. One may simply disable the period scaling. But when and how? A systematic approach has not been found to deal with this issue.

To tackle these problems, this thesis links the scheduling directly to the QoC, and then develops a hierarchical feedback scheduling framework for real-time control systems.

1.4.3 Networked Control Systems

Modern large-scale manufacturing, process control, and other systems demand increasing integration of information, communication, and control. As a result, more and more real-time control systems are implemented over data communication networks.

While control over networks becomes increasingly attractive, challenging problems exist in NCS analysis and design. The challenges include time-varying network induced delay and packet dropout. Significant effort has been made to deal with these challenging problems. For example, a simultaneous description of both network induced delay and packet loss in a unified model is considered in [Yue *et al.*, 2004]. With this model framework, stability criteria, which relate to the upper delay bound that guarantees the stability of the overall NCS, are derived [Yue *et al.*, 2004; Peng and Tian, 2006; Peng *et al.*, 2007]. Controller synthesis is also carried out to determine the controller settings under the stability criteria.

Analysis of the Dynamics of NCS Network Induced Delay

In existing methods for modelling, analysing, and synthesising NCS, there are two inherent assumptions with regard to the network induced delay:

- The network induced delay is uniformly distributed in a certain range. Much of the

existing effort is to find the upper bound of the delay for stability guarantee.

- The dynamic behaviour of the network induced delay is purely random. Consequently, stochastic models have been used to describe the dynamics of the NCS.

However, direct evidence has not been presented in the open literature to support those fundamental assumptions. Therefore, it is a significant step to develop a better understanding of the dynamic behaviour of the network induced delay in NCS.

Real-Time Queuing Protocols for Predictable NCS Network Timing

A few NCS control methodologies have been developed [Tipsuwan and Chow, 2003]. Among those methodologies, the 'queuing methodology' [Luke and Ray, 1990; Luke and Ray, 1994; Chan and Özgüner, 1995] is the only one that considers the timing behaviour of network communications explicitly in the NCS development. However, this methodology depends crucially on the accuracy of the mathematical model of the plant to be controlled. A reference to report a successful application of the methodology has not been found in the open literature.

Queuing packets to smooth out network induced delay is not a new idea in multimedia. However, the requirement in multimedia applications is different from that in NCS control. For example, throughput is important in video streaming, but is not the focus in networked control.

Network induced delay affects the accuracy of timing-dependent computations and can significantly degrade the QoC of the overall real-time control systems. To use the idea of queuing packets in NCS, the dependence of the conventional queuing methodologies on process models needs to be eliminated or reduced. This indicates the significance of the investigation into new real-time queuing protocols in this thesis.

Compensation for Packet Dropout in Networked Control

The 'queuing methodology' presented in [Luke and Ray, 1990; Luke and Ray, 1994] and also in [Chan and Özgüner, 1995] has considered packet dropout compensation. However, it has tried to implement packet dropout compensators on the controller through sophisticated algorithm computation. Implementation of such compensators relies crucially on accurate plant models,

which may not be available for many industrial processes. Also, the complex compensation algorithms may not be feasible if implemented on actuators with limited computing power.

Modelling network induced delay and packet dropout in a unified framework provides a mechanism for improvement of NCS stability conditions [Yue *et al.*, 2004; Peng *et al.*, 2007; Peng and Tian, 2006]. This methodology has received increasing interest in the last few years for dealing with various NCS conditions, in particular the network induced delay and packet dropout, in a neat way. However, methods of this kind do not compensate for packet dropout at all. They treat packet dropout as a constraint to the system. As a result, the control design is normally conservative in order to maintain the system stability.

Packet dropout results from network traffic congestions and limited network reliability. In this case, the controller and/or actuator have to make decisions with incomplete information on how to control the system. While a few schemes have been proposed for packet dropout compensation, there is still a lack of simple yet effective methods for packet dropout compensation in NCS. This shows the significance of the research on this topic.

Integrated Design of Networked Control Systems

Networked control has been investigated from various aspects, e.g., stability analysis, stabilisation, NCS control design, NCS networking, scheduling of tasks and resources, and packet dropout compensation. However, there is still a lack of effective interactions between control design and its software implementation in networked environments. Under this circumstance, an optimal solution cannot be expected to maximise the overall QoC of the NCS.

To tackle this problem, various aspects of an NCS should be considered within a uniform framework. Integrated design is an effective approach to simplify the network behaviour and consequently to maximise the QoC. This means that NCS network design, task scheduling, and software implementation should be linked directly to the QoC requirement of the system. Similarly, the control design should also be carried out together with network planning, network QoS (Quality of Service) requirements, task scheduling, and their implementation. However, an effective approach for integrated NCS design that considers all important aspects of the NCS has not been reported in the open literature. This indicates the significance of the research on this topic.

1.5 Main Contributions

Providing effective solutions to the seven research problems identified in Section 1.3, the main contributions of this thesis are summarised below:

1) **A pattern based predictive control (PPC) scheme is proposed for complex industrial processes whose mathematical models are either difficult to obtain or too complex for controller design**. It alleviates the requirement of accurate process models, and extracts pattern information of the process for predictive control design. The scheme is verified through case studies of complex reactive distillation processes for ethyl *tert*-buty ether (ETBE) synthesis. [Chapter 3, Chapter 4], [Tian *et al.*, 2003].

2) **For multiple control tasks running on a uni-processor, three strategies are proposed to reduce control latency and/or jitter: introduction of offsets into control task scheduling, decomposition of control tasks; and increasing the priority levels of the output subtasks**. The strategies are shown to be effective through case studies. [Chapter 5], [Tian *et al.*, 2006b].

3) For a multi-task control system, existing scheduling methods cannot maintain satisfactory overall control performance when the system is overloaded. **A hierarchical and quality-of-control (QoC) driven hierarchical feedback scheduling architecture is proposed to solve this problem for real-time multiple control tasks**. Case studies have demonstrated the effectiveness of the scheduling architecture. [Chapter 6, Chapter 7]

4) For typical scenarios of real-time control systems over Ethernet based networks, detailed analysis of the dynamic behaviour of the network induced delay is carried out. **It is revealed that the network induced delay is non-uniformly distributed and exhibits multi-fractal nature**. Such properties of the network delay can be employed for improving system design as well as for simplifying system analysis. [Chapter 8], [Tian *et al.*, 2007; Tian and Levy, 2008b].

5) **To address the challenging problem of the network induced delay in networked control, a queuing architecture is developed to reduce the network induced jitter, making the network induced delay more predictable**. The proposed queuing protocol is verified through comprehensive case studies. [Chapter 9], [Tian and Levy, 2008b].

6) **To address the challenging problem of the data packet dropout in networked control, three model-free strategies are developed for control packet dropout compensation**. With the mechanism of packet dropout compensation together with the queuing protocol, the sum of the network induced delay and the control computation delay is limited to within a control period. [Chapter 10], [Tian and Levy, 2008a; Tian and Levy, 2008b].

7) **It is shown that the worst-case communication delay (WCCD) can be made configurable to significantly reduce the network delay if a certain level of packet loss rate can be tolerated. Combining the queuing protocol, the WCCD configuration, packet dropout compensation, and control design, a general framework is developed to deal with network complexity and integrated design of networked control systems**. [Chapter 11], [Tian and Levy, 2008b].

1.6 Thesis Outline

The thesis begins with this introductory chapter, in which seven research problems from three related areas are identified for investigation and the main contributions of this thesis are highlighted. Then, a comprehensive literature review is given in Chapter 2. Chapters 3 through to 11 are devoted to detailed research on the identified research problems, as shown in Table 1.1. Finally, Chapter 12 concludes the thesis.

1.7 Related Publications During my PhD Study

During my PhD study, around sixty refereed publications have emanated from my research in a wide range of areas. Nine of them are used in support of this thesis (Subsection 1.7.1); and all others not covered in this thesis are mentioned in Subsection 1.7.2 and are listed in Appendix.

1.7.1 My Publications in Support of This Thesis

(1) Five Refereed Journal Papers

[Tian *et al.*, 2003] **Yu-Chu Tian**, Futao Zhao, B. H. Bisowarno, and M. O. Tadé. Pattern-based predictive control for ETBE reactive distillation. *Journal of Process Control*, 13(1):57–67,

Table 1.1: Organisation of this thesis for identified research problems.

Area	Research Problem	Chapter
(1) - Process dynamics and integrated control design for controllers		
	Problem 1: Patter predictive control of RD processes	3 (Theory)
		4 (Verification)
(2) - Integrated control task scheduling on controllers		
	Problem 2: Reducing latency and jitter in control scheduling	5
	Problem 3: Feedback scheduling of real-time control tasks	6 (Theory)
		7 (Verification)
(3) - Dynamics and integrated design of networked control systems		
	Problem 4: Analysis of NCS network dynamics	8
	Problem 5: Real-time queuing protocols for NCS	9
	Problem 6: Compensating for NCS packet dropout	10
	Problem 7: Integrated design of networked control systems	11

Feb 2003.

[Tian *et al.*, 2006b] **Yu-Chu Tian**, Qing-Long Han, David Levy, and M. O. Tadé. Reducing control latency and jitter in real-time control. *Asian Journal of Control*, 8(1):72–75, Mar 2006.

[Tian *et al.*, 2007] **Yu-Chu Tian**, Zu-Guo Yu, and Colin Fidge. Multifractal nature of network induced time delay in networked control systems. *Physics Letters A*, 361(1-2):103–107, Jan 2007.

[Tian and Levy, 2008a] **Yu-Chu Tian** and David Levy. Compensation for control packet dropout in networked control systems. *Information Sciences*, 178(5):1263–1278, Mar 2008.

[Tian and Levy, 2008b] **Yu-Chu Tian** and David Levy. Dealing with network complexity in real-time networked control. *International Journal of Computer Mathematics*, 85(8):1235–1253, Aug 2008.

(2) Three Refereed Conference Papers

[Tian *et al.*, 2006a] **Yu-Chu Tian**, Qing-Long Han, Colin Fidge, Moses O. Tadé, and Tianlong Gu. Communication architecture design for real-time networked control systems. In

Proceedings of the 4th International Conference on Communications, Circuits and Systems ICCCAS'06, pages 1840–1845, Guilin, P. R. China, 25-28 June 2006.

[Tian *et al.*, 2006c] **Yu-Chu Tian**, David Levy, Moses O. Tadé, Tianlong Gu, and Colin Fidge. Queuing packets in communication networks networked control systems. In *Proceedings of the 6th World Congress on Intelligent Control and Automation WCICA'06*, pages 205–209, Dalian, P. R. China, 21-23 June 2006.

[Tian and Levy, 2007] **Yu-Chu Tian** and David Levy. Configuring the worst-case communication delay in real-time networked control systems. In H. R. Arabnia and L T. Yang, eds., *Proceedings of the 2007 International Conference on Embedded Systems and Applications ESA'07*, pages 114–120, Las Vegas Nevada, USA, 25-28 June 2007. CSREA Press.

(3) A Journal Paper Ready for Submission

[*] **Yu-Chu Tian**, David Levy, and Ashok Agrawala. Hierarchical feedback scheduling of real-time control tasks. To be submitted, 2009.

1.7.2 My Publications Not Covered in This Thesis

For complete information about my creative and productive research, a long list of all other refereed publications during my PhD study which are not listed above is given in Appendix: My Publications Not Covered in This Thesis.

1.8 Nomenclature of This Chapter

Abbreviations

DM	Deadline-Monotonic
EDF	Earliest-Deadline-First
ETBE	Ethyl *tert*-buty ether
FP	Fixed Priority
QoC	Quality of Control
QoS	Quality of Service

RD	Reactive Distillation
RM	Rate-Monotonic
NCS	Networked Control System/s
WCCD	Worst-Case Communication Delay

Symbols

e	Control error, $e = r - y_d$
r	Set-point of the plant output
u, u_d	Control signal and its digitised form, respectively
y	Plant output
y_m, y_d	Measurement of the plant output, and its digitised form, respectively

Chapter 2

Literature Review

Real-time systems are those in which timeliness is as important as logic correctness. Missing a deadline will result in a degradation of system performance for soft real-time systems or a system failure for hard real-time systems. The requirements for real-time system design have been extensively described in a large number of papers as well as in many books, e.g., [Shaw, 2001; Burns and Wellings, 2001; Samard and Balas, 2003; Lavagno *et al.*, 2003].

As a class of real-time systems, real-time and embedded control systems have been increasingly deployed in various applications. As outlined in Chapter 1, with the focus on dynamics analysis and integrated design, this thesis will address three related areas of real-time control:

- Control design for controllers: Well designed control strategies are essential for real-time control systems to provide the desired functionality. For complex processes that cannot be well handled using either simple Proportional-Integral-Derivative (PID) control or advanced model-based control, integrated design of model-free and intelligent control is an attractive way for process operation.
- Control implementation on controllers in multi-tasking environments: Real-time control systems are typically deployed in hardware platforms with limited resources, e.g., uni-processor, resulting in various constraints in computing and scheduling of real-time control tasks of the systems. While schedulability has been the focus of conventional scheduling theory, it is not enough for real-time control systems. We need good control performance as well! In order to meet the requirements of both control performance and multi-tasking schedulability, well-designed scheduling of multiple real-time control tasks

becomes critical for these systems.

- Control design and its implementation on controllers in networked environments: Challenges exist when a real-time control system in implemented over networks. A better understanding of the dynamics of networked control is necessary for further improvement of system performance. As separate designs of control, scheduling, and networks do not provide optimal solutions to a networked control system (NCS), an integrated design of various system components is thus crucial for maximising the system performance.

Real-time systems are a vast field, so are control systems. Computer networks, which are essential for networked control, are also a very broad area. This marks the interdisciplinary nature of this research.

Because of this nature, it is unrealistic to have an exhaustive review of all related areas in a single chapter. The literature review will be performed from the above-mentioned three areas from which seven research problems have been identified in Section 1.3 for further investigation in this thesis; and only representative and directly relevant works will be reviewed. Recent developments in control of complex reactive distillation (RD) processes will be discussed in Section 2.1. Relevant works on multi-tasking scheduling for real-time control systems are summarised in section 2.2. Section 2.3 describes pertinent advances in NCS.

2.1 Dynamics and Control of Complex Reactive Distillation Processes

This section reviews the developments in control of RD processes mainly in the last ten years. It will link to the main theme of this thesis: integrated design of real-time control systems.

2.1.1 Significance of RD Processes and Their Control

Process intensification is believed to be a path for the future of chemical and process engineering demands [Charpentier, 2007]. RD processes, which integrate sequential operations into a single column, provide a good example of the path, and are becoming increasingly significant in process industry [Lin *et al.*, 2008; Taylor and Krishna, 2000; Malone and Doherty, 2000]. The potential improvements that RD can bring in capital productivity and selectivity, reduction in solvents, energy consumption, and capital investments are significant [Taylor and Krishna, 2000;

Kaymak and Luyben, 2004]. A recent report [Huang *et al.*, 2008] indicates that further internal heat integration can contribute to a substantial reduction of energy requirement and capital investment. economical, environmental, and social aspects are driving force of RD technology and its commercialisation [Harmsen, 2007].

However, the operation of RD processes is exceptionally difficult due to the requirement of tight control [Kaymak and Luyben, 2008]. It is believed that the main barrier to greater use of RD unit operations is the control strategy for achieving the operation targets which are usually coupled [Jimenez and Costa-Lopez, 2002]. Therefore, developing effective control systems for RD columns is a crucial step for the success of RD unit operations.

2.1.2 RD Process Modelling

Optimal design and control of RD processes become challenging without reliable process models. This is especially evident for dynamic simulation, optimisation, and model-based control, for which the model development may lead to a contradiction between the required model accuracy to reflect the process complexity and the feasibility of process simulations regarding the computation time. This has been motivating the research and development of RD process modelling.

There are two basic types of RD process models: equilibrium-based models, and rate-based models. Effort has been made in building good equilibrium-based models for dynamics analysis, process optimisation, and control design [Sneesby *et al.*, 1997a; Sneesby *et al.*, 1997b]. Recent research also focuses on rate-based process modelling for more accurate descriptions of RD process dynamics [Peng *et al.*, 2003; Jimenez and Costa-Lopez, 2002; Noeres *et al.*, 2004]. However, not surprisingly, rate-based models are much more complicated than conventional equilibrium-base models. Thus, simplification of the rate-based models is necessary for online computation and real-time model-based control.

Schneider and colleagues [Schneider *et al.*, 2003; Schneider and Gorak, 2001] have compared various RD modelling methods. From their comparative studies, a rigorous two-phase model is developed for simulation of RD process dynamics. Then, the model is simplified for model-based control design and online dynamic simulation.

Recently, Dalaouti and Seferlis [2006] have proposed a unified modelling approach, which

combines the rigorous non-equilibrium rate-based balance equations with the model-order reduction properties of orthogonal collocation on finite elements approximation techniques for optimal design, operation optimisation and dynamic simulation of complex staged reactive separation processes. The modelling framework has been shown to be particularly efficient in the optimal design of reactive absorption and distillation columns especially those with multiple side feed and product streams mainly due to the elimination of binary decision variables associated with the existence of column stages in any given column section.

The complexity of the rate-based models of RD processes poses difficulties in model computation. Some issues related to numerical simulation of the rate-based models are investigated in [Lextrait *et al.*, 2004] with the special emphasis on spatial discretisation in the solution of steady-state models. The rate-based models are discretised along its spatial dimensions using conventional different finite-difference schemes. The computing efficiency and its impact on the implementation of model-based control strategies are also discussed in the paper.

To develop more accurate process models, relay feedback tests are employed for identification of highly non-linear RD processes [Lin *et al.*, 2006]. It is claimed that with the identified model, good control performance can be obtained using simple PI controllers tuned with the ultimate gain and ultimate period.

RD processes behave with high non-linearity, leading to difficulties in process analysis and control design. The high non-linearity in RD process models can be reduced by using non-linear variable transformation [Tian *et al.*, 2003]. It is shown that control design incorporated with non-linear transformed variables is able to provide improved control performance than those with natural variables [Wang and Wong, 2006].

2.1.3 Control Configurations

There are five degrees of freedom in RD control and operation. Consequently, there are a large number of possible control configurations. Not all of these possible configurations are feasible in practical applications; and some configurations are more effective than others. Therefore, it is important to choose an appropriate control configuration for a specific RD process [Skogestad *et al.*, 1990; Tian *et al.*, 2003; Al-Arfaj and Luyben, 2002].

For feasible RD control configurations, the common practice is to reduce the degrees of

freedom of the control through fixing a few process variables using local control loops. For example, the feed flow rate can be maintained at the desired value through a local flow rate controller; the reflux ratio can also be fixed at the desired value through a local reflux ratio controller. Depending on how the control system is configured, there are one-point control for one degree of freedom [Tian *et al.*, 2003], two-point control for two degrees of freedom [Sneesby *et al.*, 1999; Kumar and Kaistha, 2008a], and three-point control for three degrees of freedom [Kumar and Kaistha, 2008a].

- One-point control has been designed in [Tian *et al.*, 2003] for product purity through manipulating a column temperature. A similar one-point control scheme is also investigated in [Chien *et al.*, 2005] for coupled reactor/column processes. Later, the idea of maintaining a high product purity through indirectly controlling a tray temperature is adopted in [Zeng *et al.*, 2006] for one-point control.
- Two-point control has been used in [Sneesby *et al.*, 1999] for both composition maintenance and conversion control. An investigation into dual-temperature control and one-temperature/one-composition control for different types of flowsheets has shown that simple decentralised control provides a workable solution for highly non-linear RD columns [Hung *et al.*, 2006].
- After comparing two-point and three-point control structures, Kumar and Kaistha [2008a] highlighted the effectiveness of the three-point control; however, the three-point control is designed with increased complexity and is evaluated only for an ideal RD column.

In investigation into the one-point control of coupled reactor/column processes, sensitivity analysis is performed to obtain the suitable temperature control points for the columns [Chien *et al.*, 2005]. The proposed control strategy is very simple containing only one temperature control loop in each column to indirectly control the product purity. A slow cascade outer composition loop structure is judged to be necessary using off-line composition measurements in the presence of small deviations of the product impurity compositions during disturbances.

Chen and Yu [2008] have compared one-column and two-column flowsheets for RD systems of ternary decomposition reactions. Fast measurement is shown to be essential for tight control of RD processes and parallel cascade control offers an attractive alternative for RD control.

In some RD processes, in order to maintain the control performance, controller set-points

may need to change when disturbances in the feed flow rate and catalyst activity occur. This poses a problem in control design and controller tuning. Cascade control with an inner loop and an outer loop is designed in [Wang *et al.*, 2003a] to tackle this problem.

Steady-state analysis is carried out for input-output pairings in the design of RD control systems [Singh *et al.*, 2005]. Because variable pairing and control structure selection are so critical for the overall performance of RD control systems, they become one of the most important steps in the control development. Most references about RD control discuss this issue before attempting to design actual control strategies. The literature that we will review below from other perspectives also involves substantial discussions on this issue.

2.1.4 Direct Control and Inferential Control

In RD processes, some key process variables to be maintained are not measurable easily and reliably. Product composition or purity, and reactant conversion ratio, are among those variables. Without direct measurement, direct control of those variables becomes difficult.

One way to tackle this problem is to make use of state estimators or observers to predict these variables. Several papers have been published by Olanrewaju and Al-Afraj [2008; 2006; 2005] on this topic. The idea is to develop observers or estimators from process models to estimate the composition profiles. With the observed or estimated variables, direct control can be designed for the RD processes. The above authors have claimed that the estimator-based system is robust against a moderate measurement errors and erroneous initial conditions. However, in general, estimator-based designs crucially rely on accurate process models. They do not function well without reasonably good process modelling. In the case of significant model mismatches and noisy measurements, these authors have recommended that an online analyser be integrated into the system. But the question is: if online measurement of the composition was easily available, why would a state estimator have been designed?

Another commonly adopted way to control the RD process variables that are not measurable easily and reliably is to use inferential control [Tian and Tadé, 2000]. This means that instead of seeking direct control of these variables, use indirect control of these variables through maintaining alternative process variables that are good indicators of the variables. For example, RD tray temperature is commonly used for indirect control of the product composition or purity,

and can be controlled through regulating reboiler duty [Tian *et al.*, 2003; Tian and Tadé, 2000]. Inferential control and soft sensing for general systems are discussed in many references, e.g., [Kordon *et al.*, 2003] in which the genetic algorithm is adopted.

Four alternative plant-wide control strategies are studied for RD [Tang *et al.*, 2005]. It is found that control of the product qualities by modulating two tray temperatures in the RD column and one tray temperature in the stripper is most appropriate.

In order to design effective inferential control for RD processes, determining a good location for temperature measurement is crucial. This is discussed in [Tian *et al.*, 2003] and several other references. After examining a few control schemes, Lee and colleagues have claimed that closed-loop based sensor location analysis provides a better alternative for feedback control to the open-loop one [Lee *et al.*, 2007].

Using the inferential control technology, a controller is designed for ETBE RD processes [Athimathi and Radhakrishnan, 2006]. The work shows that the control structure that organises a sensitive tray temperature in the stripping section using the reboiler duty and maintains the temperature difference of reactive trays using the reflux flow is the most suitable configuration. Then, it applies decentralised PI controller and constrained model predictive controller for the RD processes.

2.1.5 RD Control Strategies

As in many other industrial processes, the PI or PID control has been the basis of RD control system design. Many control designs have been compared with the PI or PID control.

Huang *et al.* [2004] have described the temperature control of heterogeneous reactive distillation. A detailed five-step procedure for development of the control system is presented, and a two-by-two control temperature control problem is designed using decentralised PI controllers. In the presence of disturbances, feed-forward temperature compensation is considered to be necessary to maintain the desired product composition.

Complex dynamics and various requirements in RD operation reduce the controllability of the process and suggest that the steady state benefits of reactive distillation might not be realised with a simple control. An integrated control scheme is proposed in [Sneesby *et al.*, 2000]

which permits the control objectives to be changed online in order to reflect changing economic constraints. It is shown that the careful selection of controlled and manipulated variables allows this scheme to be implemented with linear controllers only.

Due to the complex dynamics of RD processes, linear control with fixed parameters has been shown not to be satisfactory to handle its high non-linearity. It needs to be re-tuned adequately over a wide range of operating conditions. In this direction, adaptive PI control [Bisowarno *et al.*, 2004] and gain scheduling control strategies [Bisowarno *et al.*, 2003] have been proposed for an ETBE reactive distillation column. Simulation results show that the proposed control strategies outperform a standard PI controller in both set-point tracking and disturbance rejection.

In [Noeres *et al.*, 2004], a rate-based model is developed for catalytic distillation processes. The model is then simplified for off-line and online optimisation. Model-based linear controller is designed to control the RD processes in a wide range of operating conditions.

For a two-product RD column, a non-linear feedback control scheme is developed for product composition control [Han and Clough, 2006]. The control scheme is derived in the framework of Non-linear Internal Model Control.

For non-linear control design, asymptotically exact input/output-linearisation is applied in simulation studies of RD processes [Gruner *et al.*, 2003]. The resulting control law is claimed to be general. However, the control scheme requires the knowledge of the complete state of the process and therefore an observer has to be designed.

A multi-variable controller is designed for a medium-scale RD process operated in semi-batch mode through a three-step procedure: control configuration, model identification and control design, and controller simplification. The designed controller is tested on a pilot RD column [Voelker *et al.*, 2007].

In [Khaledi and Young, 2005], the non-linear behaviour of an ETBE reactive distillation column is investigated, and a two-by-two unconstrained model predictive control scheme is developed for the product purity and reactant conversion control. For the model predictive control design, a first-order plus dead time model is employed to approximate the process dynamics. The model predictive controller is shown to be capable of handling the process interactions well with good performance in both set-point tracking and disturbance rejection.

Non-linear model predictive control (NMPC) has been applied to highly non-linear RD processes [Kawathekar and Riggs, 2007]. It has been shown that NMPC provides a factor of 2-3 better performance than corresponding PI control. Non-linear model-based control is also discussed in [Balasubramhanya and Doyle, 2000] for RD columns.

RD processes exhibit complex bifurcation and multiplicities [Kumar and Kaistha, 2008c], leading to extra complexity in control design. In [Singh *et al.*, 2005], the impact of steady-state multiplicities on control structure design is highlighted. Following many previous works such as [Sneesby *et al.*, 1998], Kumar and Kaistha [2008b] have recently re-investigated the impact of steady-state multiplicities on the control of RD processes. They have shown that controlling the most sensitive reactive tray temperature results in poor robustness, while controlling a reactive tray temperature with acceptable sensitivity but larger rangeability gives better robustness. They have further demonstrated that controlling the difference in the temperature of two suitably chosen reactive trays further improves control robustness as input multiplicity is avoided.

Although steady state multiplicities occur in RD columns, a linear control is still possible if the processes are operated at a constant reflux ratio [Wang *et al.*, 2003b]. In this work, the reboiler duty is used to control the temperature of a stage just below the reaction section and near the top of the stripping section.

It has been shown that most control designs are based on an accurate process model. In practice, it is not easy to obtain such a good model, or such a model is too complicated to be applied to control design. To alleviate the requirement of process models, we have developed pattern predictive control, which extracts process patterns from process dynamics and then implements model-free pattern-based predictive control [Tian *et al.*, 2003]. **The results will be reported in Chapters 3 and 4**.

2.1.6 Integrated Design of RD Control Systems

Integrated design is necessary for RD control systems due to the complex behaviour of the RD processes, the large number of degrees of freedom, and the unavailability of the measurements of several key process variables. After a good understanding of the process dynamics is developed, decisions are made on how to pair the process variables and how to deal with the large number of degrees of freedom. Then, it is determined what and how the inferential control is

implemented. After that, control strategies are developed for the control systems. Many of the references that we have reviewed have more or less reflected the requirements of the integrated design for RD process operation. In Chapters 3 and 4, we will explicitly address the issue of integrated design of various aspects of process dynamics, pattern recognition, fuzzy logic, non-linear transformation, and predictive control for RD control.

The optimisation technique can be employed in integrated design of RD control to maximise the system performance. Dynamic optimisation has been used for optimising process design and control decisions, leading to a more economically attractive and better controlled system [Panjwani *et al.*, 2005]. In [Georgiadis *et al.*, 2002], the design and control of RD processes are tackled via two different optimisation approaches, i.e., sequential optimisation and simultaneous optimisation. It is shown that with a tight integration of various aspects, the simultaneous optimisation approach leads to a more economically beneficial and better control system.

Control design plays an important role in the development of a real-time control system. As we have seen from our review that significant effort has been made in developing the control design. The majority of the works are from control engineers, and are published in control and process related journals. Two interesting observations are:

- Interests from computer engineers have not been shown in control design; and
- Discussions on the software implementation of the RD control strategies on controllers have not been found at all in all published papers on RD control.

From these observations, it is seen that the separation between control design for controllers and its software implementation on controllers are obvious. Immediate questions are: Who (control engineers or computer engineers) implements the control onto the digital controllers? How is the control implemented on the controllers? How does the control implementation on the controllers affect the performance of the overall control system? How does the control design behave in multi-tasking environments? Is it still feasible in networked control environments?

Answering those questions require systematic research on dynamics analysis and integrated design of real-time control systems with regard to control development for controllers, control implementation on controllers, scheduling of control tasks, and networked control. This will be addressed throughout this thesis after our investigations into the RD dynamics and control design in Chapters 3 and 4.

2.2 Multi-Tasking Scheduling for Real-Time Control Systems

This section reviews recent advances in real-time scheduling, in particular, for real-time control systems. The schedulbility of control tasks is essential in all real-time systems, but is not enough for real-time control systems - we also expect good QoC (quality of control) performance. Therefore, an integrated design, the main theme of this thesis, is required for maximising the overall performance of the control systems.

2.2.1 Real-Time Scheduling Paradigms

During the last three decades, general real-time scheduling theory has advanced significantly [Sha *et al.*, 2004]. Fundamental design problems of real-time scheduling were described in [Mok, 1983; Stankovic *et al.*, 1995]. Many of these problems are still outstanding.

Real-time scheduling algorithms fall into two basic categories: static and dynamic scheduling. Static scheduling algorithms include round robin, cyclic scheduling, fixed priority (FP) with or without pre-emption, etc. For FP scheduling, there are different ways to assign priority levels, e.g., Rate-Monotonic (RM) and Deadline-Monotonic (DM):

- The RM algorithm [Liu and Layland, 1973; Lehoczky *et al.*, 1989] is an FP scheduling algorithm for periodic tasks under certain conditions, e.g., pre-emptive and uni-processor scheduling without blocking and pre-period deadlines. This means that under the same conditions, if a task set cannot be scheduled using the RM algorithm, it cannot be scheduled using any FP algorithm. RM scheduling represents one of the major real-time scheduling paradigms.
- The DM algorithm [Sprunt *et al.*, 1989] is a significant scheduling method for aperiodic tasks. It is also useful in dealing with periodic tasks with pre-period deadlines. DM scheduling is another major real-time scheduling paradigm.

Another priority assignment method is the Least-Compute-Time (LCT) algorithm. It assigns priorities in reverse order of the worst case computation time: the smaller the compute time the higher the priority.

Dynamic scheduling algorithms include those with sufficient and insufficient resources:

- Resource-sufficient dynamic scheduling. The Earliest Deadline First (EDF) algorithm [Liu and Layland, 1973; Stankovic and Ramamrithiham, 1988] is an optimal scheduling strategy under certain conditions in dynamic environments. It is also one of the major real-time scheduling paradigms. While it is normally regarded as a dynamic scheduling technique, EDF may also be used as a static list scheduler, where a list of tasks and time slots is provided to the system, which dispatches them in list order. EDF will in many cases produce a more optimal schedule than DM, but while the static form of EDF can handle periodic tasks with pre-period deadlines, it cannot handle aperiodic tasks.
- Resource-insufficient dynamic scheduling. In overload conditions, the performance of the EDF scheduling algorithm degrades rapidly. Alternatives to the EDF scheduling are the Spring scheduling algorithm [Zhao *et al.*, 1987] and other admission-control based algorithms [Stankovic and Ramamrithiham, 1988]. They represent another major real-time scheduling paradigm.

For a resource-insufficient real-time system with unpredictable workload and significant uncertainties, a new class of scheduling approaches has emerged, i.e., feedback scheduling [Lu *et al.*, 2002; Tabuada, 2007]. Feedback scheduling dynamically adjusts the workload and resources allocation for each of the tasks according to the actual QoS measurements. For QoS based scheduling, there have been a number of references, e.g., [Rajkumar *et al.*, 1998]. Feedback scheduling has become one of the major scheduling paradigms in real-time systems.

2.2.2 Reduction of Control Latency and Jitter

While real-time scheduling theory has evolved significantly [Sha *et al.*, 2004], it has been investigated mainly as a separate phase in the development of control systems. As indicated in Seto *et al.* [2001], Balbastre *el al.* [2004], and Tian *et al.* [2006b], real-time control systems are conventionally developed in two separate phases: the control design and its software implementation.

- For control design, digital control theory has been well established for fixed sampling frequency, and consequently control periods will be kept unchanged at runtime once they are determined in the control design. This makes the computing workload of the control task set unchangeable, and leads to poor use of the limited computing resources.

- For real-time scheduling, theory has been well developed under the known worst-case execution times (WCETs), fixed periods, and hard deadlines (Choi and Agrawala [2000], Samard and balas [2003], Lavagno *et al.* [2003], Balbastre *et al.* [2004], Lluesma *et al.* [2005]). However, many of such assumptions are conservative and do not reflect the real system requirements at runtime.

Thus, neither of these two separate phases can provide an optimal solution to the overall real-time control system [Tian *et al.*, 2006b]. It remains unclear how the performance of the control software affect the performance of the overall control system. We contend that without a systematic performance analysis of the digital controller as part of the system, how the performance of the overall control system can be guaranteed [Tian and Sun, 2005]. This calls for research on the co-design of control and its implementation [Seto *et al.*, 2001; Balbastre *et al.*, 2004; Tian *et al.*, 2006b].

For co-design of control and its digital implementation, Kim and Park [2003] developed a method for period and priority assignment in distributed control systems to improve the performance of the control systems. Balbastre *et al.* [2004] proposed a task model of real-time control systems in order to reduce control action interval and data acquisition interval, implying an improvement in the control performance.

Tian *et al.* [2006b] developed three strategies to reduce control latency and jitter in control implementation, i.e., introduction of offsets into control task scheduling, decomposition of control tasks into smaller subtasks; and increasing the priority levels of control output subtasks. **The detailed results of this work will be presented in Chapter 5**.

All these works have focused on reduction of control latency and/or jitter as a means of control performance improvement although the Quality-of-Control (QoC) is not quantitatively measured. This is reasonable because time delay usually results in rich dynamics in dynamic systems [Tian and Gao, 1998a] and limits the improvement of system performance. While there have been extensive studies on process time delays and significant investigations into communication delays in control systems, time delays in control computation and scheduling have not been well understood in the control community [Kim and Park, 2003].

The idea of subtask partition has also been used in Lluesma *et al.* [2006], where the performance of a control system is evaluated to examine the benefits of some task partition schemes.

2.2.3 QoC Performance and Control Task Scheduling

For real-time control systems, the primary objective is to maintain satisfactory QoC, which is characterised by some sort of control performance index [Buttazzo *et al.*, 2007]. Widely used performance indices include integral of absolute error (IAE), integral of time absolute error (ITAE), quadratic cost function, etc.

For continuous-time plants under digital control, the QoC is closely related to the control period. It has been shown that quadratic and linear approximations of control cost function can be used for on-line and optimal task scheduling [Eker *et al.*, 2000; Cervin *et al.*, 2002]. A shorter control period will lead to better QoC provided the system is not overloaded.

However, the control period cannot be made arbitrarily short because of the need to avoid an excessive workload. Also, a short control period does not help improve control performance when the output of the plant reaches the steady state.

For the lower bound of the control period, the sampling theorem requires the sampling frequency, which is the inverse of the control period, to be at least twice the bandwidth of the continuous-time output of the plant under control. A common rule of thumb is to choose the sampling frequency to be 4 to 20 times this bandwidth. Åström and Wittenmark [1997] state that the sampling period should be chosen such that $0.2 \leq \omega_0 h \leq 0.6$, where ω_0 is the natural frequency of the plant, and h is the control period. Therefore, the control task periods can be made adjustable between their upper and lower bounds in digital control applications.

An important aspect in control task scheduling is how to adjust the control periods in overload conditions. Several approaches have been proposed with different complexity of computing for period scaling [Beccari *et al.*, 2005]. In [Buttazzo *et al.*, 2002], an elastic scheduling scheme is proposed in which task utilisations are treated like springs and are compressed by enlarging periods. Cervin [2003] has developed a method incorporated with the EDF scheduling to re-scale the control periods when the processor is overloaded. With this method, in overload conditions, all periodic tasks are executed as if they were running with larger periods.

Adaptive bandwidth reservations are proposed in [Abeni *et al.*, 2005] for dynamic scheduling of real-time systems in overload conditions. The method is based on a “bandwidth” abstraction, meaning that a task is given the illusion of executing on a dedicated slower processor.

Considering the integrated design of controllers and schedulers, a method is developed in [Chandra *et al.*, 2003] to find the optimal frequencies for systems using analytically redundant controllers. The method is shown to be robust against inaccuracies in the estimation of failure rates of the controllers.

Addressing the requirements in real-time control, a method is proposed by Amirijoo *et al.* [2008] to quantify and suppress the measurement disturbance in real-time feedback control systems. The authors claim that a controller using the proposed control structure outperforms a traditional control structure with regard to performance reliability.

2.2.4 Feedback Scheduling for Real-Time Control Systems

Recently, feedback scheduling methods have been developed for scheduling control task periods to meet the QoC requirements. They have been shown to be very effective to support the specified performance of dynamic systems that are both resource insufficient and exhibit unpredictable workload [Amirijoo *et al.*, 2007].

Among much effort in this area, Cervin and colleagues [Cervin *et al.*, 2002] have summarised the related work in this area till 2002. Using quadratic and linear approximation of control cost function, they have proposed a feedforward-feedback scheduling approach for real-time control tasks [Cervin *et al.*, 2002]. A feedback scheduling scheme is proposed by Buttazzo and Abeni [2002] to automatically adjust task periods without knowing the actual computation times of tasks. Recently, Buttazzo *et al.* [2007] have investigated how to manage QoC in overloaded real-time control systems.

A feedback control scheduling framework is presented in [Lu *et al.*, 2002] for adaptive real-time systems. Different categories of real-time applications are identified where different feedback scheduling algorithms should be used.

Following the idea of [Buttazzo and Aneni, 2002], Ushio *et al.* [2007] have considered a non-linear elastic task model, where the elastic coefficient depends on the utilisation allocated to the task. The model is applied to an adaptive fair sharing controller.

In overload conditions, a method based on adaptive reservation is proposed by Abeni *et al.* [2005] for dynamic scheduling of real-time systems. Through QoS management and feedback

adjustment of the task periods, the overall QoC of a real-time control system can be maintained.

The design aspect of a real-time scheduler is considered in [Song *et al.*, 2008] for a class of embedded systems. For this purpose, a feedback controller for a reservation-based processor scheduler is designed for soft real-time systems. An analytical model for a real-time scheduler is provided in terms of a switched system with time-varying uncertainty, and a state feedback controller is developed to stabilise the switched system.

Revisiting the problem of scheduling stabilising control tasks in embedded controllers, Tabuada [2007] treats a real-time scheduler as a feedback controller that decides which task is executed at any given instant. The work shows how a simple event-triggered scheduler leads to guaranteed performance and thus relaxes traditional periodic execution requirements.

Considering that solving the optimisation problem for on-line QoC evaluation is very time-consuming and demands considerable computing power, Jin *et al.* [2007] and Xia *et al.* [2005; 2009b] have employed fuzzy logic in feedback scheduling of control tasks. Xia *et al.* [2008] and Xia and Sun [2005] have also used neural network technology to simplify the on-line QoC assessment in feedback scheduling of control systems.

Among several related works, a feedback control-based method has been developed for dynamic resource management for general distributed real-time systems [He *et al.*, 2007].

Zhu and Mueller [2008; 2005; 2005] have shown that feedback scheduling is especially useful for dynamic voltage scaling to reduce energy consumption in embedded systems. Power-aware feedback algorithms have been developed for real-time systems that adapt to dynamically changing workload. For example, they have proposed an approach combining a dynamic voltage scaling scheduler with feedback EDF scheduling [Zhu and Mueller, 2005].

Recently, feedback scheduling has also been designed for priority-driven control networks [Xia *et al.*, 2009a]. With the aid of the co-design of control and scheduling, an integrated feedback scheduler is developed to enable flexible QoC management in dynamic environments. It incorporates a cascaded feedback scheduling module for sampling period adjustment with a direct feedback scheduling module for priority modification.

Effort is also being made to decompose the overall scheduling into several levels to help simplify the scheduling problem. The concept of hierarchical scheduling has increasingly

appeared in the open literature. Various architecture designs are possible depending on how to balance the computing efficiency and computing demand.

Early in 1997, a two-level hierarchical pre-emptive scheduling model was described in which the global scheduler could be EDF [Deng *et al.*, 1997; Deng and Liu, 1997]. In this hierarchical scheduling model, a constant utilisation server was used to execute each application in an open system.

Abeni and Buttazze [1999] proposed a hierarchical feedback scheme, where an application-level feedback is used to adjust the QoC requirements of the control tasks and a system-level feedback is employed to adjust the bandwidths assigned to the tasks.

Davis and Burns [2005; 2006] have analysed a two level hierarchical system, in which both global and local schedulers used fixed priority scheduling, based on the worst-case response time. Using the same principles, Lorente and Palencia [2006] also analysed the worst-case response time for tasks under a two-level hierarchical EDF scheduling scheme.

Hierarchical scheduling of different hard real-time applications on a uni-processor was also analysed in [Zhang and Burns, 2007] where the EDF algorithm was used as the local scheduler and the global scheduler of the system could be fixed priority or EDF.

2.2.5 Gaps in Feedback Scheduling of Control Tasks

In many of the feedback scheduling methods that we have reviewed above, e.g., those in [Cervin *et al.*, 2002; Xia *et al.*, 2005; Xia *et al.*, 2009b; Xia and Sun, 2005], the focus is on the schedulability of the multiple control tasks, and the QoC performance of the control system is not directly linked to the feedback scheduling. A scheduler is implemented as a separate task with high priority to

- Evaluate the QoC periodically; and
- Re-schedule the control periods if the processor utilisation deviates from its desired value.

A trade-off is usually required when selecting the period of this scheduler task: a shorter period is favourable for the scheduler to respond to the environment changes while a longer period reduces the overhead introduced to the processor by the scheduler.

We have realised that there are two problems in these feedback scheduling methods:

1) Firstly, when the processor of the controller is overloaded, control tasks with lower priority will be blocked in fixed priority scheduling.
2) Secondly, when the actual processor utilisation is lower than the desired one, control periods will always be scaled down.

As a result of the first problem, there will be no way to evaluate the QoC performance of the blocked control loops until the feedback scheduler is run in its next period to scale up the control periods to bring down the processor workload. Consequently, the QoC of the blocked control tasks may deteriorate significantly or even become unstable. This has been well demonstrated in [Cervin *et al.*, 2002]. Reducing the period of the feedback scheduler task may improve the situation with the cost of extra workload, but will not resolve the problem completely.

It is easy to understand that no control loops will be blocked under EDF scheduling in overload conditions. This has been clearly demonstrated by Cervin [2003]. Control loops will also not be blocked under other elastic and adaptive scheduling methods, e.g., those in [Buttazzo *et al.*, 2002; Buttazzo and Aneni, 2002; Abeni *et al.*, 2005]. However, these methods are still schedulability driven, and do not directly reflect the QoC of the control system.

Because many real-time control systems have the nature of embedded systems with limited computing resources, not every existing scheduling method is well supported by a specific control computing platform. For example, in a networked multi-robot system that will be briefly discussed in Section 5.6 of Chapter 5, the V+ operating system and programming environment have been used. They support FP scheduling of multiple tasks at the application level, but do not support EDF scheduling directly. Implementing other scheduling methods in this environment will be more difficult. As another example, the real-time and object-oriented programming language Ada also provides full support to FP scheduling in application programs. While EDF dispatching has been introduced into the Ada2005 definition, there is still a lack of EDF runtime support from Ada compilers, e.g., gnat, one of the most popular Ada compilers.

For the second problem, scaling down the control periods does not really make sense for the control loops which have already reached their steady states since it does almost nothing in QoC improvement but wastes more computing resources. This has significant implications to embedded systems with limited power resources because increasing workload means more

power consumption [Zhu and Mueller, 2005].

One may argue that it is sufficient to simply disable period scaling when the actual workload is lower than the desired one. It sounds reasonable; but when and how to disable it? A clear answer to those questions has not been found in the open literature. Actually, simply disabling the period scaling does not provide an optimal solution. When a control loop reaches its steady state, we should gradually increase its control period capped by the maximum allowable value regardless what the actual processor utilisation is.

In this thesis, we have made effort to develop a priority driven and feedback scheduling framework for real-time control systems. The feedback scheduling will fill in the technical gaps analysed above by providing a new task model to ensure that the QoC performance is evaluated in each period, and a new period scaling strategy to minimise the computing demand to the control system. **The detailed results of our feedback scheduling framework will be reported in Chapters 6 and 7**.

2.3 Dynamics and Integrated Design of Networked Control Systems

With the rapid development of modern network and communication technologies, design and implementation of real-time control systems over communication data networks become technically and economically viable. In the fast growing NCS area, challenges and significant achievements co-exist. This section does not intend to provide a comprehensive survey of all aspects of the NCS research. Rather, it reviews recent NCS developments relevant to Research Problems 4 to 7 identified in Section 1.3 of Chapter 1 for investigation in this thesis. In particular, we will focus on the dynamics analysis and integrated design aspects of NCS.

2.3.1 Complex Networks and Networked Control Systems

Networked systems and applications are not a new idea. However, systematic investigations into the interactions among network components and the complex dynamics of network systems did not occur until recently. With much effort in this area, some basic concepts and approaches of complex networks have been developed in recent years to describe the connectivity, structure, and dynamics of complex systems.

Recent reports on complex networks include [da Rocha and da Costa, 2007] in which a specific hybrid system has been investigated, and [Zhanga *et al.*, 2007] that focuses on efficient packet routing in complex networks. A recent special issue on complex networks in *International Journal of Computer Mathematics*, vol. 85, no. 5, Aug 2008, is devoted to the latest developments in complex networks. It has been realised that networked environment introduces challenges to system analysis and development, and the distributed nature of many system components and services requires new technologies to guide the system design.

As a typical class of networked systems, networked control has gained increasing interest especially in the last five years. An NCS implements control over communication networks that interconnect sensors, actuators, controllers, and other components and devices (Figure 1.2). It is becoming increasingly important in industry due to the increasing demand on large-scale integration of information, communications, and control. Introduction to the fundamentals of NCS can be found in [Antsakls and Baillieul, 2007; Hespanha *et al.*, 2007; Tian *et al.*, 2006a; Martí *et al.*, 2005; Antsakls and Baillieul, 2004; Tipsuwan and Chow, 2003; Walsh *et al.*, 2002; Walsh and Ye, 2001; Lian *et al.*, 2001].

Several journals have published special issues on NCS, including *IEEE Transactions on Automatic Control*, vol. 49, Sept. 2004, and *Asia Journal of Control*, vol. 8, nos. 1-2, 2006. A recent special issue on networked control in *Proceedings of the IEEE*, vol. 95, no. 1, 2007, reflects the advances in the NCS area till 2007.

Supporting real-time traffic is an essential requirement in such networked applications. Many reports have discussed this requirement, e.g., [Tian *et al.*, 2007; Tian and Levy, 2008a; Zhang *et al.*, 2004; Huang *et al.*, 2006; Davis *et al.*, 2007], and references therein. We have recently analysed the complex behaviour of the network traffic in real-time NCS, and have found the multi-fractal nature of the real-time network traffic [Tian *et al.*, 2007].

It is interesting to notice that a Networked Control System Laboratory (NCSLab) is designed in the University of Glamorgan on http://www/ncslab.net, which provides a unified and flexible web-based interface to access test rigs located in different countries of the world [Hu *et al.*, 2008]. We are also aware that several universities from different countries, including The University of Paderborn, are discussing the possibility to establish some web-based and networked robotics laboratories. Networked control technologies play an important role in establishing such laboratory facilities.

To show how fast the NCS area is growing, Figure 2.1 depicts some statistics of the journal publications on “networked control” and “networked control systems”. This figure by no means provides an exhaustive search of the pertinent journal publications; it does show a strong trend of increasing NCS research activities over the last few years, especially after 2003. The number of publications in 2005 is doubled in 2007, and the number in the first three quarters of this year is already comparable with that of the last year.

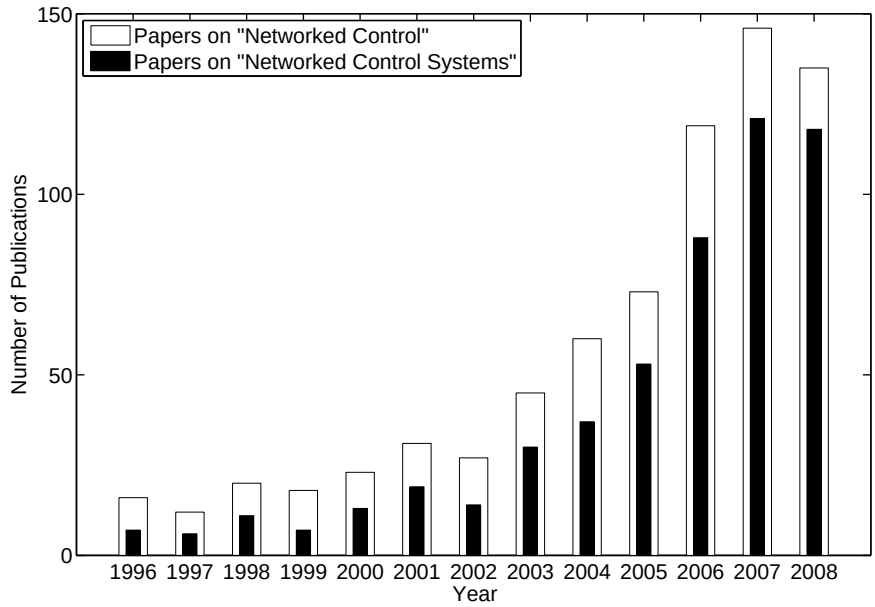

Figure 2.1: Statistics of the publications on networked control as at 29 September 2008 (Source: Web of Science).

2.3.2 Two Challenging Problems

It is well realised that control over data networks faces two challenging problems: network induced delay and packet dropout. These challenging problems appear in both wired and wireless NCS applications, and become severer in wireless NCS. For wireless NCS, other challenging problems also appear, e.g., bandwidth constraints, which will, however, not be addressed in this thesis.

The network induced delay is time-varying and unpredictable, and thus affects the accuracy of timing-dependent computing and control of the NCS. It is also related to packet sequence disruption. For general mobile IP networks, Wong *et al.* [2005] investigated the impact of

route optimisation on out-of-order packet delivery. They proposed an analytical model to assist in this investigation, and further analysed how the transmission delay distributions affect the probability of the out-of-order delivery. Their work provided a useful guideline in adjusting the routing priority of binding updates to improve the end-to-end network performance. However, their work was for general mobile IP networks, rather than specifically for NCS.

The packet loss stems from network traffic congestions and limited network reliability, and forces the controller and/or actuator to make a decision on how to control the system when a control packet is lost. Major difficulties will appear in the presence of successive packet losses. However, severe successive packet losses may be an indication of abnormal operation of the NCS, and may need to be considered separately in safety alarms and protection control.

Effort has been made to solve these two challenging problems. From the structure of the NCS (Figure 1.2), it is natural that many control scientists and engineers have focused on the study on the controller side while the network Quality-of-Service (QoS) conditions are considered as constraints. This represents a major direction in NCS research and development.

When the network induced delay is treated as a stochastic variable, mathematical models can be developed to describe the network induced delay and packet loss, separately or simultaneously. A simultaneous mathematical description of both network induced delay and packet loss in a unified model is considered in [Yue *et al.*, 2004]. This description has been shown to be more effective than previous modelling methods in dealing with network stability and control synthesis [Peng *et al.*, 2008; Peng and Tian, 2007; Peng and Tian, 2006; Jiang and Han, 2004; Yue and Han, 2006; Yue *et al.*, 2005].

With the developed simultaneous modelling of network induced delay and packet dropout, stability criteria, which relate to the upper bound of the time delay, are derived that guarantee the stability of the overall NCS [Yue *et al.*, 2004; Peng and Tian, 2006; Peng *et al.*, 2007; Zhang *et al.*, 2001]. Controller synthesis is also carried out to determine the controller settings under the stability criteria.

Overall, approaches developed from the perspective of the controller design provide stability conditions, but do not directly deal with the real-time requirement of the NCS network communications. The controllers designed from the stability criteria are usually conservative in order to guarantee the stability of the NCS.

2.3.3 Dynamics of Network Induced Delay

Two fundamental assumptions have been inherently made on network induced delay in the majority of existing literature. The first one is that the network delay is uniformly distributed with lower and upper bounds. Much effort has been made to find the upper delay bound that still maintain the stability of the NCS. The other assumption is that the network delay is purely random between the lower and upper bounds. With these assumptions, the stochastic behaviour of the network induced delay has been modelled and analysed using various techniques, e.g., the stability-focused NCS methods that have been reviewed above in Subsection 2.3.2.

Even though these fundamental assumptions are widely used in the open literature, they have not been shown to be valid through rigorous analysis. Recently, we have analysed the statistical properties of the network induced delay for some typical scenarios of real-time NCS. Our results reveal the non-uniform distribution and multi-fractal nature of NCS network traffic under certain conditions [Tian *et al.*, 2007; Tian and Levy, 2008b]. These findings enable improved stability conditions and less conservative controller design over the existing methods. **The detailed results of our analysis will be presented in Chapter 8**.

On dynamics of network traffic, early work by Leland *et al.* [1994] investigated the self-similarity nature of Ethernet traffic. Since then, self-similarity behaviour, i.e., long-term memory, has also been observed in several other types of network traffic in general network systems. However, this phenomenon has not been reported for networked control systems in which real-time requirements are essential, and will be addressed in this thesis for NCS. The multi-fractal nature of the network induced delay in real-time NCS [Tian *et al.*, 2007] reflects the self-similarity characteristics of the real-time traffic.

We have used multi-fractal analysis to investigate the dynamics of the network induced delay. Multi-fractal analysis is a useful tool to characterise the spatial inhomogeneity of both theoretical and experimental fractal patterns [Grassberger and Procaccia, 1983]. It was initially proposed to treat turbulence data, and has recently been applied successfully in many other fields, e.g., time series analysis [Pastor-Satorras, 1997; Canessa, 2000] and financial modelling [Anh *et al.*, 2000]. It also finds applications in biological problems, such as those from Anh *et al.* [2001; 2002], Yu *et al.* [2001a; 2003; 2004; 2006], and Zhou *et al.* [2005]. Some sets of physical interest have a non-analytic dependence of the dimension spectrum D_q on the

q-moments of the partition sum of the sequences.

Multi-fractality has a direct analogy to the phenomenon of phase transition in condensed-matter physics [Katzen and Procaccia, 1987]. The existence and type of phase transitions might be a useful characterisation of universality classes for the structures [Bohr and Jensen, 1987]. The concept of phase transitions in multi-fractal spectra was introduced in the study of logistic maps, Julia sets, and other systems with simple mathematical descriptions yet complex dynamics and structures. Evidence of phase transition was found in the multi-fractal spectrum of diffusion-limited aggregation [Lee and Stanley, 1988].

2.3.4 Deficiencies in Throwing Bandwidth

Conventionally, "throwing bandwidth" was an effective method to resolve many problems in network communications. Therefore, some computer/network engineers are optimistic that over-provisioning of network capacity can easily resolve the challenging problems of both network induced delay and packet dropout.

It has not been well recognised that a high bandwidth does not necessarily mean predictable communication behaviour, and real-time control does not necessarily require fast data transmission. Predictable communication behaviour is essential in real-time NCS applications. Over-provisioning may work in some applications, but is not a general solution for NCS. This can be well demonstrated from the following aspects.

1) There have been many industrial control systems implemented in fieldbus networking, e.g., Controller Area Network (CAN) [Davis *et al.*, 2007]. However, the fieldbus technology has not been widely deployed in industry [Lee and Lee, 2002]. Moreover, like all other technologies, fieldbus also has its technical limitations. For example, DeviceNet, a popular fieldbus technique based on CAN provides up to 1Mbps bandwidth with the maximum end-to-end transmission distance of 40m. For the transmission distance of 1.3km, the bandwidth on the bus is down to 50kbps; this implies that only about 25 devices can be interconnected in order to achieve similar performance to that in peer-to-peer networking, where each peer-to-peer connection can have up to 1.92kbps bandwidth with RS232 asynchronous communications for about 1km (when the 0-10mA direct current transmission is used). Over-provisioning of the capacity of such networks may be

a very expensive solution, if not impossible, especially for large-scale control systems.

2) Effort has been made to promote Ethernet based networking for simple and cost effective NCS [Lian *et al.*, 2001; Lee and Lee, 2002; Tipsuwan and Chow, 2004b]. With recent advances in network technologies, high-speed IP networks have been commercialised with the bandwidth as high as 100Mbps, 1Gbps, and even 10Gbps. Thus, the transmission speed of data packets in wired IP networks is no longer a major problem for most industrial applications. However, deterministic and predictable communication performance is not guaranteed in IP networks. If network induced delay is comparable with the process delay, the performance of the Ethernet based NCS will be largely dependent on the networking. Although high speed Ethernet has very small communication latency [Lee and Lee, 2002], it has not been recommended for time-critical NCS [Lian *et al.*, 2001] because of its limited real-time QoS.

3) Our recent studies [Tian *et al.*, 2006a] have shown that for some Ethernet based NCS networks, simply increasing the network capacity may result in more packet losses for periodic control tasks. We have interpreted this as the consequence of the burst traffic of multiple periodic control tasks. This is not acceptable in many real-time applications, e.g., safety-critical systems.

4) A large number of sensors and actuators used in various applications, especially in small and battery-powered embedded control systems, have limited computing power and do not support high communication bandwidth.

5) In wireless networked applications, over-provisioning of the bandwidth is impossible or very expensive in most cases because of the tight bandwidth resources.

Therefore, a general solution to the challenging problems of network induced delay and packet loss cannot rely on the over-provisioning of the network capacity.

From our achievements in the dynamics analysis of real-time network traffic, we have developed a general queuing methodology to smooth out the network delay [Tian and Levy, 2008b]. Aiming to provide predictable timing behaviour of the networked control, the methodology considers the real-time control requirements but is implemented as a network protocol independent of the plants to be controlled. Therefore, it does not require process models, which are essential

to most exiting NCS control technologies. **The detailed research of the real-time queuing protocol will be discussed in Chapter 9**.

2.3.5 Compensation for Control Packet Dropout

Much effort has been made to address the challenging problem of packet dropout in NCS. An active research field to deal with packet dropout is to model packet loss together with network induced delay. This is based on the observation that a packet loss leads to an increase in the network induced delay of the control loop. Thus, stability analysis and controller design can be conducted with considerations of the upper and lower bounds of the network delay. The main idea is to find the stability conditions for specific plants to be controlled and then to design the controller to meet the conditions. Along this direction, NCS stability based methods have been rapidly expanding, e.g., [Zhang *et al.*, 2001; Yue *et al.*, 2004; Peng *et al.*, 2007], Peng and Tian [2006; 2007], and many references therein. Packet dropout is considered in some references in stability analysis, but is not compensated at all in these methods.

Packet dropout compensation was considered in [Chan and Özgüner, 1995] through sophisticated probabilistic predictions of the lost packets. It was also investigated by Luke and Ray [1990; 1994] through state estimation. The concept of "queuing packets" was also presented in these references. These methods build on accurate plant models for state prediction, and thus work well only when good plant models are available. Unfortunately, this is a unrealistic assumption for many industrial applications.

Ling and Lemmon [2003] considered compensation for dropped feedback measurements in an NCS in a framework of a constrained generalised regulator problem. They claimed that their optimal dropout compensator worked better than previous dropout compensation strategies. Their compensation scheme was designed for dropped measurement packets and was directly coupled with process model and controller design.

Addressing the simultaneous compensation of network induced delay and packet dropout, Soglo and Yang [2006] designed an agent-based networked control estimator at the controller to improve the performance of the NCS. They modelled the NCS as an asynchronous dynamical system with rate constraint, and then used the bilinear matrix inequality method to solve the compensation problem. However, they inherently assumed that the network induced delays

were less than one sampling period, while the guarantee of the satisfaction of this assumption was not provided in their work.

Schenato [2006; 2008] studied optimal state estimation in NCS subject to random delay and packet loss, and used the estimation in the control design. The mathematical treatment is elegant in this work. However, like [Luke and Ray, 1994] and [Chan and Özgüner, 1995], this work also strongly relies on an accurate plant model. It is not only computationally intensive, but also difficult to implement in general NCS applications.

Some schemes for robust observer and control of general time delay systems, e.g., the method reported in [Mahmound *et al.*, 2006], may also be applicable to NCS. Similar to the method by Schenato [2006; 2008], most of these schemes require accurate plant models. Consequently, they need to be designed case by case in applications. Difficulties arise when accurate plant models are not available. In process control, there is a lack of accurate model descriptions for many complex processes, or the available models are too complicated for online computation, such as reactive distillation [Tian *et al.*, 2003], industrial crystallisation, fermentation [Yao *et al.*, 2001], simulated moving bed chromatography [Yao *et al.*, 2008], to just name a few.

Recently, we have developed simple yet effective packet dropout compensation strategies [Tian and Levy, 2008b; Tian and Levy, 2008a].We have also conducted functional analysis of simple packet loss compensators using formal methods [Fidge and Tian, 2006]. Compared with existing methods to deal with packet loss, our compensators have two unique features: (1) they do not rely on the process models, implying that the requirement of accurate process models is alleviated; and (2) they are implemented on actuators, rather than on controllers. Our idea emanates from the observation that prediction of dropped control packets from past control signals can be treated like dynamic voltage scheduling from past voltage settings [Varma *et al.*, 2003]. **The main results of the developed strategies will be presented in Chapter 10**.

2.3.6 Integrated Design of NCS

Networked control integrates control, network, and communications, and thus needs integrated design of all these aspects to maximise the performance of the system. The need for a revolutionary new approach to system co-design stems from the unique demands that will be imposed by the complex systems in the coming age of networked computational systems including

networked control [Ghosh, 2005].

Martí *et al.* [2004] have showed that the co-design of adaptive controllers and feedback scheduling policies in an NCS allows for the optimization of the overall QoC.

A co-design method is presented by Ji and Kim [2008] for dynamic optimal network-bandwidth allocation and adaptive control of NCS. It integrates adaptive control with real-time scheduling of available network resources. The same authors have also discussed a similar problem in which the co-design is carried out between the system performance and sampling periods [Ji and Kim, 2007].

Sun and El-Farra [2008] have developed a quasi-decentralised control framework for plants with distributed, interconnected units that exchange information over a shared communication network. An integrated control and communication strategy is developed to ensure the desired closed-loop stability and performance for the plant while minimising network utilisation and communication costs. This is achieved by formulating the networked closed-loop plant as a hybrid system and then obtaining the maximum allowable update period for communications.

In [Nikolakopoulos *et al.*, 2008], an integrated framework is reported for a wireless NCS with significant packet dropouts. It monitors the QoS of the network continuously. According to the QoS, it adjusts the data retransmission rate and periodically tunes the controller parameters.

A co-design approach of predictive control design and control signal transmission scheduling is proposed for a set of NCS [Zhao *et al.*, 2008]. The scheduling algorithms are designed with the guarantee of the stability of all the systems.

Some issues on integrated software implementation of NCS are investigated by Arzen *et al.* [2007]. A general component-based framework is outlined for embedded control problems over sensor networks, and a number of control-oriented components are built for control implementation and design integration. The approach is tested through a real-world application.

In [Lian *et al.*, 2006], the performance of information sharing of multiple cooperative agents over a communication network is analysed, and design methodologies are proposed to guarantee acceptable control and communication performance in a networked control system. Particular attention is paid to the co-design between control periods and the data transmission rates. Similar ideas are also used in [Chen *et al.*, 2006].

Zhang and Hristu-Varsakelis [2006] have proposed a co-design method for control and communications to stabilise an NCS, in which access to the communication medium is governed by a pair of periodic communication sequences. In their model, the communication disruptions are simplified by ignoring sensors and actuators. Their co-design method first identifies a pair of communication sequences that preserve reachability and observability, and then uses a feedback controller based on those sequences to stabilise the system. It still focuses on the controller design, and crucially relies on the models of the plant to be controlled.

It is seen that most existing integrated design methods for NCS are carried out between two aspects: the control development and transmission rate (or control period) scheduling. To ensure the performance of the control system, both aspects are tightly coupled. Tightly coupled integrated design is a good in the sense that the overall performance of the system can be optimised with respect to the available system parameters. However, the drawbacks are also obvious. For example, due to the tight coupling, more sophisticated models have to be built to deal with the complicated system analysis and design. Also, the packet dropout issue is not addressed explicitly, implying that conservative control design is required in order to maintain the system stability. Nevertheless, the great progress in integrated NCS design has provided useful experience for further development of simpler yet more effective technologies.

From our analysis of the dynamics of the network induced delay and packet dropout, we have realised that the worst-case communication delay (WCCD) in NCS can be made configurable [Tian and Levy, 2007]. This will reduce the network induced delay significantly if the NCS is tolerant of a certain level of packet loss rate. Then, we have developed a new integrated design method for real-time NCS. The method integrates the real-time queuing protocol, the packet dropout compensators, and the configuration of the WCCD into a unified framework. Through this framework, the network induced delay is limited within a single control period and becomes more predictable. This largely simplifies the system analysis and design. **The proposed integrated design framework will be discussed in detail in Chapter 11**.

2.4 Summary of This Chapter

The applications of real-time control systems are rapidly increasing, and various design aspects of the control systems have been discussed extensively in the open literature. Evidence has

shown that benefits can be achieved through integrated design that considers several aspects simultaneously, but it appears that the integrated design has not been well addressed. Especially, the implications of the implementation of the control systems on computers for the performance of the control systems have not been well analysed. Practical and effective integrated design methods are required for performance improvement and complexity reduction.

A control system will not work well without a good control design. Therefore, the majority of the research activities in real-time control systems have focused on the control strategy development mainly from control engineers. Process modelling and model-based control have been the main themes in the control design. This signifies the significance of the understanding of the plants to be controlled via mathematical descriptions. However, for many complex processes, accurate process models are either difficult to obtain or too complicated for control design. Techniques that do not tightly rely on the process models for control design are thus necessary. The pattern predictive control to be presented in this thesis is such a technique, which integrates into a unified framework various design technologies, including pattern recognition, fuzzy logic, non-linear transformation, and predictive control.

Many issues arise when control designs are implemented on a digital controller. Typically, multiple control tasks compete for computing time slices and other resources, which are limited in an embedded control system. Multi-tasking scheduling becomes significant but its effects on the control performance have not been well understood. Many computer engineers have developed various elegant scheduling algorithms for control systems. It appears that the focus of most such algorithms is still on the schedulability. It is not well recognised that schedulablity is essential but not enough in real-time control, good control performance is also expected. Therefore, there is a need to link the task scheduling to the control performance. This thesis develops two integrated design methods for multi-tasking scheduling for real-time control systems. One method links the scheduling with control latency and jitter, which are used to indirectly characterise the control performance; and the other links the scheduling directly to the QoC through a hierarchical architecture.

When a control system is designed in a networked environment, challenging problems appear, e.g., network induced delay and packet dropout. While some engineers tend to throw more bandwidth to solve these problems, this has been shown not to be a general solution. Significant effort has been made in dealing with these problems through modelling network

induced delay and packet dropout, deriving stability conditions, and then designing controllers based on the stability criteria. Integrated design is considered in NCS mainly between the control design and transmission rates. All those technical paths are complicated, and packet loss is not compensated in most existing methods. It appears to be interesting not to solve the network problems from the network perspective, but from the control perspective which gives more complicated solutions and conservative control design. Considering the real-time control requirements, this thesis develops an integrated design framework for NCS through network protocols. The framework integrates queuing protocol, packet dropout compensation, and configuration of WCCD to maximise the system performance while simplifying the system analysis and design significantly.

2.5 Nomenclature of This Chapter

Abbreviations

CAN	Controller Area Network
DM	Deadline-Monotonic
EDF	Earliest Deadline First
FP	Fixed Priority
ITAE	Integral of Time Absolute Error
LCT	Least-Compute-Time
NCS	Networked Control System/s
PI	Proportional-Integral
PPC	Pattern Predictive Control
QoC	Quality of Control
QoS	Quality of Service
RD	Reactive Distillation
RM	Rate-Monotonic
WCCD	Worst-Case Communication Delay

Chapter 3

Dynamics and Pattern Predictive Control of Reactive Distillation

---

Well designed **control strategies for controllers** are essential for a real-time control system to provide the desired functionality. For complex processes such as reactive distillation (RD), neither the widely used simple Proportional-Integral-Derivative (PID) control nor advanced model-based control can well handle the process control problems. **Integrated design of model-free and intelligent control strategies** that can capture the patterns of the process dynamics is an attractive way for process operation. From the understanding of the dynamics of the complex RD processes, this chapter develops such a control solution that integrates various aspects of process dynamics, pattern recognition, fuzzy logic, non-linear transformation, and predictive control into a unified framework.

Synthesis of ethyl *tert*-buty ether (ETBE), a high-performance fuel additive, through RD is an attractive route; but its operation and control are exceptionally difficult due to its functional combination and complex dynamics. Modern control technology greatly relies on good process models, while a reasonable RD model is too complex for control design. Moreover, RD contains considerable uncertainties that cannot be well described in process modelling. Alleviating the model requirement, this work aims to maintain ETBE purity in the RD process through developing a pattern predictive control (PPC) scheme incorporating with a linear controller. The work is carried out on a pilot-scale RD column for ETBE production.

The core content of this work has been published in [Tian *et al.*, 2003].

3.1 Process Description

In RD of ETBE, reactant conversion and product purity are two key process variables. The former is a measure of the usage of the raw materials, while the latter characterises the quality of the product. Both are directly related to the productivity of the RD process. This work addresses the maintenance of the purity of the product (ETBE) withdrawn from the bottom of the RD column.

The ETBE purity cannot be measured easily and reliably for real-time control. In our design, it will be controlled indirectly by regulating a column temperature. Therefore, the control objective is translated to the maintenance of the temperature at a desired value. For this purpose, both set-point tracking and regulation problems should be considered in the control design and deployment.

To alleviate the requirement of accurate models, this work develops a pattern predictive control (PPC) scheme incorporating with conventional proportional-integral (PI) controller for the complex RD process. Similar ideas have been employed by Zhao *et al.* [2000; 2002] for processes with delay. The process dynamics, control structure, non-linear transformation, and pattern extraction and utilisation for process prediction will be discussed in detail. Case studies will be provided in the next chapter (Chater 4) to verify the proposed PPC approach.

ETBE, $(CH_3)_2COC_2H_5$, is produced from ethanol and a mixed C_4 olefine stream containing isobutylene (typically of a cracking unit product). The dominant chemical reaction in ETBE synthesis is the reversible reaction of isobutylene and ethanol over an acid catalyst

$$(CH_3)_2C = CH_2 + C_2H_5OH \Longleftrightarrow (CH_3)_3COC_2H_5 \tag{3.1}$$

The acidic ion-exchange resin, Amberlyst 15, is used in our work. The reaction is equilibrium limited in a range of temperatures. The reaction kinetics has been investigated in [Jensen and Datta, 1995] and discussed in [Sneesby *et al.*, 1997a], from where detailed expressions for the equilibrium constant and rate equation are available.

Side reactions exist in ETBE synthesis. One is the dimerisation of isobutylene to form diisobutylene (DIB) $[(CH_3)_2C{=}CH_2]_2$. In the presence of water in the reaction environment, another side reaction is the hydration of isobutylene to form isobutanol (isobutyle alcohol)

$(CH_3)_3COH$. The side reactions are expressed by

$$(CH_3)_2C = CH_2 + (CH_3)_2C = CH_2 \iff [(CH_3)_2C = CH_2]_2 \tag{3.2}$$

$$(CH_3)_2C = CH_2 + H_2O \iff (CH_3)_3COH \tag{3.3}$$

This work is carried out on a pilot-scale RD column at Curtin University of Technology. The RD column has been built for ETBE (and MTBE) production. It is shown in Figure 3.1. With a diameter of 0.155m and a height of 4.1m, the column consists of three sections for rectifying, reaction, and stripping, respectively. It is filled with two novel packings, one of which contains the catalyst, Amberlyst 15, which is necessary for the etherification reaction. The RD process has a total condenser and a partial reboiler.

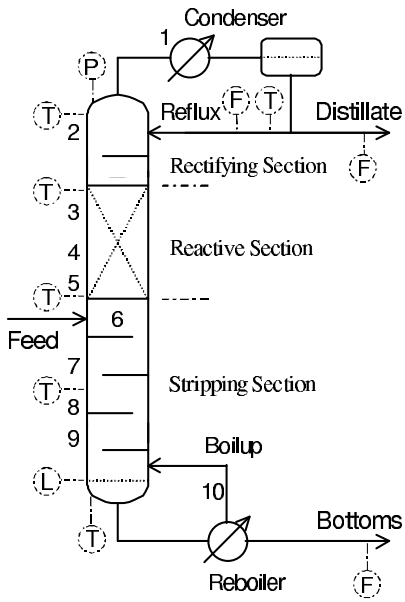

Figure 3.1: Pilot-scale RD column (T, F, P, L in dashed circles mean measurements for temperature, flow rate, pressure, and level variables).

The column is estimated to have 8 theoretic stages: 1, 3, and 4 stages in rectifying, reactive, and striping sections, respectively. The condenser and reboiler are considered as two separate stages. Therefore, there are 10 stages altogether, which are numbered from top to bottom as shown in Figure 3.1. The rectifying section has only one stage (stage 2); the reactive section has 3 stages (stages 3, 4, and 5); and the stripping section has 4 stages (stages 6 to 9). The raw material is fed at stage 6; while the inal product, ETBE, is withdrawn from stage 10 (reboiler). Measurement points are also indicated in Figure 3.1 for temperature, flow rate, pressure, and

level variables. More information about the the architecture of the RD process can be found in [Sneesby *et al.*, 1997a; Sneesby *et al.*, 1997b; Tian and Tadé, 2000].

A typical set of operating conditions of the RD process for ETBE synthesis is tabulated in Table 3.1. It will be considered as the nominal situation for design of ETBE purity control.

Table 3.1: A typical set of RD operating conditions.

Feed composition		
ETBE	29.1	mol%
Ethanol	9.1	mol%
Isobutylene	7.3	mol%
n-Butylene	54.5	mol%
Stoichiometric excess ethanol	5.0	mol%
Feed rate	0.76	L/min
Distillate rate	0.50	L/min
Reflux flow rate	2.53	L/min
Bottoms rate	0.53	L/min
Overhead pressure	950	kPa
Bottoms ether purity	90	mol%
Reboiler duty	8.45	kW
Reboiler temperature	160	°C
Stage 7 temperature	133.12	°C

3.2 Control System Configuration

RD control is challenging due to high non-linearity, strong interactions, bifurcation and multiplicity, time delay, process uncertainties, and the large number of possible control configurations. An RD process has 5 degrees of freedom for control design, i.e., 5 process variables can be manipulated (control inputs): flow rates of reflux (L), boil-up (V), distillate (D), bottoms (B), and column top vapour V_T. The reflux ratio L/D can be used instead of L for control design. In our pilot-scale RD process, the flow rates of boil-up (V) and column top vapour (V_T) are characterised by reboiler duty (Q_r) and condenser duty (Q_c), respectively,. The controlled variables are reflux accumulator and reboiler levels, column pressure, bottoms (ETBE) purity, and reactant conversion. Typical disturbances to the RD process include changes in feed flow rate (F_f) and feed composition (F_c), which is characterised by the stoichiometric ratio.

In practice, three control loops are designed for inventory and pressure control. The control configuration problem addressing which of the five degrees of freedom should be used in those three loops has been extensively discussed, e.g. [Skogestad *et al.*, 1990]. Two more loops can be designed for purity and conversion control. The resulting control configuration is conventionally named by the two independent variables that are used for composition (purity) control, e.g. LV, LB, D/V, (L/D)(V/B), etc. Among all possible control structures including ratio schemes and those with consideration of feed flow rate F_f, non-ratio schemes LV and LB has been shown to be preferred for the RD column under consideration [Sneesby *et al.*, 1997b]. This work will consider the LV configuration.

In the LV configuration of the RD process, the column pressure is maintained by manipulating the condenser duty Q_c; the inventory control for reflux accumulator and reboiler hold-up is implemented by adjusting the distillate flow rate D and bottoms flow rate B, respectively. The reflux flow rate L is fixed while the reboiler duty Q_r is manipulated for product (ETBE) purity. This is a typical one-point control problem in RD processes.

The purity control is important, while it is not easy because the purity characterised by composition is difficult to measure in real time reliably and economically. Fast and reliable measurement of the controlled variable is a basic requirement of closed-loop control.

A method to overcome this difficulty is to implement inferential control for the purity. In this method, an inferential model has to be developed to infer the purity from multiple measurements that are easily obtained, e.g. multiple column temperatures. Progress has been made in this direction [Sneesby *et al.*, 1999; Tian and Tadé, 2000].

An alternative method to overcome the difficulty is to indirectly control the purity by controlling some other process variables that are easy to obtain and are indicators of the purity. However, such indicators are not easy to find due to the unavailability of a one-to-one relationship between a single variable and the purity. In distillation column control practice, column temperatures are usually used for indirect composition control.

The reboiler temperature reflects the dynamic changes of the ETBE purity quickly, while it is not a good purity indicator since a single reboiler temperature value may correspond to multiple purity values [Tian and Tadé, 2000]. In this work, the stage 7 temperature T_7 is used for inference and control of the product (ETBE) purity for the RD process under consideration.

3.3 Process Dynamics

The steady state relationship between the ETBE purity and the reboiler duty Q_r for a fixed reflux flow rate is shown in Figure 3.2 [Tian and Tadé, 2000]. The reboiler duty Q_r = 8.45kW is around an optimal Q_r value that gives the maximum value of the ETBE purity. It is chosen as the nominal operating point.

The steady state relationships between T_7 and Q_r are investigated under different reflux flow rates and are graphically depicted in Figure 3.3. It is clearly seen from Figure 3.3 that there exists a significant non-linearity in process gain K_p (K_p is high within an operating range of Q_r but becomes small outside this range), and the reflux flow rate L affects the T_7 versus Q_r relationship in a complex manner.

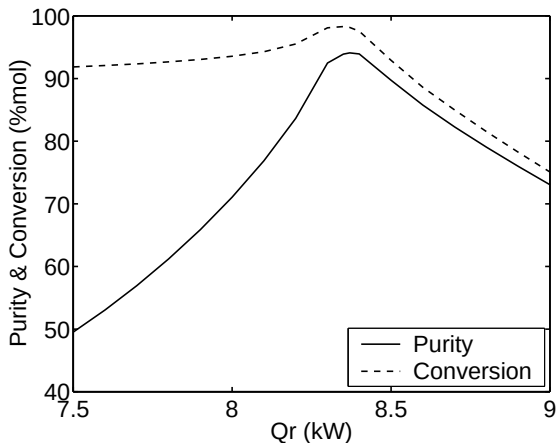

Figure 3.2: Purity and Conversion versus reboiler duty for the reflux fixed at 2.53L/min.

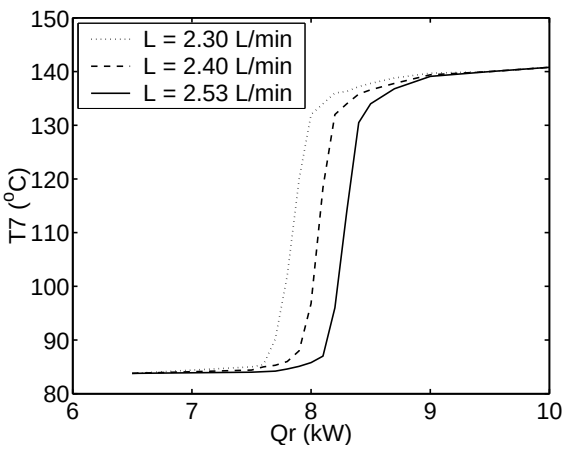

Figure 3.3: Stage 7 temperature T_7 versus reboiler duty Q_r.

The ETBE purity versus T_7 can be easily obtained from Figures 3.2 and 3.3. Although the relationship between T_7 and the purity is non-linear, T_7 determines the purity uniquely. Another advantage of using T_7 is that the sensitivity of T_7 to the purity is high [Sneesby *et al.*, 1997b].

It has been found that the RD process has changeable inertia as operating conditions change, suggesting time-varying process response speed. By convention, the inertia is characterised by a time constant or multiple time constants.

Detailed investigation also reveals considerable time delay from Q_r to T_7, implying that any manipulation in Q_r will not affect T_7 until the time delay elapses. The significance of the RD time delay is shown in Table 3.2. The existence of time delay imposes severe constraints on the control system and complicates the control system design.

Table 3.2: First-order plus delay dynamics of T_7 versus Q_r.

L	Q_r	K_p	T	d
L/min	kW	°C/kW	min	min
2.30	8.10	19.25	3.04	4.09
	8.45	3.40	1.41	2.38
	8.75	3.31	3.7	2.19
2.40	8.10	209.23	32.50	1
	8.45	4.87	3.04	4.09
	8.75	4.22	4.9	2.56
2.53	8.10	35.34	60.00	0
	8.45	49.28	15.60	7.0
	8.75	4.29	7.6	2.93
Disturbances				
$+5\%$ change in F_f		2.88	16	2
-5% change in F_f		2.88	18	2
$+5\%$ change in F_c		-30.36	38	0
-5% change in F_c		-30.60	40	5

With a fixed heat input to the reboiler, an increase in feed flow rate F_f will result in an increase in bottoms flow rate and consequently a decrease in the temperatures below column stage 6. On the other hand, the feed stoichiometric ratio characterising the feed composition has a positive effect on T_7, i.e. an increase in the ratio will lead to an increase in T_7. Changes in feed flow rate F_f and feed composition F_c are primary disturbances to the process operation. They affect T_7 dynamics significantly in a complex manner, e.g., non-linear gain, time-varying

inertia, and time delay, which are similar to but different in value from those from Q_r to T_7.

Simplified input-output process models expressed by transfer functions for manipulation and disturbances are identified under specific operating conditions. The identified results are first-order plus time delay descriptions, which are tabulated in Table 3.2. Table 3.2 shows the RD process dynamics quantitatively. For example, the process gain, time constant, and time delay all change in a wide range as process operating conditions change. While more complex expressions of T_7 dynamics can be precisely established from the first principles of mass balance and energy balance, e.g. [Sneesby *et al.*, 1997a; Sneesby *et al.*, 1997b], these types of models are difficult to use for control design.

3.4 Pattern-Based Predictive Control

Due to the complexity of the RD process dynamics, conventional control technologies, e.g. PI control, cannot provide satisfactory control performance, while the application of modern control technology requires good process models. A reasonable process model as described in [Sneesby *et al.*, 1997a; Sneesby *et al.*, 1997b] contains hundreds of equations for the 10-stage RD process under consideration. It is too complicated to be directly used for control system design. Simplified input-output process models discussed in the last section are helpful in understanding the complex process dynamics, but they are identified under some specific operating conditions and thus cannot represent the process in a wide range of operating conditions. Although it is possible to extend the transfer function models, again this will complicate the control system design. Furthermore, the RD process contains a large degree of uncertainties, which cannot be well described using any type of mathematical expressions. Therefore, techniques without using exact process models are more attractive for RD control.

Pattern predictive control (PPC) is such a method that does not rely on exact models while providing improved control performance for complex processes over conventional control algorithms. Some progress has been made in this direction, e.g. for time delay compensation [Zhao *et al.*, 2000], adaptive PI [Seem, 1998], fuzzy control [Jang and Chen, 1996], etc.

The proposed PPC system for the RD process is schematically depicted in Figure 3.4, which is a further development of the authors' previous work [Zhao *et al.*, 2000] that considers linear

processes. It consists of two main parts: a non-linear transformation $u = f(v)$ and a pattern-based predictor (PP). The former is used for input-output linearisation of the process gain, while the latter is employed to anticipate process output some (e.g. d) steps ahead. The PP utilises process feature patterns qualitatively and quantitatively, which are extracted from the controlled and manipulated variables, and is incorporated with a conventional controller G_c (e.g. PI) in the PPC system. For the RD system, y and u in Figure 3.4 correspond to T_7 and Q_r, respectively.

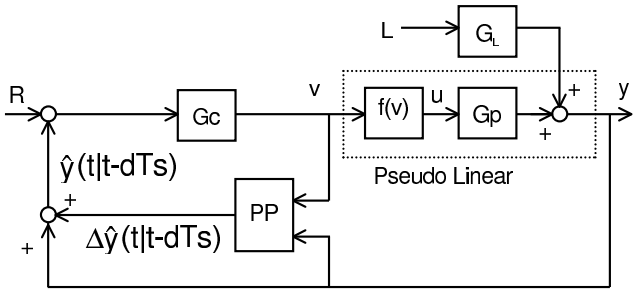

Figure 3.4: PPC system structure.

Ideally, The PP acts as a time lead component as it provides d steps ahead prediction of the controlled variable. It will effectively compensate for the time delay in the RD process and thus allows more aggressive controller settings compared with the control systems without PP. Therefore, the PPC will provide improved performance in both set-point tracking and disturbance rejection, as will be shown later for the RD process.

3.5 Non-linear Transformation

Because the process has a highly non-linear process gain, which will degrade the prediction performance of the PP, a non-linear transformation

$$u = f(v) \tag{3.4}$$

is introduced to obtain a $v \sim y$ relationship with a pseudo linear gain, where v is the new manipulated variable. This is a type of input-output linearisation [Kravaris and Kantor, 1990], which is one of the most widely used techniques for non-linear control system design.

Notice that controlling a process always requires a certain degree of understanding of the process. The rough knowledge of the process gain can be obtained *a priori*, which is sufficient

to construct such a non-linear transformation. Moreover, a linear PP can accommodate a certain degree of process uncertainties, implying that no exact non-linear transformation is required.

The process gain K_p can be determined by the derivatives of the curves in Figure 3.3. Let $g(u)$ denote the steady state input-output relationship of Figure 3.3, i.e.

$$y = g(u) \tag{3.5}$$

Ideally, $f(v)$ is designed such that y is linear to the new manipulated variable v, i.e.

$$y = g[f(v)] = bv + c \tag{3.6}$$

where b is a constant representing the gain of the input-output linearised system; c is also a constant representing the bias. It follows that $f(v)$ is the inverse of $g(bv + c)$, i.e.

$$f(v) = g^{-1}(bv + c) \tag{3.7}$$

The determination of the constants b and c is straightforward and can be done at the operating point of the RD process. Tables 3.1 and 3.2 shows that at the operating point, $Q_r = 8.45$kW, $T_7 = 133.12$°C, $K_p = 49.28$°C/kW. Suppose that these values are retained in the input-output linearised system. According to equation (3.6), we have

$$b = 49.28, c = -283.2960 \tag{3.8}$$

The solid curve of Figure 3.3 corresponds to the nominal operating conditions of the RD process. From this curve, the following non-linear transformation function can be constructed

$$f(v) = p_1 + p_2 \exp\left[p_3(bv + c - p_4)\right] + p_5 \ln(bv + c),\ v \in [7.4, 8.6] \tag{3.9}$$

where $p_1 \sim p_5$ are parameters, which are taken to be

$$[p_1, \cdots, p_5] = \begin{cases} [5.0786, -1, -0.9837, 83.3992, 0.6813], & v \leq 8.0993; \\ [3.5615, 1, 0.5623, 140.083, 1], & \text{otherwise} \end{cases} \tag{3.10}$$

As shown in equation (3.9), under the nominal operating conditions, v is limited within [7.4,8.6], resulting in a range of [6.6,10.6]kW for u. Computation of the designed non-linear transformation $f(v)$ shows that $8.45 \approx f(8.4196)$, implying that the nominal value of v is 8.4196, which has a small deviation from the nominal value of u (8.45kW). The designed $u = f(v)$ versus v is shown in Figure 3.5, in which a portrait of y versus v is also depicted. As expected, a pseudo linear relationship between y and v is obtained within $v \in [7.4, 8.6]$.

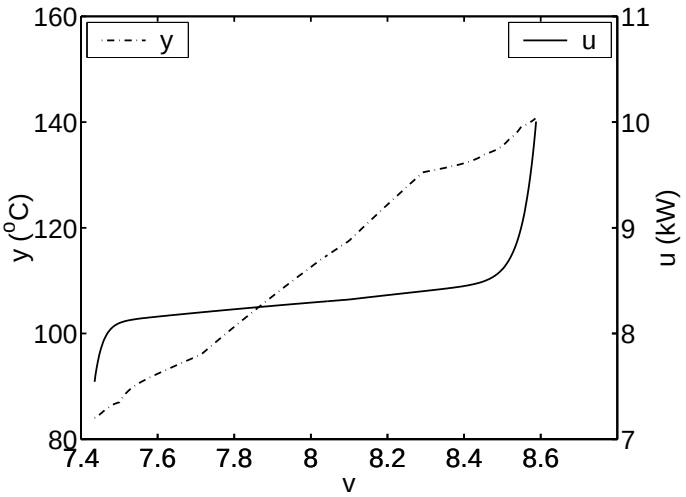

Figure 3.5: Non-linear transformation and the resulting y versus v relationship.

3.6 Selection and Extraction of Feature Patterns

Instead of using an exact process model, a PPC system utilises process feature patterns qualitatively and quantitatively for process prediction. Therefore, selection and extraction of process feature patterns are crucial in PPC design.

In process control systems, the controlled variable y is measurable and the manipulated variable u can be recorded from the controller output. Therefore, the time series $\{y(k)\}$ and $\{u(k)\}$ are available for real-time process analysis and control. The process feature patterns are extracted from these two time series. The basic ideas of the feature pattern extraction have been discussed by the authors [Zhao *et al.*, 2000] and will be further developed below.

Most chemical processes behave with an S-shaped response to a step change in manipulated variable or disturbances (Figure 3.6). The response can be characterised by four stages with different dynamic behaviour. The first stage is the *time delay* stage, in which the process

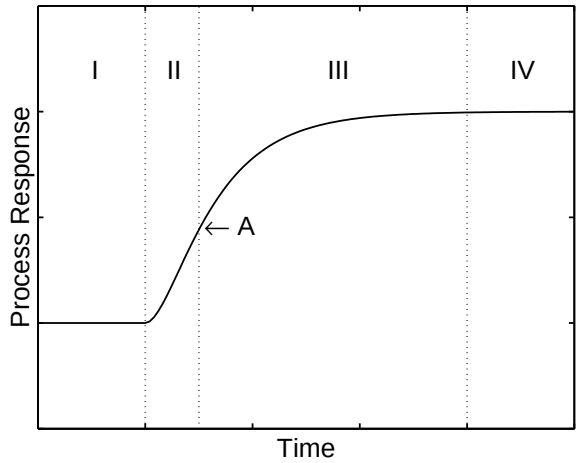

Figure 3.6: S-shape process response to a step change in either the manipulated variable or disturbances. I: time delay stage; II: accelerating increase stage; III: decelerating increase stage; IV: steady state stage; A: inflection point.

accumulates energy or materials while without any response to the input. Right after the time delay elapses is the second stage, the *accelerating increase stage*, in which the process starts to respond to the input with an accelerating rate due to the "energy-storing" effect of the first stage. The third stage, the *decelerating increase stage*, starts from an inflection point of the response and remains increasing yet with a decreasing rate due to the "energy-releasing"effect. The response increases until it reaches its steady state – this is the last stage of the response.

The difference between the values of the controlled variable at two successive sampling instants captures the incremental variation of the controlled variable. Thus, the following process feature pattern, S_1, is extracted from the time series of the controlled variable

$$S_1(k) = y(k) - y(k-1) \tag{3.11}$$

For a step change excitation, $S_1 = 0$ implies that the response is either in *time delay* stage or in *steady state* stage. Instead of using the equality $S_1 = 0$, the inequality $|S_1| < \epsilon$ should be used in practice in order to accommodate any types of noises, where $\epsilon > 0$ is a small and predetermined threshold.

It is easy to know that S_1 cannot discriminate between the *accelerating* and *decelerating increase* stages of the process response to a step change excitation as S_1 gives the same sign in both stages. This difficulty can be overcome by introducing some sort of S_1 variation as a

process feature pattern, which is denoted by S_2

$$S_2(k) = \begin{cases} 0, & |S_1(k)| < \epsilon; \\ \dfrac{1}{d}\sum_{m=1}^{d}\left[S_1(k+1-m) - S_1(k-m)\right], & \text{otherwise} \end{cases} \tag{3.12}$$

where $\epsilon > 0$ is a small threshold, d is the prediction horizon and satisfies

$$d \geq \theta / T_s \tag{3.13}$$

θ and T_s are process time delay and sampling period, respectively.

S_2 captures the fluctuating trends of S_1 in the *accelerating* and *decelerating increase* stages of the process response to a step change excitation. $S_2 > 0$ and $S_2 < 0$ imply that the response increases with an increasing and decreasing rates, respectively.

It is helpful to conceptually consider the feature patterns S_1 and S_2 as the first- and second-order differences in discrete-time systems, compared with the first- and second-order derivatives in continuous-time systems.

Time delay is a common phenomenon in chemical processes. Materials and energy fed to a process will not affect the process output within the *time delay* stage of the process response, while they will eventually change the controlled variable in the later stages. Therefore, the time series of the manipulated variable also contain information of the future process output.

Instead of the real manipulated variable u, the transformed manipulated variable v will be used in feature pattern extraction. The magnitude of v is already reflected in the feature patterns S_1 and S_2, while the incremental variation of v characterising additional excitation has not been covered by either S_1 or S_2. The following feature pattern S_3 captures the incremental variation of v during the time period $[k-d, k]$

$$S_3 = \sum_{m=1}^{d}\left[v(k+1-m) - v(k-d)\right] \tag{3.14}$$

In geometry, S_3 reflects the area bounded by the curve $v(k)$ and the straight line crossing the point $(k-d, v(k-d))$ over the time period $[k-d, k]$. It influences the process output in a complex manner, which needs more feature patterns to characterise.

The following feature pattern S_4 assists in determining the effect of S_3 on process output.

$$S_4 = \begin{cases} 0, & |S_3(k)| < \epsilon; \\ \left| \dfrac{1}{S_3} \displaystyle\sum_{m=1}^{d} m \left[v(k+1-m) - v(k-d) \right] \right|, & \text{otherwise} \end{cases} \tag{3.15}$$

3.7 Pattern-Based Fuzzy Prediction

The d steps ahead prediction of the controlled variable is carried out using the following simple formulae, which are based on the extracted process feature patterns

$$\hat{y}(k+d|_k) = y(k) + \Delta\hat{y}(k+d|_k) \tag{3.16}$$

$$\Delta\hat{y}(k+d|_k) = r_1 \left[S_1(k) + r_2 S_2(k) \right] + r_3(k) S_3(k) + h(k) \Delta\hat{y}(k|_{k-d}) \tag{3.17}$$

where $r_1 \sim r_3$ are three coefficients; $h > 0$ is an updating factor for real-time adaptation.

The updating factor h can be simply chosen to be a constant, implying that a fixed step is used. The following variable-step algorithm is employed in this work for updating h

$$h(k) = h(k-1) + \Delta h(k) \tag{3.18}$$

where $\Delta h(k)$ is designed to be 0 if $|\Delta\hat{y}(k|_{k-d})| < 0.001$, and $0.01|\Delta\hat{y}(k|_{k-d})|$ if $|\Delta\hat{y}(k|_{k-d})| > 0.1$ and $|\Delta\hat{y}(k|_{k-d}) - \Delta\hat{y}(k-1|_{k-d-1})| < 0.001$. For all other conditions, take $\Delta h(k) = 0.01\text{sign}\left[\Delta\hat{y}(k|_{k-d}) - \Delta\hat{y}(k-1|_{k-d-1})\right] \Delta\hat{y}(k|_{k-d})$.

The determination of the values of the parameters r_1 and r_2 relies on an estimate of the ratio T/θ. However, r_1 and r_2 are fixed for a specific value of T/θ. They should be proportional to the prediction horizon d and should be a function of the ratio T/θ. The following heuristic relations can be used to determine r_1 and r_2

$$r_1/d = 0.6622 + 0.0244(T/\theta) - 0.5482 \exp\left(-2.1T/\theta\right) \tag{3.19}$$

$$r_2/d = 0.2039 + 0.0047(T/\theta) - 0.1797 \exp\left(-2T/\theta\right) \tag{3.20}$$

Equation (3.20) has extended the guideline for r_2 determination in a table form as shown in

[Zhao *et al.*, 2000], where the parameter r_2 was integrated into the feature pattern S_2.

A compact, explicit, and satisfactory expression for r_3 has not been established due to the complex influences of the patterns S_3 and S_4 on the process output. However, our experience shows that r_3 should be determined based on the ratio T/θ and the feature pattern S_4. Thus, a set of fuzzy logic rules is developed to describe r_3 qualitatively and quantitatively.

Let FS_x denote the fuzzy set of $x \in \{T/\theta, S_4, r_3\}$. Each of the three fuzzy sets uses three linguistic terms: big (B), medium (M), and small (S), to represent, in an approximate and quantised way, the magnitude of the corresponding variable. Let $A_x^i \in \{B, M, S\}$ denote a specific linguistic value of FS_x in the *i*th fuzzy rule, $x \in \{T/\theta, S_4, r_3\}$. The triangle membership function $\mu(\cdot) \rightarrow [0, 1]$ is used for the fuzzy sets $FS_{T/\theta}$ and FS_{S_4} (Figure 3.7).

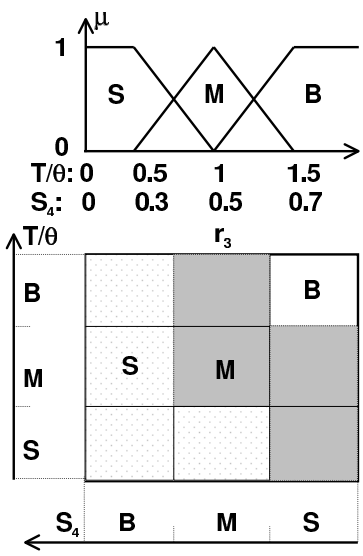

Figure 3.7: Fuzzy sets and fuzzy rules.

Corresponding to $A_{r_3}^i \in \{B, M, S\}$, three central numerical values denoted by $V(A_{r_3}^i)$ can be identified using, say, the least-squares technique. The fuzzy rules take the If-Then form, e.g.

$$\text{R}^i\text{: } \textbf{If } \left(T/\theta \text{ is } A_{T/\theta}^i\right) \text{ and } \left(S_4 \text{ is } A_{S_4}^i\right) \textbf{ Then } r_3 \text{ is } V\left(A_{r_3}^i\right) \tag{3.21}$$

Combining $FS_{T/\theta}$ and FS_{S_4}, 5 fuzzy rules are designed, which are tabulated in Table 3.3. These rules are schematically depicted in Figure 3.7. The computation of the degrees of fulfilment for these rules is also shown in Table 3.3 [Babuška, 1999].

Table 3.3: Fuzzy rules and degrees of fulfilment.

R^i	IF	THEN	β_i
R^1	S_4 is B	r_3 is S	$\beta_1 = \mu(S_4 \text{ is } S)$
R^2	(T/θ is not S) and (S_4 is M)	r_3 is M	$\beta_2 = \min\{1 - \mu(T/\theta \text{ is } S), \mu(S_4 \text{ is } M)\}$
R^3	(T/θ is S) and (S_4 is M)	r_3 is S	$\beta_3 = \min\{\mu(T/\theta \text{ is } S), \mu(S_4 \text{ is } M)\}$
R^4	(T/θ is B) and (S_4 is S)	r_3 is B	$\beta_4 = \min\{\mu(T/\theta \text{ is } B), \mu(S_4 \text{ is } S)\}$
R^5	(T/θ is not B) and (S_4 is S)	r_3 is M	$\beta_5 = \min\{1 - \mu(T/\theta \text{ is } B), \mu(S_4 \text{ is } S)\}$

Finally, the parameter r_3 is computed over the entire fuzzy rules through defuzzification

$$r_3(k) = \sum_{i=1}^{5} \left[\beta_i V\left(A_{r_3}^i\right)\right] / \sum_{i=1}^{5} \beta_i \tag{3.22}$$

3.8 Summary of This Chapter

A PPC system that integrates various aspects of pattern recognition, fuzzy logic, non-linear transformation, and predictive control has been developed for real-time control of complex RD processes. RD of ETBE is a non-conventional and complex process with high non-linearity, strong interactions, bifurcation and multiplicity, time delay, and large degree of process uncertainties. Alleviating the requirement of good process models, which are essential for modern model-based control, it utilises feature pattern-based prediction incorporated with conventional PI control. To obtain a pseudo input-output linear process gain, a non-linear transformation is designed, which needs only a rough and easily obtained knowledge of the steady state characteristics of the process. Four types of process feature patterns are extracted from the time series of the controlled variable and the transformed manipulated variable. Fuzzy logic rules driven by the extracted feature patterns are then developed for process prediction.

Case studies will be carried out in the next chapter (Chapter 4) to demonstrate the effectiveness of the developed PPC scheme.

3.9 Nomenclature of This Chapter

The same nomenclature is used in this chapter (Chapter 3) and the next chapter (Chapter 4). It is listed at the end of the next chapter (Section 4.7 of Chapter 4).

Chapter 4

Verification of Pattern Predictive Control for Reactive Distillation

---

Comprehensive case studies are conducted in this chapter to verify the pattern predictive control (PPC) scheme developed in the last chapter (Chapter 3) for reactive distillation (RD) processes. The PPC scheme is applied to a pilot-scale RD column for synthesis of ethyl *tert*-buty ether (ETBE). After discussions of the configuration of the control system, the performance of the PPC is evaluated for both set-point tracking and disturbance rejection. The results show that the PPC scheme is a promising tool for complex processes, where good process models are difficult to obtain or to implement for real-time control. The core content of this work has been published in [Tian *et al.*, 2003].

4.1 System Configuration

For case studies, the developed pattern-based predictive control (PPC) scheme is applied to a pilot-scale reactive distillation (RD) column. The RD column and its dynamics have been described in Chapter 3 Sections 3.1 through to 3.3.

The PPC strategies are used for purity maintenance in RD of ETBE. The purity is controlled indirectly by controlling the stage 7 temperature T_7 of the RD column; while T_7 is maintained by manipulating the reboiler duty Q_r. This is a one-point control problem in RD.

For the RD process under consideration, as shown in Table 3.2, $T = 15.6$min and $\theta = 7$min

under the nominal operating conditions. T_s and ϵ are set to be 1min and 10^{-5}, respectively. According to Equation (3.13), set $d = 8$. The initial value of h is taken to be 0.1.

For process prediction using Equation (3.17), r_1, r_2, and r_3 are required. Equations (3.19) and (3.20) give $r_1 = 5.6919$ and $r_2 = 1.6983$. r_3 is obtained through fuzzy logic inference.

For the specific RD control problem, $T/\theta = 15.6/7$ is already known, which is *Big* with $\mu\,(T/\theta \text{ is } B) = 1$ as shown in Figure 3.7. Therefore, the rules R^3 and R^5 in Table 3.3 are excluded and the remaining rules R^1, R^2, and R^4 are simplified to

$$\begin{aligned} &\text{R}^1\text{: } \textbf{If } (S_4 \text{ is } B) \quad \textbf{Then } r_3 \text{ is } V(S) \\ &\text{R}^2\text{: } \textbf{If } (S_4 \text{ is } M) \quad \textbf{Then } r_3 \text{ is } V(M) \\ &\text{R}^4\text{: } \textbf{If } (S_4 \text{ is } S) \quad \textbf{Then } r_3 \text{ is } V(B) \end{aligned} \tag{4.1}$$

which correspond to the first row of Figure 3.7. Consequently, β_1, β_2, and β_4 in Table 3.3 are reduced to

$$\beta_1 = \mu(S_4 \text{ is } S),\ \beta_2 = \mu(S_4 \text{ is } M),\ \beta_4 = \mu(S_4 \text{ is } S) \tag{4.2}$$

Taking into account Figure 3.7, the computation of Equation (4.2) is shown in Table 4.1.

Table 4.1: Degrees of fulfilment and r_3 for the RD purity control.

S_4	≤ 2.4	$(2.4, 4.0]$	$(4.0, 5.6]$	> 5.6
$\beta_1 = \mu(S_4 \text{ is } S)$	1	$2.5 - S_4/1.6$	0	0
$\beta_2 = \mu(S_4 \text{ is } M)$	0	$S_4/1.6 - 1.5$	$3.5 - S_4/1.6$	0
$\beta_4 = \mu(S_4 \text{ is } B)$	0	0	$S_4/1.6 - 2.5$	1
r_3	$V(S)$	$\beta_1 V(S) + \beta_2 V(M)$	$\beta_2 V(M) + \beta_4 V(B)$	$V(B)$

It is seen from Table 4.1 that $\beta_1 + \beta_2 + \beta_4 = 1$ for any specific values of S_4, implying that Equation (3.22) for r_3 computation is reduced to

$$r_3 = \beta_1 V(S) + \beta_2 V(M) + \beta_4 V(B) \tag{4.3}$$

As also shown in Table 4.1, the relationship in Equation (4.3) can be further simplified for specific values of S_4 due to the fact that one or more $\beta_i = 0$. Three values of $V(B)$, $V(M)$, and $V(S)$ have been identified to be 2.4510, 1.0622, and -1.9974, respectively.

Typical curves of $v(t)$, $y(t)$, $\hat{y}(t|t - dT_s)$, $y(t) - \hat{y}(t|t - dT_s)$, $h(t)$, and $S1$ to S_4 are shown

in Figures 4.1, 4.2, and 4.3. It is seen from these figures that the d steps ahead prediction of y matches the real y very well, and the prediction error is within ± 1.5°C, i.e., $\pm 1.25\%$ of the real y, for the v with gradual and sharp changes. The updating factor h is updated if prediction error exists, as shown in Figure 4.2. However, in this example, h (< 0.21) contributes at most 0.26°C to the prediction, implying that the adaptive term, i.e. the last term of Equation (3.17), weighs about $0.26/133 \approx 0.2\%$ of the total predicted quantity and is actually negligible. Thus, the prediction is mainly based on the feature patterns S_1 to S_4 shown in Figure 4.3.

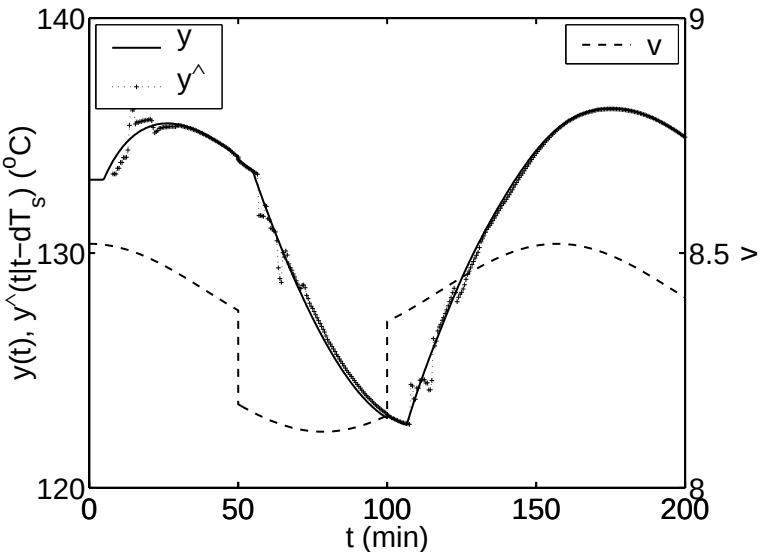

Figure 4.1: Typical curves of v, y, and $\hat{y}$.

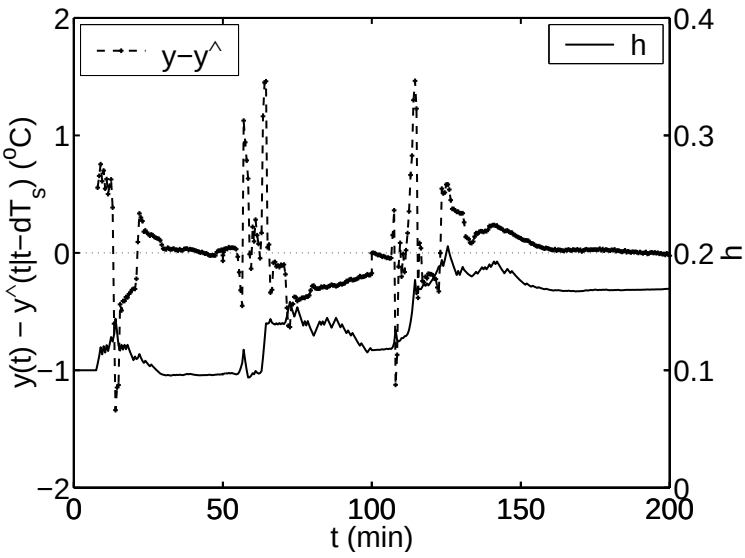

Figure 4.2: Typical $y(t) - \hat{y}(t|t - dT_s)$ and h curves corresponding to Figure 4.1.

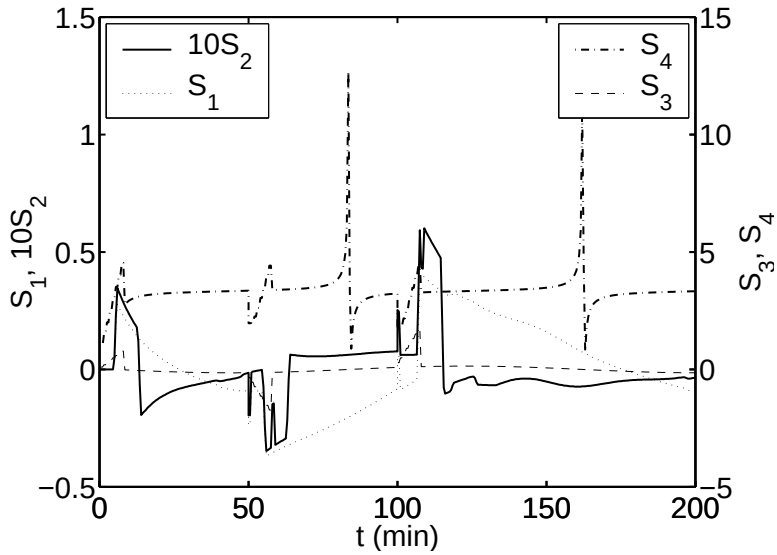

Figure 4.3: Typical curves of S_1 to S_4 corresponding to Figure 4.1.

The PP is incorporated with a conventional PI controller, which is tuned for set-point tracking for the index of the integral of time-weighted absolute error (ITAE). The controller settings are $K_c = 0.1331$ and $T_i = 22.1$min. Those settings can ensure the control performance and stability in a wide range of operating conditions.

The PPC performance is evaluated and compared with that of direct PI control. Both set-point tracking and disturbance rejection will be considered. The direct PI controller is also tuned in the range of operating conditions for set-point tracking for ITAE, giving more conservative settings $K_c = 0.0203$ and $T_i = 19.9$min, compared with the PI controller settings ($K_c = 0.1331$ and $T_i = 22.1$min) in the PPC system.

4.2 Set-Point Tracking

For evaluation of the performance of set-point tracking of the PPC system, a -5°C step change and a $+2$°C ramp change in T_7 set-point are introduced at time instants 50min and 150min, respectively.

The control results are given in Figure 4.4, which shows that less overshoot and shorter settling time are obtained from the PPC system. The PPC system improves the ITAE index by over 25% over the direct PI control, as depicted in Table 4.2 for comparisons of qualitative ITAE indices.

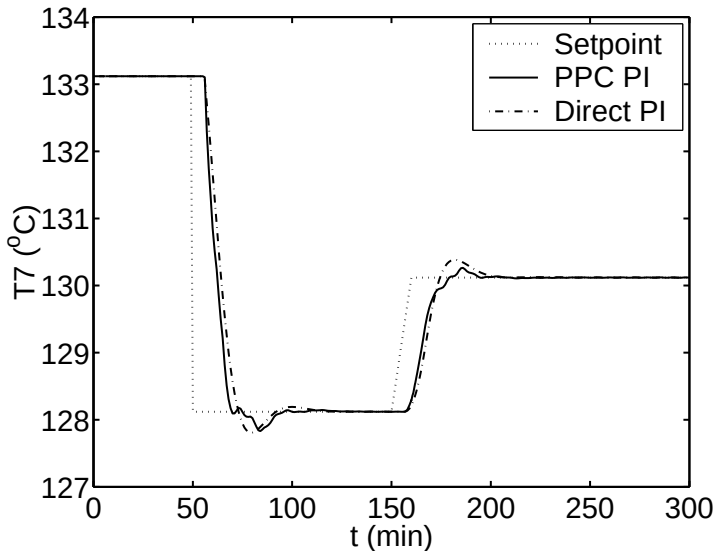

Figure 4.4: Set-point tracking.

Table 4.2: Comparisons of ITAE indices.

Magnitude	Period	Direct PI	PPC	Improve.
-5°C step in T_7	50-150min	657	479	27.1%
$+2$°C ramp in T_7	150-300min	434	313	27.9%
$+5\%$ step in F_c	50-150min	2529	1167	53.9%
-5% step in F_c	150-300min	2967	1175	60.4%
-10% step in F_f	50-300min	2461	1077	56.3%
$+10\%$ step in F_f	300-600min	2737	1210	55.8%

4.3 Rejection of Feed Composition Disturbances

Changes in feed composition, which is represented by the stoichiometric ratio, are disturbances to the RD column. $\pm 5\%$ (i.e. ± 0.25) step changes in feed composition are introduced to the process in order to test the disturbance rejection ability of the PPC system. The step changes occur at the time instants 50 and 150min, respectively.

Figure 4.5 depicts the control results, which show that PPC provides significant improvement over the disturbance rejection ability. The ITAE index improvement reaches over 50%, as shown in Table 4.2.

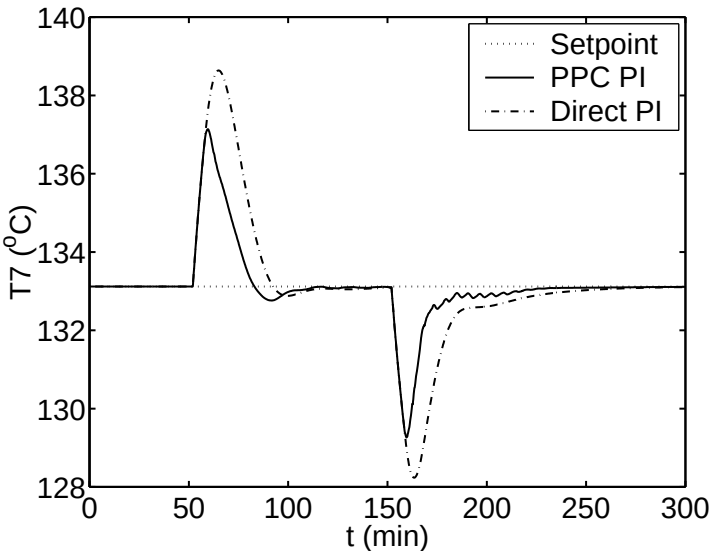

Figure 4.5: Rejection of $\pm 5\%$ disturbances in feed composition.

4.4 Rejection of Feed Rate Disturbances

Changes in feed flow rate are also disturbances to the RD column. $\mp 10\%$ (i.e. ± 0.074L/min) step changes in feed flow rate are introduced at the time instants 50 and 300min, respectively.

The control results are illustrated in Figure 4.6, which shows that disturbances in feed flow rate have severe effect on the RD process. The PPC system outperforms the direct PI control system as it rejects the disturbances much more effectively with much smaller ITAE indices (the improvement $> 55\%$), as shown in Figure 4.6 and Table 4.2. Some sort of ratio control involving the feed flow rate is an option for conventional PI control to improve the disturbance rejection performance, while previous studies have provided no incentive to configure a ratio control system in the studied RD column [Sneesby *et al.*, 1997b].

4.5 Remarks

It is worth mentioning that the RD process is severely non-linear. Therefore, system analysis and performance evaluation have to be made at the specific operating point. The nominal RD operating condition for this work is clearly listed in Table 3.1. Since this work addresses one-point control (purity control of the bottoms product), the reflux flow rate has been fixed at 2.53L/min. As a result, the degree of process non-linearity is reduced. This is why the

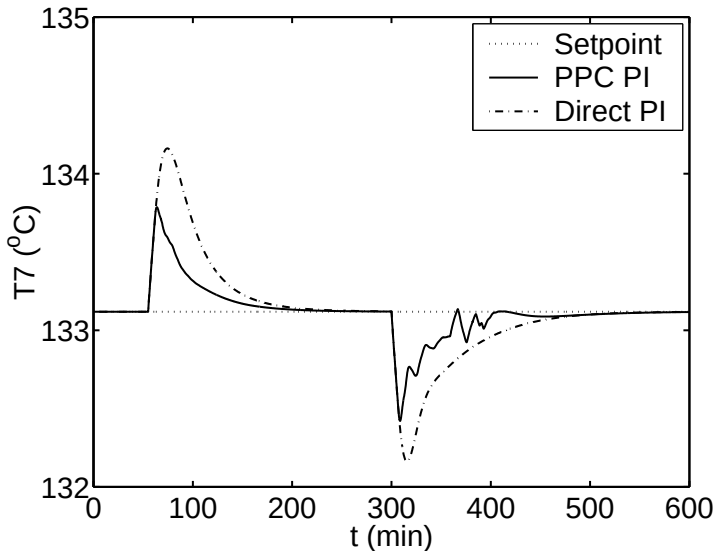

Figure 4.6: Rejection of $\mp 10\%$ disturbances in feed flow rate.

performance of the direct PI control is also acceptable in the case studies, although the PPC system does provide improvement over the direct PI control.

Through manipulating the reboiler duty Q_r, the stage 7 temperature T_7 is controlled as the indicator of the product purity. The nominal value of the purity is 90mol% (Table 3.1). The relationship between the purity and T_7 can be easily obtained from Figures 3.2 and 3.3.

4.6 Summary of This Chapter

Case studies have been conducted to verify the developed PPC scheme through integrated design of various aspects of pattern recognition, fuzzy logic, non-linear transformation, and predictive control for complex RD of ETBE. The system configuration has been discussed in detail. Then, the performance of the PPC scheme in set-point tracking and disturbance rejection has been quantitatively evaluated. For disturbance rejection, disturbances in both feed composition and feed rate are considered.

Quantitative evaluation shows that the PPC scheme incorporated with PI control improves the system performance significantly over the conventional PI control. For set-point tracking, the performance improvement is over 27%. For disturbance rejection, the performance is improved by over 50%. This indicates that the PPC is a promising tool for complex processes, where good process models are difficult to obtain or to implement for real-time control.

4.7 Nomenclature of This Chapter

The same nomenclature is used in this chapter (Chapter 4) and the last chapter (Chapter 3). It is listed below.

Abbreviations

ETBE	Ethyl *Tert*-Buty Ether
PI	Proportional-Integral
PP	Pattern Predictor/Predictive
PPC	Pattern Predictive Control
RD	Reactive Distillation

Symbols

$A_x^i \in \{B, M, S\}$	A linguistic value of FS_x for the *i*th rule, $x \in \{T/\theta, S_4, r_3\}$
B	Bottoms flow rate (L/min), or the linguistic term "Big"
$b,\ c$	Constants in the non-linear transformation
D	Distillate flow rate (L/min)
d	Number of prediction steps
f	Non-linear transformation within the PPC system
F_c	Feed composition (stoichiometric ratio)
F_f	Feed flow rate (L/min)
FS_x	Fuzzy set of the subscript variable $x \in \{T/\theta, S_4, r_3\}$
G_c	Controller
G_L	Process transfer function in disturbance path
G_p	Process transfer function in manipulation path
g	Intermediate function, at steady state $y = g(u)$
h	Updating factor for process prediction
k	Present sampling instant
K_c	Controller gain
K_p	Process gain (°C/kW)

L	Reflux flow rate (L/min) or Load
M	The linguistic term “Medium”
$p_1 \sim p_5$	Parameters for non-linear transformation f
Q_c, Q_r	Condenser duty and reboiler duty, respectively, (kW)
R	Set-point
$r_1 \sim r_3$	Parameters for prediction of the controlled variable
S	The linguistic term “Small”
$S_1 \sim S_4$	Process feature patterns
T	Process time constant
T_7	Stage 7 temperature (°C)
T_i	Controller integral time (min)
T_s	Sampling period (min)
u	Manipulated variable (e.g., reboiler duty)
V	Boilup flow rate (L/min)
v	Transformed manipulated variable in the PPC system
$V\left(A^i_{r_3}\right)$	Central numerical values of $A^i_{r_3} \in \{B, M, S\}$
V_T	Column top vapour flow rate (L/min)
y	Controlled variable (e.g., stage 7 temperature)
$\hat{y}$	d steps ahead prediction of y

Greek Letters

β_i	Degree of fulfilment for the *i*th rule
$\Delta\hat{y}$	d steps ahead prediction of the y deviation
ϵ	Small and positive threshold in S_2
μ	Membership function
θ	Process time delay (min)

Chapter 5

Reducing Control Latency and Jitter in Real-Time Control

Real-time control systems are typically deployed in hardware platforms with limited resources, e.g., uni-processor, resulting in various constraints in computing and scheduling of real-time control tasks of the systems. While schedulability has been the focus of conventional scheduling theory, it is not enough for real-time control systems. We need good control performance as well! In order to meet the requirements of both control performance and task schedulability in **implementation of control strategies on digital controllers**, well-designed scheduling of multiple real-time control tasks becomes critical. One way to address the control performance of a real-time control system with multiple control tasks is to characterise the control performance indirectly using the control latency and jitter. Then, scheduling methods are developed to minimise the latency and jitter or to make them more predictable. The resulting latency and jitter can be compensated through **integrated design of control and scheduling**.

This chapter analyses control computation latency and jitter, which have been largely ignored in the control community for real-time control. It indicates the importance of integration of control design and its software implementation explicitly and provides solutions to such integration. The focus of this work is on real-time control systems with multiple control tasks running on a uni-processor. Three strategies are proposed to reduce control latency and/or jitter: introduction of offsets into control task scheduling, decomposition of control tasks; and increasing the priority levels of output subtasks.

Part of this work has been published in [Tian *et al.*, 2006b].

5.1 Multi-tasking and Scheduling Algorithms

A periodic task is denoted by $T(c, d, p)$, where c, d, p are the worst case computation time, deadline, and period, respectively, $c \leq d \leq p$. The corresponding CPU utilisation is computed as $U = c/p$. Similarly, for n periodic tasks, each task is denoted by $T_i(c_i, d_i, p_i), i \in \{1, 2, \cdots, n\}$. The total CPU utilisation of the n tasks is $U = \sum_{i=1}^{n} c_i/p_i$. A necessary condition for schedulability on a uni-processor system is that the processor is not overloaded, i.e., $U \leq 1$.

The Rate-Monotonic (RM) algorithm is a fixed-priority (FP) scheduler that assigns the priority of each task according to its period: the shorter the period the higher the priority. With pre-emption, deadlines equal to periods, no blocking, no pre-period deadlines, and no aperiodic processes, RM scheduler is the optimal fixed-priority scheduling scheme [Liu and Layland, 1973]. This implies that under these conditions, if a task set cannot be scheduled using the RM, it cannot be scheduled using any fixed-priority algorithm.

The RM algorithm has a worst-case scheduling bound. For n periodic tasks, a sufficient condition for RM schedulability is the CPU utilisation satisfies the following relation $U \leq n\left[2^{(1/n)} - 1\right]$ provided there are no pre-period deadlines and no inter-task interference other than simple pre-emption. As the number of tasks increases, the schedulable bound decreases (and eventually approaches its limit $\ln 2 \approx 69.3$). Because the condition is not a necessary condition, sometimes a particular set of tasks will have a total CPU utilisation above the worst-case schedulable bound and still be schedulable with FP scheduling. Schedulability then depends on the specifications of the periods and execution times, which may be analysed using the completion time formula [Lehoczky *et al.*, 1989].

Deadlines of a task set are not necessarily equal to periods. In this case, an optimal fixed-priority scheduling is the Deadline-Monotonic (DM) algorithm [Leung and Whitehead, 1982]. The DM scheduler assigns priority of each task according to its deadline: the smaller the deadline the higher the priority. Under the conditions for which RM theory holds, DM and RM are equivalent.

Another FP scheduler is the Least-Compute-Time (LCT) algorithm, which assigns priorities in reverse order of the worst case computation time: the smaller the compute time the higher the priority.

While it is normally regarded as a dynamic scheduling technique, Earliest Deadline First (EDF) may also be used as a static list scheduler, where a list of tasks and time slots is provided to the system, which dispatches them in list order. EDF will in many cases produce a more optimal schedule than DM, but while the static form of EDF can handle periodic tasks with pre-period deadlines, it cannot handle aperiodic tasks.

This work considers multi-tasking scheduling in static environments. Generally, if fixed-priority scheduling is infeasible for a task set, dynamic scheduling has to be employed. An important scheduling paradigm, feedback scheduling in dynamic environments with uncertainties will be investigated in the next two chapters (Chapters 6 and 7) for real-time control.

5.2 Typical Scenarios in Control Task Scheduling

5.2.1 Single Control Task

A single control task $T(c, d, p)$ is schedulable as long as $c < d \leq p$. The base period is p; the CPU utilisation is $U = c/p$; the worst case response time is c; and the idle time in the base period is $p - c$. Therefore, the control latency is bounded by its worst case response time c, and the control jitter is within the range of the variations in the control computation delay. The control latency with the minimum jitter can be compensated through design of control strategies, e.g., [Tian and Gao, 1998c; Tian and Gao, 1999a], showing the importance of the co-design of control strategies and implementation even for a control system with a single control task.

5.2.2 Multiple Control Tasks with the Same Period

Consider a real-time control system with n control tasks $T_1(c_1, d, p)$ through $T_n(c_n, d, p)$ running on a uni-processor and having $d = p$. Assume that all tasks are released at the same time. The n tasks are schedulable as long as $\sum_{i=1}^{n} c_i \leq d = p$, and no pre-emption is required. The base period of the scheduling is p; the CPU utilisation is $U = \sum_{i=1}^{n} c_i/p$; and the idle time in the base period is $p - \sum_{i=1}^{n} c_i$. For $\forall T_i, i = 1, 2, \cdots, n$, the control latency is bounded by c_i plus a fixed task scheduling delay, and can be compensated through control design. Again, this shows the importance of integration of control design and its real-time implementation.

5.2.3 Multiple Control Tasks with Different Periods

Scheduling multiple control tasks with different periods is more complex. Consider three control tasks all with different periods, as shown in Table 5.1. The base scheduling period is 80 time units; the processor utilisation is $\sum_{i=1}^{3} c_i/p_i = 80\%$; and the idle time in the base period is 16 time units.

Non-pre-emptive and pre-emptive scheduling will give different scheduling results, and will have different effects on the performance of the overall real-time control system. They are depicted in Figure 5.1, which is based on the assumption that all tasks are released at the same time. Statistics of the scheduling are tabulated in Table 5.1.

Without pre-emption, there exists "priority inversion" in this example. This is clearly shown in Figure 5.1: the highest-priority task T_1 cannot start in the second period until the completion

Table 5.1: Response time density for the RM scheduling algorithm of three periodic control tasks $T_1(4, 10, 10), T_2(4, 16, 16)$, and $T_3(4, 40, 40)$.

Time	4	6	8	10	14	18	26
Non-pre-emptive							
T_1	0.75	0.125	0.125				
T_2	0.2	0.4	0.2	0.2			
T_3				0.5	0.5		
Pre-emptive							
T_1	1						
T_2	0.4	0.2	0.4				
T_3						0.5	0.5

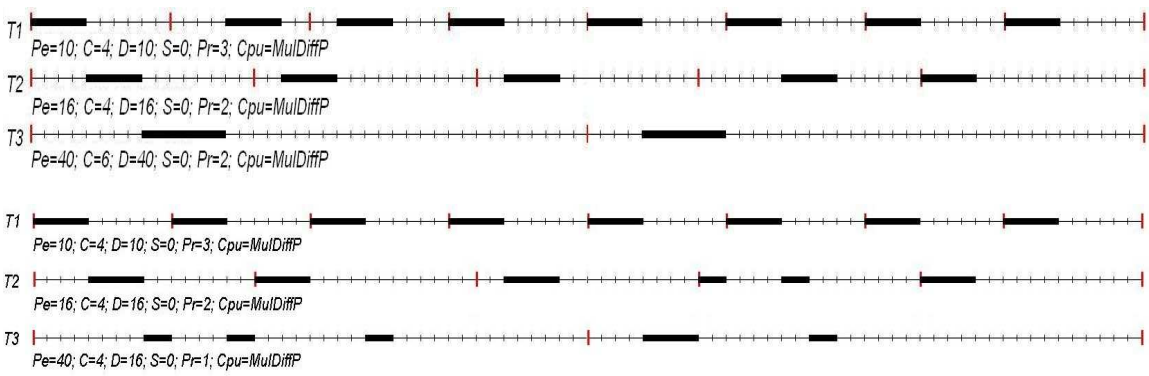

Figure 5.1: RM scheduling of three control tasks (upper plot: non-pre-emptive; lower plot: pre-emptive).

of the lowest-priority task T_3; and similarly, the highest-priority task T_1 cannot start in the third period until the lower-priority task T_2 completes.

Therefore, pre-emptive scheduling is preferred. It is seen from Figure 5.1 that tasks T_3 and T_2 are pre-empted by T_1 in the second and third period, respectively. Table 5.1 shows that pre-emption improves the system performance significantly: significant reduction of control jitter for T_1 and reduction of control jitter for T_2. However, T_3 control latency is increased significantly, which may not be acceptable in real-time control.

5.3 Introduction of Offsets to Lower-Priority Tasks

The first strategy for reducing control latency and jitter is to introduce offsets to lower-priority control tasks. For comparison, consider again the three-task example in Table 5.1. The offsets of 2 and 4 time units are introduced into T_2 and T_3, respectively. The corresponding non-pre-emptive and pre-emptive RM scheduling of the three control tasks are shown in Figure 5.2. The response time density of the scheduling are tabulated in Table 5.2.

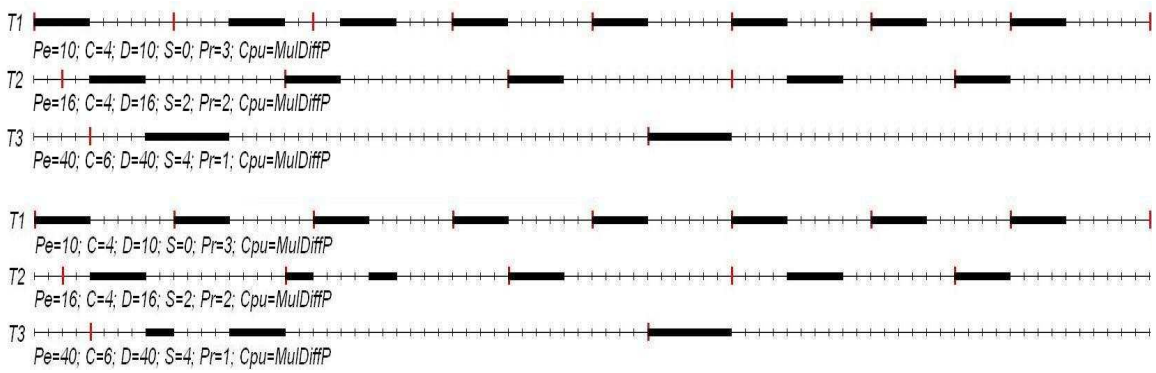

Figure 5.2: RM scheduling with offsets (upper: non-pre-emptive; lower: pre-emptive).

Now, compare Table 5.2 with Table 5.1. For non-pre-emption, T_2 with an offset has reduction in control latency and jitter. T_3 with an offset has more improvement in the upper bound of control latency (from 14 down to 10 time units), and keeps no change in control jitter.

For pre-emptive RM scheduling, it is seen from Tables 5.2 and 5.1 that there is no change in T_1 and T_2. As expected, the control latency of T_3 is reduced significantly. This is beneficial to the performance improvement of the overall control system.

Table 5.2: Response time density for three-task RM scheduling with offsets (Tasks are shown in Table 5.1).

Response Time	4	6	8	10	14
Non-pre-emptive					
T_1 (offset 0)	0.75	0.125	0.125		
T_2 (offset 2)	0.2	0.6	0.2		
T_3 (offset 4)		0.5		0.5	
Pre-emptive					
T_1 (offset 0)	1				
T_2 (offset 2)	0.4	0.2	0.4		
T_3 (offset 4)		0.5			0.5

5.4 Task Decomposition

Decomposition of a control task into smaller subtasks is feasible for control design and is also effective for reduction of control jitter. In [Balbastre *et al.*, 2004; Lluesma *et al.*, 2005], a task scheduling model with three-subtask decomposition is presented and a scheme for priority assignment is proposed. Using the same three-subtask decomposition, a different scheduling scheme is proposed here. The key step is to determine the deadline for each of the decomposed subtasks.

The detailed procedure of the scheduling scheme is described below.

Step 1. Decompose control task $T_i(c_i, d_i, p_i)$ into three subtasks $T_{i1}(c_{i1}, d_{i1}, p_i)$, $T_{i2}(c_{i2}, d_{i2}, p_i)$, and $T_{i3}(c_{i3}, d_{i3}, p_i)$, where $c_{i1} + c_{i2} + c_{i3} = c_i$ and $i \in \{1, 2, \cdots, n\}$. The first subscript of parameters represents the task identifier, while the second subscript denotes the decomposed subtask (we use 1, 2, and 3 to represent data acquisition, control computation, and control output phases, respectively).

Step 2. Determine the deadline for each of the decomposed subtasks as follows:

$$d_{i1} = \left\lceil \frac{c_{i1}}{c_i} d_i \right\rceil, d_{i2} = \left\lceil \frac{c_{i1} + c_{i2}}{c_i} d_i \right\rceil, d_{i3} = d_i \tag{5.1}$$

Because $c_i < d_i$, it follows that

$$d_{i1} > c_{i1}, d_{i2} > c_{i1} + c_{i2}, d_{i3} > c_{i1} + c_{i2} + c_{i3} \tag{5.2}$$

Step 3. Assign priority to the decomposed control subtask set using the DM algorithm. The smaller the deadline, the higher the priority.

It is worth mentioning that the deadlines for control subtasks T_{i1} and T_{i2} are soft deadlines. Therefore, missing these deadlines occasionally is still acceptable as long as the deadline for control subtask T_{i3} is not missed.

To illustrate the proposed scheduling scheme, we once again consider the three control tasks in Table 5.1. Shown in Table 5.3 are the decomposition of the control tasks, and deadlines and priority levels for decomposed subtasks. The scheduling results are tabulated in Table 5.4 with and without introduction of offsets into T_{i1}, T_{i2}, and T_{i3}. These results are comparable with those for the original control task set.

Table 5.3: Decomposition of control tasks T_1(4, 10 10), T_2(4, 16, 16), and T_3(6, 40, 40).

Task	T_1:	T_{11}	T_{12}	T_{13}	T_2:	T_{21}	T_{22}	T_{23}	T_3:	T_{31}	T_{31}	T_{33}
c		1	2	1		1	2	1		1	4	1
d		3	8	10		4	12	16		7	33	40
p		10	10	10		16	16	16		40	40	40
Priority		9	6	5		8	4	3		7	2	1

Table 5.4: Response time density for pre-emptive DM scheduling with task decomposition.

Time	4	5	6	7	8	9	14	18	26
T_1	0.17	0.58	0.25						
T_2	0.375		0.25		0.125	0.25			
T_3								0.33	0.67
T_1 offset 0	0.75			0.08	0.17				
T_2 offset 2	0.375			0.25	0.375				
T_3 offset 4			0.33				0.67		

It should be pointed out that with offset configuration, the subtask T_{12} missed its soft deadline by 1 time unit in the first and sixth periods. A careful analysis reveals that this does not affect the successful completion of the subtask T_{13} within its time frame, and at the worst case the control signal of the control task T_1 is output 4 time units earlier than its deadline. This implies that as a whole, T_1 does not miss its deadline.

5.5 Increasing Priority for Output

It is seen from last section that with control decomposition and pre-emptive DM scheduling, there exist control jitters for all three tasks. Jitters are difficult to compensate in control design and are a potential source of system instability [Tian and Gao, 1998c; Tian and Gao, 1998b; Tian and Gao, 1999a]. A scheduling method is proposed here to compress control jitters significantly. It is based on control task decomposition and the scheduling scheme developed in the last section. The basic idea is to increase the priority of control output subtasks and at the same time introduce an appropriate offset to each of control output subtasks. The scheduling procedure is discussed below.

Step 1. Apply the scheduling scheme developed in the last section to decompose the control task set, determine the deadlines for decomposed subtasks, and assign priority to the subtasks using the DM method. A set of subtasks $\{T_{ij} : i \in \{1, 2, \cdots, n\}; j \in \{1, 2, 3\}\}$ is then obtained.

Step 2. Schedule the decomposed control subtask set using the DM algorithm, and obtain the worst case response time, wrt_{i3}, for subtask $T_{i3}, i \in \{1, 2, \cdots, n\}$.

Step 3. Introduce into subtask T_{i3} an offset O_{i3}:

$$O_{i3} = wrt_{i3} - c_{i3}, i \in \{1, 2, \cdots, n\} \tag{5.3}$$

Step 4. Consider the output subtasks $T_{13}, T_{23}, \cdots, T_{n3}$. Without losing generality, suppose that their priorities are in descending order from T_{13} through T_{n3}. Increase the priorities of these output subtasks such that they are still in the same order and any of these priorities is higher than those of all remaining subtasks $\{T_{i1}, T_{i2} : i \in \{1, 2, \cdots, n\}\}$.

Step 5. Schedule the subtask set using priority based scheduling, and evaluate the scheduling results.

Let us reconsider the example discussed in Table 5.3. The resulting specifications are shown in Table 5.5, and the scheduling graph is depicted in Figure 5.3.

Table 5.5: Increasing priority for output subtasks (c, p, and d are the same as those in Table 5.3).

Task	T_1:	T_{11}	T_{12}	T_{13}	T_2:	T_{21}	T_{22}	T_{23}	T_3:	T_{31}	T_{31}	T_{33}
Priority		9	6	25		8	4	23		7	2	21
Offset				5				8				20

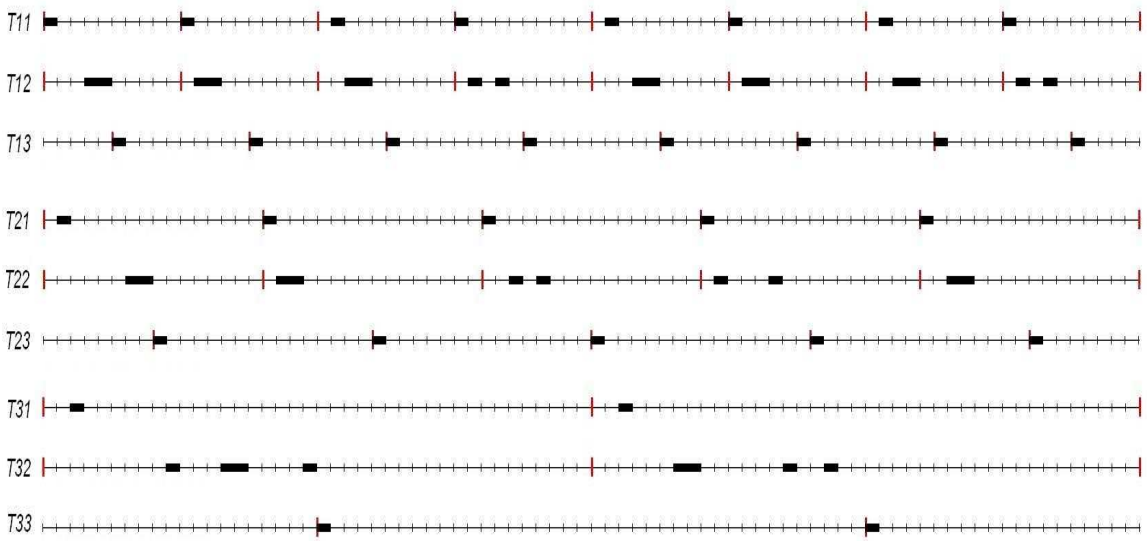

Figure 5.3: Pre-emptive scheduling with increased priority for control output subtasks.

Figure 5.3 shows that control signals are output at fixed time instants for all three control tasks. Control jitters are eliminated, implying that the control computation becomes predictable. This is favourable to real-time systems.

It is also observed from Figure 5.3 that T_1, T_2, and T_3 have fixed latencies of 6, 9, and 21, respectively. If these control latencies are not negligible, strategies can be developed in control design to compensate for the latencies. Again, this indicates the requirement of integrated design of control and its software implementation.

5.6 Applications to Multi-Robot Systems

The proposed technologies in this work have been applied to an industrial system: networked control and operation of a multi-robot system for fully automated food processing. This is the world-first production line of its kind in the world. The robot controller is the central controller running multiple tasks to control all robots and the whole system through Ethernet, CAN bus,

and other data networks. The system has been deployed to industry. Due to the confidentiality agreement with the industrial partner, further descriptions of the system are omitted.

5.7 Summary of This Chapter

Control latency and jitter resulting from control computation and scheduling in control software, and their impact on performance improvement have been analysed. This has been a largely ignored area in the control community. It is emphasised in this chapter that there is a requirement of integration of control strategy design and its software implementation on digital controllers.

Three strategies have been proposed for reduction of control latency and jitter. Introduction of offsets is a simple yet effective strategy. Decomposition of control tasks makes the control task scheduling flexible with improved performance. Increasing priorities of control output can eliminate control jitters and thus make the control computation predictable. The effectiveness of the proposed methods has been demonstrated through examples.

5.8 Nomenclature of This Chapter

Abbreviations

DM	Deadline-Monotonic
EDF	Earliest Deadline First
FP	Fixed Priority
LCT	Least-Compute-Time
RM	Rate-Monotonic

Symbols

c	Worst case computation time
d	Deadline
n	Total number of tasks
O	Offset
p	period

T	Task
U	CPU utilization
wrt	Worst response time

Subscripts

i	The first or sole subscript to c, d, O, p, T, wrt to indicate the ith task
j	The second subscript to some variables (c, d, O, T, wrt) to indicate the jth subtask decomposed from a task

Chapter 6

Hierarchical Feedback Scheduling of Real-Time Control Tasks

A real-time control system with multiple control tasks is expected to maintain good Quality of Control (QoC) as well as to meet the requirement of multi-tasking schedulability. One way that we have investigated in the last chapter to achieve this objective is to characterise the QoC indirectly using the control latency and jitter, and then to develop strategies to minimise the latency and jitter or to make them more predictable. Then, the resulting latency and jitter are compensated through integrated design of control and scheduling. An alternative way that will be explored in this chapter is to link the scheduling directly to the QoC of the control system, and then to develop dynamic and feedback scheduling policies to improve the system QoC subject to the resource constraints. This leads to tight coupling of control and scheduling, and again requires **integrated design of control and scheduling** for improved system QoC. Like the last chapter, this chapter also relates to **implementation of control on controllers**.

Compared with feedback scheduling for general real-time systems, feedback scheduling for control systems has different requirements because the primary objective of a control system is to maintain satisfactory QoC. The sampling periods are usually used as a parameter for feedback scheduling in control systems. When workload is high enough, low-priority tasks will have only a small chance to execute under fixed priority (FP) scheduling and on-line QoC monitoring for these tasks becomes difficult. The Earliest-Deadline-First (EDF) scheduling gives every task a chance to execute, but does not optimise the QoC directly. This chapter proposes a hierarchical and QoC driven feedback scheduling architecture for real-time control tasks. At the bottom

level, a task model is developed to decompose each control tasks into two smaller subtasks. At the intermediate level, the period of each task is adjusted locally according to the evaluated QoC. At the top level, through event-triggering from the QoC driven scheduling the control periods are re-scaled according to the requested workload. Both the FP and EDF policies will be considered in the proposed feedback scheduling.

6.1 Hierarchical Feedback Scheduling Architecture

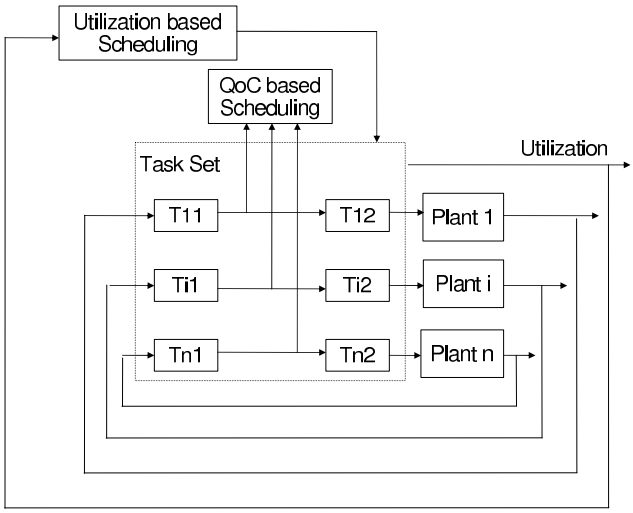

Figure 6.1: Block diagram of the hierarchical feedback scheduling structure.

The proposed hierarchical feedback scheduling architecture is shown in Figure 6.1. It consists of three levels: a task decomposition model at the bottom, QoC driven scheduling of control periods at the intermediate level, and a utilisation based scaling of periods at the top level. Both FP and EDF scheduling policies will be investigated.

The task decomposition model at the bottom level of the hierarchical feedback scheduling architecture decomposes each of the real-time control tasks into two smaller subtasks: data acquisition and QoC evaluation subtask, and control computation and output subtask. With appropriately assigned priority levels, all data acquisition and QoC evaluation subtasks will be run even when the system is overloaded. This ensures a close monitoring of the system QoC performance all the time. The evaluated QoC and the system workload are made accessible to the higher levels of the architecture for control period scheduling and re-scaling.

The intermediate level of the hierarchical feedback scheduling architecture is a QoC driven

scheduler of control periods. The control period of each control loop is allowed to be adjustable within its lower and upper bounds. The basic idea to adjust the control period of a control loop is to maintain the QoC performance of the control task within a certain level. When the QoC changes steeply, more frequent control actions will be performed; while slow changes in the QoC imply that the control period can be made bigger.

When the system is overloaded, the top-level scheduling of control periods is triggered. It simply scales up the control periods within their respective upper bounds based on the workload set-point. Unlike existing scheduling schemes, e.g., [Cervin *et al.*, 2002; Buttazzo *et al.*, 2007; Jin *et al.*, 2007; Xia *et al.*, 2005; Xia *et al.*, 2009b; Xia and Sun, 2005], the utilisation based period re-scaling scheme proposed in this work does not scale down the control periods when the system is under-loaded. This is because scaling down the control periods in this case does not help improve the QoC but wastes more system resources.

Because the data acquisition and QoC evaluation subtasks always run even in overload conditions, a hierarchical scheduling model, e.g., [Zhang and Burns, 2007], may also be appropriate and easy to use. However, our approach proposed here processes the information of each loop locally, and does not bring it up to higher levels. Consequently, the local QoC based scheduling can be effectively integrated with the data acquisition component. The top level scheduler is not always active; it is triggered only when it is necessary. Therefore, the three-level architecture is effective for saving energy and other computing resources. This is important for embedded systems with insufficient resources.

Various scheduling policies, e.g., FP, Rate-Monotonic (RM), Deadline-Monotonic (DM), and EDF, can be implemented as plug-ins in the proposed feedback scheduling architecture. This work will consider both the FP and EDF policies. The EDF is chosen because it is widely accepted as one of the major scheduling paradigms in dynamic environments. Our FP scheduling is derived from the RM scheduling, another major scheduling paradigm. The FP scheduling is widely supported in various platforms and even at application programming level.

6.2 Bottom Level: Task Model

A periodic control task is denoted by $T(c, d, p)$, where c, d, p are the worst-case execution time, deadline, and period, respectively. A single control task $T(c, d, p)$ is schedulable as long as

$c < d \leq p$, which can be interpreted as processor utilisation $U = c/p \leq 1$.

Now, consider a system with n periodic control tasks running on a uni-processor. The *i*th control task is denoted by $T_i(c_i, d_i, p_i), i \in \{1, 2, \cdots, n\}$. The total CPU utilisation of the n tasks is calculated as $U = \sum_{i=1}^{n} c_i/p_i$. It is known from scheduling theory that a necessary condition for schedulability of those tasks running on a uni-processor system is $U \leq 1$.

Because the environment of a system changes over time, the n periodic control tasks may overload the digital controller. When this happens, the overall QoC performance of the system deteriorates, and some of the control loops may even become unstable [Buttazzo *et al.*, 2007]. As demonstrated in [Cervin *et al.*, 2002], the key problem here is that some control tasks with lower priority will be blocked or significantly delayed in overload conditions when the FP scheduling is employed, making it impossible to evaluate the QoC of these loops and consequently to scale up the control periods promptly. Using similar ideas of [Balbastre *et al.*, 2004] and [Tian *et al.*, 2006b], a two-subtask decomposition model is developed in this work to resolve this problem at the bottom level of our hierarchical feedback scheduling framework.

With interrupt driven communications, decomposition of a control task into smaller subtasks is feasible for control design and also effective for reducing control jitter [Balbastre *et al.*, 2004; Lluesma *et al.*, 2005; Tian *et al.*, 2006b]. This work rectifies our approach [Tian *et al.*, 2006b] to develop a two-subtask model. While [Balbastre *et al.*, 2004; Tian *et al.*, 2006b] focus on reducing control latency and jitter, this work aims to evaluate the QoC of all control loops even in overload conditions, and then to use the QoC for scheduling of the control periods.

6.2.1 Task Decomposition

As in [Balbastre *et al.*, 2004; Tian *et al.*, 2006b], the first step of the new task decomposition approach is to decompose control task $T_i(c_i, d_i, p_i)$ into two subtasks $T_{i1}(c_{i1}, d_{i1}, p_i)$ and $T_{i2}(c_{i2}$, d_{i2}, $p_i)$, where $c_{i1} + c_{i2} = c_i$ and $i \in \{1, 2, \cdots, n\}$. It follows that

$$T_i = T_{i1} \cup T_{i2}, \forall i \in \{1, 2, \cdots, n\} \tag{6.1}$$

Two subscripts are used here to represent the attributes of the decomposed control subtasks: the first one is task identifier, and the second one indicates the decomposed subtask. For

simplicity, we use 1 to represent data acquisition and QoC evaluation phase, and 2 to denote control computation and control output phase, respectively. For example, T_{31} means the data acquisition and QoC evaluation phase of task 3.

For n control tasks, such a task decomposition gives $2n$ subtasks with data acquisition and QoC evaluation subtask set T^I and control computation and output subtask set T^{II},

$$T^I = \cup_{i=1}^{n} T_{i1}, T^{II} = \cup_{i=1}^{n} T_{i2} \tag{6.2}$$

The decomposition of the control tasks is graphically demonstrated in Figure 6.2. Priority levels of all subtasks are assigned in Figure 6.2 for FP scheduling. For each of the subtask sets T^I and T^{II}, the priority levels are determined from the RM policy.

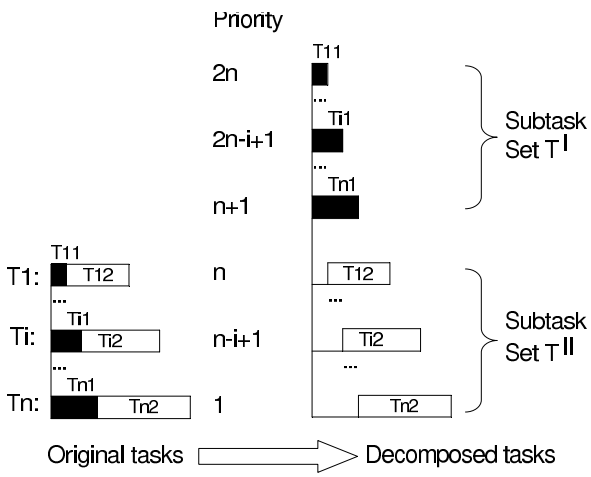

Figure 6.2: Task model for decomposition of n control tasks. The assignment of priority levels is valid only for FP scheduling.

6.2.2 Deadlines for Decomposed Subtasks

Similar to what we have done in [Tian *et al.*, 2006b], the second step of the new task decomposition approach is to determine the deadline for each of the decomposed subtasks as follows:

$$d_{i1} = \left\lceil \frac{c_{i1}}{c_i} d_i \right\rceil, d_{i2} = d_i, i \in \{1, 2, \cdots, n\} \tag{6.3}$$

It follows from the relationship $c_i < d_i$ that

$$d_{i1} > c_{i1}, d_{i2} = d_i > c_{i1} + c_{i2}, i \in \{1, 2, \cdots, n\} \tag{6.4}$$

It is worth mentioning that the deadlines d_{i1} for subtasks $T_{i1}, i = 1, 2, \cdots, n$ are soft deadlines. There is no hard requirement that these deadlines must not be missed.

6.2.3 Assignment of Priority Levels for FP Scheduling

The final step of the new task decomposition approach is to assign appropriate priority levels to the decomposed subtasks if FP scheduling is to be used.

For a system with n control tasks, in order to run each of the subtasks $\{T_{11}, T_{21}, \cdots, T_{n1}\}$ in every control period, the subtasks in the subtask set T^I are assigned priority levels higher than those of the subtasks in the subtask set T^{II}.

Without loss of generality, assume that the n tasks $\{T_1, T_2, \cdots, T_n\}$ have been arranged in the descending order of their priority levels. As shown in Figure 6.2, after decomposition of the control tasks, the subtasks in the subtask set T^{II} keep unchanged the original priority levels and order, while all subtasks in the subtask set T^I are assigned higher priority levels without changing the order of the priority. Therefore, for the ith task, we have the priority levels for the decomposed subtasks T_{i2} and T_{i1} as follows

$$\begin{aligned} &\text{Priority level:} \quad n - i + 1 \quad \text{for } T_{i2}\ \forall i \in \{1, 2, \cdots, n\} \\ &\text{Priority level:} \quad 2n - i + 1 \quad \text{for } T_{i1}\ \forall i \in \{1, 2, \cdots, n\} \end{aligned} \tag{6.5}$$

The workload of the n tasks $\{T_1, T_2, \cdots, T_n\}$ running on a uni-processor is calculated as

$$U = \sum_{i=1}^{n} c_i/p_i = U^I + U^{II}, \;\; U^I = \sum_{i=1}^{n} c_{i1}/pi, \;\; U^{II} = \sum_{i=1}^{n} c_{i2}/pi \tag{6.6}$$

where U^I and U^{II} are workload of the subtask sets T^I and T^{II}, respectively. While the workload U of all n tasks might be too heavy to make these tasks schedulable, it is almost certain that the workload U^I for the subtask set T^I is far below the full potential of the processor capacity, i.e., $U^I << 1$. This implies that even in FP scheduling, due to their higher priorities, the subtasks in the subtask set T^I will always have a chance to run in overload conditions. Therefore, the QoC can be evaluated for each of the control loops in every control period, leading to the possibility to re-scale the periods of the control tasks quickly under overload conditions. This will be discussed later in Section 6.4.

6.3 Intermediate Level: QoC Driven Scheduling of Control Periods

QoC can be characterised by some sort of performance indices [Buttazzo *et al.*, 2007], e.g. IAE, ITAE, quadratic cost function. While it is possible to evaluate a QoC index that requires high-cost computing, simplified QoC computation, e.g., linear approximation, has been shown to be effective for implementation of real-time control [Eker *et al.*, 2000; Cervin *et al.*, 2002]. In this work, the control error $\tilde{y}$, which is the difference between the set-point y_{sp} and actual output y of the controlled plant, and its one-step difference $\delta\tilde{y}$ are used to characterise the QoC.

In the work [Cervin *et al.*, 2002], the control periods of control tasks are adjusted to maintain the desired system workload. The QoC is considered indirectly, and thus is not guaranteed. The task decomposition model developed previously in Section 6.2 makes it possible to closely monitor the QoC for all control loops. When the QoC changes, the control periods are adjusted to maintain the QoC at its desired level.

For a specific control task $T_i, i \in \{1, 2, \cdots, n\}$, there are three scenarios for QoC:

1) If the QoC does not change or changes in a small region, there is no need to adjust the control period. Consequently, the workload of this control task will remain the same if other parameters, e.g., c_i, do not change.
2) If the QoC deteriorates, more frequent control actions will help avoid the QoC deteriorations. This requires reducing the control period. And
3) If the QoC is improving, less frequent control can be made to save system resources.

To help better understand the QoC driven scheduling of control periods, Figure 6.3 shows normalised output y of a plant and corresponding $\delta\tilde{y}$ under fixed sampling period of 0.25 time units. It is seen that the QoC changes significantly from one sampling period to another, especially in the first few sampling periods because of the steep change in $\tilde{y}$. In contrast, when the sampling period is allowed to be adjustable, better QoC can be maintained in each sampling period. As shown in Figure 6.4, adjusting sampling period capped with 0.5 time units will give the QoC within the pre-determined upper bound 0.08 in each period.

Under the pre-determined QoC upper bound Δ_H, the adjustment of the period can be expressed as $p_i := \dfrac{\Delta_H}{|\delta\tilde{y}|}p_i$. Considering the upper bound p_i^{max} and lower bound p_i^{min} of the

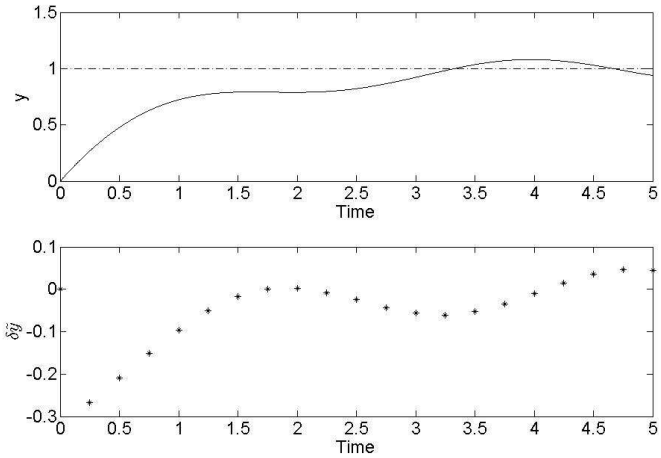

Figure 6.3: Plots of output y (upper plot) and one-step difference in control error $\delta\tilde{y}$ (lower plot) under fixed sampling period of 0.25 time units.

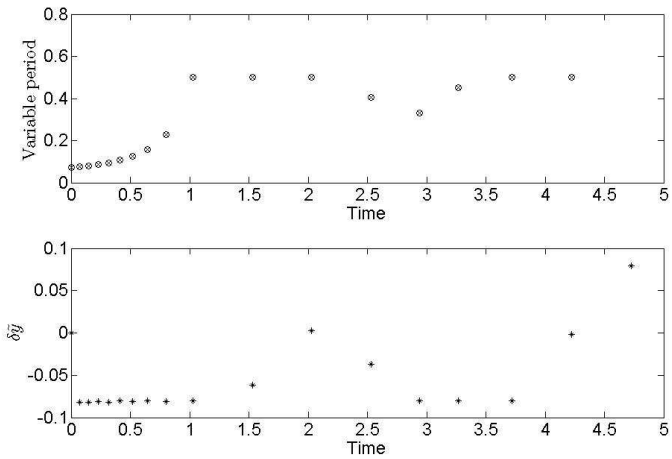

Figure 6.4: Variable sampling period with $p^{max} = 0.5$ (upper plot) and corresponding $\delta\tilde{y}$ (lower plot) with $\Delta = 0.08$.

period, the following strategy is proposed for $\forall i \in \{1, 2, \cdots, n\}$ for control period scheduling

$$p_i := \begin{cases} p_i^{max}, \text{ if } |\delta\tilde{y}| < \dfrac{p_i}{p_i^{max}}\Delta_H \text{ or } p_i = p_i^{max} \\ p_i^{min}, \text{ if } |\delta\tilde{y}| > \dfrac{p_i}{p_i^{min}}\Delta_H \text{ or } p_i = p_i^{min} \\ \dfrac{\Delta}{|\delta\tilde{y}|}p_i, \text{ else} \end{cases} \tag{6.7}$$

In implementation, the above control period scheduling strategy can be further improved for smooth operation. One improvement is to smooth out possible big fluctuations in control period

through some sort of filter. For example, Equation (6.7) can be modified to

$$p_i := \begin{cases} p_i^{max}, \text{ if } |\delta\tilde{y}| < \dfrac{p_i}{p_i^{max}}\Delta_H \text{ or } p_i = p_i^{max} \\ p_i^{min}, \text{ if } |\delta\tilde{y}| > \dfrac{p_i}{p_i^{min}}\Delta_H \text{ or } p_i = p_i^{min} \\ \epsilon p_i + (1-\epsilon)\dfrac{\Delta_H}{|\delta\tilde{y}|}p_i, 0 \leq \epsilon \leq 1, \text{ else} \end{cases} \tag{6.8}$$

where ϵ is a forgetting factor. Another improvement is to use a dead zone around $\delta\tilde{y} = \Delta_L < \Delta_H$, in which the scheduler does nothing, to avoid possible oscillations in a small range.

The algorithm for adjusting control period at intermediate level is shown in Figure 6.5.

Algorithm: Adjusting Control Period

Global c_i, p_i; //Execution time, control period

For the ith subtask T_{i1}:

  If $|\delta\tilde{y}| > \Delta_L$

    Then if $|\delta\tilde{y}| < \frac{p_i}{p_i^{max}}\Delta_H$ or $p_i == p_i^{max}$

      Then set $p_i := p_i^{max}$;

    Elseif $|\delta\tilde{y}| > \frac{p_i}{p_i^{min}}\Delta_H$ or $p_i == p_i^{min}$

      Then set $p_i := p_i^{min}$

    Else

      Set $p_i := \epsilon p_i + (1-\epsilon)\frac{\Delta_H}{|\delta\tilde{y}|}p_i, 0 \leq \epsilon \leq 1$

    End If

  End If

Figure 6.5: Intermediate level algorithm for adjusting control period.

Remark 1 *At the steady state of a controlled process, the control error tends to be zero and thus the control period of the control loop approaches it maximum value p^{max}. When the controlled output deviates from its set-point significantly, the control error becomes large and consequently the control period of the control loop tends to its minimum value p^{min}.*

6.4 Top Level: Event-Triggered Re-Scaling of Control Periods

At the top level of our hierarchical feedback scheduling architecture as shown in Figure 6.1 is an event-triggered scheduler for re-scaling of control periods. In many existing feedback schedulers, e.g., those in [Cervin *et al.*, 2002; Xia *et al.*, 2005; Xia *et al.*, 2009b; Xia and Sun, 2005],

a separate task at a high priority level must be activated. The feedback scheduler proposed here is different from these feedback schedulers in two aspects: (1) It is event-triggered; and (2) It is triggered only if the workload is higher than the desired value.

6.4.1 Event-Triggering

An advantage of using event-triggering, instead of a separate periodic task at a high priority level, is to effectively reduce the overhead of the scheduler and context switching. A periodic task will run automatically in every period, resulting in context switching every time when the task is activated. It is the only way in existing feedback schedulers to evaluate the QoC for all control loops, but does not function well under overload conditions if FP scheduling is used because some lower-priority tasks may be blocked or significantly delayed. The task decomposition model proposed in Section 6.2 makes it possible to evaluate the QoC for all control loops even when the system is overloaded. This alleviates the requirement of a periodic task at a high priority level, facilitating an event-triggering task to adjust the control periods.

Another advantage of event-triggering is its ability to respond to changes in system workload quickly. In the feedback schedulers that use a periodic task at a high level to adjust control periods, a compromise has to be made between the context switching overhead and the promptness to respond to workload changes. However, with event triggering, such a compromise is eliminated completely, implying more efficient feedback scheduling.

The algorithm, Event-Triggering, is shown in Figure 6.6 for the ith subtask T_{i1}, $i \in \{1, 2, \cdots, n\}$, which is the data acquisition and QoC evaluation subtask decomposed from the ith control task T_i. This algorithm executes for each of the subtasks in the subtask set $T^I = \{T_{11}, T_{21}, \cdots, T_{n1}\}$. It evaluates QoC, and adjusts the control period locally if necessary. Then, it updates the workload U of the overall system and also the workload U_i for the specific task

$$U := U - U_i + c_i/p_i, \;\; c_i = c_{i1} + c_{i2}, \;\; i \in \{1, 2, \cdots, n\} \tag{6.9}$$

$$U_i := c_i/p_i, \;\; c_i = c_{i1} + c_{i2}, \;\; i \in \{1, 2, \cdots, n\} \tag{6.10}$$

Finally, it checks if the system workload is heavier than its set-point, and if yes triggers the top-level scheduler to scale up the control periods for all control loops.

```
Algorithm: Event-Triggering
Global U, U_sp;    //system workload and its set-point
Global c_i, p_i;    //Execution time, control period
Global N_rq;    //Number of requests for scaling down control periods
For the ith subtask T_i1:
    Static U_i = 0;    //workload for task T_i
    Data acquisition;
    Evaluate QoC;
    Adjust control period p_i if necessary;  //Algorithm: Adjusting Control Period
    Update system workload U := U − U_i + c_i/p_i;
    Update the ith task workload U_i := c_i/p_i;
    If U > U_sp    //event happens
        N_rq := N_rq + 1;
        If N_rq > N_rq^max    //N_rq^max: predetermined upper bound of N_rq
            Reset N_rq := 0;
            Trigger the top-level scheduler  //Algorithm: ScalingUp Control Periods
        End If
    End If;
```

Figure 6.6: Algorithm for Event-Triggering.

It is worth mentioning again that when the system workload is lighter than its set-point, as long as the QoC meets the system requirement, it does not really make sense to scale down the control periods for the overall system. We even need to scale up the period to save more computing resources if a control loop reaches its steady state.

6.4.2 Scaling Up Control Periods under Overload Conditions

The top-level scheduler is triggered when the system workload is heavier than its set-point. Once triggered, the scheduler will check all control n control tasks, and will scale up the control periods or those tasks whose control periods are less than their respect maximum allowable values. The strategy to scale up the control periods is designed as

$$p_i := \frac{U}{U_{sp}} p_i \tag{6.11}$$

In this way, the workload of the system will be scaled down to its set-point U_{sp}. Similar idea has been adopted in [Cervin *et al.*, 2002], while the difference is that the scaling is used only for overload conditions in our scheme developed in this work.

The algorithm, ScalingUp Control Periods, is shown in Figure 6.7 for overload conditions.

```
Algorithm: ScalingUp Control Periods
Global U, U_sp;    //system workload and its set-point
Loop: For tasks from i = 1 to n do
    If p_i < maximum control period
        Scale up control period p_i := (U/U_sp) p_i;
        If p_i > maximum control period
            Set p_i := the maximum control period;
        End If;
    End If;
End Loop;
```

Figure 6.7: Top level algorithm for Scaling Up Control Periods.

6.5 Summary of This Chapter

Linking multi-tasking scheduling directly to QoC, a hierarchical feedback scheduling framework has been proposed for integrated design of control and scheduling of real-time control systems. The task decomposition model at the bottom level ensures that the QoC is closely monitored and evaluated all the time even under overload conditions. The intermediate level of the framework is a QoC driven scheduling of control periods. At the top level, utilisation based re-scaling of control periods is triggered only when the system is overloaded. The proposed scheduling framework will be verified in the next chapter (Chapter 7) through case studies.

6.6 Nomenclature of This Chapter

The same nomenclature is used in this chapter (Chapter 6) and the next chapter (Chapter 7). It is listed at the end of the next chapter (Section 7.10 of Chapter 7).

Chapter 7

Verification of Hierarchical Feedback Scheduling

Comprehensive case studies are carried out in this work to verify the hierarchical feedback scheduling framework developed in the last chapter (Chapter 6). The scheduling framework is applied to a control system with six control loops each of which is for a first-order plus time delay process. The performance of the control system is evaluated quantitatively under the hierarchical feedback scheduling, and compared with that under other typical scenarios. The results show that the proposed hierarchical scheduling is easy to implement and is effective to maintain the overall Quality of Control (QoC) performance of real-time control systems.

7.1 Process Dynamics and Controller Design

Consider the closed-loop control of the following first-order plus time delay process, which are common in process industry

$$G_p(s) = \frac{K_p}{T_p s + 1} e^{-\tau_p s} \tag{7.1}$$

where K_p, T_p, and τ_p are process gain, time constant, and time delay, respectively, with appropriate units. We consider six such processes with their respective settings shown in the first three columns of Table 7.1. A process model with these settings can represent the injection velocity process in a thermoplastic injection moulding machine [Tian and Gao, 1999b].

For each of the six processes, a Proportional-Integral (PI) controller G_c is employed with the transfer function

$$G_c(s) = K_c \left(1 + \frac{1}{T_c s}\right) \tag{7.2}$$

Table 7.1: Settings of six control loops with $K_p = 1$ and task execution time of 300μs, which is lumped into T_p. Priority levels are valid only for FP scheduling.

Plant	T_p	τ_p	Priority	Period	Utilisation	Nominal		PI Controller	
$G_i(s)$	(ms)	(ms)		p_i (ms)	$U_i(\%)$	p_i (ms)	U_i (%)	K_c	T_c (ms)
$G_1(s)$	8	5	1	0.8 $\sim$ 2.0	38 $\sim$ 15	1.4	21.43	1.07	8.63
$G_2(s)$	10	5	2	1.0 $\sim$ 2.5	30 $\sim$ 12	1.7	17.65	1.34	10.55
$G_3(s)$	10	6	3	1.0 $\sim$ 2.5	30 $\sim$ 12	1.7	17.65	1.14	10.74
$G_4(s)$	12	6	4	1.2 $\sim$ 3.0	25 $\sim$ 10	2.0	15.00	1.36	12.66
$G_5(s)$	12	7	5	1.2 $\sim$ 3.0	25 $\sim$ 10	2.0	15.00	1.18	12.85
$G_6(s)$	15	7	6	1.5 $\sim$ 3.8	20 $\sim$ 8	2.5	12.00	1.48	15.74
$K_p = 1$ for all plants				$\sum_i(U_i)$	168 – 67	$\sum_i(U_i)$	98.73		

where K_c and T_c are controller gain and integral time, respectively. There are many methods for PI controller tuning. In this work, the PI controller is tuned using the ITAE tuning for set-point. The standard ITAE PI settings for set-point are

$$K_c = 0.586 K_p^{-1} T_p \tau_p^{-0.916}, T_c = T_p \left(1.030 - 0.165 \frac{\tau_p}{T_p}\right)^{-1} \tag{7.3}$$

The resulting controller settings for the six control loops are depicted in the last two columns of Table 7.1. Since the process dynamics and controller design are not the main interest of this work, we will focus on scheduling of multiple control tasks on a uni-processor below.

7.2 Settings for Control Tasks

The control frequency, i.e., the inverse of the control period, for each control loop is chosen in the range of 4 to 10 times the characteristic frequency, which corresponds to -3dB magnitude decrease in the Bode diagram. Specifically, for a first-order process with time constant of T_p, the control period is chosen to be $T_p/10 \sim T_p/4$. The resulting ranges of the control periods for all six control loops are listed in Table 7.1.

Without loss of generality, assume that all six control tasks have the same execution time of 300μs, which is lumped into the process time delay τ_p. The corresponding processor utilisations for the tasks are shown in Table 7.1. The total utilisation ranges from 67% for the maximum control periods to 168% for the minimum control periods. This implies that the processor will

be overloaded when the smallest control periods are employed. In order not to overload the processor, nominal control periods are determined which give the total utilisation of just under 100%, as depicted in Table 7.1. They are still smaller than the maximum control periods, and thus are expected to deliver satisfactory control performance.

The priority levels of the control tasks are determined according to the Rate-Monotonic (RM) rules, i.e., the shorter the period, the higher the priority. If two tasks have the same period, assign the priority in an arbitrary order. As shown in Table 7.1, the priority levels of the control loops of $G_1(s)$ to $G_6(s)$ are assigned to be 1 to 6, respectively. They are valid only for fixed priority scheduling.

7.3 Scenarios to Be Tested

To evaluate the control performance, consider set-point tacking for all six loops. Set-point changes are designed to test various scenarios, as shown in Table 7.2. The TrueTime package [Ohlin *et al.*, 2007] is used to carry out the simulation studies. Both the fixed priority (FP) scheduling and Earliest-Deadline-First (EDF) scheduling will be tested.

Table 7.2: Step changes in set-point for all six control loops.

Control Loop	0ms	100ms	200ms	300ms	400ms	500ms
$G_1(s)$	$+1$	-1	$+1$	-1	$+1$	-1
$G_2(s)$	$+1$	-1	$+1$	-1	$+1$	
$G_3(s)$	$+1$	-1	$+1$	-1		
$G_4(s)$	$+1$	-1	$+1$			
$G_5(s)$	$+1$	-1				
$G_6(s)$	$+1$					

Because the QoC of a control loop largely depends on the period [Buttazzo *et al.*, 2007; Cervin *et al.*, 2002], the QoC computed using the changing control period in the range [p^{min}, p^{max}] would become incomparable for various scheduling schemes. We choose to use the same and fixed time step of 1ms to compute the integral type of control performance, e.g., ITAE, for all scheduling schemes. This implies that interpolation will be carried out for computation of the integral control performance.

7.4 Performance Evaluation under the Minimum Control Periods

When the minimum periods are used as defined in Table 7.1, the processor is overloaded with the requested utilisation of 168%.

7.4.1 The FP Scheduling

In overload conditions, the first three control loops which give the actual processor workload of 98% will execute as expected, while the other three loops will have no chance to run. Therefore, the overall performance of the system is not acceptable at all.

The FP scheduling of all six tasks are depicted in Figures 7.1 and 7.2. As shown in Figure 7.1, Task 1 with the highest priority runs well; Tasks 2 and 3 runs smoothly, which are preempted by tasks with higher priorities. Figure 7.2 shows that Tasks 1 to 3 give satisfactory performance.

Figures 7.1 and 7.2 also indicate that Task 4 does not function, and Tasks 5 and 6 never run; these three tasks do not track the set-point changes at all.

The ITAE indices of the controlled outputs of the six control loops are computed for FP scheduling, and are summarised in Table 7.3 under the MinPeriod method with FP.

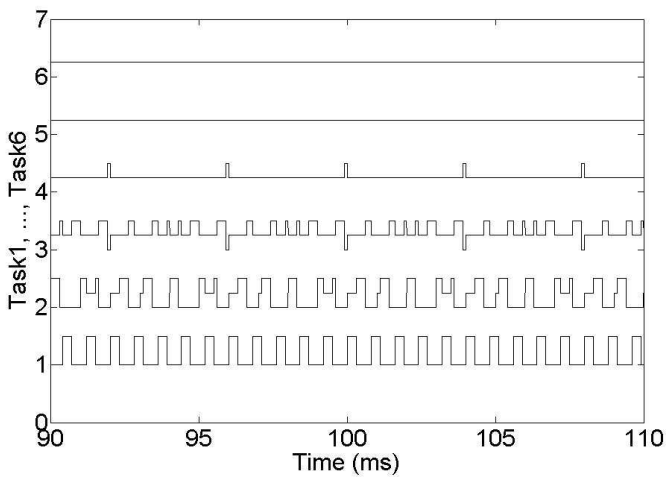

Figure 7.1: FP scheduling under the minimum control periods.

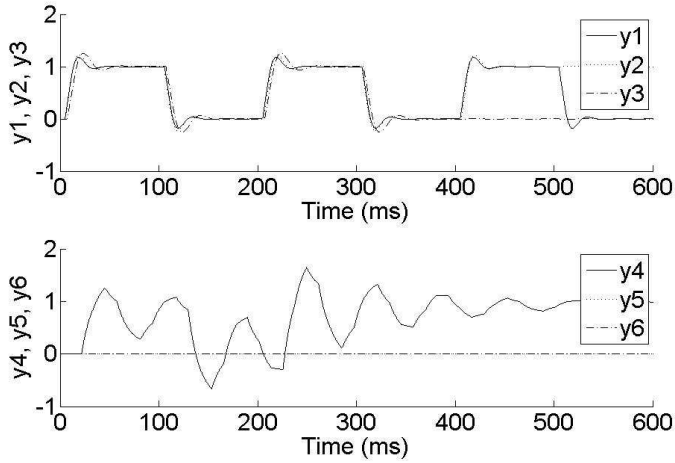

Figure 7.2: Control under the minimum control periods and FP scheduling.

Table 7.3: ITAE indices under fixed periods and FP (The results are computed using a fixed time step of 1ms).

Loop	0-100ms	100-200ms	200-300ms	300-400ms	400-500ms	500-600ms
MinPeriod with FP: overloaded (requested workload 168%)						
G1	91.17	100.53	91.424	100.69	91.624	91.274
G2	110.58	109.90	120.48	111.73	121.68	–
G3	193.65	196.24	194.91	195.84	–	–
Loops of G4, G5, and G6 are not functional.						
MaxPeriod with FP: workload 67%						
G1	127.79	154.32	153.68	156.21	154.16	152.21
G2	156.90	156.42	156.82	155.21	154.86	–
G3	210.77	212.21	211.73	211.71	–	–
G4	258.17	300.09	241.00	–	–	–
G5	305.00	362.16	–	–	–	–
G6	392.18	–	–	–	–	–
Nominal with FP: fully loaded (workload 99%)						
G1	107.14	117.07	109.47	119.57	112.26	121.51
G2	130.57	129.64	130.19	145.06	141.87	–
G3	176.66	181.62	179.00	187.94	–	–
G4	213.76	251.40	246.44	–	–	–
G5	320.78	281.44	–	–	–	–
G6	824.13	–	–	–	–	–

7.4.2 The EDF Scheduling

Under the EFD scheduling, all control tasks have chance to execute even in overload conditions of 168% requested workload. This is clearly shown in the scheduling plot of Figure 7.3. The output signals of the controlled processes under EDF scheduling are shown in Figure 7.4, and the ITAE indices of the controlled outputs are tabulated in Table 7.4 under the MinPeriod method with EDF scheduling. It is seen from Tables 7.4 and 7.3 that the EDF outperforms the FP in this specific scenario if the platform of the target system supports the EDF.

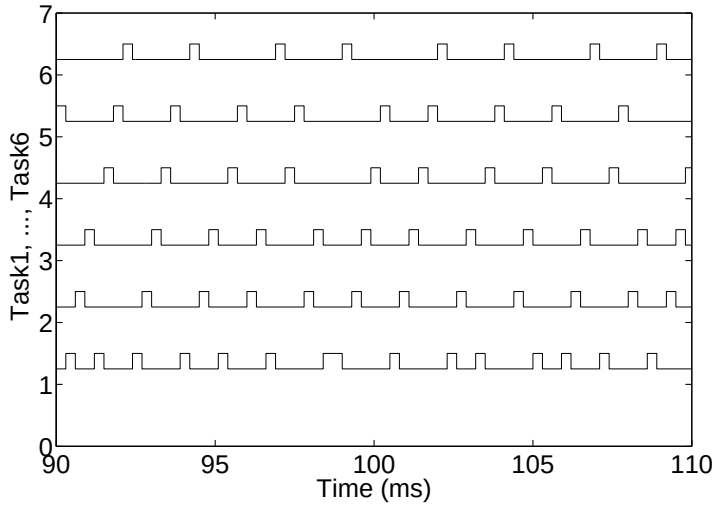

Figure 7.3: EDF scheduling under the minimum control periods.

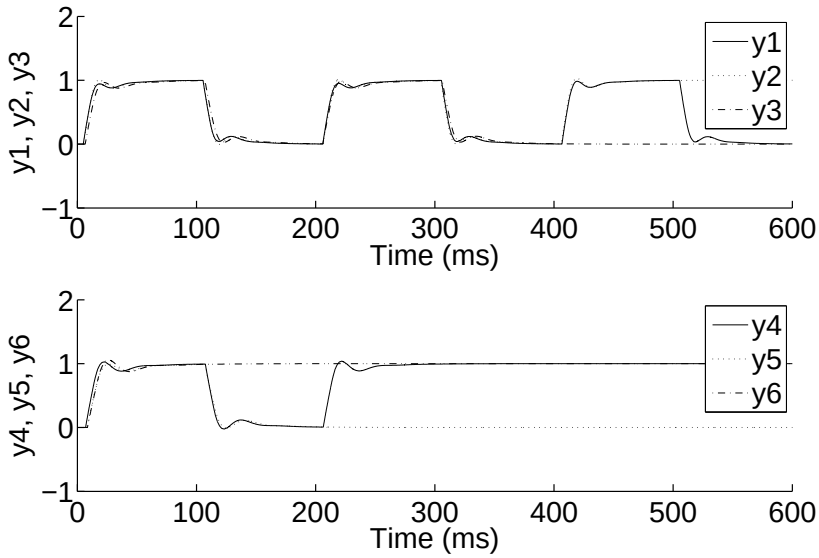

Figure 7.4: Control under the minimum control periods with EDF scheduling.

Table 7.4: ITAE indices under fixed periods and EDF (The results are computed using a fixed time step of 1ms).

Loop	0-100ms	100-200ms	200-300ms	300-400ms	400-500ms	500-600ms
MinPeriod with EDF: overloaded (requested workload 168%)						
G1	175.47	176.25	184.95	175.27	174.00	168.38
G2	179.24	180.96	187.25	175.39	170.94	-
G3	244.33	243.55	224.21	238.47	-	-
G4	232.40	239.43	214.34	–	–	–
G5	298.06	263.50	–	–	–	–
G6	308.54	–	–	–	–	–

7.5 FP Scheduling under the Maximum Periods

If the maximum control periods are implemented, the requested processor utilisation is 67%. All control tasks can execute as expected under FP scheduling. Figure 7.5 shows the corresponding multi-tasking scheduling. The output signals of the controlled processes in response to the step changes in set-points (Table 7.2) is depicted in Figure 7.6. The corresponding ITAE indices of the control loops are summarised in Table 7.3 under the MaxPeriod method with FP.

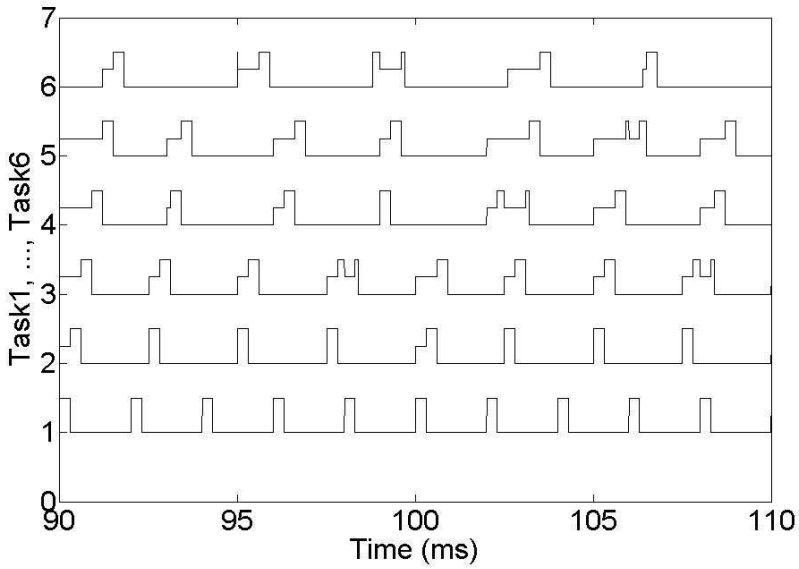

Figure 7.5: FP scheduling under the maximum control periods.

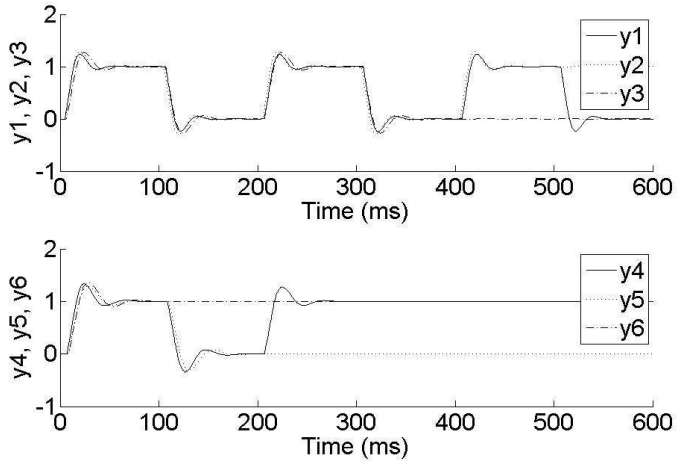

Figure 7.6: Control under the maximum control periods and FP scheduling.

7.6 FP Scheduling under Nominal Control Periods

With the settings under nominal control periods (Table 7.1), the requested processor utilisation is about 98.73%. Therefore, all six control tasks will have a chance to execute under FP scheduling. However, becuase the system is almost fully loaded, good performance cannot be expected for the control task with the lowest priority, e.g., control loop $G_6(s)$ in our case studies, because the task is often pre-empted by control tasks with higher priorities. This has been shown in Figure 7.7 for multi-tasking scheduling, and in Figure 7.8 for output signals of the controlled processes in response to the step changes in set-points (Table 7.2). As tabulated in Table 7.3, the quantitative ITAE indices indicate that compared with the MaxPeriod method with FP, the NominalPeriod method with the same scheduling policy improves the performance for all control loops except the control loop $G_6(s)$. Because the performance of the control loop $G_6(s)$ degrades significantly and the requested processor utilisation is close to 100%, the NominalPeriods method is not an idea method under FP scheduling.

7.7 Task Decomposition with Fixed Control Periods

To employ the hierarchical feedback scheduling proposed in this work, we first develop a new task model in which each of the original six control tasks is decomposed into two subtasks: data acquisition and QoC evaluation, and control computation and output. Without loss of the generality, the settings of the decomposed tasks and their priorities are shown in Table 7.5.

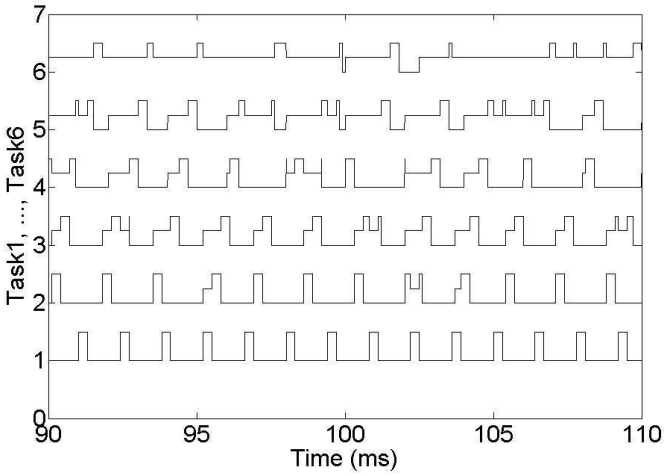

Figure 7.7: FP scheduling under nominal control periods.

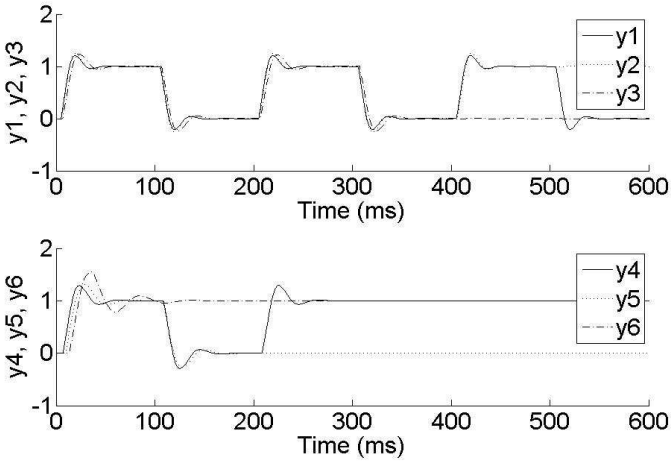

Figure 7.8: Control under nominal control periods and FP scheduling.

We have tested the new task model, in which the original six tasks are decomposed into twelve subtasks, under fixed task periods and FP. As shown in Figures 7.9 and 7.10, all subtasks T_{11} through T_{61} are executed regularly to acquire data and monitor the QoC in real-time for any settings of fixed task periods, from the minimum ones to the maximum ones.

For the minimum task periods, the subtasks T_{21} and T_{31}, which are respectively numbered as 7 and 8 in Figure 7.9, have chance to run to control processes $G_1(s)$ and $G_2(s)$, respectively. However, all other subtasks T_{41} and T_{61}, which are respectively numbered as 9 and 12 in Figure 7.9 will never run under FP, implying these loops are not functional under FP scheduling.

Table 7.5: Task model in the case studies. Top level scheduler task T_0. Sampling and control task set $T_i = T_{i1} \bigcup T_{i2}$, $i = 1, 2, \cdots, 6$. Control periods are the same as those in Table 7.1. Priority levels are valid only for FP scheduling.

Settings	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$	$G_5(s)$	$G_6(s)$
	T_{11}, T_{12}	T_{21}, T_{22}	T_{31}, T_{32}	T_{41}, T_{42}	T_{51}, T_{52}	T_{61}, T_{62}
Priority	2, 8	3, 9	4, 10	5, 11	6, 12	7, 13
Period (ms)	0.8 - 2.0	1.0 - 2.5	1.0 - 2.5	1.2 - 3.0	1.2 - 3.0	1.5 - 3.8
Top level scheduler task T_0 with the highest priority of 1.						
Execution times $c_0 = 10\mu$s; $c_{i1} = 50\mu$s and $c_{i2} = 250\mu$s, $i = 1, 2, \cdots, 6$.						
Deadlines $d_0 = 1$ms; d_{i1} and d_{i2}, $i = 1, 2, \cdots, 6$, are calculated using Eqn. (6.3).						
Other Settings: $\Delta_H = 0.05; \Delta_L = 0.01, \epsilon = 0.9, U_{sp} = 0.92, N_{rqsp} = 5$.						

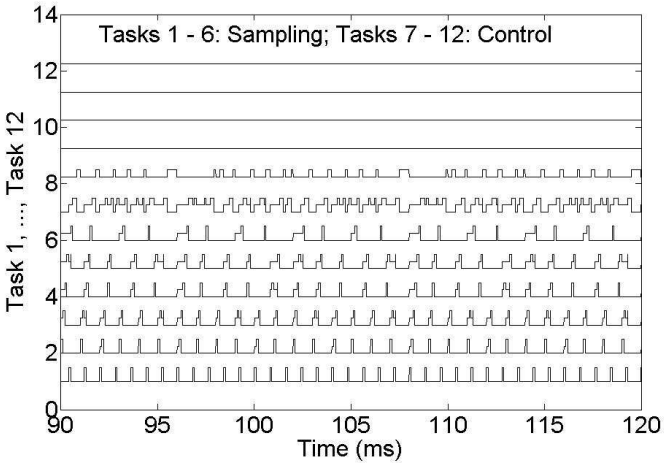

Figure 7.9: Task decomposition with the minimum control periods and FP.

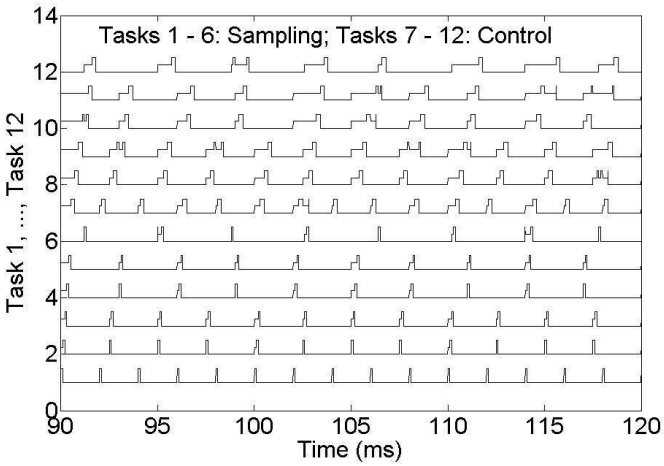

Figure 7.10: Task decomposition with the maximum control periods and FP.

For the maximum task periods, all subtasks T_{21} through T_{61}, which are respectively numbered as 7 through 12 in Figure 7.10, can be periodically executed under FP as expected although the QoC performance can be improved significantly.

7.8 Task Decomposition with Variable Control Periods

With the task decomposition model, we are able to employ the hierarchical feedback scheduling method proposed in this to improve the scheduling and QoC performance of the control system. Some settings of the scheduling are tabulated in Table 7.5.

7.8.1 The FP Scheduling Based on Task Decomposition

We start to execute all tasks under FP using the maximum periods shown in Tables 7.1 and 7.5. The initial processor utilisation is 67%. With the on-line evaluation of the QoC of the controlled processes, the periods of all tasks are adjusted on-line and dynamically. Figure 7.13 depictes the adjustment of the task periods, where p_i represents the period of the subtasks T_{i1} and T_{i2}, $i = 1, 2, \cdots, 6$. Figure 7.12 shows the plot of the requested processor utilisation. Adjustment of the periods can also be observed from this plot they directly link with the requested utilisation.

To clearly show that all tasks have been made schedulable under the variable-period FP scheduling, Figure 7.13 depicts the plot of the FP scheduling over the time. Process responses to step changes in set-points under variable-period FP scheduling of the decomposed tasks are shown in Figure 7.14. The ITAE indices of the controlled processes under variable-period FP scheduling of the decomposed tasks are tabulated in Table 7.6.

We have the following discussions:

1) When step changes in set-points of all six processes are introduced at 0ms, big control errors appear to all processes (Figure 7.14). The intermediate level scheduler reduces the periods to improve the QoC (Figure 7.11); and the requested utilisation increases till the processor is almost fully loaded (Figure 7.12). When the requested utilisation is over the desired value ($U_{sp} = 92\%$ in this example) and the number of requests for scaling up the periods reaches the pre-determined value ($N_{rq} = 5$ in this example), the top level scheduler is activated to increase the periods (Figure 7.13). When all processes are approaching their steady states, the control

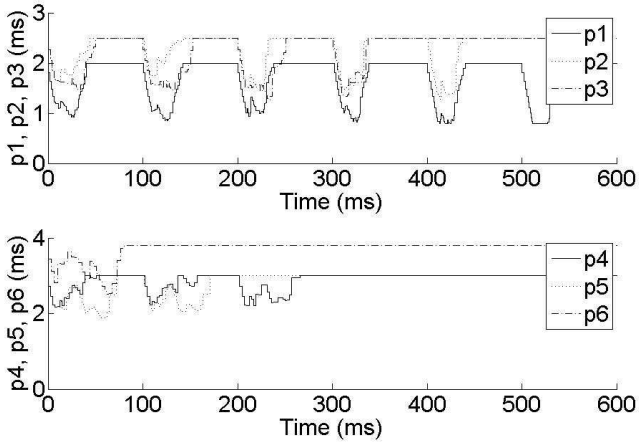

Figure 7.11: Dynamical adjustment of task periods under FP.

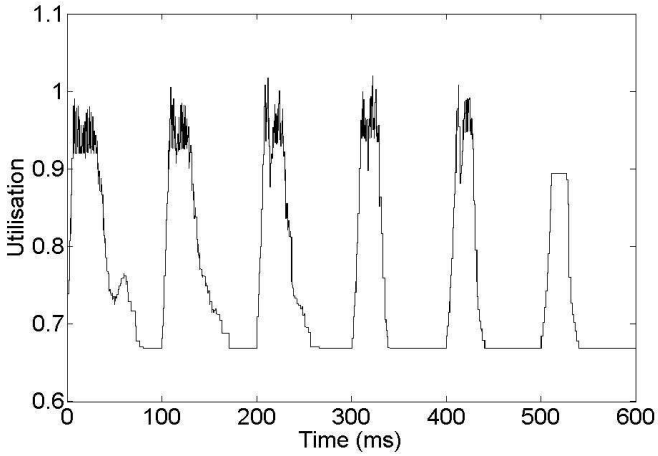

Figure 7.12: Dynamical adjustment of processor utilisation under FP.

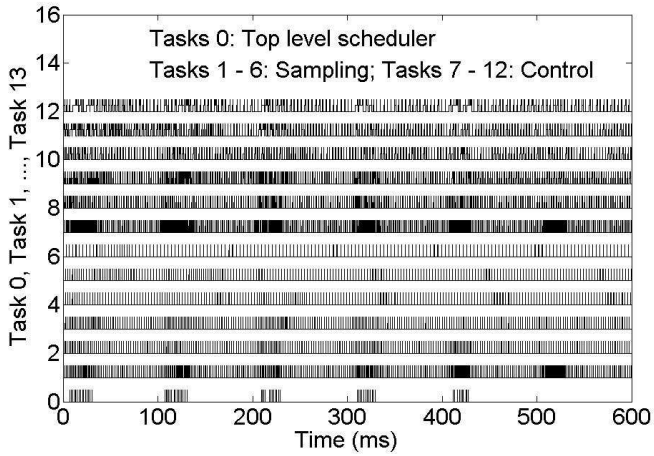

Figure 7.13: Variable-period FP scheduling of the decomposed tasks.

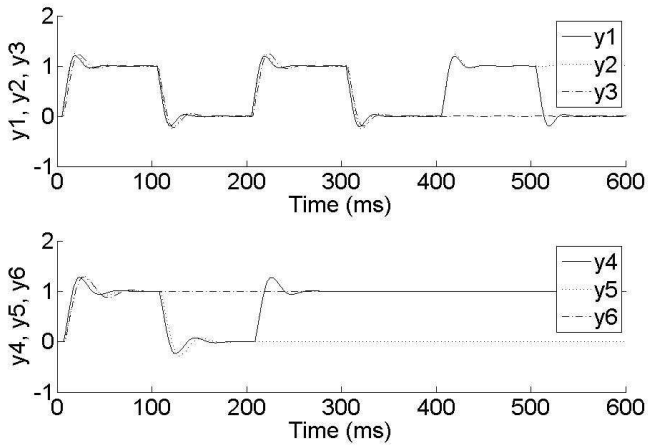

Figure 7.14: Process responses to step changes in set-points under variable-period FP scheduling of the decomposed tasks.

Table 7.6: ITAE indices under variable periods and flexible workload (The results are computed using a fixed time step of 1ms. The average workload is 77%).

Loop	0-100ms	100-200ms	200-300ms	300-400ms	400-500ms	500-600ms
ThisWork with FP						
G1	105.72	106.90	104.13	103.35	106.86	101.53
G2	133.77	153.61	124.13	154.48	128.81	–
G3	168.45	169.62	177.28	155.82	–	–
G4	214.81	235.28	250.39	–	–	–
G5	310.61	313.50	–	–	–	–
G6	375.79	–	–	–	–	–
ThisWork with EDF						
G1	99.30	122.82	110.43	103.18	101.87	104.87
G2	123.66	122.38	117.80	130.33	122.83	–
G3	166.98	183.25	177.02	192.79	–	–
G4	206.43	236.52	224.12	–	–	–
G5	274.77	248.61	–	–	–	–
G6	316.93	–	–	–	–	–

errors become smaller and smaller. The QoC can be well maintained with bigger periods, and thus the intermediate level scheduler gradually increases the periods to their maximum values corresponding to the requested utilisation of about 67% (Figure 7.12).

2) At 100ms, step changes are introduced into the set-points of the first five processes, resulting in big control errors for these processes (Figure 7.14). The control periods will be

dynamically adjusted for these processes for QoC improvement. Because $G_6(s)$ is at its steady state and can be well controlled using the maximum period, more processor resources can be used for controlling the first five processes (Figure 7.11).

3) Similarly, as shown in Figures 7.11 to 7.14, when step changes are introduced to the set-points of some of the six processes at 200ms, 300ms, and 400ms, the proposed scheduling strategies under FP can well adjust the control periods of the corresponding control tasks.

4) It is also seen from Figures 7.12 and 7.13 that the top level scheduler will not be activated when the requested utilisation is below its set-point U_{sp}. It is likely to be activated when it is required to improve the QoC, e.g., when there are changes in set-point for one or more processes. Our results show that from 0ms to 100ms, the top level scheduler is only activated for 15 times in the first 32ms, and remains idle for the remaining times.

5) It is also observed from Figure 7.12 that unlike many existing scheduling methods, the proposed method does not maintain the processor utilisation at a desired value but normally below a desired value. Over the time-span from 0ms to 600ms, the average requested processor utilisation is about 77%. This saves system resources without sacrifice of the system QoC.

6) A comparison between Tables 7.6 and 7.3 shows that, overall, the ITAE indices under the variable-period FP scheduling based on task decomposition are quite comparable with those under the fixed-period FP scheduling with nominal periods for the control loops of G_1 to G_5, and are improved significantly for the control loop of G_6. The control system under the proposed scheduling with FP consumes only 77% of the processor resources on average, compared to the constant 98% under the FP scheduling with fixed nominal periods.

7) Comparisons between Tables 7.6 and 7.4 reveal that the ITAE indices under the variable-period FP scheduling with task decomposition are better than those under fixed-period EDF scheduling without task decomposition.

7.8.2 The EDF Scheduling Based on Task Decomposition

Now, let us embed the EDF into our hierarchical feedback scheduling framework. Again, we start to execute all tasks under EDF using the maximum periods shown in Tables 7.1 and 7.5, and then to dynamically adjust the control periods to maintain satisfactory QoC. Figure 7.15

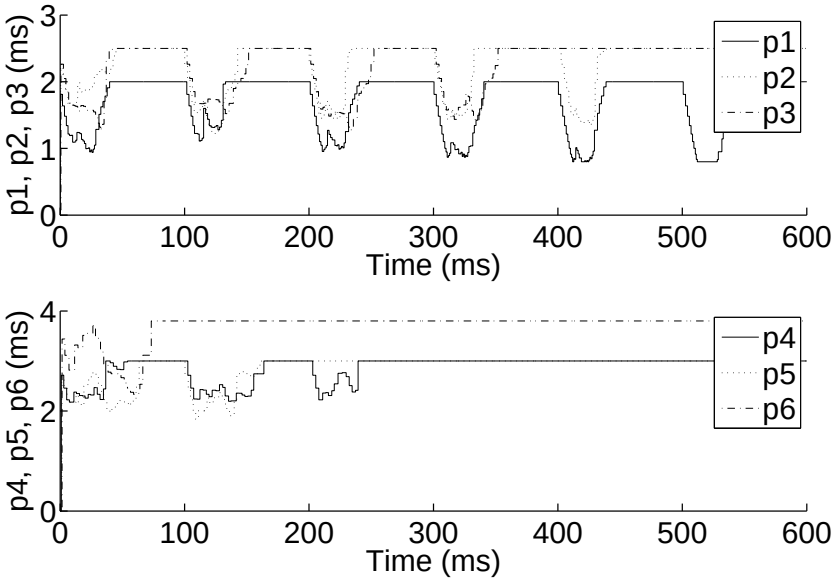

Figure 7.15: Dynamical adjustment of task periods under EDF.

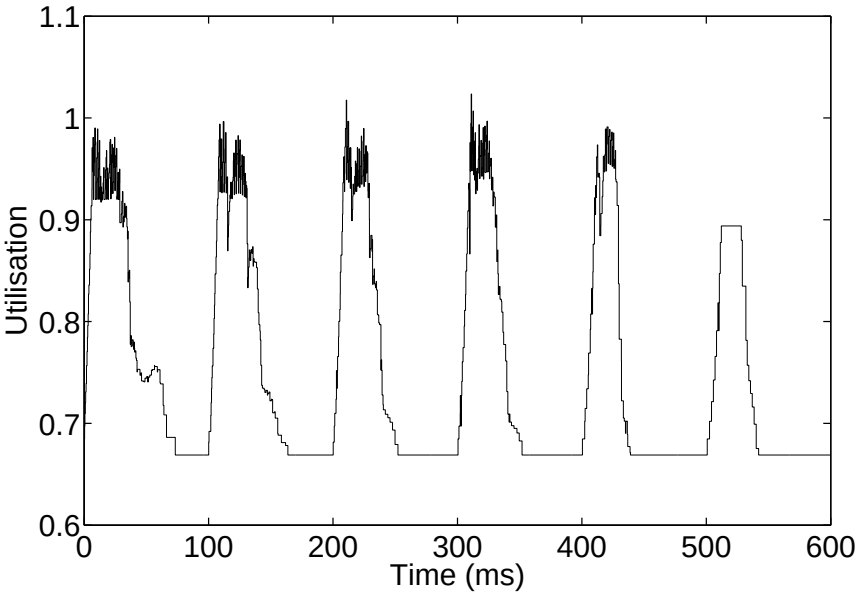

Figure 7.16: Dynamical adjustment of processor utilisation under EDF.

shows the dynamic scheduling process for the control periods. The dynamic adjustment of the requested process utilisation is illustrated in Figure 7.16.

Comparisons between Figures 7.15 and 7.11 for periods, and between Figures 7.16 and 7.12 for workload reveal that the EDF and FP in the proposed feedback scheduling scheme have similar patterns of period adjustment and workload adaptation.

The ITAE indices of the controlled processes under variable-period EDF scheduling of the decomposed tasks are summarised in Table 7.6. It is seen from this table that both FP and EDF provide quite comparable ITAE performance for the first four control loops $G_1(s)$ to $G_4(s)$. For the last two control loops $G_5(s)$ and $G_6(s)$, the QoC performance is slightly improved under EDF with increased scheduling complexity. Overall, both FP and EDF work well in the proposed feedback scheduling framework, while the EDF is slightly better.

7.9 Summary of This Chapter

Case studies have been carried out to verify the hierarchical feedback scheduling framework developed in the last chapter (Chapter 6). In the case studies, six processes are controlled by a uni-processor controller that uses the proposed scheduling framework with FP or EDF. The QoC performance of the control system is evaluated quantitatively. The results show that the FP with task decomposition outperforms the EDF without task decomposition. It is also demonstrated that with task partitioning, both FP and EDF in the proposed hierarchical feedback scheduling give similar QoC performance, while the EDF performs slightly better with increased scheduling complexity and overhead.

7.10 Nomenclature of This Chapter

The same nomenclature is used in this chapter (Chapter 7) and the last chapter (Chapter 6). It is listed below.

Abbreviations

EDF	Earliest-Deadline-First
FP	Fixed Priority
IAE	Integral of Absolute Error
ITAE	Integral of Time Absolute Error
PI	Proportional-Integral
QoC	Quality of Control

Symbols

c	Worst case computation time
d	Deadline
G	Controller or its transfer function
K	Gain
N	Number of requests for scaling down control periods
n	Total number of tasks
p	period
s	Laplace transform operator
T	Task, or time constant for plant (T_p), or integral time for controller (T_c)
T^I	Subtask set $T^I = \cup_{i=1}^{n} T_{i1}$
T^{II}	Subtask set $T^{II} = \cup_{i=1}^{n} T_{i2}$
U	CPU utilisation
U^I	Workload of the subtask T^I
U^{II}	workload of the subtask T^{II}
y	Output of the controlled variable
$\tilde{y}$	Control error

Greek Letters

Δ_H	QoC upper bound
Δ_L	Dead zone around $\delta\tilde{y}$. $\Delta_L < \Delta_H$
$\delta\tilde{y}$	One-step difference of $tildey$
ϵ	Forgetting factor
τ	Time delay

Superscripts

max	Maximum value or upper bound (for p, p_i and N)
min	Minimum value or lower bound (for p and p_i)

Subscripts

c	Controller
i	The first or sole subscript to c, d, p, T to indicate the ith task
j	The second subscript to some variables (c, d, T) to indicate the jth subtask decomposed from the ith task
p	Plant
rq	Requests (for N)
sp	Set-point for U and y

Chapter 8

Multi-Fractal Nature of Network Induced Delay in Networked Control Systems

---

Modern large-scale and real-time control systems are implemented over data networks as the communication media. Network induced time delay in a networked control system (NCS) is time-varying and unpredictable, and thus affect the accuracy of timing-dependent computing and control of the NCS. It is a challenging problem in NCS research and development. In most existing research on stability and stabilisation of NCS, the network induced delay is commonly treated as a stochastic variable, and is inherently assumed to be uniformly distributed in a certain range. However, recent studies have revealed that the control designed from these assumptions is conservative in order to maintain the system stability. A better understanding of the **dynamics of real-time networked control** is necessary for further improvement of system Quality of Control (QoC) performance.

In order to analyse the dynamics of network induced delay in NCS applications, typical scenarios of real-time networked control over TCP/IP networks are modelled and simulated. Network traffic data sets have been obtained, which show irregular dynamic behaviour. Our work has indicated that NCS network induced delay has a non-uniform distribution. Furthermore, analysing the network traffic data sets has revealed the multi-fractal nature of the network induced delay for the investigated scenarios of the real-time networked control.

Part of this work has been published in [Tian and Levy, 2008b] and [Tian *et al.*, 2007].

8.1 Hierarchical Control Systems

Let us begin with an investigation into a networked control system (NCS) for a middle-scale process or multiple small-scale processes. The control system has a typical hierarchical structure. It is implemented over an IEEE802.3 based network. The system is modelled and simulated using the open source package Network Simulator ns2 [UCB/LBNL/VINT Groups, 2008], which is a discrete event driven simulator for TCP/IP networks.

8.1.1 Control System Architecture

The NCS we focus on in the case studies has a three-layer hierarchical topology, as depicted in Fig. 8.1, which is typical in industrial control systems [Tian and Tadé, 2002]. The system has been investigated previously in our preliminary studies [Tian *et al.*, 2006c; Tian *et al.*, 2006a]. On the top of the hierarchy are management computers. Control computers, i.e., controllers, are in the middle. Sensors and actuators, and the plant to be controlled, are at the bottom. There are n smart sensors and m smart actuators, respectively. In practice, $n \geq m$. We set $n = 30$ and $m = 20$ here to simulate a middle-scale industrial process or multiple small-scale industrial processes. Also, we use 5 control computers and 5 management computers in the NCS.

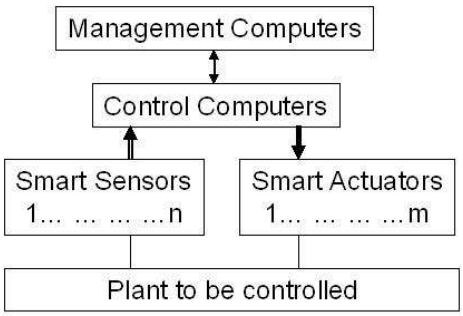

Figure 8.1: Hierarchical topology of the modelled NCS.

For ease of discussions, the following notations are used to represent various NCS devices:

S1, S2, $\cdots$, Sn:	Smart Sensors, $n = 30$
A1, A2, $\cdots$, Am:	Smart Actuators, $m = 20$
C1, C2, $\cdots$, C5:	5 control computers
M1, M2, $\cdots$, M5:	5 management computers
P1, P2, $\cdots$, Pq:	P2P connection devices, q is configurable in our model

Without loss of the generality, we assign C1 as the central controller, and C5 as a local control server. C2, C3, and C4 are used for display and other purposes.

It is assumed that there are 20 control loops, corresponding to the 20 actuators. The arrangement of the control loops is tabulated in Table 8.2.

Table 8.2: Control Loops.

Control Loop	1	2	...	10	11	12	···	20
Sensor(s)	1	2	...	10	11	13	···	29
					12	14	···	30
Actuator	1	2	...	10	11	12	···	20

8.1.2 Network Architecture

Various LAN (local area network) architecture designs are possible to interconnect NCS hosts and devices. This chapter considers a two-segment architecture with 10Mbps network capacity. Other network architecture designs will be investigated in the next chapter (Chapter 9).

In the two-segment architecture, put all management and control computers in a segment, and all smart sensors and actuators in the other segment. This architecture design is shown in Figure 8.2. The corresponding representation in ns2 is depicted in Figure 8.3.

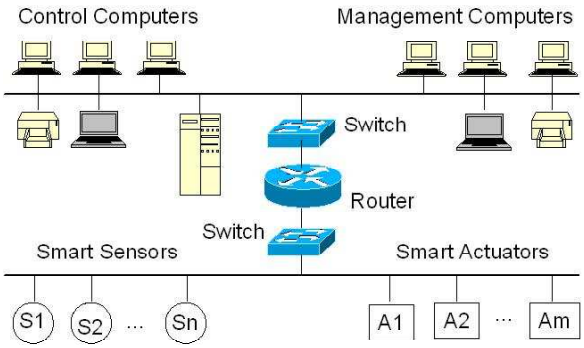

Figure 8.2: Two-segment architecture of the NCS Network.

8.1.3 Traffic Flow Specifications

For traffic flow specifications, we suppose that control tasks are periodic with the period of 200ms. Typical traffic flows in the NCS are specified below.

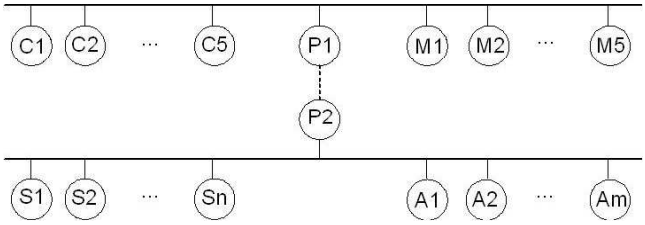

Figure 8.3: Representation of the two-segment architecture in ns2.

1) Within a control period, each of the smart sensors sends a packet of 200 bytes in the first 50ms to the controller (C1), and also to the local server (C5).

2) From 100ms to 200ms, the controller sends a control packet of 200 bytes to each of the actuators, and a packet of 1k bytes to the local server C5.

3) In each control period, C5 sends packets of 2k bytes to each of C2, C3, and C4 for local information processing, monitoring, and displaying.

4) Each of C2, C3, and C4 sends queries to the local server C5 and gets responses from C5 within an interval of 5s. The sizes of the queries and responses for each of C2, C3, and C4 are 1k and 10k bytes, respectively.

5) Within a session of 10min, each management computer sends queries of 60k bytes to the local server C5, and gets responses of 600k bytes from C5.

From the above specifications, traffic flows are calculated and tabulated in Table 8.3 for the modelled NCS. It is worth mentioning that for each TCP packet transmitted over the NCS network, there is a 40-byte overhead for TCP and IP headers.

In all cases, assume that the network propagation time is 4ms for all LANs and P2P links. The modelled NCS is simulated for 10s using ns2 under Unix. All traffic flows over the network are monitored and recorded in trace files. Then, extract information from the trace files and analyse and evaluate the performance the NCS network.

8.2 Non-Uniform Distribution of Network Induced Delay

Selected results of the performance analysis for the NCS are summarised in Table 8.4.

Table 8.3: Traffic Flow Specifications.

No.	Traffic Flow			Flow Rate
TCP1.	0–50ms:	each S1 $\sim$ Sn $\Rightarrow$ C1,	200 bytes	$\rightarrow$ 8kbps
TCP2.	0–50ms:	each S1 $\sim$ Sn $\Rightarrow$ C5,	200 bytes	$\rightarrow$ 8kbps
TCP3.	100–200ms:	C1 $\Rightarrow$ each A1$\sim$Am,	200 bytes	$\rightarrow$ 8kbps
TCP4.	100–200ms:	C1 $\Rightarrow$ C5,	1k bytes	$\rightarrow$ 40kbps
TCP5.	0–200ms:	C5 $\Rightarrow$ each C2$\sim$C4,	2k bytes	$\rightarrow$ 80kbps
TCP6.	0–5s:	each C2$\sim$C4 $\Rightarrow$ C5,	1k bytes	$\rightarrow$ 1.6kbps
TCP7.	0–5s:	C5 $\Rightarrow$ each C2$\sim$C4,	10k bytes	$\rightarrow$ 16kbps
TCP8.	0–10min:	each M1$\sim$M5 $\Rightarrow$ C5,	60k bytes	$\rightarrow$ 800bps
TCP9.	0–10min:	C5 $\Rightarrow$ each M1$\sim$M5,	600k bytes	$\rightarrow$ 8kbps
Packet Size (bytes) - TCP1,2,3,6: 200; TCP4,5,7,9: 1k; TCP8: 100				
10Mbps capacity and 4ms propagation time for all links				

Table 8.4: Selected results for TwoSegments/10Mbps NCS.

Total number of recorded receive events:	25 600	
All measurement/control packets received?	Yes	
Received bits/s within each local area (bps)		Network Utilisation
Computer side:	1 313 760	$< 13.2\%$
Sensor/actuator side:	901 120	$< 9.1\%$
P2P link:	901 120	$< 9.1\%$

Average throughput (bps):	C1	C5	Aj (Actuator)
	323 232	355 680	9 632

Sensor-to-controller delay and jitter (ms)						
	S01	S10	S20	S25	S30	All
Min delay	16.64	16.71	16.71	16.71	16.81	16.30
Max delay	69.79	77.92	93.34	91.04	56.65	93.34
Jitter	53.15	61.25	76.64	74.33	39.84	77.05
Controller-to-actuator delay and jitter (ms)						
	A01	A05	A10	A15	A20	All
Min delay	17.50	17.35	18.42	19.48	19.12	16.93
Max delay	28.92	29.85	31.37	32.86	34.16	34.16
Jitter	11.42	12.50	12.96	13.38	15.04	17.23

It is seen from Table 8.4 that the traffic load of the NCS network is less than 15%. However, network induced delay and jitter is significant. The sensor-to-actuator delay ranges from 16.30ms to 93.34ms, and the controller-to-actuator delay changes from 16.93ms to 34.16ms.

With consideration of both sensor-to-controller delay and controller-to-actuator delay, the total NCS communication latency is between 33.23ms and 127.50ms, resulting in a jitter of 94.27ms.

In most existing research on stability and stabilisation of NCS, it is inherently assumed that the network induced delay is uniformly distributed in a certain range. However, our investigation does not support this assumption. To support this claim, Figure 8.4 shows a plot of network induced sensor-to-controller delay. The corresponding delay distribution is depicted in Figure 8.5, which clearly shows the non-uniform distribution of the delay.

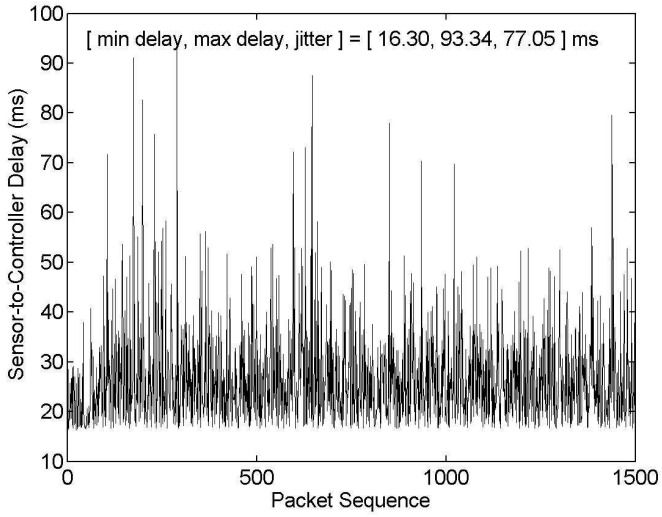

Figure 8.4: Sensor-to-controller delay.

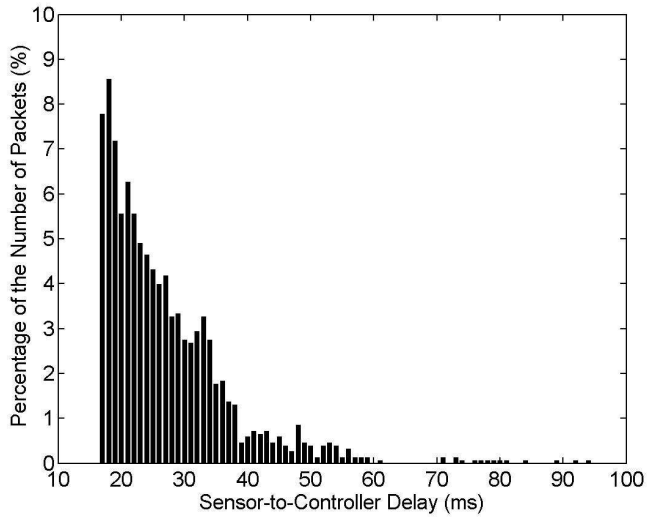

Figure 8.5: Distribution of the sensor-to-controller delay.

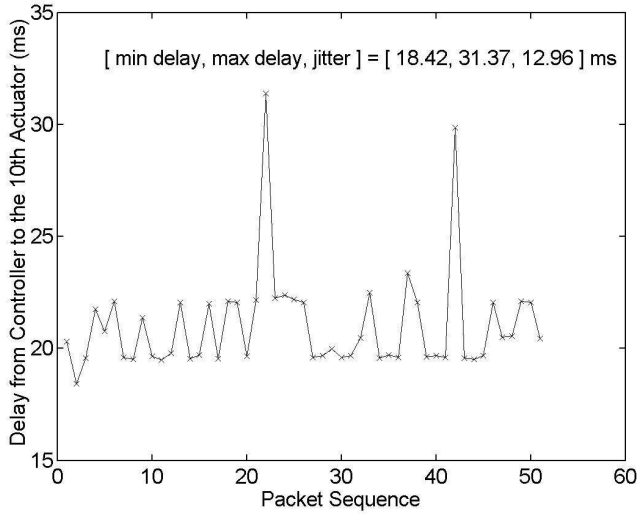

Figure 8.6: Delay from the controller to actuator 10.

Compared with the sensor-to-controller delay, the control-to-actuator delay exhibits more regular behaviour with small jitter (Figure 8.6). This is due to the same setting of the control period for all control tasks; and the setting means no sensor-to-controller traffic sharing the network bandwidth when the controller sends control packets to the actuators. Nevertheless, the controller-to-actuator delay is also non-uniformly distributed.

Further analysis of the data sets in Figure 8.4 will reveal multi-fractual nature. To facilitate the analysis, let us discuss the theoretic background of time series analysis in the next section.

8.3 Theoretical Background of Time Series Analysis

8.3.1 R/S Analysis

Denote the dynamics of the network traffic shown in Figure 8.4 as $x = \{x_k\}_{k=1}^{N}$, where N is the length of the sequence. This sequence can be treated as fractal records in time. Hurst invented the R/S analysis method to study such sequences [Hurst, 1951]. Later, Mandelbrot [Mandelbrot, 1982] and Feder [Feder, 1988] further developed this method in fractal theory.

For any fractal records in time $x = \{x_k\}_{k=1}^{N}$ and any $2 \leq n \leq N$, define

$$< x >_n = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{8.1}$$

$$X(i,n) = \sum_{u=1}^{i}[x_u - < x >_n] \tag{8.2}$$

$$R(n) = \max_{1\le i\le n} X(i,n) - \min_{1\le i\le n} X(i,n) \tag{8.3}$$

$$S(n) = [\frac{1}{n}\sum_{i=1}^{n}(x_i - < x >_n)^2]^{1/2} \tag{8.4}$$

Hurst found that

$$R(n)/S(n) \ \sim \ (\frac{n}{2})^H \tag{8.5}$$

where H is called the *Hurst exponent*.

As n changes from m to N, we obtain $N-m+1$ points in $\ln(n)$ v.s. $\dfrac{\ln(R(n)}{S(n))}$ plane. Then, we can calculate the Hurst exponent for the time series using the least-square linear fit.

The Hurst exponent is usually used as a measure of complexity. The trajectory of the record is a curve with a fractal dimension $D = 2 - H$ [Feder, 1988, p. 149]. Hence a smaller H means a more complex system. When applied to fractional Brownian motion, if $H > \frac{1}{2}$, the system is said to be *persistent*, which means that if for a given time period t the motion is along one direction, then in the time succeeding t it is more likely that the motion will follow the same direction. For a system with $H < \frac{1}{2}$, the opposite holds, that is, the system is *antipersistent*. But when $H = \frac{1}{2}$ the system produces Brownian motion, which is random.

8.3.2 Multi-fractal Analysis

First, we define a measure from a positive time series as is done for the length sequence of a genome [Yu *et al.*, 2001b]. Let $T_t,\ t = 1, 2, \cdots, N$, be the time series. Define

$$F_t = \frac{T_t}{\sum_{j=1}^{N} T_j} \tag{8.6}$$

to be the frequency of T_t. It follows that $\sum_t F_t = 1$. Now we can define a measure μ on interval $[0,1]$ by $d\mu(x) = Y(x)dx$, where

$$Y(x) = N \times F_t, \text{ when } \ x \in \left[\frac{t-1}{N}, \frac{t}{N}\right] \tag{8.7}$$

It is easy to see that $\int_0^1 d\mu(x) = 1$ and $\mu\left([\frac{(t-1)}{N}, \frac{t}{N}]\right) = F_t$.

The most common numerical implementations of multi-fractal analysis are the so-called *fixed-size box-counting algorithms* [Halsey *et al.*, 1986]. In the one-dimensional case, for a given measure μ with support $E \subset \mathbf{R}$, we consider the *partition sum*

$$Z_\epsilon(q) = \sum_{\mu(B)\neq 0} [\mu(B)]^q \tag{8.8}$$

$q \in \mathbf{R}$, where the sum runs over all different non-empty boxes B of a given side ϵ in a grid covering of the support E, that is,

$$B = [k\epsilon, (k+1)\epsilon] \tag{8.9}$$

The scaling exponent $\tau(q)$ is defined by

$$\tau(q) = \lim_{\epsilon\to 0} \frac{\log Z_\epsilon(q)}{\log \epsilon} \tag{8.10}$$

The generalised fractal dimensions of the measure are defined as

$$D_q = \frac{\tau(q)}{q-1}, \quad \text{for } q \neq 1 \tag{8.11}$$

and

$$D_q = \lim_{\epsilon\to 0} \frac{Z_{1,\epsilon}}{\log \epsilon}, \quad \text{for } q = 1 \tag{8.12}$$

where $Z_{1,\epsilon} = \sum_{\mu(B)\neq 0} \mu(B) \log \mu(B)$.

The generalised fractal dimensions are numerically estimated through a linear regression of $\frac{1}{q-1} \log Z_\epsilon(q)$ against $\log \epsilon$ for $q \neq 1$, and similarly through a linear regression of $Z_{1,\epsilon}$ against $\log \epsilon$ for $q = 1$. D_1 is called the *information dimension* and D_2 the *correlation dimension*. The D_q of the positive values of q gives relevance to the regions where the measure is large. The D_q of the negative values of q deals with the structure and the properties of the most rarefied regions of the measure.

Following the idea of the thermodynamic formulation of multi-fractal measures, Canessa [Canessa, 2000] has derived an expression for the "analogous" specific heat as

$$C_q \equiv -\frac{\partial^2 \tau(q)}{\partial q^2} \approx 2\tau(q) - \tau(q+1) - \tau(q-1) \tag{8.13}$$

He has shown that the form of C_q resembles a classical phase transition at a critical point for financial time series. We will discuss the property of C_q for NCS network induced delay in the next section.

8.4 Multi-fractal Nature of Network Induced Delay

The sequence of data in Figure 8.4 for network induced delay can be treated as fractal records in time. For this sequence of data, the Hurst exponent is calculated. The graph of the R/S analysis of the delay time series is shown in Figure 8.7.

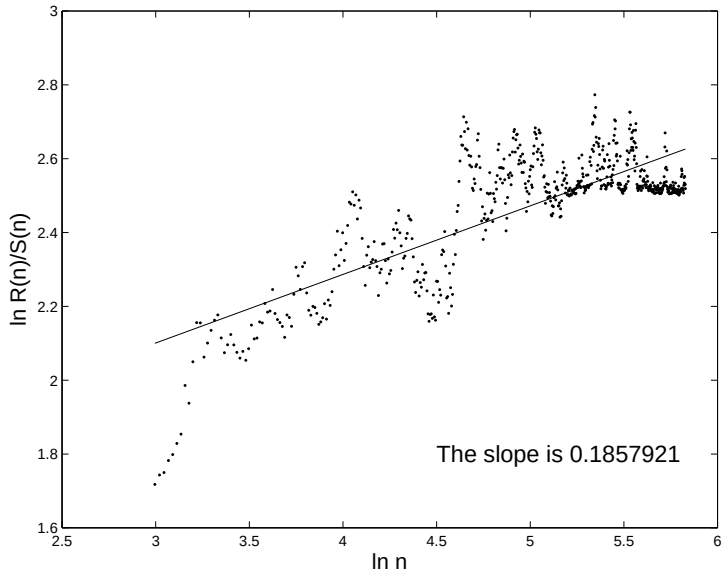

Figure 8.7: R/S analysis of the data. $m = 40, N = 680$.

Then, the generalised dimensions of the delay time series were computed. The D_q vs q curve is shown in Figure 8.8. It is seen from this figure that the D_q spectra is multi-fractal-like and sufficiently smooth for the C_q vs q curve to be meaningful. Depicted in Figure 8.9 is the C_q vs q curve corresponding to D_q in Figure 8.8. It can be seen from Figure 8.9 that it resembles a classical phase transition at a critical point.

From the values of the Hurst exponent, D_q spectra and related C_q curve, it can be concluded that the network induced delay has multi-fractal nature and exhibits long-range correlation.

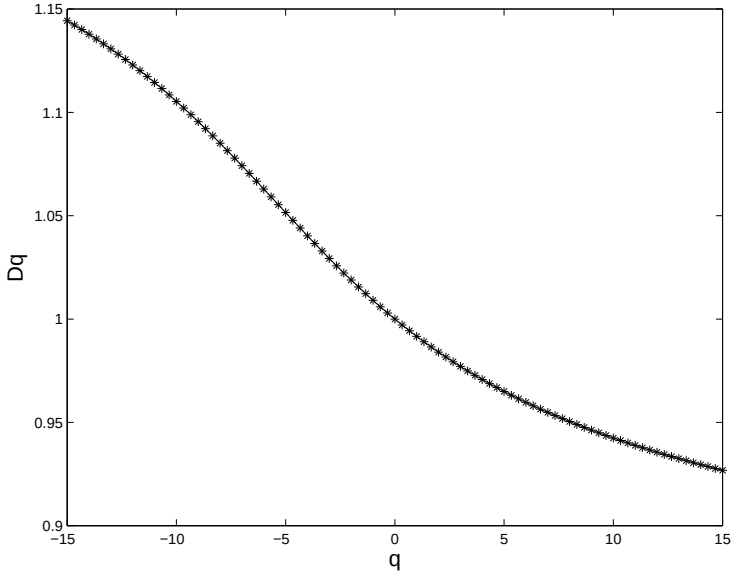

Figure 8.8: Generalised dimensions of the data.

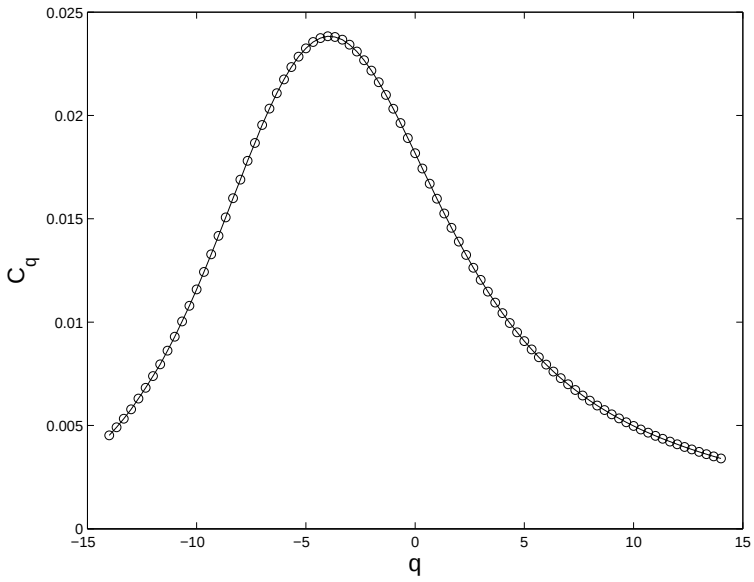

Figure 8.9: “Analogous” specific heat of the data.

8.5 Summary of This Chapter

The dynamics of network induced delay in real-time NCS has been analysed. The results have revealed non-uniform distribution and multi-fractal nature of the network delay. They do not support the theoretic background of existing NCS research in two aspects: (1) Most existing research on NCS stability and stabilisation has inherently assumed a uniform distribution of

network delay. With the finding of the non-uniform distribution of the network delay, less conservative stability criteria and improved control design can be made possible through employing some information of the delay distribution. (2) Another common assumption in NCS theoretic research is the randomness of the network induced delay. However, the multi-fractal nature of network induced delay implies long-range correlation of the delay. It further suggests that the traffic irregularity we have observed does not represent short-term randomness in the networked induced delay. These properties may be used to improve NCS modelling and design.

8.6 Nomenclature of This Chapter

Abbreviations

IP	Internet Protocol
TCP	Transport Control Protocol
NCS	Networked Control System/s

Symbols

A1 $\sim$ Am	Smart actuators
C1 $\sim$ C5	Control computers
C_q	Specific heat
D_q	Fractal dimension
F_t	Frequency of time series T_t
H	Hurst exponent
k	Integer
M1 $\sim$ M5	Management computers
m	Integer variable, or the number of smart actuators
N	Length of sequence
n	The number of data, or the number of smart sensors
q	Real number
$\mathbf{R}$	One-dimensional Euclidean space
$R(n)$	$R(n) = \max_{1\leq i\leq n} X(i,n) - \min_{1\leq i\leq n} X(i,n)$

S1 $\sim$ Sn	Smart sensors
$S(n)$	$S(n) = \left[\frac{1}{n}\sum_{i=1}^{n}(x_i - < x >_n)^2\right]^{1/2}$
T_t	Time series
t	Integer, $t = 1, 2, \cdots, N$
u	Integer
$X(i,n)$	$X(i,n) = \sum_{u=1}^{i}[x_u - < x >_n]$
x	Set of a sequence of data
x_i, x_k, x_u	The i-, k- and kth data in a sequence of data, respectively
$< x >_n$	Mathematical expectation for n sequence data
$Y(x)$	Used for defining μ, $d\mu(x) = Y(x)dx$
Z	Partition sum

Greek Letters

ϵ	Side length of a non-empty box
μ	A measure on interval $[0, 1]$
τ	Scaling exponent

Acknowledgment. Assistance from Zu-Guo Yu, a co-author of [Tian *et al.*, 2007] where part of this work has been published, in computation for Figures 8.7 to 8.9 is acknowledged.

Chapter 9

Real-Time Queuing Protocol for Networked Control Systems

Time-varying network induced delay in real-time networked control systems (NCS) has been analysed in detail in the last chapter (Chapter 8). It is a challenging problem in real-time NCS research and development. Over-provisioning of network capacity is not a general solution to this problem as it cannot provide any guarantee for predictive communication behaviour, which is a basic requirement for many real-time applications. As separate design of control, scheduling, and networks does not provide optimal solutions to an NCS, **integrated design of network, control, and other NCS components** is thus crucial for maximising the Quality of Control (QoC) performance of the NCS.

Providing such an integrated design solution, this work proposes a real-time protocol for NCS applications. A queuing architecture is developed to reduce the network induced jitter, making the network induced delay more predictable. Furthermore, case studies of NCS applications are given to demonstrate how to roughly estimate the timing parameters and calibrate the proposed queuing protocol under typical scenarios using Ethernet-based networking technologies. The NCS communication performance of the proposed queuing protocol is also evaluated in the case studies, showing the applicability of the queuing protocol in both hard and soft real-time control systems.

The core content of this work has been published in [Tian and Levy, 2008b].

9.1 Queuing Protocol

Queuing packets to smooth out network induced delay is not a new idea in multimedia and even in NCS, e.g., [Luke and Ray, 1990; Luke and Ray, 1994; Chan and Özgüner, 1995]. However, the requirement in multimedia applications is different from that in NCS. For example, throughput is important in video streaming, but is not the focus in networked control as normally measurement and control packets are very small. For existing queuing methods for NCS, there has been no report on their success in real applications. This work will rectify our real-time queuing protocol, which was originally presented in [Tian *et al.*, 2006a; Tian *et al.*, 2006c], to deal with time-varying network induced delay.

Our philosophy in development of the queuing protocol is based on the following aspects:

1) Keep the general networks unchanged at the transport, network, data-link and physical layers to maintain the simplicity, scalability, and interconnectivity of the networks. However, a real-time queuing protocol is introduced on top of the transport layer to meet the requirement of predictable timing behaviour for real-time control.

2) Use two queues to smooth out the network induced time-varying delay and jitter.

3) The resulting predictable communication delay is then compensated through control design. Therefore, a co-design of network and control will maximise the control performance of the real-time NCS.

From our preliminary studies [Tian *et al.*, 2006a; Tian *et al.*, 2006c], the queuing architecture shown in Figure 9.1 implements the above mentioned ideas.

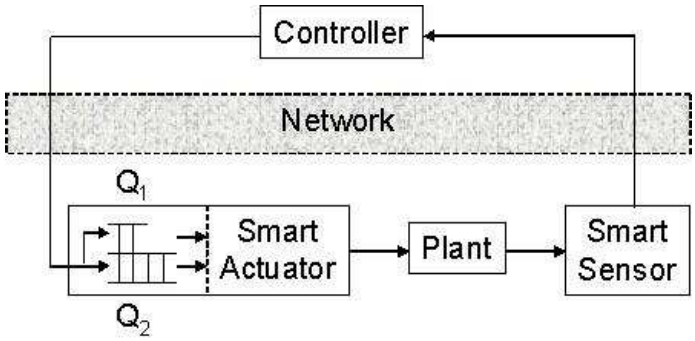

Figure 9.1: Real-time queuing architecture for networked control.

Compared with conventional networked control shown in Figure 1.2 (Chapter 1), the proposed NCS scheme in Figure 9.1 has two parallel queues, Q_1 and Q_2, on the actuator. Queue Q_1 stores only one control packet, and is used to enqueue the received control packet. Queue Q_2 is a first-in-first-out (FIFO) queue, with a capacity of a few, e.g., two or three, packets; and is designed to deal with packet dropout of control packets. The issue of packet dropout compensation will be investigated in the next chapter (Chapter 10).

For real-time control, it is expected that the control signals are sent to the plant under control at predictable time instants. However, due to the sensor-to-controller and controller-to-actuator jitters, control signals will arrive at the actuators in variable time intervals. Using the queue Q_1, the control packet is simply enqueued if it arrives too early. Later, at a fixed time instant in a control period, dequeue the packet from Q_1 and send it to the plant. In this way, network induced delay and jitter can be smoothed out. Consequently, the network induced delay becomes predictable, favourable to real-time networked control.

Events and timeline settings for Q_1 operations are proposed below in the next section.

9.2 Events and Timelines

For periodic control tasks, events and timelines of the queuing protocol in a control period are depicted in Figure 9.2 [Tian *et al.*, 2006c; Tian *et al.*, 2006a], and are explained below in detail.

1) The control task commences at time instant T_O;

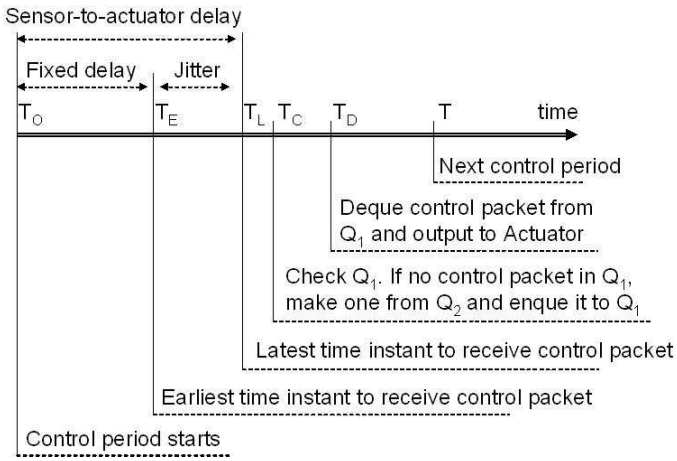

Figure 9.2: Timelines for the queuing architecture.

2) The controller receives a measurement packet from the sensor over the NCS network, calculates the corresponding control action, and then outputs a control packet to queue Q_1 through the NCS network;

3) In Figure 9.2, the earliest possible time instant T_E, at which the control packet arrives at queue Q_1, is described by

$$T_E = T_O + \min(t_{sc}) + t_{ctr} + \min(t_{ca}) \tag{9.1}$$

where t_{sc}, t_{ctr}, and t_{ca} are sensor-to-controller delay, the worst case control computation time, and controller-to-actuator delay, respectively.

4) Without packet dropout, the latest possible time instant T_L, at which the control packet reaches queue Q_1, is,

$$T_L = T_E + t_j \tag{9.2}$$

where t_j is the sum of sensor-to-controller and controller-to-actuator jitters, i.e.,

$$t_j = \max(t_{sc}) - \min(t_{sc}) + \max(t_{ca}) - \min(t_{ca}) \tag{9.3}$$

5) The actuator checks Q_1 at fixed time instant $T_C \geq T_L$. If Q_1 is empty, the control packet has either been dropped or has arrived too late. In this case the actuator makes a decision on how to control the plant based on past control packets queued in Q_2. Strategies will be developed in the next chapter (Chapter 10) to deal with packet dropout.

6) Finally, dequeue the control packet from Q_1 and send it to the plant at fixed time $T_D > T_C$;

7) At time instant $T > T_C$, a new control period commences.

9.3 Timeline Requirements for Real-Time Control

Several timing parameters shown in Figure 9.2 need to be calibrated in order to achieve predictable timing behaviour and satisfactory control performance of the overall NCS. The basic requirements among these parameters for real-time systems are:

$$T_O < T_E < T_L \leq T_C < T_D \leq T. \tag{9.4}$$

Equation (9.4) can be interpreted differently for hard and soft real-time systems, respectively. Hard real-time systems require that Equation (9.4) is always met for the queuing architecture to be applicable. Any network design that does not satisfy Equation (9.4) will not allow the applications of the proposed method in hard real-time systems. However, for soft real-time systems, Equation (9.4) can be occasionally, but not frequently, missed out, e.g., $T_L > T$ occurs occasionally. This explains why Ethernet has applications in soft real-time control systems [Lee and Lee, 2002].

The following scenario is a good explanation to these two interpretations. In an inverted pendulum system, the control period $T < 1$ms, while the network induced delay is larger than 1ms in switched Ethernet with TCP transmission. Since the condition in Equation (9.4) is rarely satisfied, it would be difficult to use switched Ethernet with TCP transmission in the control of the inverted pendulum system regardless of either soft or hard real-time control design.

The control period, T, is determined from the frequency characteristics of the plant to be controlled. It should not be too small to avoid overloading the controller and communication network. It is easy to understand that a too big T is also not acceptable because the dynamics of the plant may not be reconstructed from the sampled measurements.

It is seen from Figure 9.2 and Equation (9.1) that $T_E - T_O = \min(t_{sc}) + t_{ctr} + \min(t_{ca})$ represents the fixed part of the network induced delay and control computing time t_{ctr}. From the control design point of view, T_E, the earliest time instant for actuators to receive control packets and enqueue them, is not a design parameter. However, T_E normally changes with different control algorithms and network designs. Therefore, it is necessary to optimise control algorithms and network architectures in order to reduce this fixed time delay.

The latest time instant for actuators to receive control packets and enqueue them, T_L, is chosen such that it covers the major part of the jitter or its full range or its full range. We have shown show how to achieve such a T_L under acceptable level of packet loss rate [Tian and Levy, 2007].

At time instant $T_C \geq T_L$, check whether queue Q_1 is empty or full. In practice, T_C should be a little later than T_L, giving the processor a time slice of duration $T_C - T_L$ for task switching. For ease of theoretical analysis, we may simply set $T_C = T_L$.

At time instant T_D, dequeue the control packet from Q_1 and output the control signal to the plant. The choice of T_D relies on how much time is required to deal with control packet

dropout. The constraint is that the computing of the algorithm dealing with packet dropout should complete within the time interval $T_D - T_C$ for hard real-time systems. This constraint can be alleviated occasionally for soft real-time systems.

9.4 Estimate of Timeline Parameters

In order to check whether the requirements of Equation (9.4) are fulfilled, it is necessary to estimate any two parameters of T_E, T_L, and t_j. Two possible solutions to the parameter estimation are: theoretical calculations and simulation analysis.

Equations (9.1) - (9.3) are expressions of the timing parameters; while more detailed descriptions are required for actual calculations. Theoretical calculations of time delay for simple networks are possible under simplified assumptions [Lee and Lee, 2002]. However, such calculations have significant limitations for NCS design. For example, they depends crucially on the configurations of the networks and thus do not scale well.

Therefore, we propose to estimate the key timing parameters through network modelling and simulation analysis. With network modelling and simulation, it is easy to analyse the best and worst communication delays under various conditions of network traffic and configurations, such as the timing parameters in Equations (9.1) through to (9.4): T_E, T_L, max(t_{sc}), min(t_{sc}), max(t_{ca}), min(t_{ca}), etc. Moreover, modelling and simulation can also help evaluate the timing behaviour of the NCS network communications. This will be shown below in Section 9.5 through case studies.

9.5 Case Studies

This section carries out case studies for a middle-scale process or multiple small-scale processes to be controlled over an NCS network, and shows how the real-time queuing protocol can be applied. Because the focus of this work is on real-time communications for NCS, we will evaluate the NCS network performance under various scenarios. The issues of controller design and control performance are beyond the scope of this work and thus will not be discussed.

9.5.1 Case Study 1: NCS with a TwoSegments/10Mbps Network

This case study has been modelled in the last chapter (Section 8.1 of Chapter 8). The NCS is a hierarchical control system with 30 sensors, 20 actuators, 5 control computers, 5 management computers, and other devices. It controls 20 loops. The system architecture, NCS network architecture, and network traffic specifications have been described in detail in Section 8.1 of Chapter 8.

Section 8.2 of Chapter 8 has given the results of the NCS network delay. It is seen from Table 8.4 that the sensor-to-actuator delay ranges from 16.30ms to 93.34ms, and the controller-to-actuator delay changes from 16.93ms to 34.16ms. Considering both delays, we can see that the total NCS communication latency is between 33.23ms and 127.50ms, resulting in a jitter of 94.27ms. Both the latency and jitter are significant.

Using the notations in Equations (9.1) through (9.4), we have

$$\begin{aligned} &\min(t_{sc}) = 16.30\text{ms}, \quad \max(t_{sc}) = 93.34\text{ms} \\ &\min(t_{ca}) = 16.93\text{ms}, \quad \max(t_{ca}) = 34.16\text{ms} \\ &t_j = 94.27\text{ms} \end{aligned} \tag{9.5}$$

Now let us show how to use the rough estimates in Equation (9.5) to calibrate the timeline parameters in the queuing protocol. We have set the control period $T = 200$ms. For timeline settings in Figure 9.2 and Equations (9.1) to (9.4), let $T_O = 0$ms. From Equations (9.1) and (9.5), it follows that

$$T_E = 33.23\text{ms} + t_{ctr} = 88.23\text{ms} \tag{9.6}$$

where $t_{ctr} = 50$ms is the control computing time. We will try to cover the full range of the communication jitter to avoid any control packet dropouts. According to Equations (9.2), (9.3), (9.5) and (9.6), we need to set

$$T_L \geq T_E + t_j = 127.50\text{ms} + t_{ctr} = 177.50\text{ms} \tag{9.7}$$

For the control period of 200ms, it is seen from Equations (9.4) and (9.7) that we still have around 22.50ms for checking the queue Q_1, dealing with possible packet dropout, and

outputting the control signal to the plant. In this example, we can simply set

$$T_C = 180\text{ms}, T_D = 190\text{ms}. \tag{9.8}$$

This will allow an extra $T_C - T_L = 2.50$ms time for control computing (t_{ctr}) and packet transmissions. We also have $T_D - T_C = 10$ms time to deal with possible packet dropout. The total time delay for the control computation and packet transmission in each of the control loops is $T_D = 190$ms, which is fixed for all control periods.

Figure 9.3 shows the resulting timing behaviour of the modelled NCS (Refer to Table 8.2 for control loop arrangement). It is seen from Figure 9.3 that deterministic and predictable timing behaviour is achieved for the NCS by using the proposed queuing protocol.

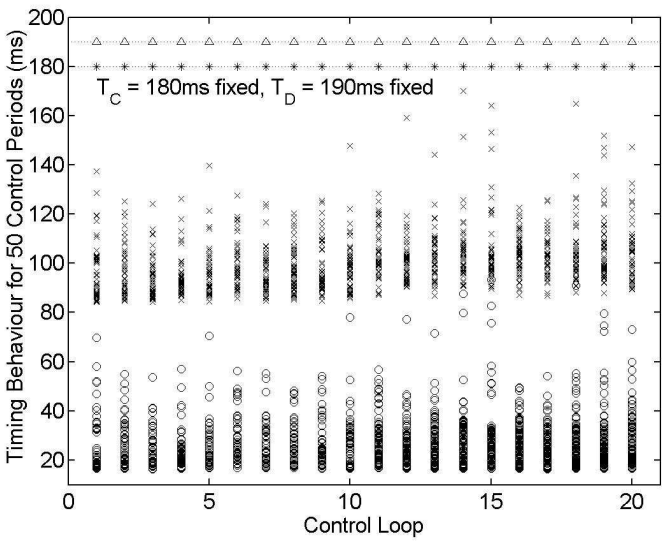

Figure 9.3: Timing behaviour of the NCS with the queuing protocol for case study 1 ('O': measurement received by controller; 'X': Control received by actuator; '*': T_C; 'Δ': T_D).

9.5.2 Case Study 2: NCS with a FourSegments/10Mbps Network

This case study is the same as Case Study 1 described in the last section except four segments instead of two segments are deployed in the NCS network. The network architecture and its ns2 representation are depicted in Figures 9.4 and 9.5, respectively.

Simulation results for this FourSegments/10Mbps case study are summarised in Table 9.1 [Tian *et al.*, 2006a]. Compared with Table 8.4 for TwoSegments/10Mbps architecture, Table

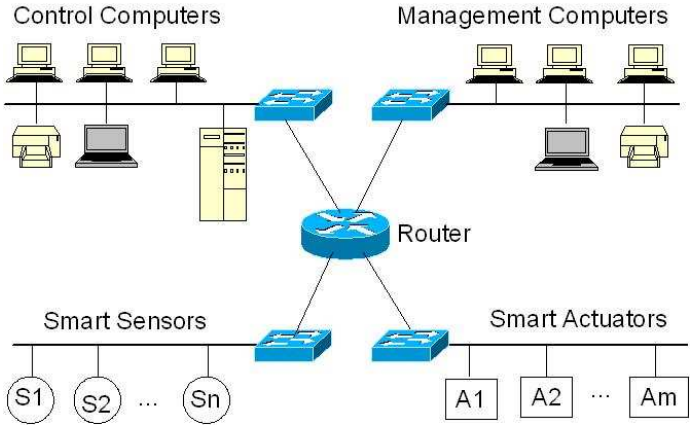

Figure 9.4: Four-segment architecture of the NCS Network.

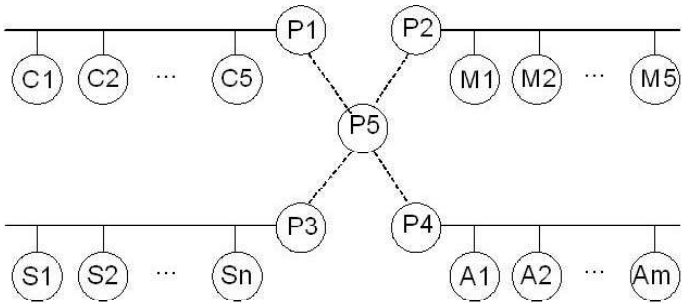

Figure 9.5: Representation of the four-segment architecture in ns2.

9.1 shows noticeable performance degradation in network induced delay and jitter when using the Foursegments/10Mbps architecture. Therefore, if the communication performance is the first priority, the two-segment architecture investigated in Section 8.1 of Chapter 8 is a better option. However, the four-segment architecture is good for network design and implementation, administration, and maintenance.

From Table 9.1, we have the following rough estimates

$$\begin{aligned} &\min(t_{sc}) = 16.12\text{ms}, \quad \max(t_{sc}) = 90.86\text{ms} \\ &\min(t_{ca}) = 21.69\text{ms}, \quad \max(t_{ca}) = 57.29\text{ms} \\ &t_j = 110.34\text{ms} \end{aligned} \tag{9.9}$$

Compared with the estimates in Equation (9.5) for Case Study 1, the estimates here show larger network induced delay (about 20ms larger) and jitter (about 16ms larger). This is due to the NCS network architecture of more LANs. As in Case Study 1, these rough estimates in Equation (9.9) are useful for calibration of timeline parameters in the queuing protocol.

Table 9.1: Selected results for FourSegments/10Mbps NCS.

Total number of recorded receive events			34 420			
All measurement/control packets received?			Yes			
Received bits/s within each local area (bps)			Network Utilisation			
Control computer side:			1 313 760			13.1%
Manag. comput. side, P2-P5:			51 040			<1.0%
Sensor side, P3-P5 link:			675 840			6.8%
Actuator side:			192 640			1.9%
P1–P5 link:			952 160			9.5%
P4–P5 link:			225 280			2.3%
Average throughput (bps):		C1	C5	Aj (Actuator)		
		323 232	355 680	9 632		
Sensor-to-controller delay and jitter (ms)						
	S01	S10	S20	S25	S30	All
Min delay	20.84	20.90	20.85	21.46	20.83	16.12
Max delay	52.81	50.01	81.01	76.11	60.88	90.86
Jitter	31.97	29.11	60.16	54.65	40.05	74.74
Controller-to-actuator delay and jitter (ms)						
	A01	A05	A10	A15	A20	All
Min delay	21.69	22.63	23.41	24.47	25.54	21.69
Max delay	53.24	54.10	55.16	56.22	57.29	57.29
Jitter	31.56	31.46	31.75	31.75	31.75	35.60

Again, set $T_O = 0$. Using similar calibration process to that in Case Study 1, we have

$$T_E = 37.81\text{ms} + t_{ctr} = 87.81\text{ms} \tag{9.10}$$

where $t_{ctr} = 50$ms is the control computing time. To cover the full range of the jitter to avoid packet loss, according to Equations (9.2), (9.3), (9.9) and (9.10), we need to configure

$$T_L \geq T_E + t_j = 148.15\text{ms} + t_{ctr} = 198.15\text{ms}. \tag{9.11}$$

For the control period of 200ms, it is seen from Equations (9.4) and (9.7) that we still have the flexibility of around 11.06ms to do some extra work. In this example, we can simply set

$$T_C = 199\text{ms}, T_D = T = 200\text{ms}. \tag{9.12}$$

This give the NCS extra $T_C - T_L = 850\mu$s for control computing (t_{ctr}) and packet transmissions. We also have $T_D - T_C = 1$ms, which can be used for packet dropout compensation. The overall delay of the control computation and communications is fixed at 200ms for each loop. The timing of the NCS with the queuing protocol is shown in Figure 9.6. Figure 9.6 clearly show that deterministic and predictable timing behaviour is achieved for the NCS.

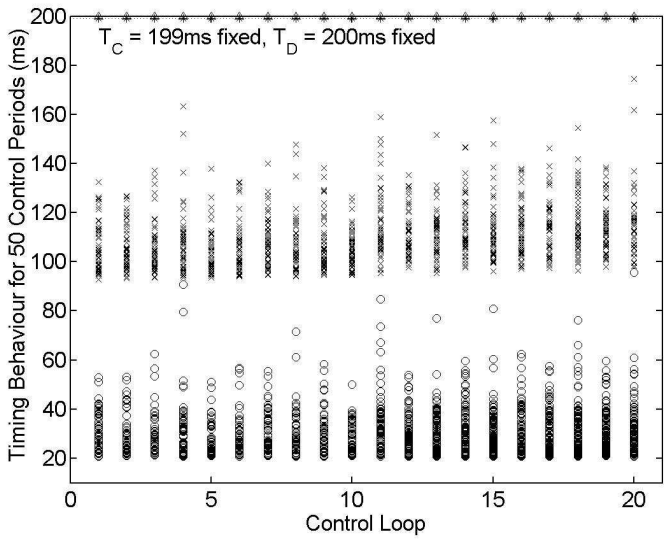

Figure 9.6: Timing behaviour of the NCS with the queuing protocol for case study 2 ('O': measurement received by controller; 'X': control received by actuator; '*': T_C; 'Δ': T_D).

9.6 Summary of This Chapter

Integrating various technologies into a unified framework, a real-time queuing protocol has been proposed for real-time NCS applications. Two of the key concepts in the protocol have been discussed in this work: the queuing architecture with two parallel queues on actuators, and timeline settings and calibrations for the queuing protocol. With applications of this queuing protocol, more predictive timing behaviour can be achieved for general NCS networks. Several typical scenarios of Ethernet based TCP/IP networking have been evaluated for a real-time NCS. Our work has indicated that adding hierarchy into the NCS network will introduce extra control latency and jitter. The overall work has demonstrated that co-design of network and control is necessary in order to maximise the real-time control performance of an NCS.

9.7 Nomenclature of This Chapter

Abbreviations

IP	Internet Protocol
TCP	Transport Control Protocol
NCS	Networked Control System/s

Symbols

A1 $\sim$ Am	Smart actuators
C1 $\sim$ C5	Control computers
M1 $\sim$ M5	Management computers
m	The number of smart actuators
n	The number of smart sensors
Q_1	Queue to store one control packet
Q_2	First-in-first-out (FIFO) queue to store a few control packets
S1 $\sim$ Sn	Smart sensors
T	Control period
T_C	Time instant to deal with control packet dropout
T_D	Time instant to dequeue Q_1
T_E	Earliest time instant to receive control packet
T_O	Time instant at which a control period commences
T_L	Latest time instant to accept control packet
t_{ca}	Controller-to-actuator delay
t_{ctr}	Control computation time
t_j	Sum of sensor-to-controller and controller-to-actuator jitters
t_{sc}	Sensor-to-controller delay

Chapter 10

Compensation for Control Packet Dropout in Networked Control Systems

---

Like time-varying network induced delay that have been investigated in the last two chapters (Chapters 8 and 9), data packet dropout is another challenging problem in real-time networked control systems. Applying the real-time queuing protocol that we developed in the last chapter, we are able to limit the sum of the network induced communication delay and the control computation delay to within a control period. This one-period delay is further guaranteed by improved **integrated design of real-time networked control** through embedding packet dropout compensation into the queuing protocol.

This chapter proposes to compensate for the control packet dropout at the actuator using past control signals. Three model-free strategies for control packet dropout compensation, namely, PD (proportional plus derivative), PD2 (Proportional plus up to the second-order derivative), and PD3 (proportional plus up to the third-order derivative) are developed. They are suitable for a large number of NCS without the need to tune the compensator parameters. The proposed dropout compensation schemes are demonstrated through numerical examples.

The core content of this work has been published in two papers [Tian and Levy, 2008a; Tian and Levy, 2008b].

10.1 Philosophy for Control Packet Dropout Compensation

A queuing protocol has been developed in the last chapter for NCS (Figure 9.1 of Chapter 9). Control packet dropout refers to the situation where, in a control period, no control packet is received by the actuator before the latest time instant to enqueue a control packet to queue Q_1.

From the timelines in Figure 9.2 for the queuing protocol, it is seen that T_L is the latest *possible* time instant to receive control packets without packet dropout. After this time instant, some control packets may not be delivered successfully. T_L is also the cut-off, i.e., latest *allowable*, time instant for Q_1 to accept control packets in a control period. Even if some control packets can reach the actuator after this time instant, they are purposely dropped.

Before developing strategies to deal with control packet dropout, we would like to clarify the following two aspects, which are crucial for our development:

1) How much computing power and other resources are available for calculations of packet dropout compensation? and
2) How much chance will various packet dropout scenarios likely occur?

To answer the first question we have noted that smart actuators normally have limited computing power and resources, and consequently sophisticated algorithms are difficult to implement in the actuators. Therefore, we have aimed to develop simple and model-free methods to resolve the control packet dropout problem. In this work, we will further extend our solution presented in [Tian *et al.*, 2006a; Tian *et al.*, 2006c].

The following discussions help answer the second question. As mentioned previously, an NCS network should be designed such that the packet loss rate is low and sustained sequence of successive dropped packets will unlikely happen under normal conditions. Therefore, it is senseless to discuss how to compensate for, say 50%, packet loss rate, which simply means a bad network design. For a rough quantitative analysis, if the packet loss rate of the network is lr, the rate of n successive packet losses can be estimated as $(lr)^n$. For example, 5% packet loss rate (this is significant for NCS networks), two and three successive packet losses will happen with the chance of 2.5% and 1.25%, respectively. This tells us to focus on the compensation for a single packet loss and two/three successive packet dropouts; there is only a small chance for four or more successive packet dropouts to occur.

10.2 Mechanism for Control Packet Dropout Compensation

For real-time networked control, having the mechanism to output control signals at a fixed time instant in the real-time queuing protocol leads to more predictive timing behaviour of the NCS. This compensates for time-varying network delay, but does not solve the problem of control packet dropout. What will happen if a control packet is not received by the actuator by the time instant T_L? In this case, Q_1 is empty and no control signal can be output to the plant. We need to "make" a control signal! This is what we will investigate in detail in the next few sections.

When a control packet is not received by the actuator by the time instant T_L, a component of packet dropout compensation in the NCS is activated to estimate the dropped packet. In this way, the overall delay from the sensor through the controller to the actuator is still limited to within one control period. Therefore, effective strategies for control packet dropout compensation need to be embedded into the queuing protocol to ensure its successful applications in NCS.

The key concepts in our simple solution are:

1) To use the FIFO queue Q_2 to buffer a few control packets to allow the smart actuator to recover a dropped control packet; and
2) To construct a control signal from Q_2 when a control packet is dropped.

Suppose that the control packet in control period k is not received by the actuator by the pre-specified time instant T_L, but there are a few past control packets, $u_{k-1}, u_{k-2}, \cdots$, available in queue Q_2 for prediction of the dropped packet. We can use these past control packets to construct a packet $\hat{u}_k$ for use in the current control period. The accent over 'u' implies that the packet is an estimate. A simple and general form of $\hat{u}_k$ can be

$$\hat{u}_k = f(u_{k-1}, u_{k-2}, u_{k-3}, \cdots), \tag{10.1}$$

where f : R $\rightarrow$ R is a nonlinear function.

In the next few sections, new compensation strategies will be developed which use both past control signals and their derivative information. As a common practice, pre-processing of data packets is necessary in the presence of noises when applying these strategies.

10.3 PD Prediction of Dropped Control Packets

The general formulation of our compensation strategies for dropped control packets is given in Equation (10.1). If the rate of change in control u is approximately constant, from the principle of motion we may choose a particular function $f(\cdot)$ in Equation (10.1) as

$$\hat{u}_k = u_{k-1} + K_{u1} T u_{k-1}^{(1)}, K_{u1} \in [0, 1] \tag{10.2}$$

where $u_{k-1}^{(1)}$ is the first-order derivative of u at time instant $(k-1)T$, and is approximated by

$$u_{k-1}^{(1)} = (u_{k-1} - u_{k-2})/T \tag{10.3}$$

Equation (10.2) is a one-step prediction scheme through a proportional term plus a derivative term using two past control signals u_{k-1} and u_{k-2}. It is a proportional-derivative (PD) compensator. Two special cases of Equation (10.2) are:

- $K_{u1} = 0$; this corresponds to the case $\hat{u}_k = u_{k-1}$, implying that he actuator simply reuses the last control signal queued in Q_2 when the current control packet is dropped; and

- $K_{u1} = 1$; this gives $\hat{u}_k = u_{k-1} + T u_{k-1}^{(1)}$, implying that if u changed in the last control period, it likely changes again at the same rate in current control period.

We have analysed the dynamics of these two special cases through functional analysis [Fidge and Tian, 2006]. Reusing the last control packet, i.e., setting $K_{u1} = 0$ in (10.2), provides a conservative control. Setting $K_{u1} = 1$ in (10.2) gives better control in most cases, but may result in unacceptable behaviour in the presence of repetitive pattern of packet losses. However, sustained repetitive pattern of dropped packets will unlikely happen under normal conditions.

The strategy in Equation (10.2) looks simple. However, we will indicate in the following that the scheme is actually a combination of PI (proportional-integral) and PD compensations from the viewpoint of the controlled variable!

Assume that the measurement series of the controlled variable y are y_{k-1}, y_{k-2}, $\cdots$. The set-point of y is y_r. A PI controller is used to control the plant. The proportional coefficient and

integral time of the PI controller are K_c and T_i, respectively. It follows that

$$\begin{aligned} &u_{k-1} = K_c\left(\Delta y_{k-1} + \frac{1}{T_i}\sum_{j=-\infty}^{k-1}\Delta y_j\right), \quad u_{k-2} = K_c\left(\Delta y_{k-2} + \frac{1}{T_i}\sum_{j=-\infty}^{k-2}\Delta y_j\right) \\ &\Delta y_j = y_r - y_j, j = \cdots, k-2, k-1 \end{aligned} \tag{10.4}$$

Substituting Equation (10.4) into Equation (10.2), we have

$$\begin{aligned} &\hat{u}_k = K_c\left(\Delta y_{k-1} + \frac{1}{T_i}\sum_{j=-\infty}^{k-1}\Delta y_j\right) + \bar{K}_c\left[\Delta y_{k-1} + \bar{T}_d\left(\Delta y_{k-1}\right)^{(1)}\right] \\ &\bar{K}_c = K_{u1}K_c/T_i, \bar{T}_d = T_iT \end{aligned} \tag{10.5}$$

where $\left(\Delta y_{k-1}\right)^{(1)}$ approximates the first-order derivative of Δy_{k-1} at time instant $(k-1)T$

$$\left(\Delta y_{k-1}\right)^{(1)} = \frac{\Delta y_{k-1} - \Delta y_{k-2}}{T} \tag{10.6}$$

Equation (10.5) shows that if the plant is controlled by a PI controller, the one-step ahead estimation $\hat{u}_k$ for a dropped control packet is a PI+PD (proportional-derivative) prediction.

- The PI part is $K_c\left(\Delta y_{k-1} + \frac{1}{T_i}\sum_{j=-\infty}^{k-1}\Delta y_j\right)$ with the proportional coefficient K_c and integral time T_i, respectively, and is calculated at time instant $(k-1)T$.
- The PD part is $\bar{K}_c\left[\Delta y_{k-1} + \bar{T}_d\left(\Delta y_{k-1}\right)^{(1)}\right]$ with the proportional coefficient $\bar{K}_c$ and derivative time $\bar{T}_d$, respectively. It captures the characteristics of the rate of change in y to form a one-step ahead prediction to the control increment.

Rearranging Equation (10.5), we have

$$\begin{aligned} &\hat{u}_k = \tilde{K}_c\left[\Delta y_{k-1} + \frac{1}{\tilde{T}_i}\sum_{j=-\infty}^{k-1}\Delta y_j + \tilde{T}_d\left(\Delta y_{k-1}\right)^{(1)}\right] \\ &\tilde{K}_c = K_c\left(1 + K_{u1}/Ti\right), \tilde{T}_i = T_i + K_{u1}, \tilde{T}_d = \frac{K_{u1}T_i}{T_i + K_{u1}}T \end{aligned} \tag{10.7}$$

It is seen from Equation (10.7) that the PD compensator in (10.2) can also be interpreted as a proportional-integral-derivative (PID) predictive control.

The mode of PI+PD shown in Equation (10.5) or PID shown in Equation (10.7) explains

why the PD compensator (10.2) can have good prediction ability. It also reveals the complicated predictive control mechanism behind the simple PD compensation in Equation (10.2).

10.4 PD2 Prediction of Dropped Control Packets

In many cases, the rate of change in control u varies significantly over the time. The prediction of dropped packets should be able to capture this accelerated or decelerated change in u. Again, from the principle of motion we may choose a particular function $f(\cdot)$ in Equation (10.1) as

$$\hat{u}_k = u_{k-1} + K_{u1} T u_{k-1}^{(1)} + \frac{1}{2} K_{u2} T^2 u_{k-1}^{(2)} \quad K_{u1} \in [0,1], K_{u2} \in [0,1] \tag{10.8}$$

where $u_{k-1}^{(1)}$ is the first-order derivative of u at the time instant $(k-1)T$ and can be approximated by Equation (10.3), $u_{k-1}^{(2)}$ is the second-order derivative of u at the time instant $(k-1)T$ and can be estimated from

$$u_{k-1}^{(2)} = \frac{u_{k-1} - 2u_{k-2} + u_{k-3}}{T^2} \tag{10.9}$$

Therefore, the dropout compensation scheme in Equation (10.8) is a PD plus second-order-derivative compensator. The compensator is abbreviated as PD2 in this work, where D2 indicates up to the second-order derivative.

Again, assume that the plant is controlled by a PI controller with Equation (10.4). Substituting Equation (10.4) into Equation (10.8), we have

$$\begin{cases} \hat{u}_k &= K_c \left(\Delta y_{k-1} + \dfrac{1}{T_i} \displaystyle\sum_{j=-\infty}^{k-1} \Delta y_j \right) \\ & \quad + \bar{K}_c \left[\Delta y_{k-1} + \bar{T}_d \left(\Delta y_{k-1} \right)^{(1)} \right] + \frac{1}{2} K_c K_{u2} T^2 \left(\Delta y_{k-1} \right)^{(2)} \\ \bar{K}_c &= K_{u1} K_c / T_i, \ \bar{T}_d = \left(T_i + \dfrac{1}{2} \dfrac{K_{u2}}{K_{u1}} \right) T \end{cases} \tag{10.10}$$

where $(\Delta y_{k-1})^{(1)}$, which approximates the first-order derivative of Δy_{k-1} at time instant $(k-1)T$, is described in Equation (10.6), and $(\Delta y_{k-1})^{(2)}$ is an approximation of the second-order derivative of Δy_{k-1} at the time instant $(k-1)T$,

$$(\Delta y_{k-1})^{(2)} = \frac{(\Delta y_{k-1} - 2\Delta y_{k-2} + \Delta y_{k-3})}{T^2} \tag{10.11}$$

It is seen from Equation (10.10) that when the plant is controlled using a PI controller, the one-step ahead estimation $\hat{u}_k$ for a dropped control packet is a PI+PD+D2 prediction, where D2 means the second-order derivative.

- The PI part is $K_c\left(\Delta y_{k-1}+\frac{1}{T_i}\sum_{j=-\infty}^{k-1}\Delta y_j\right)$ with the proportional coefficient K_c and integral time T_i, respectively; it reuses the last control action as the control baseline in the current control period if the current control packet is dropped;
- The PD part is $\bar{K}_c\left[\Delta y_{k-1}+\bar{T}_d\left(\Delta y_{k-1}\right)^{(1)}\right]$ with the proportional coefficient $\bar{K}_c$ and derivative time $\bar{T}_d$, respectively; it captures the characteristics of the rate of change in the controlled variable y; and
- The D2 part is $\frac{1}{2}K_cK_{u2}T^2\left(\Delta y_{k-1}\right)^{(2)}$, which captures the characteristics of the acceleration or deceleration of the controlled variable y.

Rearranging Equation (10.5) gives

$$\begin{aligned}
&\hat{u}_k=\tilde{K}_c\left[\Delta y_{k-1}+\frac{1}{\tilde{T}_i}\sum_{j=-\infty}^{k-1}\Delta y_j+\tilde{T}_d\left(\Delta y_{k-1}\right)^{(1)}+\tilde{T}_d^{(2)}\left(\Delta y_{k-1}\right)^{(2)}\right]\\
&\tilde{K}_c=K_c\left(1+K_{u1}/Ti\right),\ \tilde{T}_i=T_i+K_{u1},\\
&\tilde{T}_d=\frac{K_{u1}T_i+\frac{1}{2}K_{u2}}{T_i+K_{u1}}T,\ \tilde{T}_d^{(2)}=\frac{1}{2}\frac{K_{u2}T_i}{T_i+K_{u1}}T^2
\end{aligned}\tag{10.12}$$

Equation (10.12) reveals that the PD2 compensation scheme in (10.8) for control packet dropout implements a PID2 predictive control. Since the second-order derivative of past control u, and consequently second-order derivative of past controlled variable y, are considered explicitly in the scheme, more accurate prediction can be expected than that from the simple extrapolation (10.2) at the cost of more computational demand.

If we set $K_{u2}=0$, the PD2 in Equation (10.8) is reduced to the PD in Equation (10.2). In practice, we may simply set $K_{u2}=K_{u1}=1$.

10.5 PD3 Prediction of Dropped Control Packets

In the scheme shown in Equation (10.8), the PD2 prediction of dropped control packets makes use of the acceleration or deceleration information of past control signals. Obviously, the

acceleration or deceleration is time-varying. To capture the characteristics of the rate of change in the acceleration or deceleration, the following function $f(\cdot)$ is chosen for Equation (10.1)

$$\begin{aligned} &\hat{u}_k = u_{k-1} + K_{u1}Tu_{k-1}^{(1)} + \tfrac{1}{2}K_{u2}T^2u_{k-1}^{(2)} + \tfrac{1}{6}K_{u3}T^3u_{k-1}^{(3)} \\ &K_{u1} \in [0,1], K_{u2} \in [0,1], K_{u3} \in [0,1] \end{aligned} \tag{10.13}$$

where $u_{k-1}^{(1)}$, the first-order derivative of u at the time instant $(k-1)T$, is approximated by Equation (10.3); $u_{k-1}^{(2)}$, the second-order derivative of u at the time instant $(k-1)T$, is estimated from (10.9); and $u_{k-1}^{(3)}$ is the third-order derivative of u at the time instant $(k-1)T$, and can be approximated using

$$u_{k-1}^{(3)} = \frac{u_{k-1} - 3u_{k-2} + 3u_{k-3} - u_{k-4}}{T^3} \tag{10.14}$$

Therefore, the dropout compensation scheme in Equation (10.13) is a PD3 compensator, where D3 means up to the third-order derivative.

When the plant is controlled by a PI controller of the form in Equation (10.4), we have

$$\begin{aligned} \hat{u}_k &= K_c\left(\Delta y_{k-1} + \frac{1}{T_i}\sum_{j=-\infty}^{k-1}\Delta y_j\right) + \bar{K}_c\left[\Delta y_{k-1} + \bar{T}_d\left(\Delta y_{k-1}\right)^{(1)}\right] \\ &\quad + \frac{1}{2}K_c\left(K_{u2} + \frac{1}{3}\frac{K_{u3}}{T_i}\right)T^2\left(\Delta y_{k-1}\right)^{(2)} + \frac{1}{6}\frac{K_cK_{u3}}{T_i}T^3\left(\Delta y_{k-1}\right)^{(3)} \\ \bar{K}_c &= K_{u1}K_c/T_i, \bar{T}_d = \left(T_i + \frac{1}{2}\frac{K_{u2}}{K_{u1}}\right)T \end{aligned} \tag{10.15}$$

where $(\Delta y_{k-1})^{(1)}$ and $(\Delta y_{k-1})^{(1)}$ are the first- and second-order derivatives of Δy_{k-1} at time instant $(k-1)T$, respectively; and $(\Delta y_{k-1})^{(3)}$ represents an approximation of the third-order derivative of Δy_{k-1} at the time instant $(k-1)T$,

$$\left(\Delta y_{k-1}\right)^{(3)} = \frac{\left(\Delta y_{k-1} - 3\Delta y_{k-2} + 3\Delta y_{k-3} - \Delta y_{k-4}\right)}{T^3} \tag{10.16}$$

It is seen from Equation (10.15) that if the plant is controlled using a PI controller, the one-step ahead estimation $\hat{u}_k$ is a PI + PD + D2 + D3 prediction, where D2 and D3 represent the second- and third-order derivatives, respectively. Compared with (10.10) for PD2, (10.15) for PD3 keeps the PI and PD terms unchanged, but enhances the D2 term and introduces the D3 term.

Rearranging Equation (10.15) gives

$$\begin{aligned}
\hat{u}_k &= \tilde{K}_c \left[\Delta y_{k-1} + \tfrac{1}{\tilde{T}_i} \textstyle\sum_{j=-\infty}^{k-1} \Delta y_j + \tilde{T}_d \left(\Delta y_{k-1} \right)^{(1)} \right. \\
&\quad \left. + \tilde{T}_d^{(2)} \left(\Delta y_{k-1} \right)^{(2)} + \tilde{T}_d^{(3)} \left(\Delta y_{k-1} \right)^{(3)} \right] \\
\tilde{K}_c &= K_c \left(1 + \tfrac{K_{u1}}{Ti} \right), \tilde{T}_i = T_i + K_{u1} \\
\tilde{T}_d &= \frac{K_{u1} T_i + \frac{1}{2} K_{u2}}{T_i + K_{u1}} T, \tilde{T}_d^{(2)} = \frac{1}{2} \frac{K_{u2} T_i + \frac{1}{3} K_{u3}}{T_i + K_{u1}} T^2, \tilde{T}_d^{(3)} = \frac{1}{6} \frac{K_{u3} T_i}{T_i + K_{u1}} T^3
\end{aligned} \tag{10.17}$$

Equation (10.17) shows that the PD3 compensator in Equation (10.13) is a PID3 predictive control. The scheme considers up to the third-order derivative of past control u and y explicitly, and is thus able to capture not only the dynamics of u but also the dynamics of y.

It is also observed that the PD3 in Equation (10.13) is reduced to the PD2 in Equation (10.8) if K_{u3} is set to be 0. In practice, we may simply set $K_{u3} = K_{u2} = K_{u1} = 1$.

10.6 Illustrative Examples

This section gives some examples to demonstrate the effectiveness of the proposed strategies of packet dropout compensation. Because the focus is on the control packet dropout compensation, detailed discussions on control design and controller tuning will be omitted.

10.6.1 Time Delay Process and Its Networked Control

Consider the following time delay process, which is common in process industry

$$G_p(s) = \frac{K_p}{T_p s + 1} e^{-\tau_p s}, K_p = 1, T_p = 2, \tau_p = 0.3 \tag{10.18}$$

where K_p, T_p, and τ_p are process gain, time constant, and time delay, respectively, with appropriate units.

For digital control of process (10.18), the control period is chosen to be $T = 0.2$ time units. A PI controller of the form (10.4) is used to control the process. It is tuned using ITAE (integral of timed absolute error) for set-point. The standard ITAE PI settings for set-point are

$$K_c = 0.586 K_p^{-1} T_p \tau_p^{-0.916}, T_i = T_p \left(1.030 - 0.165 \tau_p / T_p \right)^{-1} \tag{10.19}$$

These standard PI settings do not consider any network induced delay and control computation delay. Therefore, they need to be modified for NCS applications if the network induced delay or the control computation delay or the sum of both is significant.

Denote the sum of the network induced delay and the control computation delay by τ_c, which is normally time-varying and becomes infinity when a measurement or control packet is dropped. Applying the proposed real-time queuing protocol to this example, we can achieve the upper bound of τ_c to be T. Especially, if setting $T_D = T$ in the timeline of the proposed queuing architecture, we can achieve a fixed τ_c as

$$\tau_c = T_D = T = 0.2 \tag{10.20}$$

This fixed τ_c can be compensated easily through controller design. To fully compensate for the fixed $\tau_c = 0.2$ time units, we change the standard ITAE PI settings in (10.19) into

$$K_c = 0.586 K_p^{-1} T_p (\tau_p + \tau_c)^{-0.916}, T_i = T_p \left(1.030 - 0.165 \frac{\tau_p + \tau_c}{T_p}\right)^{-1} \tag{10.21}$$

Substituting the parameter values in Equations (10.18) and (10.20) into (10.21) gives

$$K_c = 2.0863, T_i = 2.0228 \tag{10.22}$$

10.6.2 Significant Step Changes in Set-Point and Load Disturbance

Significant external signals are fed into the NCS for evaluation of the system performance: (1) A unit step change in set-point is introduced at $t = 1$ to the closed-loop NCS; and (2) A negative unit step change in load disturbance is introduced at $t = 10$ to the NCS.

Without control packet dropout, the results of the networked control of the process are depicted in Figure 10.1. The upper plot of Figure 10.1 shows the closed-loop responses of the system to step changes in set-point and load disturbances, respectively; while the lower plot of the figure illustrates the corresponding control action through a zero-order hold.

Figure 10.1 shows that in response to the significant step change in set-point, a big jump in u is generated for fast set-point tracking. It will be seen later that this big jump in u requires special considerations in packet dropout compensation. Apart from this big jump, the control

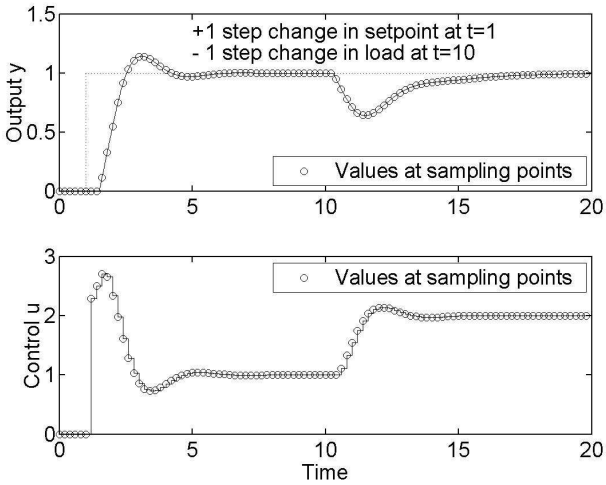

Figure 10.1: Closed-loop responses to step changes in set-point and load disturbance when there is no packet dropout.

signal jumps up or down in smaller steps.

For the significant step change in load disturbances, the process smooths out the step change. Consequently, the corresponding control action is much smoother than that for tracking the significant step change in set-point, as shown in Figure 10.1.

Various strategies for packet dropout compensation are simulated in the presence of the large external step change signals described above. The one-step ahead predictions of u, together with the estimation errors, are shown in Figure 10.2.

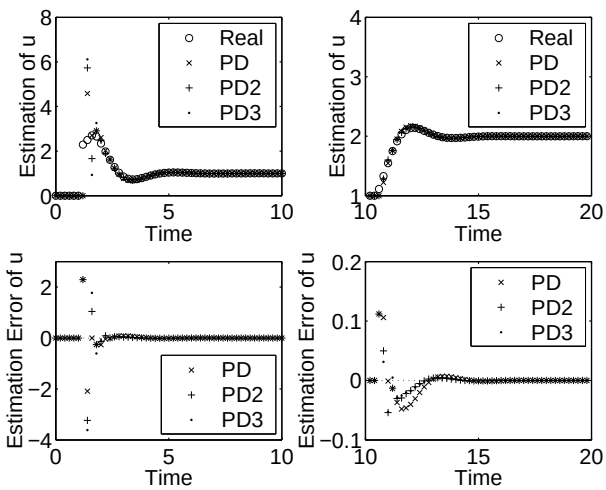

Figure 10.2: One-step ahead prediction of u for step changes in set-point (at $t = 1$) and load disturbance (at $t = 10$).

As shown in Figure 10.2, for the significant step change in set-point, the big jump in u results in necessary adjustments of the prediction of u over three control periods. The errors of the prediction for all three compensators are obvious during this adjustment, and are suppressed significantly after this. Therefore, the big jump in u requires special considerations in implementing a compensator for packet dropout. The following observations justify our statement.

- Difficulties resulting from the significant change in set-point are not unique to the problem discussed here for compensation of dropped packets. They are common in control design and other possible forms of compensators for packet dropout.
- In many applications, changes in set-point are known in advance [Tian and Gao, 1999b]. Passing on this information to the packet dropout compensator will help reduce the prediction error of u significantly in the first few control periods.

For the significant step change in load disturbances, Figure 10.2 shows that all three compensators, i.e., PD, PD2, and PD3, give satisfactory performance. Compared with the PD compensator, PD2 gives smaller prediction error of u. PD3 further improves the performance over the PD2. To evaluate the performance of the three compensators quantitatively, the SSE (sum of the square error) index defined below is computed for each of the compensators

$$\text{SSE Index} = \sum_k (\text{prediction error of } u \text{ at } t = kT)^2 \tag{10.23}$$

It is calculated from $t = 10$ when the step change in load is introduced till $t = 20$. The SSE results are tabulated in Table 10.1. They indicate that PD2 improves PD by over 36%, and PD3 improves PD by over 43%. Since the step-change load disturbances are the worst ones, the conclusion drawn here is similar for other forms of load disturbances although actual performance improvement may vary.

10.6.3 Sinusoidal Change in Set-Point

Let us consider a sinusoidal change in set-point, sin(t). A sinusoidal change in set-point can simulate time-varying set-point change that is smoother than the step change. Without control packet dropout, the results of the networked control of the process are depicted in Figure 10.3.

Table 10.1: SSE indices under various strategies for packet dropout compensation.

Strategy	PD	PD2	PD3
Step change in load	0.0330	0.0209	0.0187
Improvement over PD	–	36.76%	43.27%
Sinusoidal change in set-point	0.2867	0.0801	0.0728
Improvement over PD	–	72.06%	74.61%

The upper plot of Figure 10.3 shows the closed-loop responses of the system to the set-point change; while the lower plot of the figure illustrates the corresponding control action.

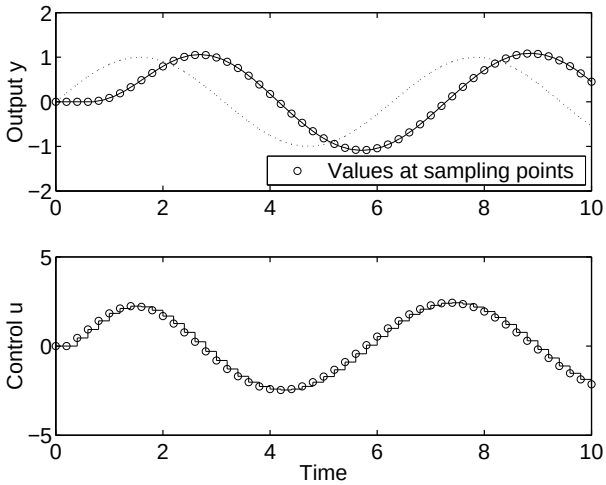

Figure 10.3: System responses to sinusoidal changes in set-point when there is no packet dropout.

For the three packet dropout compensators proposed in this work, the one-step ahead predictions of u, together with the estimation errors, are shown in Figure 10.4 for the sinusoidal changes in set-point. Figure 10.4 shows that PD2 and PD3 have similar performance. They both behave significantly better than the PD scheme. Quantitative computation of the SSE indices shows that PD2 improves PD by over 68%, and PD3 improves PD2 by over 70%.

10.6.4 Higher-Order Processes

The proposed real-time queuing protocol and packet dropout compensators have been shown above to be effective for networked control of first-order plus delay processes. They are also effective for networked control of higher-order plus delay processes since most higher-order processes can be approximated by a first-order plus delay model.

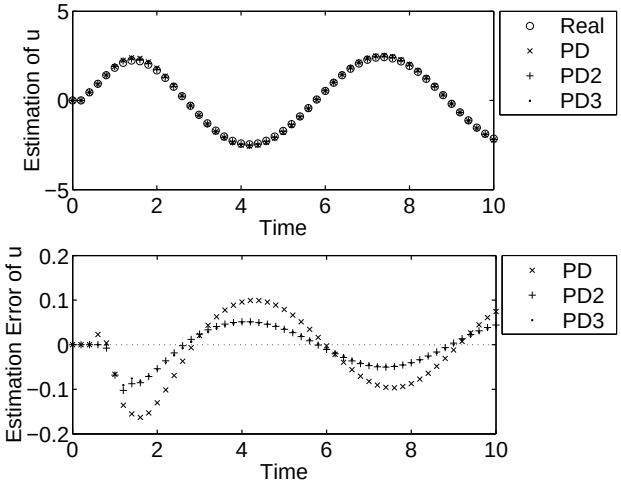

Figure 10.4: One-step ahead prediction of u for sinusoidal changes in set-point.

Let us consider a class of higher-order time delay processes governed by

$$G_p(s) = \frac{1}{(2s+1)(s+1)} e^{-0.3s} \tag{10.24}$$

For controller design and tuning, one expects to reduce the above higher-order plus delay model into a first-order plus delay one. There are a number of methods for model reduction. Among these methods, Skogestad's half rule is simple and effective [Skogestad, 2003]. The half rule evenly distributes the largest neglected (denominator) time constant (lag) to the effective delay and the smallest retained time constant. Applying the half rule to the higher-order plus delay model in Equation (10.24) yields the following reduced first-order plus delay model

$$G_p = \frac{K_p}{T_p s + 1} e^{-\tau_p s}, K_p = 1, T_p = 2.5, \tau_p = 0.8 \tag{10.25}$$

This reduced model (10.25) is used to design and tune the controller. Without consideration of network induced delay and control computation delay, the ITAE PI controller settings for the process are computed from Equation (10.19).

Set the control period $T = 0.2$. Applying the queuing protocol, we have the upper bound of $\tau_c = T$, where τ_c is the sum of the network induced delay and the control computation delay. Especially, when setting $T_D = T$ in the protocol, we can achieve a fixed $\tau_c = T_D = T = 0.2$ time units. This fixed delay can be compensated through a co-design of network and control, e.g., from (10.21) we have the ITAE PI controller settings of $K_c = 1.3565$ and $T_i = 2.5934$.

To test the performance of the NCS, a unit step change in set-point and a negative step change in load disturbance at $t = 15$ are introduced, respectively. Without control packet dropout, the responses of the NCS to these step changes are depicted in Figure 10.5. The corresponding control action u, which is computed at discrete sampling points and output through a zero-order hold, is also shown in the figure. It is seen from Figure 10.5 that satisfactory performance of the networked control has been achieved from the proposed real-time queuing protocol together with the effective model reduction and controller design.

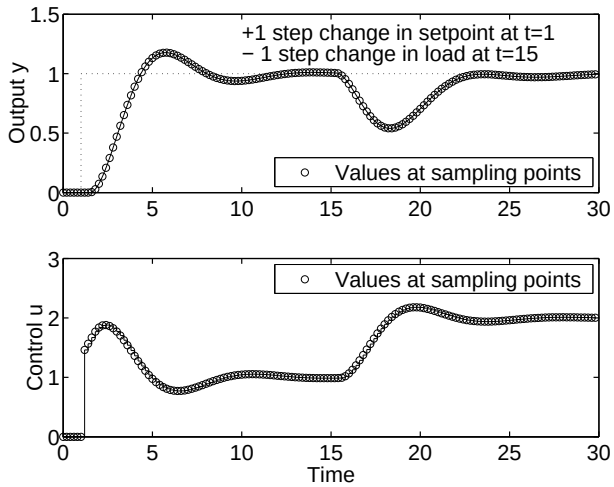

Figure 10.5: NCS responses to step changes in set-point and load disturbance for process (10.24) when there is no packet dropout.

The packet dropout compensators proposed in this work are simulated in the presence of step changes in both set-point and load disturbance. The one-step ahead predictions of u, together with the estimation errors, are shown in Figure 10.6. It is seen from Figure 10.6 that all compensators give acceptable one-step ahead predictions of u. Although the set-point step change causes a big jump in u and consequently results in an obvious estimation error of u, this estimation error occurs only in a single control period and vanishes quickly. Again, knowing set-point changes in advance will help reduce the estimation error [Tian and Gao, 1999b]. When both set-point tracking and load disturbance rejection are considered, PD2 is a good choice.

10.7 Remarks

One possible difficulty in applying the proposed compensators is the necessary adjustments in the first few control periods after a significant step change is introduced into the set-point. This

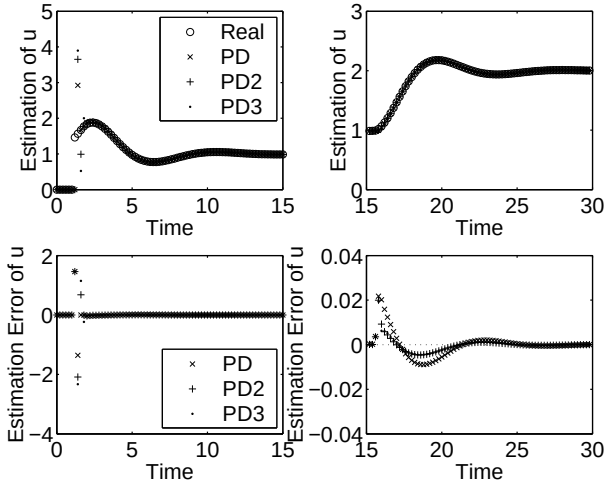

Figure 10.6: One-step ahead prediction of u for step changes in set-point (at $t = 1$) and load disturbance (at $t = 15$) for networked control of (10.24).

requires a big jump in the control signal for fast set-point tracking. This difficulty is not unique to the problem discussed in this work for packet dropout compensation; it is common to control design and other possible forms of packet dropout compensators. Knowing the set-point change in advance will help reduce the prediction error of the control signal significantly.

Another consideration in using the proposed compensators is measurement noise. A high level of noise will result in fluctuations in control signal. These fluctuations may be further amplified by the derivative action of the compensators. This problem is also not unique to the problem of packet dropout compensation; it is common to predictor design and implementation. Pre-processing and filtering of measured data are necessary in most control systems.

It is noted that the PD2 scheme improves the PD scheme significantly under typical scenarios discussed in this work; the PD3 has improvement over the PD2 at the cost of increased computational demand and potential risk of noise amplification. Therefore, the PD2 is recommended for actual implementation of packet dropout compensation.

If sustained packet losses occur, the proposed compensators will not function. It is our belief that the packet dropout, which results from unreliable networking, should be first addressed in the design of networks. This is a better option than implementing sophisticated packet dropout compensation. Therefore, sustained sequences of successive dropped packets, which will unlikely happen under normal conditions, might be an indication of abnormal conditions or an unsatisfactory network design. This is an issue of integrated design for the overall NCS.

10.8 Summary of This Chapter

Existing methodologies either do not compensate for control packet dropout at all or attempt to compensate for it at the central controller through sophisticated algorithms that strongly rely on accurate process models. With the application of the real-time queuing protocol that we developed for NCS, three model-free schemes have been proposed for packet dropout compensation. They are all implemented at the smart actuator, and are computed from past control signals. With the proposed compensators, together with the real-time queuing protocol, the network induced delay in NCS becomes fixed and is limited to within a control period. This will help achieve predictable dynamics of NCS communications and control. The fixed and relatively small network induced delay can be compensated through control strategy design. Examples have been given to demonstrate these concepts and methods.

10.9 Nomenclature of This Chapter

Abbreviations

ITAE	Integral of Timed Absolute Error
PD	Proportional plus Derivative
PD2	Proportional plus up to the Second-Order Derivative
PD3	Proportional plus up to the Third-Order Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
NCS	Networked Control System/s
SSE	Sum of Square Error

Symbols

$f(\cdot)$	function
G_c, G_p	Controller and plant transfer functions, respectively
K_c, $\bar{K}_c$	Controller gain, and equivalent controller gain, respectively
K_p	Process gain

K_{u1}, K_{u2}, K_{u3}	Proportional coefficients
k	Integer representing the kth control period
lr	Loss rate
n	Integer
Q_1, Q_2	Two queues in the queuing protocol
s	Laplace transform operator
T	Control period
t	Time variable
T_D	Time instant to dequeue Q_1
$\bar{T}_d$	Equivalent derivative time
T_i, $\bar{T}_i$	Integral time, and equivalent integral time, respectively
T_L	Latest time instant to accept control packet
T_p	Process time constant
u, $\hat{u}$	Control signal, and its estimate, respectively
y, y_r	Controlled variable, and its set-point, respectively

Greek Letters

Δy_j	Control error $(= y_r - y_j)$
τ_c	Sum of network delay and control computation delay
τ_p	Process time delay

Superscripts

(i)	The first-, second, or third-order derivative for $i = 1, 2$, or 3

Subscripts

j, k	Integers representing the jth and kth control periods, respectively

Chapter 11

Integrated Design of Real-Time Networked Control Systems

From our effort in the understanding of the non-uniform distribution of network induced delay (Chapter 8), in developing the queuing protocol (Chapter 9), and in dealing with control packet dropout (Chapter 10) for networked control systems (NCS), a general framework is developed in this work to deal with network complexity and **integrated design of real-time NCS**. When the complex traffic of an NCS is treated as stochastic and bounded variables, simplified yet improved methods for robust stability and control synthesis can be developed to guarantee the stability of the systems.

Integrated design of network and control is an effective approach to simplify the network behaviour and consequently to maximise the Quality of Control (QoC) performance of the overall NCS. It combines the queuing protocol, packet dropout compensators, and control design. Consequently, the network induced delay can be limited to within a single control period, significantly simplifying the network complexity as well as system analysis and design. The resulting delay can be further reduced significantly if the system is tolerant of a certain level of packet loss rate.

Part of this work has been published in [Tian and Levy, 2008b; Tian and Levy, 2007].

11.1 Networked Control and Scheduling

An NCS typically has multiple control tasks; and thus is related to multi-tasking scheduling. In scheduling theory, the worst-case execution time of a task is used in order to achieve predictive system behaviour (Chapter 5) [Balbastre *et al.*, 2004; Samard and Balas, 2003; Shaw, 2001]. It is conventionally treated as a constant. To a large extent, the worst-case execution time determines the schedulability and QoC of multiple control tasks.

It is worth mentioning that unlike conventional multi-tasking scheduling, NCS communication events are not pre-emptive once they have happened. Once a packet is sent out, the transmission cannot be stopped. Thus, reducing the worst-case communication delay is crucial to ensure the schedulability and performance of the communication and control tasks of NCS.

Applying the multi-tasking scheduling theory in NCS, we have used the worst-case communication delay (WCCD) in our real-time queuing architecture (Chapter 9) [Tian *et al.*, 2006a; Tian *et al.*, 2006c]. Especially, we use the WCCD as a constant timing parameter to smooth out the time-varying communication delay for predictive real-time computing and control of NCS.

Recent studies have shown that the network delay in a real-time NCS has non-uniform distribution [Tian and Levy, 2008b; Tipsuwan and Chow, 2004a] and multi-fractal nature (Chapter 8) [Tian *et al.*, 2007]. However, these findings have not been effectively utilised in NCS analysis and design. Through extensive investigations into various scenarios of NCS, we have recently observed that most communication delays are far below the WCCD, implying that the use of the WCCD in the conventional sense will result in tight scheduling and conservative QoC.

We will show how to make the WCCD configurable. The main idea is to significantly reduce the value of the WCCD in the real-time queuing protocol under certain level of packet loss rate. The packet losses can be compensated through some simple strategies. This also highlights the necessity of the integrated design of networking and control.

11.2 Further Analysis of NCS Network Delay

Let us re-consider Case Study 1 carried out in Section 9.5 of Chapter 9 for a hierarchical NCS with Two-Segments/10Mbps network architecture. The system has been modelled in

Section 8.1 of Chapter 8. The NCS is a hierarchical control system with 30 sensors, 20 actuators, 5 control computers, 5 management computers, and other devices. It controls 20 loops. The system architecture, Two-Segments/10Mbps network architecture, and network traffic specifications have been described in detail in Section 8.1 of Chapter 8.

Selected results of our simulation and performance analysis for the modelled NCS have been tabulated in Table 8.4 of Chapter 8. Controller-to-actuator delay of the NCS is graphically shown in Figure 8.6. Also, Figures 8.4 depicts sensor-to-controller delay, which has non-uniform distribution and multi-fractal nature as analysed in Chapter 8.

Table 8.4 indicates that the sensor-to-actuator delay ranges from 16.30ms to 93.34ms, and the controller-to-actuator delay changes from 16.93ms to 34.16ms. Considering both sensor-to-controller delay and controller-to-actuator delay, we can see that the total NCS communication latency is between 33.23ms and 127.50ms, resulting in a jitter of 94.27ms. Both the latency and jitter are significant. The WCCD here is 127.50ms, which leaves us only 72.50ms for all other tasks in a control period of $T = 200$ms.

Although the WCCD is as high as 127.50ms, we have observed some delay spikes in Figure 8.4, implying that most sensor-to-controller delays are far below the worst-case value. This has also been clearly shown in a plot of the distribution of the delay in Figure 8.5. The accumulated percentage of the delay is given in Figure 11.1.

It is seen from Figures 8.4, 8.5, and 11.1 that in most cases, small delays are dominant while large delays are exiguous. Therefore, the probability of small communication delays is higher than that of large delays. A detailed analysis of the accumulated distribution of the sensor-to-controller delay is summarised in Table 11.1. It is observed from Table 11.1 that 95% sensor-to-controller delays fall below 47ms; 99% delays are less than 60ms. These delays are far below the worst-case value of 93.34ms.

Table 11.1: Accumulated distribution of the sensor-to-controller delays.

Loss rate (%)	0	0.50	1.0	1.9	3.3	5.0
Accumulated % of delays	100	99.5	99.0	98.1	96.7	95.0
Delay (ms)	93.34	77	60	54	50	47
Delay reduction (ms)	-	16.34	33.34	39.34	43.34	46.34
Delay reduction (%)	-	17.51	35.72	42.15	46.43	49.65

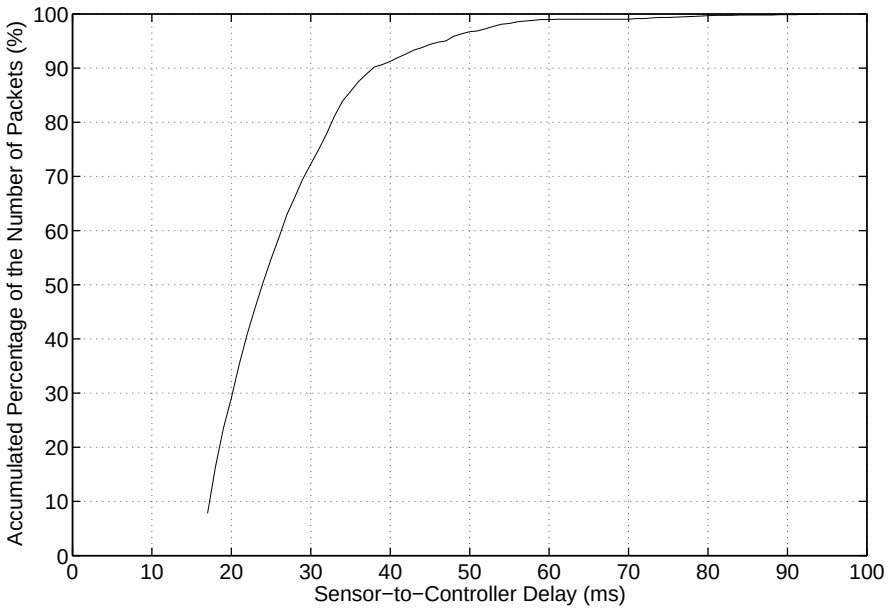

Figure 11.1: Accumulated percentage of the sensor-to-controller delays.

11.3 Configuring the WCCD

Our approach for configuring the WCCD in real-time NCS is based on our observations and analysis in previous sections.

From Table 11.1, if the NCS is tolerant of a 5% packet loss rate, the worst-case sensor-to-controller delay can be reduced from 93.34ms down to 47ms, representing a reduction of 46.34ms. This means that the WCCD can be reduced from 127.50ms down to 81.16ms (a 36% improvement) under the level of 5% packet loss rate.

If the desired level of the packet loss rate for sensor-to-controller packets is 1.9%, the improvement in the WCCD is better than 39.34ms. The 39.34ms drop of the WCCD from 127.50ms represents a significant improvement of nearly 31%.

Further analysis from Table 11.1 shows that a 1% loss rate of the measurement packets allows a drop of 33.34ms in the WCCD (a 26% reduction). A 0.5% loss rate of the measurement packets corresponds to a reduction of 16.34ms in the WCCD (a 23% improvement). All these improvements are significant with the low loss rates.

The overall design process for the WCCD is just like to put a horizontal line in Figure 8.4.

Move the line up or down to adjust the threshold for different levels of packet loss rate for the measurement packets. Or, thinking about putting a vertical line in Figure 8.5 or Figure 11.1, move the line to the left or right for an acceptable level of packet loss rate.

The fundamental requirements of the proposed approach for configuring the WCCD are:

1) The communication delay is non-uniformly distributed and most of the delays are far below the worst one – This is verified in our investigations in Chapter 8 and also in [Tipsuwan and Chow, 2004a].
2) The small level of packet loss rate can be tolerated – this is guaranteed through our simple packet dropout compensators (Chapter 10).

11.4 Integrated Design of NCS

From our recent development, this work aims to develop a general framework to deal with the NCS network complexity. The basic idea is the integrated design of network and control. The procedure of the framework is summarised below.

Step 1. Apply the real-time queuing protocol in the NCS network to make the dynamic behaviour of the network traffic more predictable, thus simplifying the network dynamics. The most important achievement is the predictability of the network induced delay. (Chapter 9).

Step 2. Configure the WCCD under an acceptable level of packet loss rate to significantly reduce the network induced delay in networked control. (Section 11.3).

Step 3. For any control packet loss, activate a packet dropout compensator to predict the lost control signal. A control packet that is received after the control period ends is treated as a lost packet in that control period. (Chapter 10).

Step 4. Through the above three steps, the network induced delay is limited to within a single control period, largely simplifying the system analysis and design. Then, the predictable network induced delay is compensated through controller design. Many controller strategies can be used for predictable delay compensation, e.g., [Tian and Gao, 1998c; Tian and Gao, 1998b].

11.5 Case Studies

Again, consider Case Study 1 discussed in Section 9.5 of Chapter 9 for an NCS with Two-Segments/10Mbps network architecture. It has been further studied in Sections 11.2 and 11.3.

Using the integrated design framework proposed above in Section 11.4, we apply the real-time queuing protocol first to smooth out the time-varying network induced delay. This gives the timing behaviour shown in Figure 9.3 for the closed-loop NCS. The time instant T_C for dealing with packet dropout is fixed at 180ms, and the time instant T_D for control output is fixed at 190ms.

Next, allowing 1% loss rate for control packets, we will be able to reduce the WCCD by 33.34ms. This is a 26% reduction in the WCCD. As a result, both T_C and T_D can be reduced by the same amount. The timing behaviour of NCS network is depicted in Figure 11.2, which clearly shows the improvement of the delay performance in comparison with Figure 9.3.

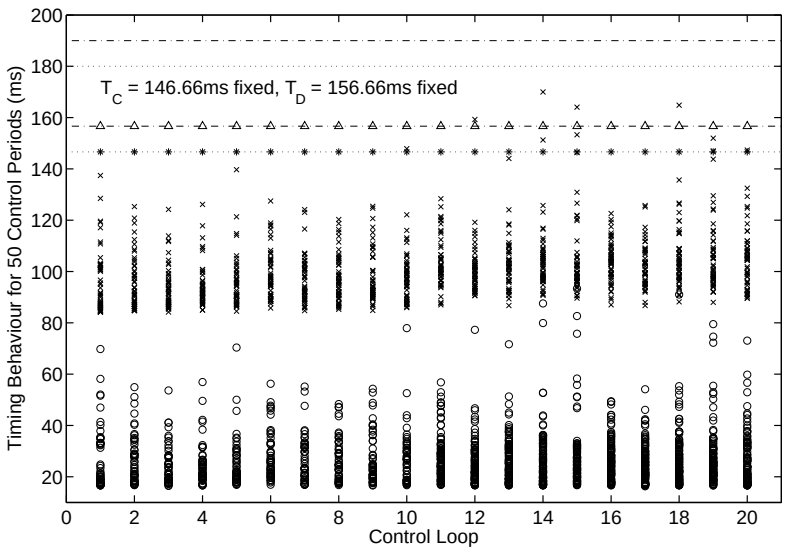

Figure 11.2: Timing behaviour of the NCS with 1% packet loss rate ('O': measurement packets received by controller; 'X': Control packets received by the actuator; '*': T_C; 'Δ': T_D).

Furthermore, because some packets will be dropped, a simple packet dropout compensator is employed to ensure smooth operation of the control system.

Finally, due to the fact that the network delay has been limited to within a single control period, simplified stability analysis and design can be carried out with improved performance.

11.6 Summary of This Chapter

The non-uniform distribution of the communication delay in NCS makes it possible to reduce the WCCD with a sacrifice for a certain level of packet loss rate. An approach has been proposed to configure the WCCD for much less conservative system design and multi-tasking scheduling in real-time NCS. When a small level of packet loss rate can be tolerated, a threshold can be set as a configurable WCCD, which is far below the real worst-case communication delay.

Furthermore, a general framework has been developed in this work to deal with the network complexity and integrated design of NCS for real-time applications. It consists of four main steps: 1) Apply a real-time queuing protocol to achieve predictable network induced delay; 2) Configure the worst-case communication delay; 3) Activate packet loss compensation when a packet is lost; and 4) Compensate for the predictable network induced delay in control design. The framework has emphasised integrated design of network, scheduling, and control. The resulting network induced delay is limited to within a single control period, significantly simplifying the network complexity as well as system analysis and design while improving the performance of the overall NCS.

11.7 Nomenclature of This Chapter

Abbreviations

NCS	Networked Control System/s
WCCD	Worst-Case Communication Delay

Symbols

Q_1	Queue to store one control packet
T	Control period
T_C	Time instant to deal with control packet dropout
T_D	Time instant to dequeue Q_1

Chapter 12

Conclusions and Future Work

12.1 Summary of the Research

As a type of real-time systems, real-time control systems are widely deployed in various applications. However, challenging problems exist in system design and implementation, and pose performance limitations to the control systems. In a real-time control system, two of the most important issues are timing and QoC (Quality of Control) performance. We have addressed these two broad issues in this thesis through comprehensive research on **dynamics analysis and integrated design** for **seven research problems** identified from **three related areas**.

The first area that we have investigated is about **control design for controllers**. The control strategy development is for maintenance and improvement of QoC performance of the process control. Well designed control system architecture together with appropriately developed control strategies is essential for the overall control system.

- To facilitate the research on control design, the complex, and industrially significant, reactive distillation processes, have been investigated. Model-free pattern predictive control (Research Problem 1 in Section 1.3) has been developed through integrated design of pattern recognition, fuzzy logic, non-linear transformation, and predictive control. The developed control strategies have been verified on a pilot-scale reactive distillation column. This is **the first main contribution** of this thesis.

A good control design gives good control performance only if it is well implemented in

software on the controller. The software implementation of control strategies has been a largely ignored area in the control system development. **The second area** that we have studied is about **control implementation on controllers** with regard to multi-tasking scheduling. **Integrated design of control and scheduling** has been carried out through linking the scheduling directly or indirectly to the QoC of the control system.

- Because time delay and jitter affect the dynamics and timing of real-time control significantly, they can be used to indirectly characterise the system QoC. With this understanding, it is important to reduce control latency and jitter for a control system. Therefore, the problem of reducing control latency and jitter (Research Problem 2 in Section 1.3) has been explored in this thesis. Strategies have been developed to tackle this problem and have been successfully applied to real-time networked control and operation of a large-scale industrial robotic system. This is **the second main contribution** of this thesis.
- Feedback scheduling makes it possible to directly evaluate the QoC performance and consequently to re-schedule the computing and network resources whenever necessary. However, this may not be always feasible especially in overload conditions. The problem of hierarchical feedback scheduling (Research Problem 3 in Section 1.3) has been raised for study in this thesis. A task model and a hierarchical feedback scheduling framework have been proposed to solve the problem effectively and efficiently. This is **the third main contribution** that we have claimed in this thesis.

With the integration of information, communication, and control in modern real-time control applications, networked control is becoming increasingly significant. However, there are still major difficulties in analysis, design, and implementation of real-time networked control. **The third area** that we have investigated in this thesis is about **control design and implementation in networked environments** with the focus on **dynamics analysis and integrated design** of control, network, and resource scheduling.

- Time-varying network induced delay is one of the major difficult problems in networked control. To develop a better understanding of the dynamics of the network delay, we have made effort to analyse the behaviour of the delay (Research Problem 4 in Section 1.3). Our research has revealed the non-uniform distribution and multi-fractal nature of the network delay in real-time networked control. This is **the forth main contribution**

claimed in this thesis.

- When both QoC and network QoS (Quality of Service) are considered, the network induced delay problem can be tackled from the co-design of network and control. This is Research Problem 5 defined in Section 1.3. Introducing artificial delay, a real-time queuing protocol has been proposed to smooth out the time-varying network induced delay, leading to more predictable communication behaviour which is favourable for real-time applications. This is claimed as **the fifth main contribution** of this thesis.

- Packet dropout is another major difficult problem in analysis an design of a network control system (NCS), and has been investigated systematically in this thesis (Research Problem 6 in Section 1.3). Simple packet dropout compensators have been developed for implementation on the actuators to predict lost control packets from past control signals. This achievement is **the sixth main contribution** of this thesis.

- Integrated design of network and control has been considered as one of the major research topics in this thesis (Research Problem 7 in Section 1.3). Making use of the non-uniform distribution property of the NCS network delay, the worst-case communication delay (WCCD) is made configurable for significant reduction of the delay if the system is tolerant of a certain level of packet loss rate. With combined applications of the queuing protocol, the packet loss compensators, the WCCD configuration, and control design, an integrated NCS design framework has been developed to maximise the NCS performance improvement. It limits the network induced delay to within a single control period, leading to significant simplification of NCS analysis and considerable improvement of system QoC. This is claimed as **the seventh main contribution** of this thesis.

In conclusion, we have fully completed the research outlined in Chapter 1 and have presented our main contributions in this thesis.

12.2 Future Work

The presented work in this thesis on dynamics analysis and integrated design of real-time control systems suggests many interesting research directions.

While all technologies developed in this thesis have been verified via simulation studies, experimental testing, or practical applications, there is significant potential to apply the technologies in more practical control systems. This would be useful for promoting new technologies in real-time control for simplified system analysis and design and also for QoC improvement.

The queuing protocol developed in this thesis introduces artificial delay to smooth out network induced delay. It is effective for a large class of real-time control systems that require accurate timing. There also exist many control systems, e.g., industrial process control applications, in which imposing the control signal to the plant once it is available may be preferable. In this case, how to analyse and design the network, scheduling, and control in an integrated framework to ensure the QoS for the control network and the QoC for the control tasks is still difficult, and is worthy to be investigated.

Networked control has been deeply studied in this thesis. However, we have not specified wireless networks explicitly. Wireless communication is a fast growing area, and is becoming part of our daily life. Two of the challenging problems in networked control, i.e., time-varying networked induced delay and packet dropout, which we have addressed, become more evident and severer in wireless networked applications. Research and development of innovative technologies for wireless networked control are emerging. The outcomes presented in this thesis would be a good basis for investigation into wireless NCS.

Finally, there have been major difficulties in modelling and simulating NCS in a uniform software environment and platform. In this thesis, we have used several software packages for complementary modelling and simulation of networked control, for example, ns2 (or Opnet) for networks, Matlab/TrueTime for process dynamics and scheduling. However, time-driven Matlab/TrueTime is not suitable for comprehensive simulation of complex networks, and event-driven ns2 is not designed for computation of continuous-time dynamics. Effort is being made in three directions: (1) Simulate networks in ns2 (or Opnet), export the network dynamics to a file, and then read this file into Matlab for control system simulation; (2) Embed continuous-time simulation into event-driven ns2 by adding continuous-time simulation agents; and (3) Build an interface between Matlab and ns2 (or Opnet) for direct information exchange. We are making progress in all these three directions. It would be worthwhile, although difficult, to find a simple and convenient mechanism for modelling and simulation of networks, scheduling, and process control simultaneously in a uniform environment.

Literature Cited

[Abeni and Buttazzo, 1999] L. Abeni and G. Buttazzo. Adaptive bandwidth reservation for multimedia computing. In *Proceedings of the 6th IEEE International Conference on Real-Time Computing Systems and Applications*, pages 70–77, Hong Kong, P. R. China, Dec 1999. IEEE Computer Society, Washington, DC, USA.

[Abeni *et al.*, 2005] L. Abeni, T. Cucinotta, G. Lipari, L. Marzario, and L. Palopoli. QoS management through adaptive reservations. *Real-Time Systems*, 29(2-3):131–155, Mar 2005.

[Al-Arfaj and Luyben, 2002] M. A. Al-Arfaj and W. L. Luyben. Comparative control study of ideal and methyl acetate reactive distillation. *Industrial and Engineering Chemistry Research*, 41(16):3784–3796, Aug 2002.

[Amirijoo *et al.*, 2007] M. Amirijoo, J. Hansson, S. H. Son, and S. Gunnarsson. Experimental evaluation of linear time-invariant models for feedback performance control in real-time systems. *Real-Time Systems*, 35(3):209–238, Apr 2007.

[Amirijoo *et al.*, 2008] Mehdi Amirijoo, Joergen Hansson, Svante Gunnarsson, and Sang H. Son. Quantifying and suppressing the measurement disturbance in feedback controlled real-time systems. *Real-Time Systems*, 40(1):44–76, Oct 2008.

[Anh *et al.*, 2000] V. V. Anh, Q. M. Tieng, and Y. K. Tse. Cointegration of stochastic multifractals with application to foreign exchange rates. *International Transactions in Operational Research*, 7(4-5):349–363, July-Sept 2000.

[Anh *et al.*, 2001] V. V. Anh, K. S. Lau, and Z. G. Yu. Multifractal characterization of complete genomes. *Journal of Physics A - Mathematical and General*, 34(36):7127–7139, Sept 2001.

[Anh *et al.*, 2002] V. V. Anh, K. S. Lau, and Z. G. Yu. Recognition of an organism from fragments of its complete genome. *Physics Review E*, 66(3):Art. No. 031910, Sept 2002.

[Antsakls and Baillieul, 2004] Panos Antsakls and John Baillieul. Guest editorial: Special issue on networked control systems. *IEEE Transactions on Automatic Control*, 49(9):1421–1423, Sept 2004.

[Antsakls and Baillieul, 2007] P. Antsakls and J. Baillieul. Guest editorial: Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1):5–8, Jan 2007.

[Arzen *et al.*, 2007] Karl-Erik Arzen, Antonio Bicchi, Gianluca Dini, Stephen Hailes, Karl H. Johansson, John Lygeros, and Anthony Tzes. A component-based approach to the design of networked control systems. *European Journal of Control*, 13(2-3):261–279, Mar-June 2007.

[Aström and Wittenmark, 1997] K. J. Aström and B. Wittenmark. *Computer-Controlled Systems*. Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 1997.

[Athimathi and Radhakrishnan, 2006] Sachidanandam Athimathi and Thota K. Radhakrishnan. Control system design for a single feed ETBE reactive distillation column. *Chemical Engineering and Technology*, 29(10):1137–1154, Oct 2006.

[Babuška, 1999] R. Babuška. An overview of fuzzy modeling and model-based fuzzy control. In H.B. Verbruggen and R. Babuška, editors, *Fuzzy Logic Control Advances in Applications*, pages 3–35. World Scientific, 1999.

[Balasubramhanya and Doyle, 2000] L. S. Balasubramhanya and F. J. Doyle. Nonlinear model-based control of a batch reactive distillation column. *Journal of Process Control*, 10(2-3):209–218, Apr-June 2000.

[Balbastre *et al.*, 2004] Patricia Balbastre, Ismael Ripoll, Josep Vidal, and Alfons Crespo. A task model to reduce control delay. *Real-Time Systems*, 27(3):215–236, Sept 2004.

[Beccari *et al.*, 2005] G. Beccari, S. Caselli, and F. Zanichelli. A technique for adaptive scheduling of soft real-time tasks. *Real-Time Systems*, 30(3):187–215, July 2005.

[Bisowarno *et al.*, 2003] B. H. Bisowarno, Yu-Chu Tian, and Moses O. Tadé. Model gain scheduling control of an ethyl tert-butyl ether reactive distillation column. *Industrial and Engineering Chemistry Research*, 42(15):3584–3591, July 2003.

[Bisowarno *et al.*, 2004] B. H. Bisowarno, Yu-Chu Tian, and Moses O. Tadé. Adaptive control of an ETBE reactive distillation column. *Journal of Chemical Engineering of Japan*, 37(2):210–216, Feb 2004.

[Bohr and Jensen, 1987] T. Bohr and M. H. Jensen. Order parameter, symmetry-breaking, and phase-transitions in the description of multifractal sets. *Physics Review A*, 36(10):4904–4915, Nov 1987.

[Burns and Wellings, 2001] Alan Burns and Andy Wellings. *Real-Time Systems and Programming Languages*. Addison Wesley Longmain, 2001.

[Buttazzo and Aneni, 2002] G. Buttazzo and L. Aneni. Adaptive workload management through elastic scheduling. *Real-Time Systems*, 23(1-2):7–24, July-Sept 2002.

[Buttazzo *et al.*, 2002] G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni. Elastic scheduling for flexible workload management. *IEEE Transactions on Computers*, 51(3):289–302, Mar 2002.

[Buttazzo *et al.*, 2007] G. Buttazzo, M. Velasco, and P. Martí. Quality-of-control management in overloaded real-time systems. *IEEE Transactions on Computers*, 56(2):253–266, Feb 2007.

[Canessa, 2000] E. Canessa. Multifractality in time series. *Journal of Physics A - Mathematical and General*, 33:3637–3651, May 2000.

[Cervin *et al.*, 2002] Anton Cervin, Johan Eker, Bo Bernhardsson, and Karl-Erik Årzén. Feedback-feedforward scheduling of control tasks. *Real-Time Systems*, 23(1-2):25–53, July-Sept 2002.

[Cervin, 2003] Anton Cervin. *Integrated control and real-time scheduling*. PhD Thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, Apr 2003.

[Chan and Özgüner, 1995] H. Chan and Ü. Özgüner. Closed-loop control of systems over communications network with queues. *International Journal of Control*, 62(3):493–510, Sept 1995.

[Chandra *et al.*, 2003] R. Chandra, X. Liu, and L. Sha. On the scheduling of flexible and reliable real-time control systems. *Real-Time Systems*, 24(2):153–169, Mar 2003.

[Charpentier, 2007] Jean-Claude Charpentier. In the frame of globalization and sustainability, process intensification, a path to the future of chemical and process engineering (molecules into money). *Chemical Engineering Journal*, 134(1-3):84–92, Nov 2007.

[Chen and Yu, 2008] Chin-Shih Chen and Cheng-Ching Yu. Effects of relative volatility ranking on design and control of reactive distillation systems with ternary decomposition reactions. *Industrial and Engineering Chemistry Research*, 47(14):4830–4844, July 2008.

[Chen *et al.*, 2006] Liang Chen, Jianming Zhang, and Shuqing Wang. Scheduling and control co-design for delay compensation in the networked control system. *Asian Journal of Control*, 8(2):124–134, June 2006.

[Chien *et al.*, 2005] I. L. Chien, Y. P. Teng, H. P. Huang, and Y.T. Tang. Design and control of an ethyl acetate process: coupled reactor/column configuration. *Journal of Process Control*, 15(4):435–449, June 2005.

[Choi and Agrawala, 2000] Seonho Choi and Ashok K. Agrawala. Scheduling of real-time tasks with complex constraints. In Günter Haring, Christoph Lindemann, and Martin Reiser, editors, *Performance Evaluation: Origins and Directions*, volume 1769 of *Lecture Notes in Computer Science*, pages 253 – 282. Springer-Verlag Berlin, Heidelberger Platz 3, D-14197 Berlin, Germany, 2000.

[da Rocha and da Costa, 2007] L. E. C. da Rocha and L. da Costa. An adaptive routing strategy for packet delivery in complex networks. *New Journal of Physics*, 9:108, Apr 2007.

[Dalaouti and Seferlis, 2006] N. Dalaouti and P. Seferlis. A unified modeling framework for the optimal design and dynamic simulation of staged reactive separation processes. *Computers and Chemical Engineering*, 30(8):1264–1277, June 2006.

[Davis and Burns, 2005] R. I. Davis and A. Burns. Hierarchical fixed priority pre-emptive scheduling. In *Proceedings RTSS 26th IEEE International Real-Time Systems Symposium*, pages 389–398, Miami, FL, USA, 5-8 Dec 2005. IEEE Computer Society, Washington DC, USA.

[Davis and Burns, 2006] R. I. Davis and A. Burns. Resource sharing in hierarchical fixed priority pre-emptive systems. In *Proceedings of 2006 IEEE 27th Real-Time Systems Symposium*, pages 257–270, Rio de Janeiro, Brazil, Dec 5-8 2006. IEEE Computer Society, Washington DC, USA.

[Davis *et al.*, 2007] Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. Controller area network (CAN) schedulability analysis: refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, Apr 2007.

[Deng and Liu, 1997] Z. Deng and J.W.-S. Liu. Scheduling real-time applications in an open environment. In *Proceedings of the 18th IEEE Real-Time Systems Symposium*, pages 308–319, San Francisco, CA, USA, June 1997. IEEE Computer Society, Los Alamitos, CA, USA.

[Deng *et al.*, 1997] Z. Deng, J.W.-S. Liu, , and J. Sun. A scheme for scheduling hard real-time applications in open system environment. In *Proceedings of the 9th Euromicro Workshop on Real-Time Systems*, pages 191–199, Toledo, Spain, June 1997. IEEE Computer Society, Los Alamitos, CA, USA.

[Eker *et al.*, 2000] J. Eker, P. Hagander, and K. Erik Årzén. A feedback scheduler for real-time control tasks. *Control Engineering Practice*, 8(12):1369–1378, Dec 2000.

[Estrin *et al.*, 2002] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, Jan 2002.

[Feder, 1988] J. Feder. *Fractals*. Plenum Press, New York and London, 1988.

[Fidge and Tian, 2006] Colin Fidge and Yu-Chu Tian. Functional analysis of a real-time protocol for networked control systems. In *Automated Technology for Verification and Analysis. 4th International Symposium, ATVA 2006*, volume 4218 of *Lecture Notes in Computer Science: ATVA 2006*, pages 446–460, Beijing, P. R. China, 23-26 October 2006. Springer-Verlag Berlin, Heidelberg, Berlin, Germany.

[Georgiadis *et al.*, 2002] M. C. Georgiadis, M. Schenk, E. N. Pistikopoulos, and R. Gani. The interactions of design, control and operability in reactive distillation systems. *Computers and Chemical Engineering*, 26(4-5):735–746, May 2002.

[Ghosh, 2005] Sumit Ghosh. Large-scale systems design: a revolutionary new approach in software hardware co-design. In *Artificial Intelligence and Simulation. 13th International Conference on AI, Simulation, and Planning in High Autonomy Systems, AIS 2004*, volume 3397 of *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, pages 270 – 274, Jeju Island, South Korea, 4-6 Oct 2005. Springer-Verlag, Berlin, Germany.

[Grassberger and Procaccia, 1983] P. Grassberger and I. Procaccia. Characterization of strange attractors. *Physics Review Letters*, 50(5):346–349, Jan 1983.

[Gruner *et al.*, 2003] S. Gruner, K. D. Mohl, A. Kienle, E. D. Gilles, G. Fernholz, and M. Friedrich. Nonlinear control of a reactive distillation column. *Control Engineering Practice*, 11(8):915–925, Aug 2003.

[Halsey *et al.*, 1986] T. C. Halsey, M. H. Jensen, L. P. Kadanoff, I. Procaccia, and B. I. Schraiman. Fractal measures and their singularities - the characterization of strange sets. *Physical Review A*, 33(2):1141–1151, Feb 1986.

[Han and Clough, 2006] Myungwan Han and David E. Clough. Nonlinear model based control of two-product reactive distillation column. *Korean Journal of Chemical Engineering*, 23(4):540–546, July 2006.

[Harmsen, 2007] G. Jan Harmsen. Reactive distillation: The front-runner of industrial process intensification - a full review of commercial applications, research, scale-up, design and operation. *Chemical Engineering and Processing*, 85(A9):1268–1280, Sept 2007.

[He *et al.*, 2007] Tian He, John A. Stankovic, Michael Marley, Chenyang Lu, Ying Lu, Tarek Abdelzaher, Sang Son, and Gang Tao. Feedback control-based dynamic resource management in distributed real-time systems. *Journal of Systems and Software*, 80(7):997–1004, July 2007.

[Hespanha *et al.*, 2007] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, Jan 2007.

[Hu *et al.*, 2008] Wenshan Hu, Guo-Ping Liu, David Rees, and Yuliang Qiao. Design and implementation of Web-based control laboratory for test rigs in geographically diverse locations. *IEEE Transactions on Industrial Electronics*, 55(6):2343–2354, June 2008.

[Huang *et al.*, 2004] S. G. Huang, C. L. Kuo, S. B. Hung, Y. W. Chen, and C. C. Yu. Temperature control of heterogeneous reactive distillation. *AIChE Journal*, 50(9):2203–2216, Sept 2004.

[Huang *et al.*, 2006] Yaqing Huang, Roch Guérin, and Pranav Gupta. Supporting excess real-time traffic with active dropqueue. *IEEE/ACM Transactions on Networking*, 14(5):965–977, Oct 2006.

[Huang *et al.*, 2008] Kejin Huang, San-Jang Wang, and Wenming Ding. Towards further internal heat integration in design of reactive distillation columns - Part III: Application to a MTBE reactive distillation column. *Chemical Engineering Science*, 63(8):2119–2134, Apr 2008.

[Hung *et al.*, 2006] S. B. Hung, M. J. Lee, Y. T. Tang, Y. W. Chen, I. K. Lai, W. J. Hung, H. P. Huang, and C. C. Yu. Control of different reactive distillation configurations. *AIChE Journal*, 52(4):1423–1440, Apr 2006.

[Hurst, 1951] H. E. Hurst. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–808, 1951.

[Jang and Chen, 1996] M. J. Jang and C. L. Chen. Fuzzy successive modelling and control for time-delay system. *International Journal of System Science*, 27(12):1483–1490, Dec 1996.

[Jensen and Datta, 1995] K. L. Jensen and R. Datta. Ethers from ethanol. 1. Equilibrium thermodynamic analysis of the liquid phase ethyl *tert*-butyl ether reaction. *Industrial and Engineering Chemistry Research*, 34(1):392–399, Jan 1995.

[Ji and Kim, 2007] Kun Ji and Won-Jong Kim. Stochastic optimal control and network co-design for networked control systems. *International Journal of Control Automation and Systems*, 5(5):515–525, Oct 2007.

[Ji and Kim, 2008] Kun Ji and Won-jong Kim. Optimal bandwidth allocation and QoS-adaptive control co-design for networked control systems. *International Journal of Control Automation and Systems*, 6(4):596–606, Aug 2008.

[Jiang and Han, 2004] X. Jiang and Q. L. Han. Network-induced delay-dependent H_∞ controller design for a class of networked control systems. *Asian Journal of Control*, 8(2):97–106, June 2004.

[Jimenez and Costa-Lopez, 2002] L. Jimenez and J. Costa-Lopez. The production of butyl acetate and methanol via reactive and extractive distillation. II. Process modeling, dynamic simulation, and control strategy. *Industrial and Engineering Chemistry Research*, 41(26):6735–6744, Dec 2002.

[Jin *et al.*, 2007] Hong Jin, Danli Wang, Hongan Wang, and Hui Wang. Feedback fuzzy-DVS scheduling of control tasks. *Journal of Supercomputing*, 41(2):147–162, Aug 2007.

[Katzen and Procaccia, 1987] D. Katzen and I. Procaccia. Phase-transitions in the thermodynamic formalism of multifractals. *Physics Review Letters*, 58(12):1169–1172, Mar 1987.

[Kawathekar and Riggs, 2007] Rohit Kawathekar and James B. Riggs. Nonlinear model predictive control of a reactive distillation column. *Control Engineering Practice*, 15(2):231–239, Feb 2007.

[Kaymak and Luyben, 2004] Devrim B. Kaymak and William L. Luyben. Quantitative comparison of reactive distillation with conventional multi-unit reactor/column/recycle systems for different chemical equilibrium constants. *Industrial and Engineering Chemistry Research*, 43(10):2493–2570, May 2004.

[Kaymak and Luyben, 2008] Devrim B. Kaymak and William L. Luyben. Quantitative comparison of dynamic controllability between a reactive distillation column and a conventional multi-unit process. *Computers and Chemical Engineering*, 32(7):1456–1470, July 2008.

[Khaledi and Young, 2005] R. Khaledi and B. R. Young. Modeling and model predictive control of composition and conversion in an ETBE reactive distillation column. *Industrial and Engineering Chemistry Research*, 44(9):3134–3145, Apr 2005.

[Kim and Park, 2003] Hyoung Yuk Kim and Hong Seong Park. Period and priority assignment method for DCS design. *Asian Journal of Control*, 5(3):422–432, Sept 2003.

[Kordon *et al.*, 2003] Arthur K. Kordon, Guido F. Smits, Alex N. Kalos, and Elsa M. Jordaan. Robust soft sensor development using genetic programming. In Riccardo Leardi, editor, *Nature-Inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks*, volume 23 of *Data Handling in Science and Technology*. Elsevier, 2003.

[Kravaris and Kantor, 1990] C. Kravaris and C. J. Kantor. Geometric methods for nonlinear process control: 1. Background; 2. Controller synthesis. *Industrial and Engineering Chemistry Research*, 29(12):2295–2323, Dec 1990.

[Kumar and Kaistha, 2008a] M. V. Pavan Kumar and Nitin Kaistha. Decentralized control of a kinetically controlled ideal reactive distillation column. *Chemical Engineering Science*, 63(1):228–243, Jan 2008.

[Kumar and Kaistha, 2008b] M. V. Pavan Kumar and Nitin Kaistha. Role of multiplicity in reactive distillation control system design. *Journal of Process Control*, 18(7-8):692–706, Aug-Sept 2008.

[Kumar and Kaistha, 2008c] M. V. Pavan Kumar and Nitin Kaistha. Steady-state multiplicity and its implications on the control of an ideal reactive distillation column. *Industrial and Engineering Chemistry Research*, 47(8):2778–2787, Apr 2008.

[Lavagno *et al.*, 2003] Luciano Lavagno, Grant Martin, and Bran Selic. *UML for Real Design of Embedded Real-Time Systems*. Kluwer Academic Publishers, Boston, 2003.

[Lee and Lee, 2002] Kyung Chang Lee and Suk Lee. Performance evaluation of switched Ethernet for real-time industrial communications. *Computer Standards and Interfaces*, 24(5):411–423, Nov 2002.

[Lee and Stanley, 1988] J. Lee and H. E. Stanley. Phase-transition in the multifractal spectrum of diffusion-limited aggregation. *Physics Review Letters*, 61(26):2945–2948, Dec 1988.

[Lee *et al.*, 2007] H. Y. Lee, H. P. Huang, and I. L. Chien. Control of reactive distillation process for production of ethyl acetate. *Journal of Process Control*, 17(4):363–377, Apr 2007.

[Lehoczky *et al.*, 1989] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: exact characterization and average case behavior. In *Proceedings, Real Time Systems Symposium (Cat. No.89CH2803-5)*, pages 166 – 71, Los Alamitos, CA, USA, 1989.

[Leland *et al.*, 1994] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similarity nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–5, Feb 1994.

[Leung and Whitehead, 1982] J. Leung and J. Whitehead. On the complexity of fixed priority scheduling of periodic, real-time tasks. *Performance Evaluation*, 2(4):237–250, Dec 1982.

[Lextrait *et al.*, 2004] S. Lextrait, R. B. Eldridge, and T. F. Edgar. Steady-state rate-based simulation of packed reactive distillation: Spatial discretization. *Industrial and Engineering Chemistry Research*, 43(14):3855–3869, July 2004.

[Lian *et al.*, 2001] Feng-Li Lian, James R. Moyne, and Dawn M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems Magazine*, 21(1):66–83, Feb 2001.

[Lian *et al.*, 2006] Feng-Li Lian, John K. Yook, Dawn M. Tilbury, and James Moyne. Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems. *IEEE Transactions on Industrial Informatics*, 2(1):12–24, Feb 2006.

[Lin *et al.*, 2006] Y. D. Lin, H. P. Huang, and C. C. Yu. Relay feedback tests for highly nonlinear processes: Reactive distillation. *Industrial and Engineering Chemistry Research*, 45(12):4081–4092, June 2006.

[Lin *et al.*, 2008] Yu-Der Lin, Jun-Hong Chen, Jian-Kai Cheng, Hsiao-Ping Huang, and Cheng-Ching Yu. Process alternatives for methyl acetate conversion using reactive distillation. 1. Hydrolysis. *Chemical Engineering Science*, 63(6):1668–1682, Mar 2008.

[Ling and Lemmon, 2003] Q. Ling and M. D. Lemmon. Optimal dropout compensation in networked control systems. In *Proceedings - 42nd IEEE International Conference on Decision and Control*, volume 1, pages 670–675, Maui, HI, USA, 9-12 December 2003. IEEE, Piscataway, NJ, USA.

[Liu and Layland, 1973] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of ACM*, 23(1):46–68, 1973.

[Lluesma *et al.*, 2005] M. Lluesma, P. Balbastre, I. Ripoll, and A. Crespo. Comparison and analysis of two proposals for reducing control delays. In M. Colnaric, W. A. Halang, and M. Wegrzyn, editors, *Real-Time Programming 2004, A Proceedings Volume from the 28th IFAC/IFIP Workshop, Istanbul, Turkey, 6-8 September 2004*, pages 55–60. Elsevier IFAC Publications, 2005.

[Lluesma *et al.*, 2006] Manuel Lluesma, Anton Cervin, Patricia Balbastre, Ismael Ripoll, and Alfons Crespo. Jitter evaluation of real-time control systems. In *Proceedings - 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2006*, pages 257–60, Sydney, Australia, 16-18 Aug 2006. IEEE Computer Society, Los Alamitos, CA, USA.

[Lorente and Palencia, 2006] J. L. Lorente and J. C. Palencia. An EDF hierarchical scheduling model for bandwidth servers. In *Proceedings - 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2006*, pages 261–266, Sydney, Australia, 16-18 Aug 2006.

[Lu *et al.*, 2002] C. Y. Lu, J. A. Stankovic, S. H. Son, and G. Tao. Feedback control real-time scheduling: framework, modeling, and algorithms. *Real-Time Systems*, 23(1-2):85–126, July-Sept 2002.

[Luke and Ray, 1990] R. Luke and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5):903–905, Sept 1990.

[Luke and Ray, 1994] R. Luke and A. Ray. Experimental verification of a delay compensation algorithm for integrated communication and control systems. *International Journal of Control*, 59(6):1357–1372, June 1994.

[Mahmound *et al.*, 2006] M. S. Mahmound, P. Shi, J. Q. Yi, and J. S. Pan. Robust observers for neutral jumping systems with uncertain information. *Information Sciences*, 176(16):2355–2385, Aug 2006.

[Malone and Doherty, 2000] M. F. Malone and M. F. Doherty. Reactive distillation. *Industrial and Engineering Chemistry Research*, 39(11):3953–3957, 2000.

[Mandelbrot, 1982] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1982.

[Martí *et al.*, 2004] P. Martí, J. Yepez, M. Velesco, R. Villa, and J. M. Fuertes. Managing quality-of-control in network-based control systems by controller and message scheduling co-design. *IEEE Transactions on Industrial Electronics*, 51(6):1159–1167, Dec 2004.

[Martí *et al.*, 2005] P. Martí, R. Villá, J. M. Fuertes, and G. Fohler. Networked control systems overview. In Richard Zurawski, editor, *The Industrial Information Technology Handbook*. CRC Press, Dec 2005.

[Mok, 1983] A. K. Mok. Fundamental design problems of distributed systems for the hard real-time environment, 1983. Technical Report: TR-297.

[Nikolakopoulos *et al.*, 2008] George Nikolakopoulos, Athanasia Panousopoulou, and Anthony Tzes. Experimental controller tuning and QoS optimization of a wireless transmission scheme for real-time remote control applications. *Control Engineering Practice*, 16(3):333–346, Mar 2008.

[Noeres *et al.*, 2004] C. Noeres, K. Dadhe, R. Gesthuisen, S. Engell, and A. Gorak. Model-based design, control and optimisation of catalytic distillation processes. *Chemical Engineering and Processing*, 43(3):421–434, Mar 2004.

[Ohlin *et al.*, 2007] Martin Ohlin, Dan Henriksson, and Anton Cervin. *TrueTime 1.5 - Reference Manual*. Department of Automatic Control, Lund University, Jan 2007.

[Olanrewaju and Al-Arfaj, 2005] M. J. Olanrewaju and M. A. Al-Arfaj. Development and application of linear process model in estimation and control of reactive distillation. *Computers and Chemical Engineering*, 30(1):147–157, Nov 2005.

[Olanrewaju and Al-Arfaj, 2006] M. J. Olanrewaju and M. A. Al-Arfaj. Estimator-based control of reactive distillation system: Application of an extended Kalman filtering. *Chemical Engineering Science*, 61(10):3386–3399, May 2006.

[Olanrewaju and Al-Arfaj, 2008] Moshood J. Olanrewaju and Muhammad A. Al-Arfaj. Design and implementation of state observers in reactive distillation. *Chemical Engineering Communications*, 195(3):267–292, Mar 2008.

[Panjwani *et al.*, 2005] P. Panjwani, M. Schenk, M. C. Georgiadis, and E. N. Pistikopoulos. Optimal design and control of a reactive distillation system. *Engineering Optimization*, 37(7):733–753, Oct 2005.

[Pastor-Satorras, 1997] R. Pastor-Satorras. Multifractal properties of power-law time sequences: application to rice piles. *Physics Review E*, 56(5):5284–5294, Nov 1997.

[Peng and Tian, 2006] Chen Peng and Yu-Chu Tian. Robust H_∞ control of networked control systems with parameter uncertainty and state-delay. *European Journal of Control*, 12(5):471–480, Sept–Oct 2006.

[Peng and Tian, 2007] Chen Peng and Yu-Chu Tian. Networked H_∞ control of linear systems with state quantization. *Information Sciences*, 177(24):5763–5774, Dec 2007.

[Peng *et al.*, 2003] J. J. Peng, T. F. Edgar, and R. B. Eldridge. Dynamic rate-based and equilibrium models for a packed reactive distillation column. *Chemical Engineering Science*, 58(12):2671–2680, June 2003.

[Peng *et al.*, 2007] Chen Peng, Yu-Chu Tian, and Dong Yue. Network quality-of-service based guaranteed cost control for networked control systems. *Dynamics of Continuous, Discrete and Impulsive Systems B - Applications and Algorithms*, 14(2):233–247, 2007.

[Peng *et al.*, 2008] Chen Peng, Yu-Chu Tian, and Moses O. Tadé. State feedback controller design of networked control systems with interval time-varying delay and nonlinearity. *International Journal of Robust and Nonlinear Control*, 18(12):1285–1301, Oct 2008.

[Rajkumar *et al.*, 1998] R. Rajkumar, Chen Lee, J.P. Lehoczky, and D.P. Siewiorek. Practical solutions for QoS-based resource allocation problems. In *Proceedings 19th IEEE Real-Time Systems Symposium (Cat. No.98CB36279)*, pages 296 – 306, Los Alamitos, CA, USA, 1998.

[Samard and Balas, 2003] Tariq Samard and Gray Balas. *Software-Enabled Control*. IEEE Press/Wiley-Interscience, Piscataway NJ/Hoboken NJ, 2003.

[Sanz and Alonso, 2001] R. Sanz and M. Alonso. CORBA for control systems. *Annual Reviews in Control*, 25:169–181, 2001.

[Schenato, 2006] L. Schenato. Optimal estimation in networked control systems subject to random delay and packet loss. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, USA, 13-15 Dec 2006. IEEE.

[Schenato, 2008] Luca Schenato. Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Transactions on Automatic Control*, 53(5):1311 – 1317, June 2008.

[Schneider and Gorak, 2001] R. Schneider and A. Gorak. Model optimization for the dynamic simulation of reactive absorption processes. *Chemical Engineering and Technology*, 24(10):979–989, Oct 2001.

[Schneider *et al.*, 2003] R. Schneider, F. Sander, and A. Gorak. Dynamic simulation of industrial reactive absorption processes. *Chemical Engineering and Processing*, 42(12):955–964, Dec 2003.

[Seem, 1998] J. E. Seem. A new pattern recognition adaptive controller with application to HVAC systems. *Automatica*, 34(8):969–982, Aug 1998.

[Seto *et al.*, 2001] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin. Trade-off analysis of real-time control performance and schedulability. *Real-Time systems*, 21(3):199–217, Nov 2001.

[Sha *et al.*, 2004] L. Sha, T. Abdelzaher, K. E. Arzen KE, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok. Real time scheduling theory: a historical perspective. *Real-Time Systems*, 28(2-3):101–155, Nov-Dec 2004.

[Shaw, 2001] Alan C. Shaw. *Real-Time Systems and Software*. John Wiley & Sons, Inc., New York, 2001.

[Singh *et al.*, 2005] B. P. Singh, R. Singh, M. V. P. Kumar, and N. Kaistha. Steady-state analyses for reactive distillation control: an MTBE case study. *Journal of Loss Prevention in the Process Industries*, 18(4-6):283–292, Jul-Nov 2005.

[Skogestad *et al.*, 1990] S. Skogestad, P. Lundström, and E. W. Jacobsen. Selecting the best distillation control configuration. *AIChE Journal*, 36(5):753–764, May 1990.

[Skogestad, 2003] S. Skogestad. Simple analytical rules for model reduction and PID controller tuning. *Journal of Process Control*, 13:291–309, 2003.

[Sneesby *et al.*, 1997a] M. G. Sneesby, M. O. Tadé, R. Datta, and T. N. Smith. ETBE synthesis via reactive distillation. 1. Steady state simulation and design aspects. *Industrial and Engineering Chemistry Research*, 36(5):1855–1869, May 1997.

[Sneesby *et al.*, 1997b] M. G. Sneesby, M. O. Tadé, R. Datta, and T. N. Smith. ETBE synthesis via reactive distillation. 2. Dynamic simulation and control aspects. *Industrial and Engineering Chemistry Research*, 36(5):1870–1881, May 1997.

[Sneesby *et al.*, 1998] M. G. Sneesby, Moses O. Tadé, and T. N. Smith. Multiplicity and pseudo-multiplicity in MTBE and ETBE reactive distillation. *Chemical Engineering Research and Design*, 76(A4):525–531, May 1998.

[Sneesby *et al.*, 1999] M. G. Sneesby, M. O. Tadé, and T. N. Smith. Two-point control of a reactive distillation column for composition and conversion. *Journal of Process Control*, 9(1):19–31, Feb 1999.

[Sneesby *et al.*, 2000] M. G. Sneesby, Moses O. Tadé, and T. N.. Smith. A multi-objective control scheme for an ETBE reactive distillation column. *Chemical Engineering Research and Design*, 78(A2):283–292, Mar 2000.

[Soglo and Yang, 2006] A. B. Soglo and X. Yang. Compensation for network-induced delays and packet dropping in control systems. *European Journal of Control*, 12(3):296–306, 2006.

[Song *et al.*, 2008] Insop Song, Sehjeong Kim, and Fakhreddine Karray. A real-time scheduler design for a class of embedded systems. *IEEE-ASME Transactions on Mechatronics*, 13(1):36–45, Feb 2008.

[Sprunt *et al.*, 1989] B. Sprunt, L. Sha, and J. P. Lehoczky. Aperiodic task scheduling for hard real-time systems. *Real-Time Systems*, 1(1):27–60, June 1989.

[Stankovic and Ramamrithiham, 1988] J. A. Stankovic and K. Ramamrithiham. *Hard Real-Time Systems*. IEEE Press, 1988.

[Stankovic *et al.*, 1995] John A. Stankovic, Marco Spuri, Marco Di Natale, and Giorgio C. Buttazzo. Implications of classical scheduling results for real-time systems. *Computer*, 28(6):16–25, June 1995.

[Sun and El-Farra, 2008] Yulei Sun and Nael H. El-Farra. Quasi-decentralized model-based networked control of process systems. *Computers and Chemical Engineering*, 32(9):2016–2029, Sept 2008.

[Tabuada, 2007] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, Sept 2007.

[Tang *et al.*, 2005] Y. T. Tang, H. P. Huang, and I. L. Chien. Plant-wide control of a complete ethyl acetate reactive distillation process. *Journal of Chemical Engineering of Japan*, 38(2):130–146, Feb 2005.

[Taylor and Krishna, 2000] R. Taylor and R. Krishna. Modelling reactive distillation. *Chemical Engineering Science*, 55(2):5183–5229, 2000.

[Tian and Gao, 1998a] Yu-Chu Tian and Fuong Gao. Adaptive control of chaotic continuous-time systems with delay. *Physica D*, 117(1-4):1–12, June 1998.

[Tian and Gao, 1998b] Yu-Chu Tian and Furong Gao. Compensation of dominant and variable delay in process systems. *Industrial and Engineering Chemistry Research*, 37(3):982–986, Mar 1998.

[Tian and Gao, 1998c] Yu-Chu Tian and Furong Gao. Double-controller scheme for control of processes with dominant delay. *IEE Proceedings: Control Theory and Applications*, 145(5):479–484, Sept 1998.

[Tian and Gao, 1999a] Yu-Chu Tian and Furong Gao. Control of integrator processes with dominant time delay. *Industrial and Engineering Chemistry Research*, 38(8):2979–2983, Aug 1999.

[Tian and Gao, 1999b] Yu-Chu Tian and Furong Gao. Injection velocity control of thermoplastic injection molding via a double controller scheme. *Industrial and Engineering Chemistry Research*, 38(9):3396–3406, Sept 1999.

[Tian and Levy, 2007] Yu-Chu Tian and David Levy. Configuring the worst-case communication delay in real-time networked control systems. In H. R. Arabnia and L T. Yang, editors, *Proceedings of the 2007 International Conference on Embedded Systems and Applications ESA'07*, pages 114–120, Las Vegas Nevada, USA, 25-28 June 2007. CSREA Press.

[Tian and Levy, 2008a] Yu-Chu Tian and David Levy. Compensation for control packet dropout in networked control systems. *Information Sciences*, 178(5):1263–1278, Mar 2008.

[Tian and Levy, 2008b] Yu-Chu Tian and David Levy. Dealing with network complexity in real-time networked control. *International Journal of Computer Mathematics*, 85(8):1235–1253, Aug 2008.

[Tian and Sun, 2005] Yu-Chu Tian and You-Xian Sun. Editorial: advanced control and real-time systems. *Developments in Chemical Engineering and Mineral Processing*, 13(3/4):209–210, 2005.

[Tian and Tadé, 2000] Yu-Chu Tian and Moses O. Tadé. Inference of conversion and purity for ETBE reactive distillation. *Brazilian Journal of Chemical Engineering*, 17(4-7):617–625, Dec 2000.

[Tian and Tadé, 2002] Yu-Chu Tian and Moses O Tadé. Processing systems engineering: successful application in an industrial process. *Developments in Chemical Engineering and Mineral Processing*, 10(1/2):181–196, 2002.

[Tian *et al.*, 2003] Yu-Chu Tian, Futao Zhao, B. H. Bisowarno, and Moses O. Tadé. Pattern-based predictive control for ETBE reactive distillation. *Journal of Process Control*, 13(1):57–67, Feb 2003.

[Tian *et al.*, 2006a] Yu-Chu Tian, Qing-Long Han, Colin Fidge, Moses O. Tadé, , and Tianlong Gu. Communication architecture design for real-time networked control systems. In *Proceedings of the 4th International Conference on Communications, Circuits and Systems ICCCAS'06*, pages 1840–1845, Guilin, P. R. China, 25-28 June 2006.

[Tian *et al.*, 2006b] Yu-Chu Tian, Qing-Long Han, David Levy, and M. O. Tadé. Reducing control latency and jitter in real-time control. *Asian Journal of Control*, 8(1):72–75, Mar 2006.

[Tian *et al.*, 2006c] Yu-Chu Tian, David Levy, Moses O. Tadé, Tianlong Gu, and Colin Fidge. Queuing packets in communication networks networked control systems. In *Proceedings of the 6th World Congress on Intelligent Control and Automation WCICA'06*, pages 205–209, Dalian, P. R. China, 21-23 June 2006.

[Tian *et al.*, 2007] Yu-Chu Tian, Zu-Guo Yu, and Colin Fidge. Multifractal nature of network induced time delay in networked control systems. *Physics Letters A*, 361(1-2):103–107, Jan 2007.

[Tipsuwan and Chow, 2003] Yodyium Tipsuwan and Mo-Yuen Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11(10):1099–1111, Oct 2003.

[Tipsuwan and Chow, 2004a] Y. Tipsuwan and M. Y. Chow. Gain scheduling middleware: a methodology to enable existing controllers for networked control and teleoperation I: network control. *IEEE Transactions on Industrial Electronics*, 51(6):1218–1227, Dec 2004.

[Tipsuwan and Chow, 2004b] Yodyium Tipsuwan and Mo-Yuen Chow. On the gain scheduling for networked PI control over IP network. *IEEE Transaction on Mechatronics*, 9(3):491–498, Sept 2004.

[UCB/LBNL/VINT Groups, 2008] UCB/LBNL/VINT Groups. NS Network Simulator. http://www.isi.edu/nsnam/ns/, 2008. Accessed on 8 September 2008.

[Ushio *et al.*, 2007] Toshimitsu Ushio, Haruo Kohtaki, Masakazu Adachi, and Fumiko Harada. Adaptive fair sharing control real-time systems using nonlinear elastic task models. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E90A(6):1154–1161, June 2007.

[Varma *et al.*, 2003] A. Varma, B. Ganesh, M. Sen, S. R. Choudhury, L. Srinivasan, and J. Bruce. A control-theoretic approach to dynamic voltage scheduling. In *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, CASES 2003 Proceedings*, pages 255–266, San Jose, California, USA, 30 Oct - 1 Nov 2003. ACM.

[Voelker *et al.*, 2007] Marten Voelker, Christian Sonntag, and Sebastian Engell. Control of integrated processes: a case study on reactive distillation in a medium-scale pilot plant. *Control Engineering Practice*, 15(7):863–881, July 2007.

[Walsh and Ye, 2001] Gregory C. Walsh and Hong Ye. Scheduling of networked control systems. *IEEE Control Systems Magazine*, 21(1):57–65, Feb 2001.

[Walsh *et al.*, 2002] G. C. Walsh, Y. Hong, and G. B. Linda. Stability analysis of networked control systems. *IEEE Transactions on Control System Technology*, 10(3):438–446, May 2002.

[Wang and Wong, 2006] S. J. Wang and D. S. H. Wong. Control of reactive distillation production of high-purity isopropanol. *Journal of Process Control*, 16(4):385–394, Apr 2006.

[Wang *et al.*, 2003a] S. J. Wang, D. S. H. Wong, and E. K. Lee. Control of a reactive distillation column in the kinetic regime for the synthesis of n-butyl acetate. *Industrial and Engineering Chemistry Research*, 42(21):5182–5194, Oct 2003.

[Wang *et al.*, 2003b] S. J. Wang, D. S. H. Wong, and E. K. Lee. Effect of interaction multiplicity on control system design for a mtbe reactive distillation column. *Journal of Process Control*, 13(6):503–515, Sept 2003.

[Wikipedia, 2008] Wikipedia. Kyoto Protocol. http://en.wikipedia.org/wiki/Kyoto_Protocol, 2008. Accessed on 18 September 2008.

[Wong *et al.*, 2005] Y. C. Wong, T. P. Wang, and Y. B. Lin. Effects of route optimization on out-of-order packet delivery in mobile ip networks. *Information Sciences*, 169(3-4):263–278, Feb 2005.

[Xia and Sun, 2005] F. Xia and Y. X. Sun. Neural network based feedback scheduling of multitasking control systems. In *Knowledge-Based Intelligent Information and Engineering Systems - 9th International Conference, KES 2005, Proceedings Part II*, volume 3682 of *Lecture Notes in Artificial Intelligence*, pages 193–199, Melbourne, Australia, 14-16 Sept 2005. Springer-Verlag, Berlin, Germany.

[Xia *et al.*, 2005] F. Xia, X. F. Shen, L. P. Liu, Z. Wang, and Y. X. Sun. Fuzzy logic based feedback scheduler for embedded control systems. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005. Proceedings Part II*, volume 3645 of *Lecture Notes in Computer Science*, pages 453–462, Hefei, China, 23-26 Aug 2005. Springer.

[Xia *et al.*, 2008] Feng Xia, Yu-Chu Tian, Youxian Sun, and Jinxiang Dong. Neural feedback scheduling of real-time control tasks. *International Journal of Innovative Computing, Information and Control*, 4(11):2965–2975, Nov 2008.

[Xia *et al.*, 2009a] Feng Xia, Youxian Sun, and Yu-Chu Tian. Feedback scheduling of priority-driven control networks. *Computer Standards and Interfaces*, 31(3):539–547, Mar 2009.

[Xia *et al.*, 2009b] Feng Xia, Yu-Chu Tian, Moses O. Tadé, and Jinxiang Dong. Fuzzy feedback scheduling of resource-constrained embedded control systems. *International Journal of Innovative Computing, Information and Control*, 5(2):311–321, Feb 2009.

[Yao *et al.*, 2001] Hongmei Yao, Yu-Chu Tian, Moses O. Tadé, and H. M. Ang. Variations and modelling of oxygen demand in amino acid production. *Chemical Engineering and Processing*, 40(4):401–409, July 2001.

[Yao *et al.*, 2008] Hongmei Yao, Yu-Chu Tian, and Moses O. Tadé. Using wavelets for solving SMB separation process models. *Industrial and Engineering Chemistry Research*, 47(15):5585–5593, Aug 2008.

[Yu *et al.*, 2001a] Z. G. Yu, V. V. Anh, and K. S. Lau. Measure representation and multifractal analysis of complete genomes. *Physics Review E*, 64(3):Art. No. 031903, Sept 2001.

[Yu *et al.*, 2001b] Z. G. Yu, V. V. Anh, and K. S. Lau. Multifractal characterisation of length sequences of coding and noncoding segments in a complete genome. *Physica A*, 301(1-4):351–361, Dec 2001.

[Yu *et al.*, 2003] Z. G. Yu, V. V. Anh, and K. S. Lau. Multifractal and correlation analyses of protein sequences from complete genomes. *Physics Review E*, 68(2):Art. No. 021913, Aug 2003.

[Yu *et al.*, 2004] Z. G. Yu, V. V. Anh, and K. S. Lau. Chaos game representation of protein sequences based on the detailed hp model and their multifractal and correlation analyses. *Journal of Theoretical Biology*, 226(3):341–348, Feb 2004.

[Yu *et al.*, 2006] Z. G. Yu, V. V. Anh, K. S. Lau, and L. Q. Zhou. Clustering of protein structures using hydrophobic free energy and solvent accessibility of proteins. *Physics Review E*, 73(3):Art. No. 031920, Mar 2006.

[Yue and Han, 2006] D. Yue and Q. L. Han. Network-based robust H_∞ filtering for uncertain linear systems. *IEEE Transactions on Signal Processing*, 54(11):4293–4301, Nov 2006.

[Yue *et al.*, 2004] D. Yue, Q. L. Han, and C. Peng. State feedback controller design of networked control systems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 51(11):640–644, Nov 2004.

[Yue *et al.*, 2005] D. Yue, Q. L. Han, and J. Lam. Network-based robust H_∞ control of systems with uncertainty. *Automatica*, 41(6):999–1007, June 2005.

[Zeng *et al.*, 2006] K. L. Zeng, C. L. Kuo, and I. L. Chien. Design and control of butyl acrylate reactive distillation column system. *Chemical Engineering Science*, 61(13):4417–4431, July 2006.

[Zhang and Burns, 2007] Fengxiang Zhang and Alan Burns. Analysis of hierarchical EDF pre-emptive scheduling. In *Proceedings - 28th IEEE International Real-Time Systems Symposium, RTSS 2007*, pages 423–434, Tucson, AZ, USA, 3-6 Dec 2007. IEEE.

[Zhang and Hristu-Varsakelis, 2006] Lei Zhang and Dimitrios Hristu-Varsakelis. Communication and control co-design for networked control systems. *Automatica*, 42(6):953–958, June 2006.

[Zhang *et al.*, 2001] W. Zhang, M. S. Branicky, , and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, Feb 2001.

[Zhang *et al.*, 2004] S. J. Zhang, A. Burns, J. Chen, and E. S. Lee. Hard real-time communication with the timed token protocol: Current state and challenging problems. *Real-Time Systems*, 27(3):271–295, Sept 2004.

[Zhanga *et al.*, 2007] Huan Zhanga, Zonghua Liua, Ming Tanga, and P. M. Huib. An adaptive routing strategy for packet delivery in complex networks. *Physics Letters A*, 364(3-4):177–182, Apr 2007.

[Zhao *et al.*, 1987] W. Zhao, K. Ramamrithiham, and J. A. Stankovic. Preemptive scheduling under time and resource constraints. *IEEE Transactions on Computers*, C-36(8):949–960, Aug 1987.

[Zhao *et al.*, 2000] F. Zhao, J. Ou, and W. Du. Pattern-based fuzzy predictive control for a chemical process with dead time. *Engineering Applications of Artificial Intelligence*, 13(1):37–45, Feb 2000.

[Zhao *et al.*, 2002] Futao Zhao, Yu-Chu Tian, Moses O. Tadé, and Hui Li. A time-delay compensation strategy for processes with uncertainties. *Computers and Chemical Engineering*, 26:1437–1447, Oct 2002.

[Zhao *et al.*, 2008] Y. B. Zhao, G. P. Liu, and D. Rees. Integrated predictive control and scheduling co-design for networked control system. *IET Control Theory and Applications*, 2(1):7–15, Jan 2008.

[Zhou *et al.*, 2005] L. Q. Zhou, Z. G. Yu, J. Q. Deng, V. V. Anh, and S. C. Long. A fractal method to distinguish coding and non-coding sequences in a complete genome based on a number sequence representation. *Journal of Theoretical Biology*, 232(4):559–567, Feb 2005.

[Zhu and Mueller, 2005] Y. F. Zhu and F. Mueller. Feedback EDF scheduling of real-time tasks exploiting dynamic voltage scaling. *Real-Time Systems*, 31(1-3):33–63, Dec 2005.

[Zhu and Mueller, 2008] Yifan Zhu and Frank Mueller. Exploiting synchronous and asynchronous DVS for feedback EDF scheduling on an embedded platform. *ACM Transactions on Embedded Computing Systems*, 7(1):Art. No. 3, Dec 2008.

Appendix: My Publications Not Covered in This Thesis

During my PhD study, I have also published other four first-authored refereed papers. Moreover, as at 08 September 2008, I have published forty-seven other refereed papers in collaboration with my colleagues, collaborators, and students. These publications reflect my creative and productive research in a wide range of areas. They are, however, not covered in this Thesis.

Since several papers of ours are currently under revision or review, it is anticipated that this long list of refereed publications will become even longer when I submit this Thesis.

The asterisk (*) indicates that the corresponding paper is not cited in this thesis.

Four First-Authored Papers Not Covered in This Thesis

[*] **Yu-Chu Tian**, David Levy, and Tianlong Gu. Implementing a process model for real-time applications. *Complex'04 (The 7th Asia-Pacific Conference on Complex Systems)*, pages 287–296, Cairns, Australia, 6–10 Dec 2004.

[*] **Yu-Chu Tian**, Moses O. Tadé, and Hongmei Yao. Optimization in modelling, operation and control of a fermentation process. In Luus R ed., *Recent Developments in Optimization and Optimal Control in Chemical Engineering*, pages 155–177, 2002. Research Signpost.

[*] **Yu-Chu Tian**, Moses O. Tadé, and David Levy. Constrained control of chaos. *Physics Letters A*, 296(2-3):87–90, Apr 2002.

[Tian and Tadé, 2002] **Yu-Chu Tian** and Moses O. Tadé. Process systems engineering: successful application in an industrial process. *Developments in Chemical Engineering and Mineral Processing*, 10(1/2):181–196, 2002.

Other Forty-Seven Papers Not Covered in This Thesis

[*] Chen Peng, Dong Yue, and **Yu-Chu Tian**. New approach on robust delay-dependent H-infinity control for uncertain T-S fuzzy systems with interval time-varying delay. *IEEE Transactions on Fuzzy Systems*, in press, 2009.

[*] Chen Peng, Dong Yue, and **Yu-Chu Tian**. Delay distribution based robust H-infinity control of NCS with uncertainties. *Asian Journal of Control*, in press, 2009.

[*] Hongmei Yao, Tonghua Zhang, Moses O. Tadé,and **Yu-Chu Tian**. Wavelet based approaches and high resolution methods for complex process models. In *Computational Chemistry*, Nova Publishers, in press, 2009.

[Xia *et al.*, 2008a] Feng Xia, Youxian Sun, and **Yu-Chu Tian**. Feedback scheduling of priority-driven control networks. *Computer Standards and Interfaces*, 31(3):539–547, Mar 2009.

[Xia *et al.*, 2008c] Feng Xia, **Yu-Chu Tian**, Moses O. Tadé, and Jinxiang Dong. Fuzzy feedback scheduling of resource-constrained embedded control systems. *International Journal of Innovative Computing, Information and Control*, 5(2):311-321, Feb 2009.

[*] Hui Miao and **Yu-Chu Tian**. Robot path planning in dynamic environments using a simulated annealing based approach. In *Proceedings of the 10th Int Conf on Control, Automation, Robotics and Vision - ICARCV'2008*, pages 1253–1258, Hanoi, Vietnam, 17-20 Dec 2008.

[Xia *et al.*, 2008b] Feng Xia, **Yu-Chu Tian**, Youxian Sun, and Jinxiang Dong. Neural feedback scheduling of real-time control tasks. *International Journal of Innovative Computing, Information and Control*, 4(11)::2965–2975, Nov 2008.

[*] Chen Peng, **Yu-Chu Tian**, and Engang Tian. Improved delay-dependent robust stabilization conditions of uncertain T-S fuzzy systems with time-varying delay. *Fuzzy Sets and Systems*, 159(10): 2713–2729, Oct 2008.

[*] Tonghua Zhang, Moses O. Tadé, **Yu-Chu Tian**, and Hong Zang. High-resolution method for numerically solving PDEs in process engineering. *Computers and Chemical Engineering*, 32(10):2403–2408, Oct 2008.

[Peng *et al.*, 2008] Chen Peng, **Yu-Chu Tian**, and Moses O. Tadé. State feedback controller design of networked control systems with interval time-varying delay and nonlinearity.

International Journal of Robust and Nonlinear Control, 18(12):1285–1301, Oct 2008.

[*] Chen Peng and **Yu-Chu Tian**. Improved delay-dependent robust stability criteria for uncertain systems with interval time-varying delay. *IET Control Theory and Applications*, 2(9):752–761, Sept 2008.

[*] Feng Xia, **Yu-Chu Tian**, Youxian Sun, and Jinxiang Dong. Control-theoretic dynamic voltage scaling for embedded controllers. *IET Computers and Digital Techniques*, 2:377–385, Sept 2008.

[*] Tonghua Zhang, **Yu-Chu Tian**, and Moses O Tadé. Wavelet-based collocation method for stiff systems in process engineering. *Journal of Mathematical Chemistry*, 44(2):501–513, Aug 2008.

[Yao *et al.*, 2008] Hongmei Yao, **Yu-Chu Tian**, and Moses O. Tadé. Using wavelets for solving SMB separation process models. *Industrial and Engineering Chemistry Research*, 85(8):1235–1253, Aug 2008.

[*] Chen Peng, **Yu-Chu Tian**, and Dong Yue. Delay-dependent robust stability analysis for neutral systems with discrete and distributed delays. In *Proceedings of the 7th World Congress on Intelligent Control and Automation - WCICA'08*, pages 916-921, Chongqing, China, 25-27 June 2008.

[*] Tonghua Zhang, Moses O. Tadé, **Yu-Chu Tian**; Y. Zhang, and J. Utomo. Wavelet approach incorporated with optimization for solving stiff systems. *Journal of Mathematical Chemistry*, 43(4):1533–1548, May 2008.

[*] Chen Peng and **Yu-Chu Tian**. Delay-dependent robust stability criteria for uncertain systems with interval time-varying delay. *Journal of Computational Applied Mathematics*, 214(2):480–494, Apr 2008.

[*] Guosong Tian, Colin Fidge, and **Yu-Chu Tian**. High-precision relative clock synchronization using time stamp counters. *Proc the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2008)*, pages 69–78, Belfast, Northern Ireland, 31 March - 4 April 2008.

[*] Hong Zang, Tonghua Zhang, **Yu-Chu Tian**, Moses O. Tadé. Limit cycles for the Kukles system. *Journal of Dynamical and Control Systems*, 14(2):283–298, Apr 2008.

[*] Feng Xia, W. Zhao, Y. Sun, and **Yu-Chu Tian**. Fuzzy logic control based QoS management in wireless sensor/actuator networks. *Sensors*, 7(12), 3179-3191, Dec 2007.

[*] Tonghua Zhang, **Yu-Chu Tian**, and Moses O. Tadé. On the zeros of the Abelian integrals for a class of perturbed Liénard systems. *International Journal of Bifurcation and Chaos*, 17(9):3281–3287, Sept 2007.

[*] Feng Xia, **Yu-Chu Tian**, Y. Li, and Youxian Sun. Wireless sensor/actuator network design for mobile control applications. *Sensors*, 7(10):2157–2173, Oct 2007.

[Peng and Tian, 2007] Chen Peng and **Yu-Chu Tian**. Networked H_∞ control of linear systems with state quantization. *Information Sciences*, 177(24):5763–5774, Dec 2007.

[*] Li Gui, **Yu-Chu Tian**, and Colin Fidge. Performance evaluation of IEEE 802.11b wireless networks for real-time networked control systems. In H. R. Arabnia and L T. Yang, eds., *Proceedings of the 2007 International Conference on Embedded Systems and Applications ESA'07*, pages 121–126, Las Vegas Nevada, USA, 25-28 June 2007. CSREA Press.

[*] Tonghua Zhang, **Yu-Chu Tian**, Moses O. tadé, and J. Utomo. Comments on "The Computation of Wavelet-Galerkin Approximation on a Bounded Interval". *International Journal for Numerical Methods in Engineering*, 72(2):244–251, Oct 2007.

[Peng *et al.*, 2007] Chen Peng, **Yu-Chu Tian**, and Dong Yue. Network quality-of-service based guaranteed cost control for networked control systems. *Dynamics of Continuous, Discrete and Impulsive Systems B - Applications and Algorithms*, 14(2):233–247, 2007.

[*] Tonghua Zhang, Moses O. Tadé, and **Yu-Chu Tian**. Linear estimate of the number of limit cycles for a class of non-linear systems. *Chaos, Soliton and Fractals*, 31(4):804–810, Feb 2007.

[Peng and Tian, 2006] Chen Peng and **Yu-Chu Tian**. Robust H_∞ control of networked control systems with parameter uncertainty and state-delay. *European Journal of Control*, 12(5):471–480, Sept–Oct 2006.

[*] Tonghua Zhang, Moses O. Tadé, **Yu-Chu Tian**, and M. Han. On the zeros of the Abelian integrals for a class of Linard systems. *Physics Letters A*, 358(4):262–374, Oct 2006.

[*] Chen Peng and **Yu-Chu Tian**. Robust controller design of networked control systems with nonlinear uncertainties. *Proc 3rd Int Conf on Impulse Dynamic Systems and applications*, pages 1525–1529, 2006

[Fidge and Tian, 2006] Colin Fidge and **Yu-Chu Tian**. Functional analysis of a real-time protocol for networked control systems. In *Automated Technology for Verification and Analysis. 4th International Symposium, ATVA 2006*, volume 4218 of *Lecture Notes in Computer Science: ATVA 2006*, pages 446–460, Beijing, P. R. China, 23-26 October 2006. Springer-Verlag Berlin, Heidelberg, Berlin, Germany.

[*] Alex Campbell, Binh Pham, and **Yu-Chu Tian**. A framework for spatio-temporal data analysis and hypothesis exploration. *iMESs Summit on Environmental Modelling and Software*, Vermont, USA Software, 9-12 July 2006

[*] Xiefu Jiang, Qing-Long Han, and **Yu-Chu Tian**. Controller design for networked fuzzy systems. *Proc The 6th World Congress on Intelligent Control and Automation - WCICA'06*, pages 210–214, Dalian, China, 21-23 June 2006.

[*] Qing-Long Han, **Yu-Chu Tian** and D. Han. Robust absolute stability criteria for a class of uncertain Lur'e systems of neutral type. *Proc The 6th World Congress on Intelligent Control and Automation - WCICA'6*, pages 2339–2343, Dalian, China, 21-23 June 2006.

[*] Alex Campbell, Binh Pham, **Yu-Chu Tian**. Delay-embedding approach to multi-agent system analysis and calibration. In A. Zerger and R. M. Argent, editors, *MODSIM 2005 International Congress on Modelling and Simulation*. pages 113–119, Modelling and Simulation Society of Australia and New Zealand, Melbourne, Australia, 12-15 December 2005. http://www.mssanz.org.au/modsim05/papers/campbell.pdf.

[*] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Application of side reactors on ETBE reactive distillation . *Chemical Engineering Journal*, 99(1):35–43, May 2004.

[*] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Interaction of Separation and Reactive Stages on ETBE Reactive Distillation Columns. *AIChE Journal*, 50(3):646–653, Mar 2004.

[Bisowarno *et al.*, 2004] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Adaptive control of an ETBE reactive distillation column. *Journal of Chemical Engineering Japan*, 37(2):210–216, Feb 2004.

[*] A. B. Campbell, B. Pham, and **Y.-C. Tian**. Mining ecological data using cellular automata. *ACRI 2004 (The 6th Int. Conf. on Cellular Automata for Research and Industry)*, Amsterdam, The Netherlands, 25-27 October 2004. In Peter M. A. Sloot, Bastien Chopard,

and Alfons G. Hoekstra, editors, *Lecture Notes in Computer Science*, Vol. 3305, Springer-Verlag Heidelberg, ISSN 0302-9743, 2004.

[*] J. Hu, M. Shi, and **Yu-Chu Tian**. A new switching function for variable structure control. *Proc. WCICA04 (World Congress on Intelligent Control and Automation)*, pages 1163–1166, Hangzhou, China, 2004.

[Bisowarno *et al.*, 2003] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Model gain scheduling control of an ethyl tert-butyl ether reactive distillation column. *Industrial and Engineering Chemistry Research*, 42(15):3584–3591, July 2003.

[*] L. Li, **Yu-Chu Tian**, Moses O. Tadé, *et al*. Apoptosis of tumour cells by temperature and anti-tumour drug: microscopic and macroscopic investigations. *Journal of Thermal Biology*, 28(4):321–329, May 2003.

[*] T. Gu, R. Dong, and **Yu-Chu Tian**. Continuous Petri nets augmented with maximal and minimal firing speeds. *Proc IEEE International Conference on Systems, Man and Cybernetics*, pages 1493–1498, Washington DC, Oct 5-8 2003.

[Zhao *et al.*, 2002] Futao Zhao, **Yu-Chu Tian**, Moses O. Tadé, and Hui Li. A time-delay compensation strategy for processes with uncertainties. *Computers and Chemical Engineering*, 26:1437–1447, Oct 2002.

[*] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Adaptive control of an ETBE reactive distillation column. *Proc of the 9th APCCHE Congress and CHEMECA 2002*, Paper #545, New Zealand, 2002.

[*] B. H. Bisowarno, **Yu-Chu Tian**, and Moses O. Tadé. Effects of separation stages on ETBE reactive distillation performance. *Proc of the 9th APCCHE Congress CHEMECA 2002*, Paper #778, New Zealand, 2002.

[*] N. J. Lockwood, Moses O. Tadé, and **Yu-Chu Tian**. Multicomponent optimisation of reactive distillation systems an ETBE case study. *Proc of the 9th APCCHE Congress CHEMECA 2002*, Paper #780, New Zealand, 2002.