WIRELESS NETWORKED CONTROL SYSTEM COORDINATION AGENT

by

Hazem Mohamed Saad Ibrahim

B.Sc. in Electrical Engineering
Zagazig University, Egypt, 2006

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of

Master of Science in Engineering

in the Graduate Academic Unit of Electrical and Computer Engineering

Supervisor: James H. Taylor, Ph.D., Electrical and Computer Engineering

Examining Board: Brent Petersen, Ph.D., Electrical and Computer Engineering
Chris Diduch, Ph.D., Electrical and Computer Engineering
John DeDourek, Ph.D., Computer Science

This thesis is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

March, 2010

To my lovely parents, my brother, my two sisters, and Dodo my nephew who I have not seen yet.

Abstract

Wireless networked control systems have begun to gain acceptance during the last decade, largely due to the increased flexibility and lower costs they promise to provide. The pace of application has been held back, however, by the reluctance of industry to make the accommodations necessary to allow wireless paths to be incorporated in process control loops, thus limiting the potential applications and benefits of wireless systems. The problem is that there are conflicts between maintaining the performance (and particularly stability) of a control loop, which can be degraded by slow data rates and delays in a wireless path, and the usual objectives in managing a wireless sensor network, namely freedom to configure the network to maximize efficiency and to adjust data rates in the network to conserve energy consumption in the network nodes, which are very often battery powered.

This conflict has been addressed by developing a wireless networked control system coordination agent, as part of an intelligent supervisory control system, which will grant the wireless sensor network gateway as much latitude in meeting its objectives as possible while maintaining the performance of control loops that incorporate wireless paths.

Acknowledgements

I would like to thank especially and foremost my research supervisor Prof. James H. Taylor for his generous support, patience, and guidance throughout my research work. Without his great help and encouragement this thesis would not appear. I owe him a huge amount of appreciation.

This project is supported by Atlantic Canada Opportunities Agency (ACOA) under the Atlantic Innovation Fund (AIF) program. I gratefully acknowledge that support and the collaboration of the Cape Breton University (CBU).

I would like to thank PAWS staff members, especially the program manager Jeff Slipp, James Nicholson and Martin J. Murillo for their co-operation, help, and support to develop this thesis.

I would also like to express my sincere appreciation to the faculty and staff of the Department of Electrical and Computer Engineering at University of New Brunswick for their patience and continuous encouragement.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

Symbols

D	Diameter of the reactor (tank)
A_B	Area of JCSTR base
A_H	Area of heat transfer
C_p	Heat capacity
ρ	Fluid density
U	Heat transfer coefficient
V_j	Heating fluid volume
F_{in}	Mixture inflow into the reactor
T_s	Settling time
T_{in}	Temperature of the mixture feed
F_{out}	Mixture outflow from reactor
T_{out}	Mixture outflow temperature
F_{jout}	Heating water outflow
T_{jout}	Temperature of heating water outflow
F_{jin}	Heating water inflow
T_{jin}	Temperature of the heating fluid feed
H	Mixture height in the tank
T	Temperature inside the tank
x_1	Level of mixture in the reactor

x_2	Temperature in the reactor
x_3	Temperature in the jacket
u_1	Tank mixture outflow
u_2	Jacket heating water inflow
$\bar{y}$	Desired output
Q	Heat transfer rate
$\tau_{sL(i)}$	Sampling time i for level control loop
$\tau_{sT(i)}$	Sampling time i for temperature control loop
$\tau_{d_{max}}$	Maximum time delay
τ_{dL}	Time delay for level control loop
τ_{dT}	Time delay for temperature control loop
τ_{dL-max}	Maximum time delay for level control loop
τ_{dT-max}	Time delay for temperature control loop
τ_{s-LC}	Sampling time for level control loop
τ_{s-TC}	Sampling time for temperature control loop
C_{S_1}	Configuration for sensor path (loop 1)
C_{A_1}	Configuration for actuator path (loop 1)
C_{S_2}	Configuration for sensor path (loop 2)
C_{A_2}	Configuration for actuator path (loop 2)
ϵ_{OS}	Specified tolerance for % OS

Abbreviations

A	Actuator
CBU	Cape Breton University
CNA	College of North Atlantic
Ctrl	Control
DCS	Distributed control system
DL	Data layer
EMI	Electromagnetic interference
FDIA	Fault detection isolation and accommodation
GBN	Generalized binary noise
GPS	Generalized parity space
ICAM	Intelligent control and asset management
ID	Identification
JCSTR	Jacketed continuous stirred-tank reactor
LC	Level control
LQI	Link quality indicator
MAS	Multi-agent system
MEMS	Micro electro-mechanical system
NDDR	Nonlinear dynamic data reconciliation
NSR	Normal sampling rate
OS	Overshoot
PAWS	Petroleum applications of wireless systems
PC	Personal computer
PI	Proportional plus integral control
PEM	Prediction error minimization
RF	Radio frequency
RIM	Radio irregularity model

RSR	Reduced sampling rate
RSSI	Received signal strength indicator
S	Sensor
S/A	Sensor/Actuator
TC	Temperature control
TCP	Transmission control protocol
TCP/IP	Transmission control protocol/Internet protocol
UDB	User Datagram Protocol
UIL	User interface layer
UNB	University of New Brunswick
USB	Universal serial bus
WINTeR	Wireless industrial sensor network testbed for radio-harsh environments
WN	Wireless network
WNCS	Wireless networked control systems
WNCSCA	Wireless networked control system coordination agent
WSN	Wireless sensor network

Chapter 1

INTRODUCTION

1.1 Motivation

During the past two decades, a large amount of research has been done on distributed control systems that incorporate wireless sensor networks, or what are called Wireless Networked Control Systems (WNCSs) [15]. The source of that interest can be traced to many advantages achieved by eliminating the restrictions of traditional point-to-point wired control architectures, such as a reduction in wiring, rapid deployment, flexible installation, fully mobile operation, and improved freedom in placement of sensors and controllers [15, 6, 7, 4]. Networked control systems are becoming a mandatory technology for many military, commercial, and industrial applications. In such systems, distributed sensors, controllers, and actuators are exchanging information over a communication network. New research in wireless networked control allows engineers to support a number of control applications that were previously difficult to achieve or afford.

Due to the fast development of Micro Electro-mechanical Systems (MEMS) and wireless communication devices, engineers can integrate small sensors, actuators, tiny processors, batteries, and wireless communication devices into small packages which can be used in wireless sensor and actuator networks, or, more simply, wireless sensor networks (WSNs). These WSN nodes can then be distributed in large numbers

to self-organize into networks that serve a wide range purposes, including off-shore petroleum applications, environmental monitoring, industrial process control and intelligent systems for any application. Improved technology and stricter requirements make the development of WNCSs more difficult, however. Part of the problem arises from inflexibility in the imposition of strict requirements on data rates, jitter, latency and data loss to ensure control system performance on one hand [15, 6], and effective protocols for WSN robustness and efficiency [7, 4] on the other.

Developing a distributed control system over a wireless sensor network is a challenging task because it is necessary to satisfy pressing requirements from both fields, communication networks and control systems. The performance of the closed-loop system with wireless links (sensor-to-controller, controller-to-actuator) is one of the most important requirements in industrial control systems, so it is the main concern of this thesis. Although modest data rates, network delay, packet loss, and jitter are generally accepted in communication networks, there are strict limits as to what can be accepted for closed-loop control systems over wireless sensor networks. In distributed control systems, the network design objective should be to optimize the control performance. In WNCSs the network design objectives must include optimizing control performance, or at least maintaining stability-related constraints such as maximum acceptable percent overshoot. On the other hand, from the WSN perspective it is desirable to conserve node battery power and to have complete flexibility to configure the network to promote the efficient use of resources; therefore, slower data rates and more delay may be accepted to achieve these goals. Thus, there are distinct tradeoffs between network communications and controller performance.

From the control perspective, the more the controller knows about the system, the better the control performance is. This can be done by increasing the number of sensors or sending sensor measurements more frequently [7]. We are addressing this problem by developing a Wireless Networked Control System Coordination Agent (WNCSCA). The main responsibility of the WNCSCA is data rate and path delay management, both of which have a major impact on controller performance over the

WSN, and the energy consumption of the WSN nodes. Node energy conservation is a critical issue in WSNs for node and network life, as the nodes are usually powered by batteries, and in many cases the replacement of these power sources is difficult, inconvenient, and/or expensive. Our WNCSCA will coordinate with a supervisory control system and the WSN gateway to allow as much network optimization and flexibility as possible based on the performance requirements of the WNCS.

1.2 Objectives and thesis outline

This thesis focuses on the development of the **Wireless Networked Control System Coordinator Agent**, which will be part of the multi-agent intelligent supervisory system for Integrated Control and Asset Management (ICAM) for petroleum production facilities. The ICAM [19] system is explained in the following chapter. This agent will serve to mediate between the ICAM system and the Gateway of the Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments (WINTeR). WINTeR is an open-access, multi-user experimental testbed, currently under development by our research partners at Cape Breton University (CBU) under the project named Petroleum Applications of Wireless Systems (PAWS[1]), to support implementation and evaluation of wireless sensor networks for industrial applications in radio-harsh environments such as on- and off-shore petroleum processing facilities [21]. This agent will form important enabling technology for using WSNs in process control applications.

1.2.1 Objective

During the operation of the ICAM system, its agents impose different specifications or requirements on the WSN to complete their functions properly. For example, some agents require different data rates from specific sensors and actuators in specific

[1]The Petroleum Applications of Wireless Systems (PAWS) project is funded by the Government of Canada, through the Atlantic Innovation Fund (AIF).

regimes, such as start-up mode, set-point change mode and steady state mode for supervising and monitoring the plant. During start-up mode, the initial process variable transients must decay under closed-loop control, so appropriate data rates and path delays must be imposed. Once the process reaches the desired steady-state set point the model identification agent may perturb the plant with pseudo random signals (to excite all the modes in the plant), gather data and apply its model identification algorithm, perhaps needing a different data rate than before. After that, the process may remain settled in steady state, in which case loops can be opened[2] and data rates reduced so ICAM can monitor the process; as long as there are no disturbances or set-point changes slow sampling can continue and the Gateway can manage its operations freely and conserve energy accordingly. In many industrial systems a process may be in steady state for long periods of time, with infrequent set-point changes requiring closed-loop control, so this strategy would allow the WSN to managed in a more optimal way much of the time.

In summary, the various modes of operation require tighter or looser constraints on data rates and path delays; the WNCSCA mediates between ICAM and the WSN Gateway to allow both the control system and the WSN to meet their objectives as flexibly and well as possible. The interface between the ICAM system and the Gateway of WINTeR is portrayed in figure 1.1.

The objectives of this thesis are as follows:

1. Develop a method to check the configuration of the WSN to determine whether or not it meets control-loop performance requirements; based on that analysis the WNCSCA will accept or reject the WSN configuration.

2. Develop a method to manage energy consumption strategically, in partnership with the Gateway, by allowing reduction of the data rates for sensor and actuator nodes as much as possible without degrading control loop performance,

[2]Opening control loops momentarily to handle data drop-outs has been suggested by Kawka et al. [6]; our strategy of opening control loops to alleviate strict control-related WSN constraints is new.

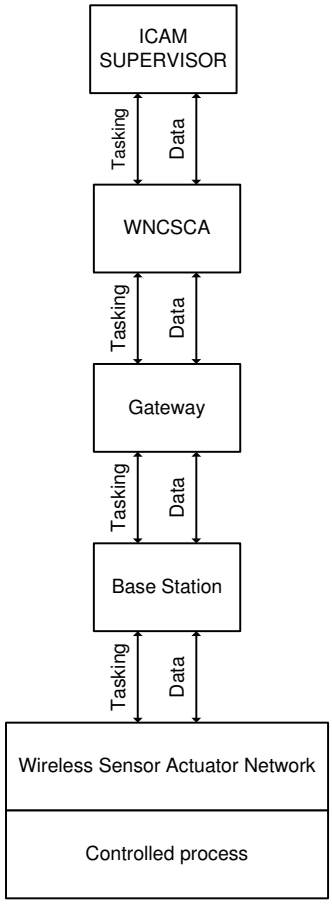

Figure 1.1: Wireless Networked Control System Coordinator Agent framework

based on monitoring the process state to determine its behavior (mode), e.g., are the states in steady state or transient conditions (this information is provided by ICAM's Steady-state Agent).

3. Design, implement, and test the WNCSCA that interfaces and mediates between the ICAM system and the gateway of the wireless sensor network.

Achieving these objectives comprises the contributions of this thesis.

1.2.2 Thesis outline

In chapter 2 the ICAM system is presented. In chapter 3 the Jacketed Continuous Stirred-Tank Reactor (JCSTR) and Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments (WINTeR) are presented. In chapter 4 the Wireless Networked Control System Coordinator Agent Communication Scheme with ICAM and the WSN Gateway is presented. Finally, in chapter 5 contribution, future work, and conclusion are discussed.

Chapter 2

ICAM System

2.1 Introduction

The Petroleum Applications of Wireless Systems (PAWS) research project is being pursued by several universities in Atlantic Canada for oil and gas applications. The main objective of the UNB PAWS project is to develop an intelligent control and asset management system. The overall PAWS project is divided into two areas: the first one is led by Cape Breton University (CBU), which is focused on implementing WSN protocols (at the physical, data link, network, and transport layers), security, channel modeling, RF circuit design, antenna design, and applications such as alarm management and condition-based monitoring for Wireless Sensor Networks (WSNs). The second one is led by University of New Brunswick (UNB) which is oriented to intelligent management and control of data to implement commercially viable wireless process control systems with applications within the energy sector, both onshore and offshore. For more information about the PAWS project see the corresponding web sites [14].

In order to achieve reliability, accuracy, and efficiency in a modern process control system, extensive supervisory monitoring and control are required. Several actions must be taken, including steady-state detection, nonlinear dynamic data reconciliation (NDDR), fault detection, isolation and accommodation (FDIA), process

model identification, and supervisory control. In order to maximize the efficiency of the process control, a multi-agent system (MAS), able to manage, supervise, and integrate all these tasks, had been designed and implemented. This system, developed by the University of New Brunswick (UNB) as part of the PAWS project, is the Intelligent Control and Asset Management system (ICAM system) [23]. ICAM is a system for supervising and managing industrial processes and which tries to reduce the maintenance and production costs, enhance safety, improve utilization of manufacturing equipment, and improve production.

2.2 ICAM system prototype

Data are acquired in real-time from an external plant or from a process simulation model. In this thesis, data are acquired from a nonlinear process simulator called Jacketed Continuous Stirred-tank Reactor (JCSTR). The JCSTR is explained in chapter 3. This data is transmitted to the data statistical pre-processor and reconciliation block. This component consists of two different agents. The first agent is the Steady-state Agent, which determines if the plant variables are either at a steady state or a transient state, and the second agent is the Nonlinear Dynamic Data Reconciliation Agent, which reduces noise, removes outliers and ensures that energy and material balance is achieved. The processed data are stored in a real-time database. The diagram shown in Figure 2.1 visualizes the simplified ICAM prototype [16].

The Model Identification Agent uses generalized binary noise (GBN) signals for testing and perturbing the process inputs and for collecting control relevant information about the dynamics of process and its environment. If there is no model available, or if a significant change in the process operating point occurs, the Model Identification Agent is executed. Once the new model is obtained, the process model parameters are updated and loaded in the NDDR and FDIA Agents [10].

At this time, data are received by the FDIA Agent to determine if the system is being affected by sensor or actuator faults. The FDIA Agent classifies the type

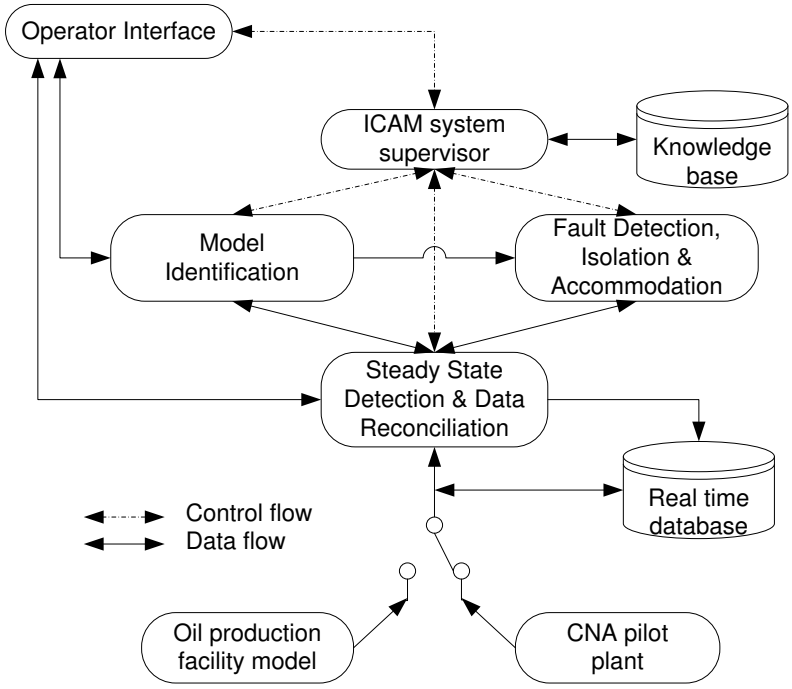

Figure 2.1: ICAM system prototype [18].

and size of fault and accommodates the fault if this has an impact on a sensor. The FDIA Agent informs the supervisor if a fault has occurred in order to proceed with the appropriate actions [11].

All agents inform the supervisor about every event that occurs. This enables the supervisor to control, monitor, and observe the system. The supervisor also sends appropriate process data to an operator interface, to allow the operator to take decisions according to the system status and its requirements. The external plant for this part of the project is a simulator that represents an oil production facility, which separates oil well fluids into crude oil, sales gas, and water. The plant itself is at the College of North Atlantic (CNA) [16].

2.3 Detailed ICAM agent descriptions

The ICAM system prototype consists of five reactive agents:

1. Pilot Plant Agent: The Pilot Plant Agent represents an oil production facility which separates oil well fluids into crude oil, sales gas, and water. The pilot plant can be run under different scenarios: the first scenario is the default scenario, which runs the model at its nominal operating point. The second scenario allows changes in the set points. The third scenario applies perturbations to the plant for model identification and the last scenario represents the plant when it is affected by faults in its sensors or actuators. The four scenarios illustrate the performance and the logical behavior of the ICAM system prototype during different circumstances [16].

2. Statistical Pre-processing Agent: This agent actually consists of two different parts. The first is the Steady-state Agent, which determines if the plant is in a steady or transient state and reports this state to the supervisor. The second agent is the NDDR Agent, which reduces noise, removes undesired discrepancies such as outliers and missing data, and reconciles the data with respect to material and energy balance [8].

3. Model Identification Agent: This agent is functioning in two cases: if there is no linearized model for the nonlinear process, or if there is a significant set-point change that makes the previously identified model invalid. First, the perturbations signals are generated and they are applied as set-point variations to excite each control loop and generate the controller output (plant inputs) and the plant outputs. Then, the linearized state space model and the corresponding percentage of fitting for each output are calculated using the prediction error/-maximum likelihood method PEM in MATLAB® for the reconciled input-output measurements [13].

4. Fault Detection, Isolation and Accommodation Agent: The FDIA Agent uses the generalized parity space (GPS) approach to create a set of directional residuals. From those residuals the faults can be detected and isolated. This agent is in charge of establishing the size and type of fault, and which sensor or actuator is being affected. If the fault is present in a sensor, the algorithm is also able to

accommodate the fault by correcting the sensor data. For further description about the FDIA Agent refer to [10, 11, 13].

5. Wireless Networked Control System Coordination Agent: The WNCSCA interfaces between the ICAM supervisor and the Gateway of the WSN at Cape Breton University (CBU) to achieve intelligent energy management under the constraint that operating control loops must perform acceptably. WNCSCA is responsible for managing the data rate of the WSN and the control loop delays by a new way that gives the WSN Gateway the freedom to reconfigure with out degrading the performance of the closed loop control in such away that reserves and reduce energy of sensors nodes.

2.4 WNCSCA interaction with the ICAM system

In order to show the importance of the ICAM system, it should be connected to a WSN. In this thesis the ICAM system is connected to WINTeR, which is under development by our research partner at Cape Breton University (CBU). Moreover, to connect the ICAM system to a WSN there are many requirements that must be satisfied on both sides. The main objective of this thesis is to implement the WNCSCA which is in charge of satisfying these requirements on both sides: the ICAM system and the Gateway of the WSN.

Performance of the control closed-loop system over the WSN is one of the most important requirements which should be preserved during the operation of the ICAM system. Data rate management also is one important factor which will be managed because it has a great impact on both the performance of the control closed-loop system over the WSN and on the life of the sensor node itself because the nodes are powered by batteries and in many cases the replacement of the power resource is difficult, inconvenient and/or expensive. Sensor node energy conservation is a critical issue in wireless sensor networks for node and network life. The configuration of the WSN also has a great impact on the induced delay time of the control packets of the

closed-loop system over the WSN, which directly affects control system performance.

Chapter 3

Process Simulation Model (JCSTR) and Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments (WINTeR)

The main objective of the ICAM system is to develop commercially viable wireless process control systems within the energy sector both onshore and offshore. The ICAM system needs real time data that are collected from either an external plant or from a process simulation model which simulates the petroleum application plant. The JCSTR simulator model was embedded in the Wireless Industrial Sensor Network Testbed For Radio-Harsh Environments (WINTeR) at Cape Breton University to run experiments with wireless-in-the-loop. The physical characteristics of the JCSTR and WINTeR hardware and software architecture are discussed in the following sections of this chapter.

3.1 Nonlinear JCSTR simulator

In this thesis, the Jacketed Continuous Stirred-Tank Reactor (JCSTR) has been chosen and implemented as a process simulator which reproduces the nonlinear nature of a typical petroleum application plant. The JCSTR simulator was implemented as a MATLAB® function by coding the differential equations for the nonlinear JCSTR and the PI (proportional/integral) controller in a suitable way to be used with MATLAB® integrators such as ode45, and as a Simulink® model using MATLAB® code in an S-function which models the physical process, digital controller blocks and zero-order holds. There are two process simulators currently linked to WINTeR, one a high-order, highly complex model of the CNA pilot plant [18] which has five control loops, and the other a simpler model of a JCSTR; the JCSTR has been chosen and implemented as a process simulator to better allow us to study issues related to the performance of control systems over a wireless sensor network. The data rates of sensors in a WSN affect the WSN nodes' energy consumption as well as the performance of a control system implemented with wireless communication links in its closed loops form. The main goal of this chapter is to present a practical way to determine the minimum data rate (maximum sampling time τ_{s-max}) that maintains acceptable performance of the closed loop control system over the wireless communication links of the WSNs, thus allowing the energy consumption of the WSN nodes to be reduced by not requesting a faster rate.

It is known that the value of maximum sampling time (τ_{s-max}) depends on the data delay or latency τ_d induced in the control loop paths by the WSN, which depends on the WSN configuration that comes from the number of hops in these paths, communication delay, congestion in the network, and other delay elements. If we account for delay using a conservative (worst-case) design, based, for example, on the maximum number of hops and the maximum delay per hop, then an unreasonably small τ_{s-min} may be demanded; this we should avoid. Therefore, we assume that τ_d can be determined by the Gateway of the WSN by using time stamps, ping tests or counting hops in the WSN configuration. The reconfiguration of the WSN will

require a redetermination or readjusting of τ_d and $\tau_{s-min}(\tau_d)$, so a practical method used for readjusting the sampling time according to the encountered time delay for each control loop over the WSN should be relatively easy to execute and fast, as shown in Chapter 4.

This JCSTR simulator will facilitate development of the WNCSCA, thus forming an important enabling technology for the use of WSNs in process control applications. The WNCSCA will not only maintain the **stability** of the closed-loop control system but also guard against the **performance** of the closed loop control system degrading excessively. On the other hand, node energy conservation is a critical issue in WSNs in terms of node and network life, as the nodes are usually powered by batteries, and in many cases the replacement of these power sources is difficult, inconvenient, and/or expensive. Our WNCSCA will coordinate with ICAM and the WSN Gateway to allow as much network optimization and flexibility as possible under constraints imposed by the requirements of the WNCS.

3.1.1 JCSTR physical model

The JCSTR is depicted in figure 3.1. As shown in figure 3.1, the tank inlet flow is received from another process unit. There is also a heating fluid is circulated through the jacket in order to heat the fluid inside the tank. The main objective of the JCSTR is to control the mixture level H (the level control or LC) and the mixture temperature T (the temperature control TC) inside the tank reactor. This can be done by controlling the jacket inlet valve flow rate and the tank outlet valve flow rate.

In order to derive the physical modeling equations of the tank and jacket temperature, the following assumptions were made:

- Perfect mixing of fluids in the tank and the jacket.
- Liquids inside the tank and the jacket have constant density and heat capacity.
- The tank inlet flow rate, tank inlet temperature and jacket inlet temperature may change; these are uncontrolled inputs, assumed here to be constants.

- The tank outlet flow rate and jacket flow rate are control inputs (actuation).
- The rate of the heat transfer from the jacket to the tank is governed by the equation $Q = UA_H(T_j - T)$ where U is the overall heat transfer coefficient, T_j is the heating fluid temperature, and A_H is the area of heat transfer which is given by:

$$A_H = A_B + \pi DH \tag{3.1}$$

$$A_B = \frac{\pi D^2}{4} \tag{3.2}$$

where A_B is the area of the tank base and D is the diameter of the tank reactor.

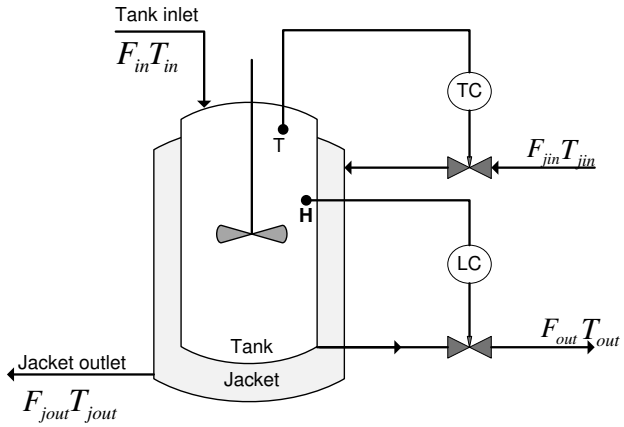

Figure 3.1: Jacketed continuously stirred-tank reactor.

The JCSTR simulator is a third order model which has two input variables, three state variables (outputs), and two control variables. The state variables are liquid height in the tank, H, temperature inside the tank, T, and the temperature inside the jacket T_j . The control variables are F_{out} and F_{jin}. The secondary input variables are the mixture inflow, F_{in}, mixture outflow, F_{out}, and the temperature of the mixture feed in the the tank, T_{in}.

The following ordinary differential equation set describes the dynamic behavior

and the physical nonlinear model of the JCSTR:

$$\dot{H} = \frac{1}{A_B}(F_{in} - F_{out}) \tag{3.3}$$

$$\dot{T} = \frac{F_{in}(T_{in} - T)}{A_B H} + \frac{U A_H (T_j - T)}{A_B H \rho C_p} \tag{3.4}$$

$$\dot{T}_j = \frac{F_{jin}(T_{jin} - T_j)}{V_j} - \frac{U A_H (T_j - T)}{V_j \rho C_p} \tag{3.5}$$

where the subscripts in, out, j refer to inlet, outlet and jacket respectively. Table 3.1 summarizes the constants and variables which were used in the physical modeling of the JCSTR simulator.

Table 3.1: JCSTR constants and variables

Parameter	Value	Unit	Characteristic
D	5	m	Diameter of the reactor (tank)
A_B	19.6350	m^2	Area of JCSTR base
H	dynamic value	m	Height of fluid in the reactor
A_H	dynamic value	m^2	Area of heat transfer
C_p	4186.8	j/kg·K	Heat capacity
ρ	997.95	kg/m^3	Density
U	851.74	W/m^2·K	Heat transfer coefficient
V	dynamic value	m^3	Volume of fluid in tank = $A_B H$
V_j	9	m^3·K	Heating water volume
F_{in}	0.1	m^3/s	Mixture inflow in reactor
T_{in}	283	K	Temperature of the mixture feed
F_{jout}	0.15	m^3/s	Heating fluid outflow
T_{jin}	419	K	Temperature of the heating fluid feed
x_1	H	m	Level of mixture in reactor
x_2	T	C	Temperature in the reactor
x_3	T_j	C	Temperature in the jacket
u_1	F_{out}	m^3	Tank mixture outflow
u_2	F_{jin}	m^3	Jacket heating fluid inflow

3.1.2 Controller design for the JCSTR

The JCSTR model that has been built is a nonlinear system. In order to design a controller to a nonlinear system, the linearization about the set point technique has been used. Most process control systems are required to keep the state of the dynamic process at a certain constant value. This value is usually called the set point, in the process control industry. This is achieved by providing control signals (plant inputs) that have two components: The first part or feedforward term $\bar{u}$ generates the control signal needed to keep the process at the set point in steady state, and the second component or feedback term δu generates the incremental control needed to return the process to the set point if it should deviate from it [5]. We can visualize the control system in these applications as two different subsystems. The structure of the control system designed in this way is shown in figure 3.2.

For the mathematical representation of the method of the linearization about the set point that was used, the nonlinear plant model (JCSTR) can be defined as:

$$\dot{x} = f(x, u) \tag{3.6}$$

$$y = g(x, u) \tag{3.7}$$

Suppose that $\bar{y}$ is the desired output for the control system, which is assumed to be constant. In our case, the desired outputs are the level, H, and the temperature, T, of the mixture inside the tank reactor. Suppose also that $\bar{x}$ and $\bar{u}$ are values of the state and control, respectively, that produce this constant output. Finally, suppose that $\bar{x}$ is an equilibrium state. Then equations 3.6 and 3.7 at the equilibrium state become:

$$0 = f(\bar{x}, \bar{u}) \tag{3.8}$$

$$\bar{y} = g(\bar{x}, \bar{u}) \tag{3.9}$$

The determination of $\bar{y}$ and $\bar{u}$ is an algebraic problem. The structure of the control system designed by linearization about a set point (linear control of nonlinear plant) is shown in figure 3.2. The function $U(.)$ is called the feedforward control law,

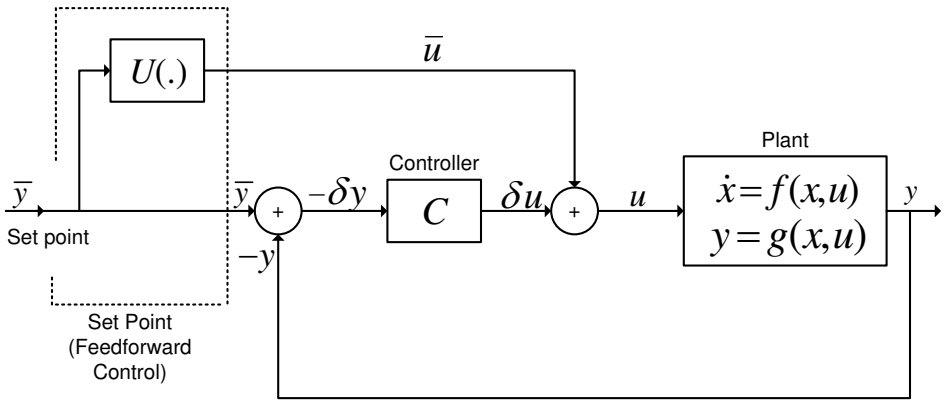

Figure 3.2: Structure of control system designed by linearization about a set point.

because it determines the setting of the control on basis of the desired output. To control the deviations of the actual state y from the set point $\bar{y}$, another subsystem is required. This subsystem is called feedback controller, because it generates the corrective control signals based on the actual measured performance. Figure 3.2 visualizes the structure of that system [5].

The feedback controller operates on the difference δy between the actual state y and the desired output $\bar{y}$ to generate a corrective control signal δu. Then, the corrective control signal is summed with the feedforward control signal $\bar{u}$ to produce the actual control signal u as input to the plant.

By substituting from equations 3.1 and 3.2 into 3.3, 3.4, and 3.5 and considering that the state variables of the system are level (fluid height), H, tank temperature, T, and jacket fluid temperature, T_j, i.e., $x = [H\ T\ T_j]^T$ and the control input $u = [F_{out}\ F_{jin}]^T$. Finally, the uncontrolled inputs are F_{in}, T_{in} and T_{jin}, i.e. $w = [F_{in}\ T_{in}\ T_{jin}]^T$. We can find the three state variables ordinary differential equa-

tions:

$$\dot{x_1} = \frac{1}{A_B}(F_{in} - u_1) \tag{3.10}$$

$$\dot{x_2} = \frac{F_{in}(T_{in} - x_2)}{A_B x_1} + \frac{U(x_3 - x_2)}{x_1 \rho\, C_p} + \frac{U\pi D(x_3 - x_2)}{A_B \rho\, C_p} \tag{3.11}$$

$$\dot{x_3} = \frac{u_2(T_{jin} - x_3)}{V_j} - \frac{UA_B(x_3 - x_2)}{V_j \rho\, C_p} - \frac{U\pi D x_1(x_3 - x_2)}{V_j \rho\, C_p} \tag{3.12}$$

In order to calculate the equilibrium states $\bar{u}_1$, $\bar{u}_2$ and $\bar{x}_3$, we need to make $\dot{x_1}$, $\dot{x_2}$ and $\dot{x_3}$ equal to 0. From that substitution we found that:

$$\bar{u}_1 = F_{in} \tag{3.13}$$

$$\bar{u}_2 = \frac{F_{in}(\bar{x}_2 - T_{in})}{T_{jin} - \bar{x}_3} \tag{3.14}$$

$$\bar{x}_1 = \bar{y}_1 \text{ (Set-point given)} \tag{3.15}$$

$$\bar{x}_2 = \bar{y}_2 \text{ (Set-point given)} \tag{3.16}$$

$$\bar{x}_3 = \bar{x}_2 + \frac{F_{in}\rho\, C_p(\bar{x}_2 - T_{in})}{UA_B + U\pi D\bar{x}_1} \tag{3.17}$$

In order to design the feedback controller which regulates the system to maintain the system output at the desired input values, the small signal linearization technique has been used. Given a dynamical system and output equation $\dot{x} = f(x, u),\ y = g(x, u)$, some values for u_0 and w_0 and the corresponding equilibrium x_0, then we can calculate the corresponding output value, $y_0 = h(x_0, u_0, w_0)$. Define the perturbation variables $\delta x = x - x_0$, $\delta u = u - u_0$, $\delta w = w - w_0$, $\delta y = y - y_0$.

If the perturbations are small and if continuous partial derivatives exist at (x_0, u_0, w_0) the behavior of the original system near the operating point x_0 is similar to that of $\delta\dot{x} = A\delta x + B\delta u + E\delta w$ and $\delta y = C\delta x + D\delta u + F\delta w$ where:

$$A = \left[\frac{\delta f}{\delta x}\right]_{x_0,u_0,w_0}, \qquad B = \left[\frac{\delta f}{\delta u}\right]_{x_0,u_0,w_0}, \qquad E = \left[\frac{\delta f}{\delta w}\right]_{x_0,u_0,w_0} \tag{3.18}$$

$$C = \left[\frac{\delta g}{\delta x}\right]_{x_0,u_0,w_0}, \qquad D = \left[\frac{\delta g}{\delta u}\right]_{x_0,u_0,w_0}, \qquad F = \left[\frac{\delta g}{\delta w}\right]_{x_0,u_0,w_0} \tag{3.19}$$

The procedure in equations 3.18 and 3.19 is called small signal linearization, since it provides a good model for the systems dynamic behavior only if the perturbation variables δx, δu, δu are small. By partially differentiating equations 3.10, 3.11, and 3.12 with respect to x_1, x_2, and x_3 , we evaluated the state space representation of the plant which involves the four matrices A, B, C, and D, where the partial derivatives are evaluated at the set point:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \qquad B = \begin{bmatrix} \frac{-1}{A_B} & 0 \\ 0 & 0 \\ 0 & \frac{T_{jin}-\bar{x}_3}{V_j} \end{bmatrix} \tag{3.20}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, and \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.21}$$

where

$$a_{21} = \frac{-F_{in}(T_{in} - \bar{x}_2)}{A_B \bar{x}_1^2} - \frac{U(\bar{x}_3 - \bar{x}_2)}{\rho\, C_p \bar{x}_1^2} \tag{3.22}$$

$$a_{22} = \frac{-F_{in}}{A_B \bar{x}_1} - \frac{U}{\bar{x}_1 \rho\, C_p} - \frac{U\pi D}{A_B \rho\, C_p} \tag{3.23}$$

$$a_{23} = \frac{U}{\bar{x}_1 \rho\, C_p} + \frac{U\pi D}{A_B \rho\, C_p}) \tag{3.24}$$

$$a_{31} = \frac{-U\pi D(\bar{x}_3 - \bar{x}_2)}{V_j \rho\, C_p} \tag{3.25}$$

$$a_{32} = \frac{UA_B + U\pi D\bar{x}_1}{V_j \rho\, C_p}) \tag{3.26}$$

$$a_{33} = \frac{-\bar{u}_2}{V_j} - \frac{U(A_B + \pi D\bar{x}_1)}{V_j \rho\, C_p} \tag{3.27}$$

$$\tag{3.28}$$

Note that we ignored the secondary plant input w; since we are only considering w equal to a constant it too can be eliminated from consideration except in the evaluation of A, B, C, D. By substituting from table 3.1 into A, B, C, and D, the numerical values of A and B can be calculated as following:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.0036 & -0.0012 & 0.0005 \\ -0.0575 & 0.0076 & -0.0254 \end{bmatrix}, B = \begin{bmatrix} -0.0509 & 0 \\ 0 & 0 \\ 0 & 3.0396 \end{bmatrix}$$

Therefore, the plant state space representation is as follows:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0.0036 & -0.0012 & 0.0005 \\ -0.0575 & 0.0076 & -0.0254 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} -0.0509 & 0 \\ 0 & 0 \\ 0 & -6.9626 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

By converting A, B, C, and D from the state space form to its transfer function equivalent form, we got $G(s)$ where $G(s)$ is a 2×2 matrix of transfers functions where:

$$\mathrm{G(s)}= \begin{bmatrix} \frac{-0.05093}{s} & \frac{-0.0001852s - 3.31e-006}{s^3 + 0.02661s^2 + 2.641e-005s} \\ 0 & \frac{0.001453}{s^2 + 0.02661s + 2.641e-005} \end{bmatrix}$$

.

By designing suitable PI controllers for the transfer functions $G(s)_{11}$ and $G(s)_{22}$ in the main diagonal elements of the plant transfer function $G(s)$ by using the root locus technique, we found the controllers that give a fast response with low overshoot as follows:

$$C_{11}(s) = 0.05 + \frac{3.0e-005}{s}$$
$$C_{22}(s) = -0.4 + \frac{-9.2e-004}{s}$$

Therefore, the controller transfer function is as following:

$$C(s)= \begin{bmatrix} C_{11}(s) & 0 \\ 0 & C_{22}(s) \end{bmatrix}$$

.

3.1.3 JCSTR simulator test

In order to test the process simulator functionality, step inputs are applied to the level and the temperature of the mixture inside the tank reactor. By considering the initial conditions of the states for the JCSTR simulator as shown in table 3.2, and applying two successive set-point changes for the level and the temperature inside the reactor as follows:

- The level set point changes at $t = 90$ minute from 7 to 7.5 m.
- The mixture temperature inside the reactor changes at $t = 120$ minute from 52 C to 56 C.

The step responses of those changes in the set points are shown in figures 3.3, 3.4, and 3.5 respectively. The PI control gains could probably be tuned to improve performance, however, given the strongly nonlinear and coupled behavior of the temperature loop an overshoot of about 30% for T is not bad.

Table 3.2: JCSTR states' initial conditions

State initial condition	Value	Unit	Characteristic
x_{1_0}	6.6	m	Level of mixture in reactor
x_{2_0}	50	C	Temperature in the reactor
x_{3_0}	143	C	Temperature in jacket

3.1.4 JCSTR simulator modification

The JCSTR simulator has very slow dynamics, as can be noticed in figure 3.3 when the level desired output was changed from 6.6 to 7 meter at $t = 0$, the settling time for this response is $T_s = 43.23$ minutes to settle with in $\pm\delta = 5\%$. When the level set point was also changed from 7 to 7.5 meter at $t = 90$ minutes, the settling time for this response is $T_s = 48.4$ minutes to settle with in $\pm\delta = 5\%$. A slow response can also be noticed in figure 3.4 for the mixture temperature. Therefore, the simulator has a

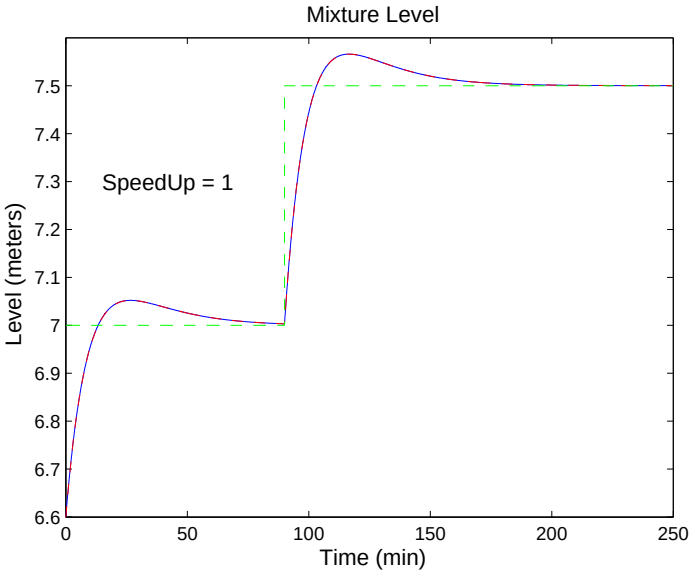

Figure 3.3: Mixture level inside the reactor vs. time.

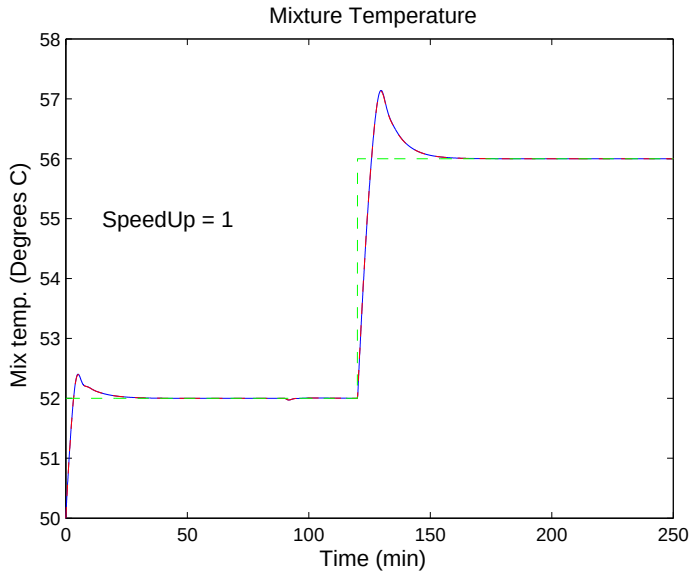

Figure 3.4: Mixture temperature inside the reactor vs. time.

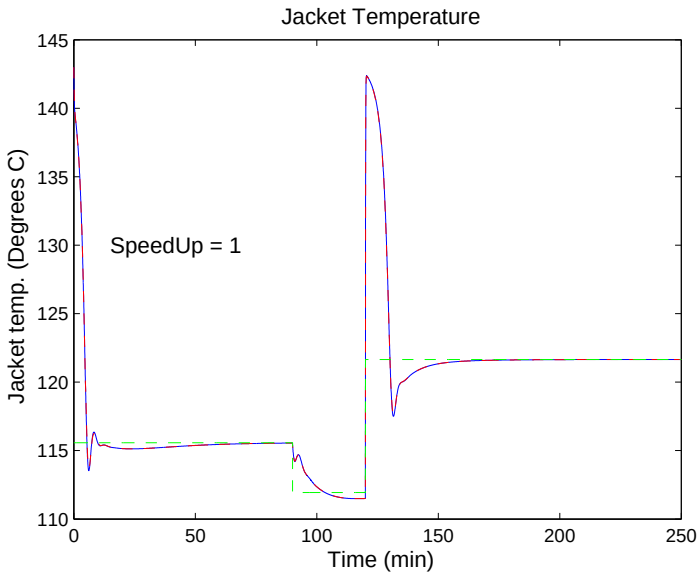

Figure 3.5: Temperature inside the jacket vs. time.

very slow dynamics in real time. That period of time is too long to be implemented in a real time version of that simulator which will be embedded in WINTeR to run our experiments.

The huge reactor tank dimensions are the reason of the slow dynamics of that simulator, 5 m in diameter and 10 m in height, that is a huge amount of liquid for which to control level and temperature. Therefore, a scale factor to speed up the dynamics was added which is called SpeedUp. The value of the SpeedUp factor is 50. The differential equations of the nonlinear JCSTR model were multiplied by the SpeedUp factor, then the dynamics are sped up by 50 times. That roughly corresponds to the dimensions of the reactor (height, diameter) being reduced by 1/50. This is an effective way to decrease the model response times, but it is nonphysical.

In order to test the new version of the process simulator functionality, set point changes are applied to the level and the temperature of the mixture inside the reactor tank. By considering the same initial conditions of the states for the JCSTR simulator as shown in table 3.2 and applying the same two successive step inputs for the level and the temperature inside the reactor in the new time scale (120 minute in

figure 3.4 equal 2.4 minute now) use get the step responses shown in figures 3.6, 3.7, and 3.8 respectively.

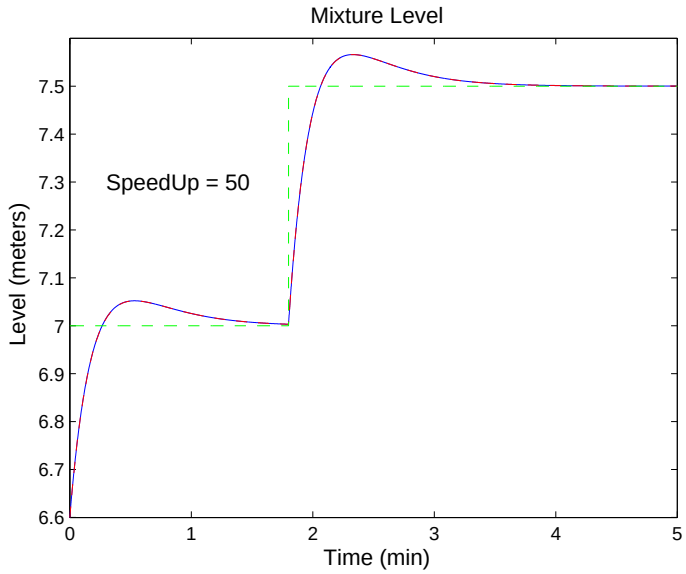

Figure 3.6: Mixture level inside the reactor vs. time (SpeedUP version).

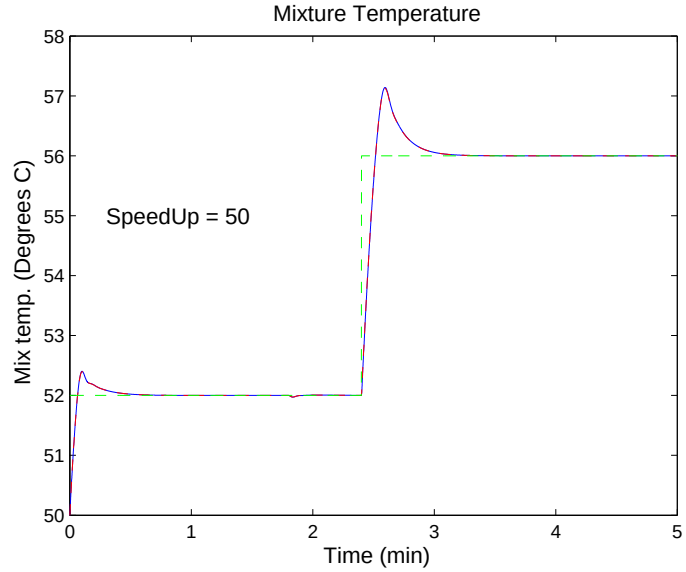

Figure 3.7: Mixture temperature inside the reactor vs. time (SpeedUP version)

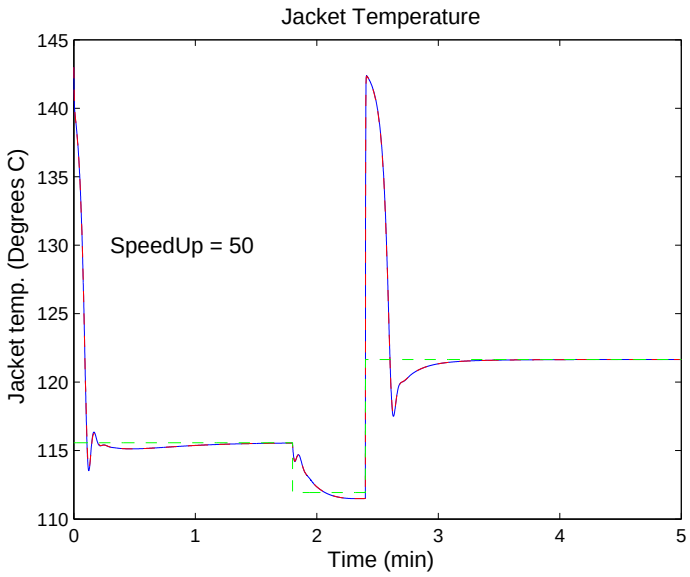

Figure 3.8: Temperature inside the jacket vs. time (SpeedUP version).

3.1.5 JCSTR Simulink® model

In order to simulate and understand the effects of the sampling rate and the time delay on the performance of a closed-loop system, two Simulink® models for the JCSTR have been implemented to enable us to change the sampling rate and the time delay of the two sensors involving in the two control closed loop in the JCSTR. We followed this procedure:

1. JCSTR Simulink® model with continuous controllers: The JCSTR simulator has been implemented as a MATLAB® S-function, to be suitable for embedding in WINTeR. An S-Function was used to implement the JCSTR (as a physical model) as well as the two controllers, as shown in figure 3.9. The step response plots which are depicted in figure 3.10 and 3.11 show that the JCSTR Simulink® model has the same behavior and characteristics as the JCSTR simulator which was implemented by a MATLAB® M-file (function), shown in figures 3.6 and 3.7.

2. JCSTR Simulink® model with digital controllers: In order to simulate the effects of the sampling time of the sensors on the performance of the control closed

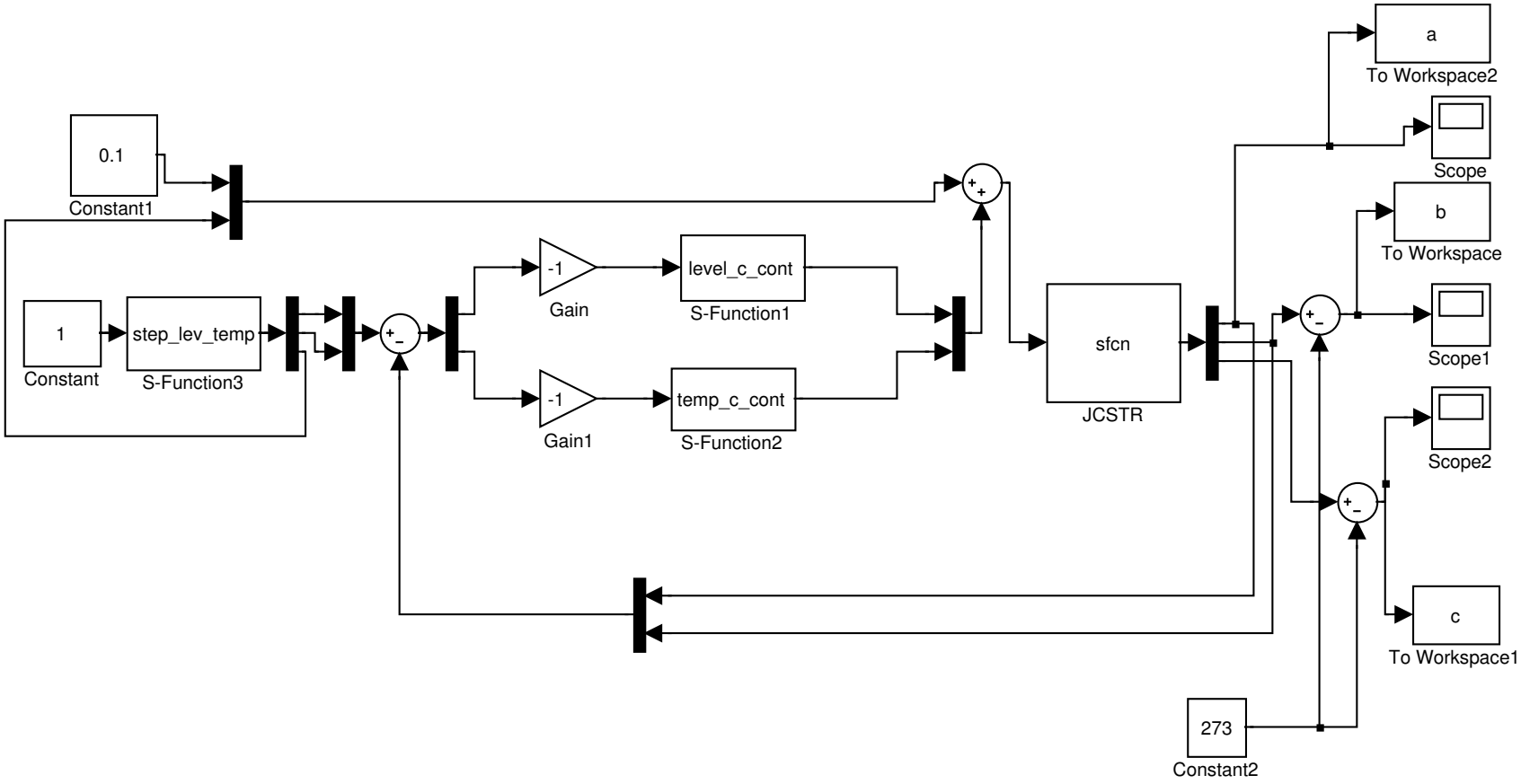

Figure 3.9: JCSTR Simulink model.

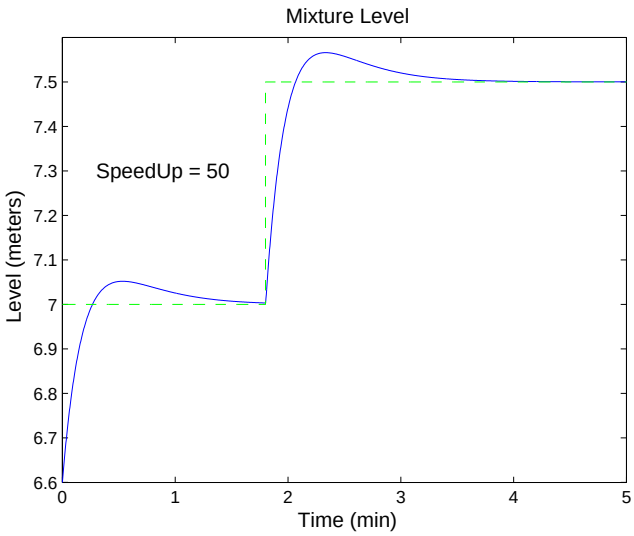

Figure 3.10: Mixture level inside the reactor vs. time (Simulink, continuous controller).

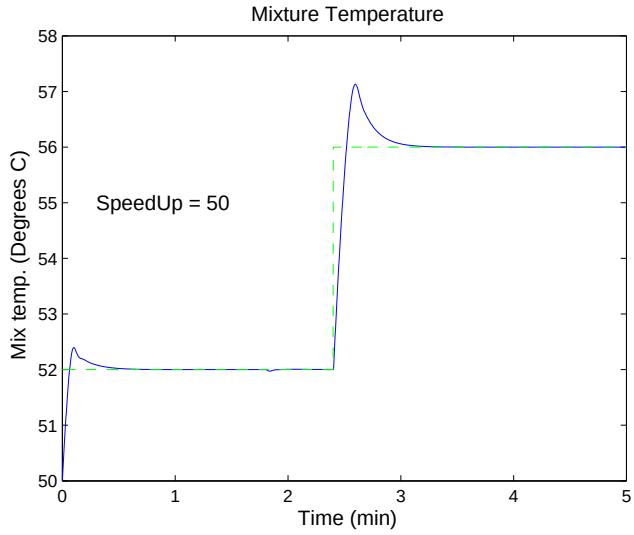

Figure 3.11: Mixture temperature inside the reactor vs. time (Simulink, continuous controller).

loop, and find the maximum sampling time rate that achieves an acceptable performance for the closed control system (which affects the power consumption of the nodes‘ batteries of the wireless sensor networks) the mixture level controller and the mixture temperature controller have been digitalized and tuned with 0.1 second sampling rate and a Zero-Order Hold was included as shown in figure 3.12. The step response plots which are depicted in figure 3.13 and 3.14 show that the JCSTR Simulink® model with digital controllers has the same basic behavior and characteristics of the JCSTR Simulink® model which has been implemented with continuous controllers (Figures 3.10, 3.11) but it has less overshoot, which is one of the advantages of the digital controller. These % OSs will be considered as the benchmark percent overshoots; they are 10.8 for the level loop and 11.0 for the mixture temperature loop.

3. Changing the sampling time: The sampling time of the mixture level sensor has been varied to check its effect on the control closed loop performance. In this thesis, the judgement factor on the acceptable performance of the closed loop control has been chosen to be in between $23:27$ %OS[1] (acceptable %OS band). By fixing the sampling time of the temperature sensor to its benchmark value (0.1 s) and varying the sampling time for the mixture level closed loop τ_{sL} , we found that the maximum sampling rate that makes the mixture level control closed loop stable is 0.25 second, which gives 24.9% overshoot as shown in figure 3.15. If the sampling time increases more than that value, for example the sampling rate is 0.3 second, it gives a low damping oscillation (poor/unacceptable) response with 40% overshoot which is not accepted, as shown in figure 3.16.

   A similar study was done for the mixture temperature closed loop control. The sampling time of the mixture temperature sensor was varied to check its effect on the performance of the control closed loop. By fixing the sampling time of the level sensor to its benchmark and changing (tuning) the sampling time for the mixture temperature closed loop, we found that the maximum sampling

[1]Of course a more conservative specification such as design %OS=0 and acceptable %OS=10 could be used.

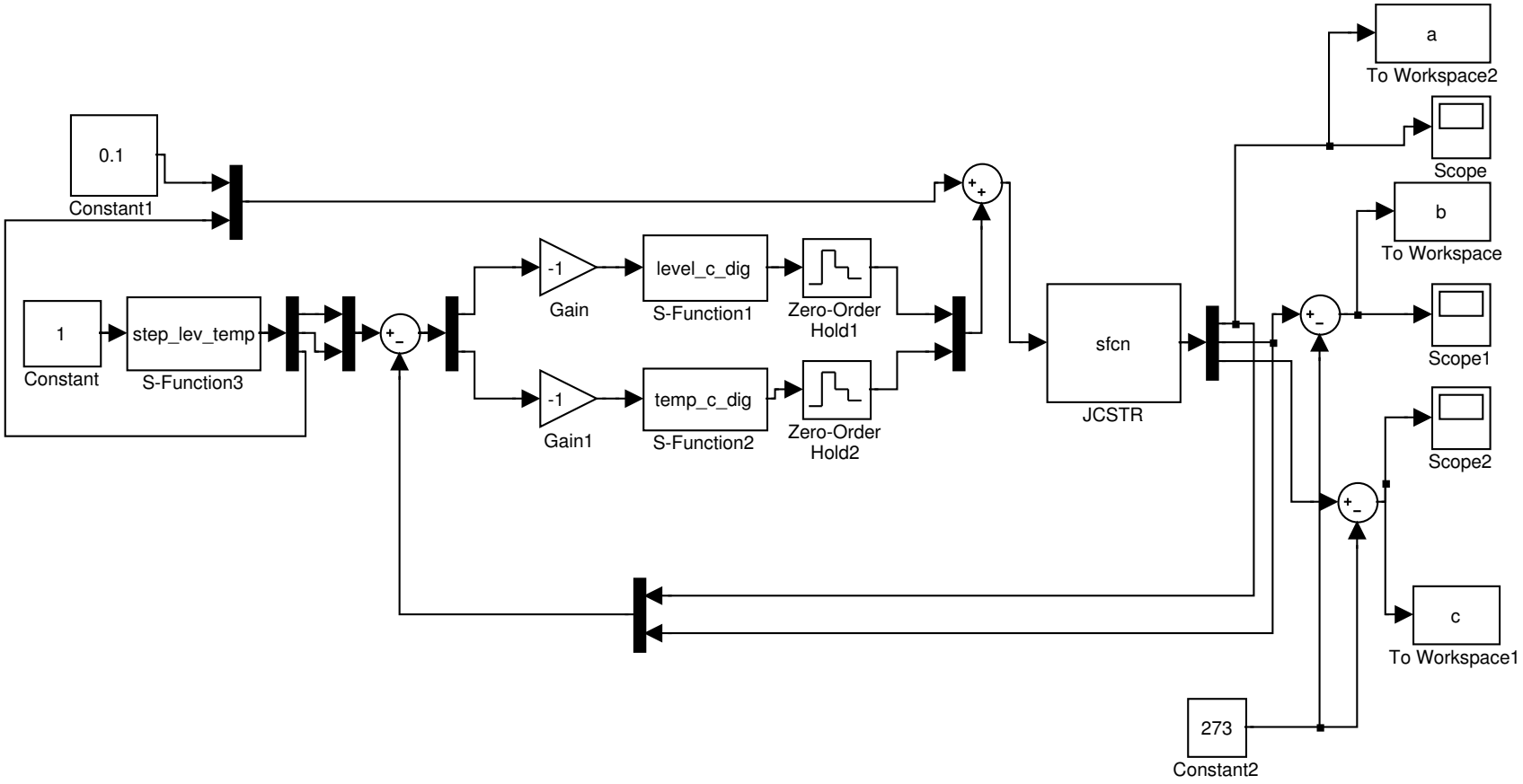

Figure 3.12: JCSTR Simulink model (with digital controllers).

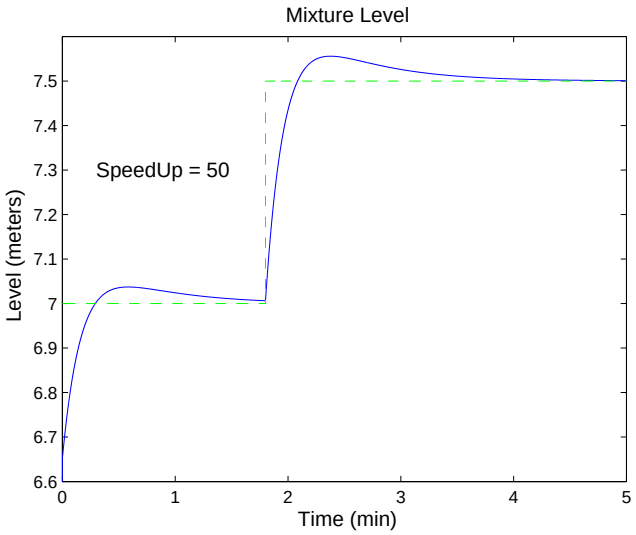

Figure 3.13: Mixture level inside the reactor vs. time (digital controller).

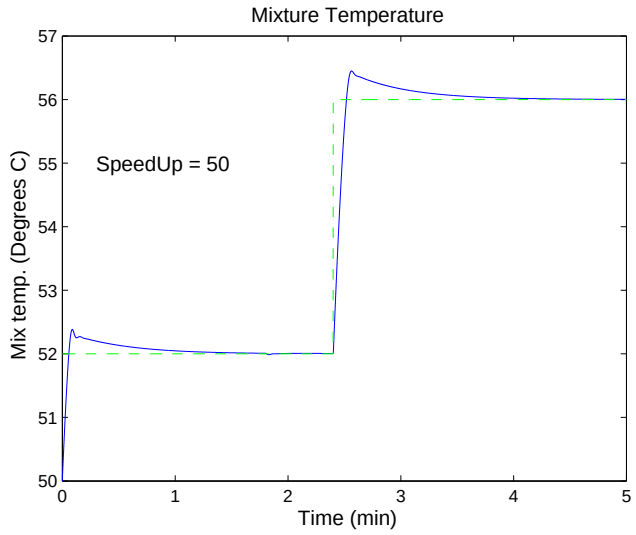

Figure 3.14: Mixture temperature inside the reactor vs. time (digital controller).

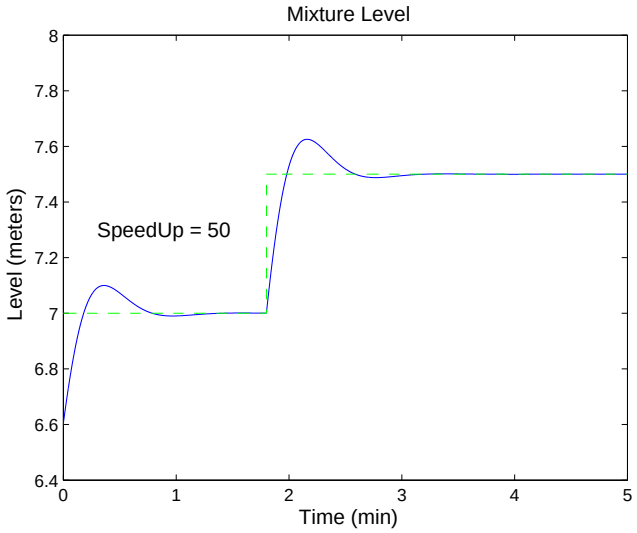

Figure 3.15: Mixture level inside the reactor vs. time with $\tau_{sL} = 0.25$ s (24.9% overshoot).

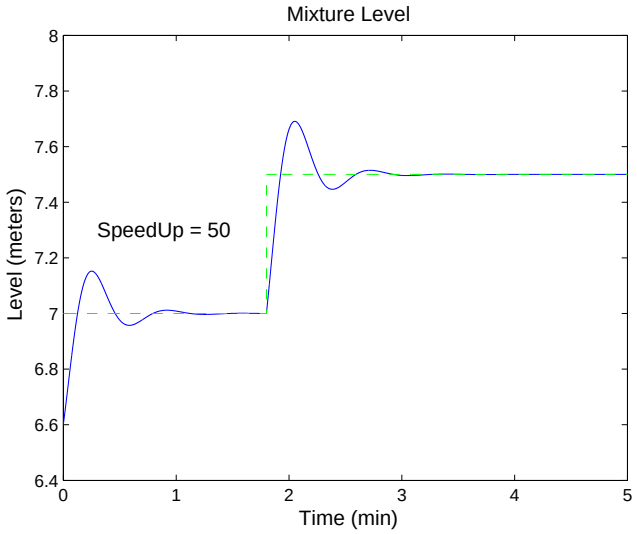

Figure 3.16: Mixture level inside the reactor vs. time with $\tau_{sL} = 0.3$ s (40% overshoot).

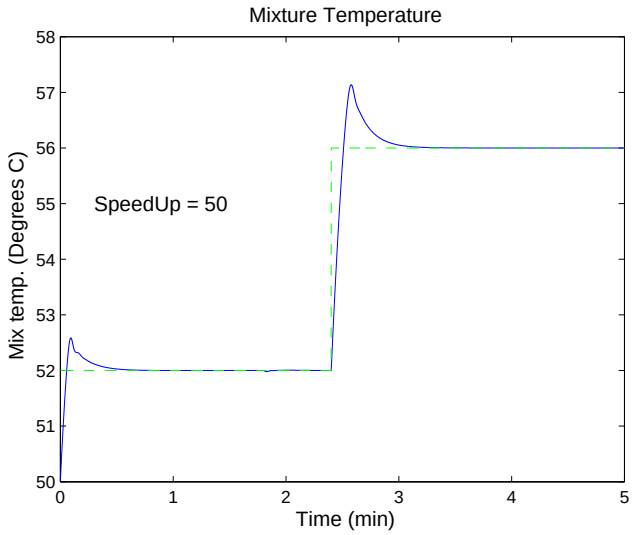

Figure 3.17: Mixture temperature inside the reactor vs. time with $\tau_{sT} = 0.3$ s (28.5% overshoot).

rate that makes the mixture level control acceptable is 0.3 second, which gave 28.5% overshoot as shown in figure 3.17. If the sampling time increases more than that value, for example the sampling rate is 0.35 second, it gives poor unacceptable response with 51.2% overshoot as shown in figure 3.18.

4. Adding time delay elements: after adjusting the lowest sampling rate that gives an acceptable response based on the percentage overshoot time delay elements were inserted in the feedback paths of the level and temperature loops as shown in figure 3.19. By resetting the sample time of the level and temperature sensor to their benchmark values and increasing the time delay element 1 and time delay element 2 gradually starting from zero time delay, we can find the maximum time delay τ_{dL-max} and τ_{dT-max} which lead to marginally acceptable performance.

   The measured time delays will be compared by the actual time delay of the transmitted control data packets through the WSN which is caused by many factors such as congestion in the network, packet transmission time delay, network configuration, etc as discussed in Chapter 4. By running the above experiment

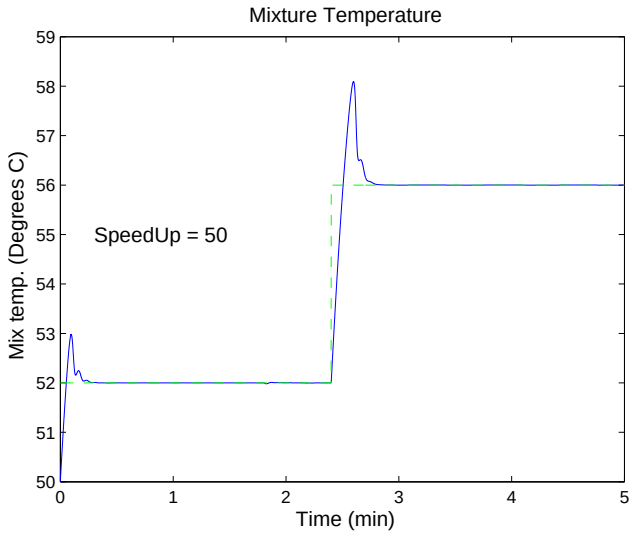

Figure 3.18: Mixture temperature inside the reactor vs. time with $\tau_{sT} = 0.35$ s (51.2% overshoot).

for both loops, we found that the maximum time delay element for the level control loop is $\tau_{dL-max} = 0.253$ second, which gives 25% overshoot, and the maximum time delay element for the temperature control loop is $\tau_{dT-max} = 0.182$ s which gives 27.6% overshoot as shown in figure 3.20 and 3.21 respectively. But when the time delay element increases more than these maximum values τ_{dL-max} and τ_{dT-max}, the closed loop system gives too much overshoot (more than 30%).

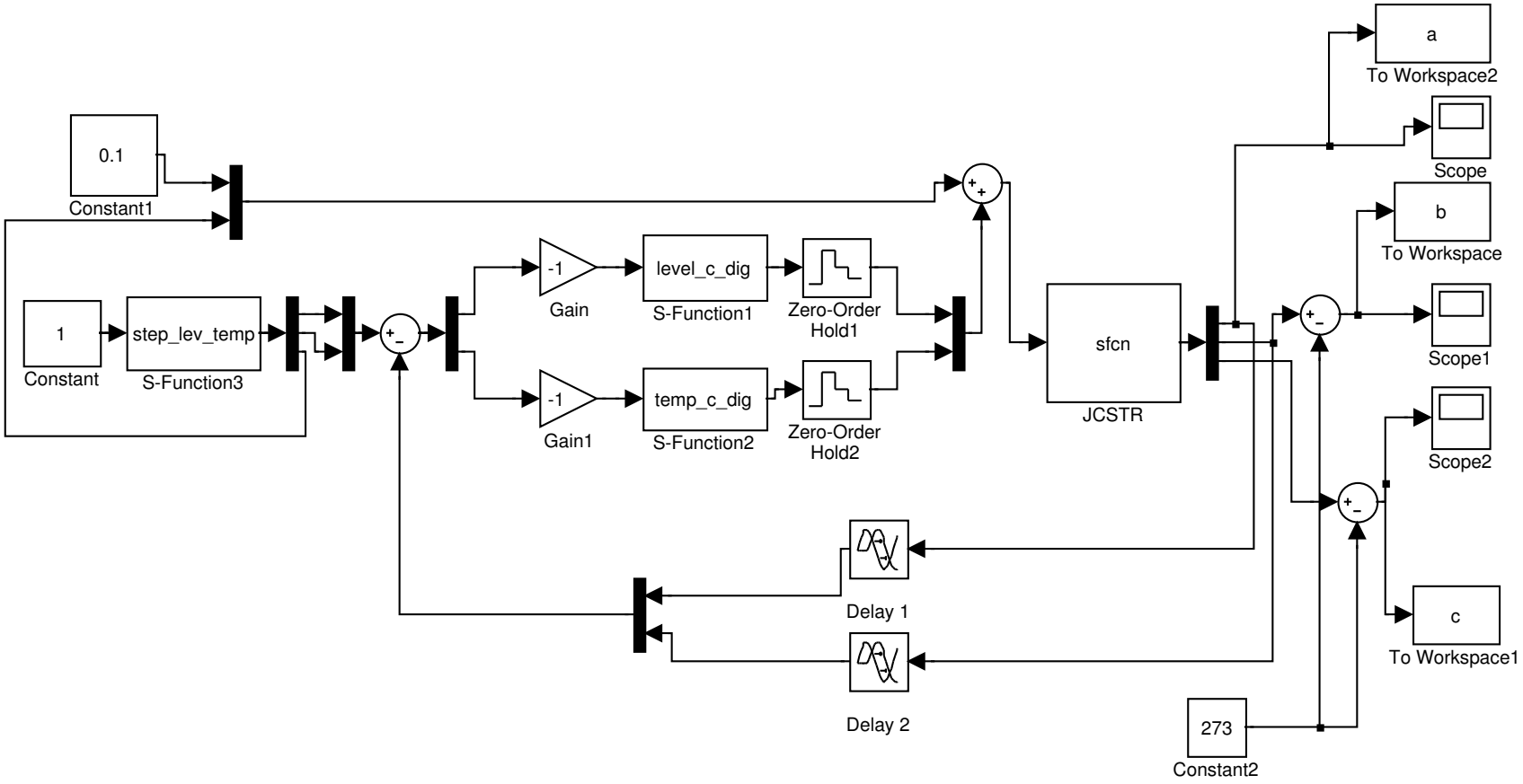

Figure 3.19: JCSTR Simulink model (with digital controllers) and time delay elements in the feedback paths.

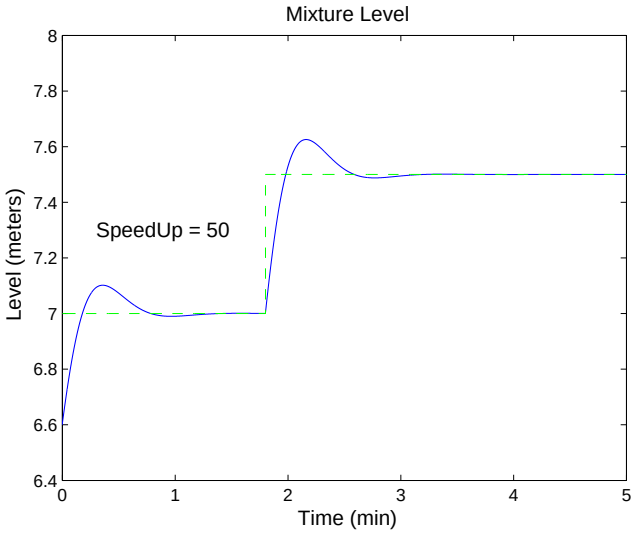

Figure 3.20: Mixture level inside the reactor vs. time with 0.253 s time delay (25% overshoot).

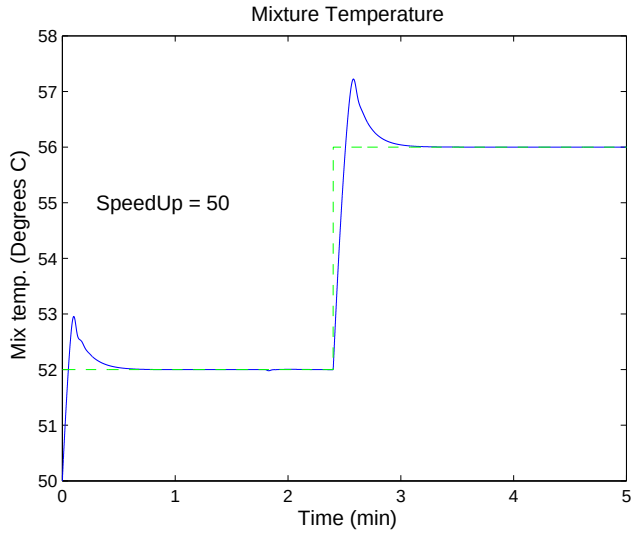

Figure 3.21: Mixture temperature inside the reactor vs. time with 0.182 s time delay (27.6% overshoot).

3.2 WINTeR

In order to develop robust and reliable wireless sensor networks which can operate in the harsh environment of the petroleum field including shore-based refineries and offshore-based facilities, the Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments (WINTeR) was implemented. WINTeR is an open-access multi-user experimental testbed that supports the development and evaluation of WSNs [21]. WINTeR is a testbed that supports the research and development of WSNs technologies, including the physical layer implementation, propagation model measurement, simulation of industrial processes, and cross-layer optimization. This testbed is under development by our research partners at Cape Breton University.

3.2.1 WINTeR hardware architecture

The diagram of the testbed is depicted in figure 3.22. The testbed is composed of five components:

1. Physical Structure: which provides a real model for the radio harsh environment which exists in petroleum fields by including pipes, beams, tanks, and reflecting metal surfaces.

2. Testbed Server: which provides remote access options to the testbed, allowing multiple researchers to access, create jobs, execute jobs, and log data from anywhere in the world by using the Internet.

3. Data Generation Server: which generates data to be transmitted to the WSN motes deployed in different places in the testbed, as shown in figure 3.22. Any process control simulator can be uploaded on the data generation server. The JCSTR simulator will be uploaded on the data generation server to run future experiments.

4. Mote Platform: which consists of a mote, DC power source, embedded micro

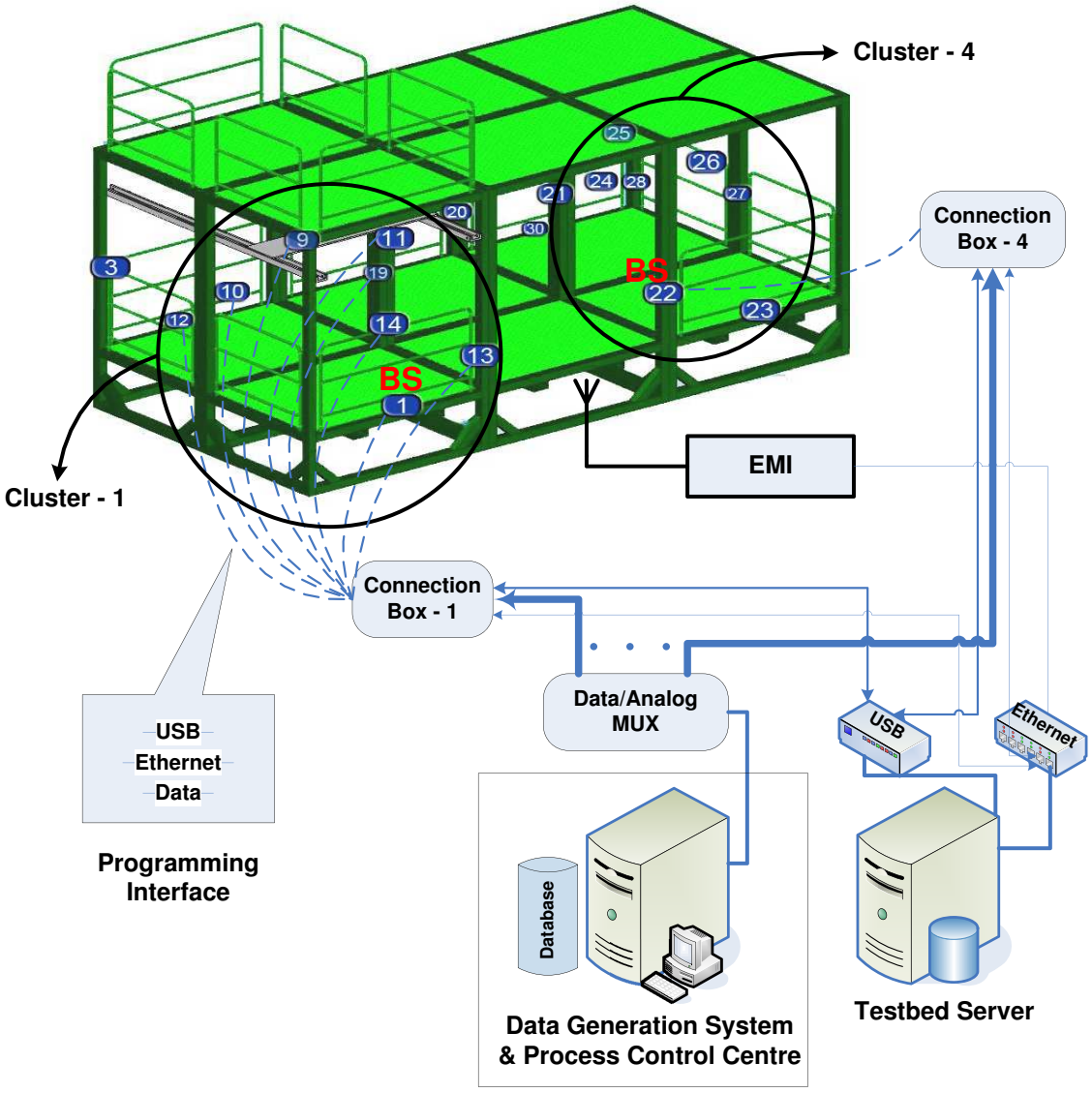

Figure 3.22: WINTeR hardware architecture [23].

processor, micro power meter, programmable attenuator, and industrial enclosure.

5. EMI (Electromagnetic Interference) Generator: which transmits real electromagnetic interference signals during the experiments trying to mimic the actual noise and interference signals in industrial environments.

3.2.2 WINTeR software architecture

The software architecture of the testbed can be divided into six layers, as shown in figure 3.23.

1. User Layer (UL): is used to enable users to connect to WINTeR. Users can connect to the testbed from anywhere in the world by using the Internet.

2. User Interface Layer (UIL): is used to enable researchers to utilize the multiple functions of the testbed. It accepts user inputs for creating jobs, and gives the results of those jobs. There are multiple components in this layer which allow users to create their own jobs (by the Job Creator), configure the schedule for those jobs (by the Scheduler), and graph the results of their experiments.

3. Execution Layer (EL): consists of a Job Daemon module, EMI module, Fading module, Data generation module, and Report generator module. Those modules manage, control, and help job execution.

4. Data Layer (DL): is used for permanent and temporary storage. The DL is implemented as a MYSQL database. There are three sections of the Data Layer. The first one is the Configuration Database, which is used to store data for users' accounts, job configurations, and scheduling. The second one is the Output Database, which stores temporary data and interfaces between the Data Interface Layer and Execution Layer. The third one is the Users Database which keeps track of all testbed‘s users.

5. Device Interface Layer (DIL): is used as a switch which directs data between the DL and the End Device Layer (EDL). The DIL consists of five processes. The first process is Programmer Output, which is in charge of programming the wireless devices by using USB or Ethernet Back Channels; it obtains the mote type from the database and invokes the corresponding programmer function. The second process is the EMI Output process, which invokes the name of the user-specified interface signals and programs the Vector Signal Generator device to produce the required electromagnetic interference signal. The third process is the Transducer I/O process which is a LabVIEW™ process that sends the digital signals to a specific channel on the D/A card for a specific mote. The fourth one is the Power Meter process, which accepts the power consumption information from motes (batteries level) and motes states, and then logs it in the user database. The fifth process is Data Logger process, which receives messages from the wireless devices, sorts the data, and logs it in the user database. Important statistics can be traced and captured, such as dropped packets, packet time delay, power consumption as a function of mote state, received signal strength indicator (RSSI), and link quality indicator (LQI).

6. End Device Layer (EDL): consists of all physical devices which are used for implementing the wireless sensor networks (creating inputs or outputs) such as sensors and actuators.

3.2.3 WINTeR applications

Multiple research aspects concerning WSNs can be studied on WINTeR, such as routing protocols, security, power management, and cross layer optimization. Some of those aspects concern the following:

- Routing protocols: The testbed supports conducting routing protocol research for a WSN, where the testbed can be used as a vehicle to verify, validate, and tune energy efficient communication protocols. Researchers can upload their

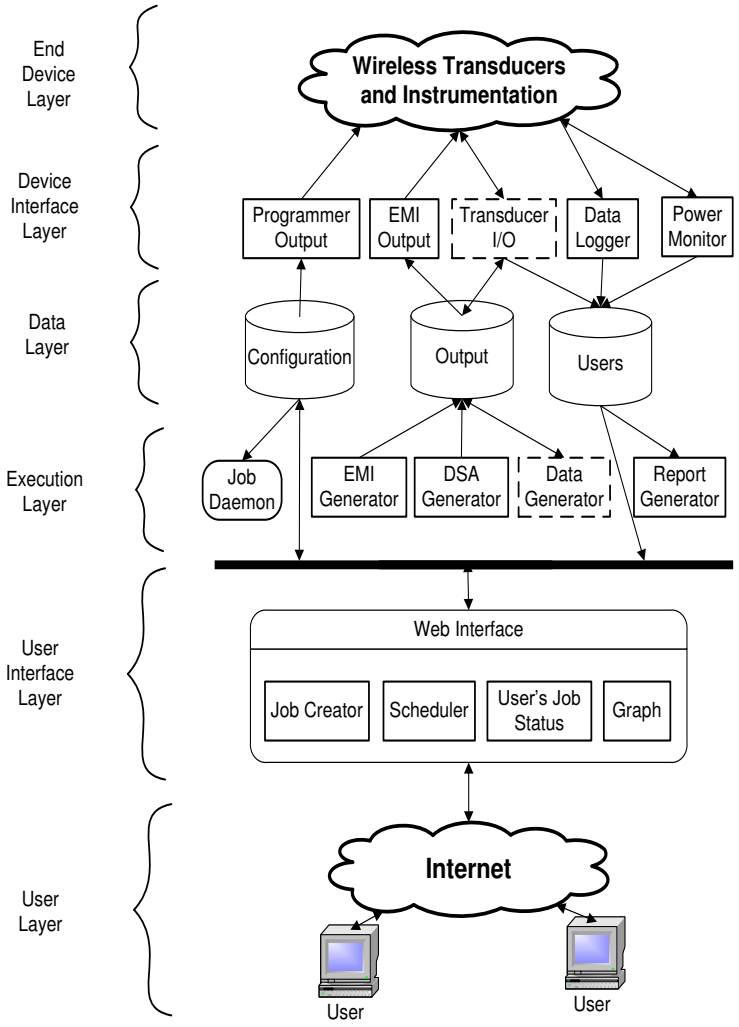

Figure 3.23: WINTeR software architecture [21].

own TinyOS code for WSNs to the testbed and configure the testbed according to their research requirements by using the user-defined channel models (number of sensors, type of sensors, RSSI, EMI, etc). Then, they can run their code and get the results in textual or graphical formats. The testbed supports designing and testing energy efficient data communication protocols and supports the investigation of realistic power consumption models such as the radio irregularity model (RIM).

- Security: This is a very important issue in wireless sensor networks in general, especially in WSNs which are involved in the petroleum industry due to the existence of industrial espionage and hacker attacks. The testbed supports testing newly developed security protocols for WSNs. Researchers can download their own new security protocol on the testbed and run multiple attacks as required to break into the network, such as adding an eavesdropping node (non-invasive attack) or deploying malicious nodes that gain network trust and send malicious data to the network. By this way the designed security protocol can be tested, evaluated, and verified in a practical environment similar to an industrial one.

- Power consumption modeling: Computer simulation cannot be used to evaluate power consumption in a WSN properly. In order to get a true power consumption model you need to have your routing protocol running on a real WSN. The micro power meter which is embedded in each mote is used to measure and monitor power consumption as a function of the mote state, such as sleep, wakeup, receive, and transmit [21].

- Validation of wireless technology for plant: Industry needs a reliable WSN under radio-harsh environment conditions. An industrial process can be modeled and downloaded on the testbed. At this time researcher can run simulations for that model and get realistic results that reflect the performance of the process. In this thesis a JCSTR simulator was supposed to be available to run experiments

to investigate control loop performance over the wireless sensor network; this did not happen.

- Cross-layer optimization: It is impossible to provide an overall optimized solution by simply combining optimized solutions of the individual layers [22]. A combination of technologies is supported by the testbed, which can be used at various of layers to investigate the cross-layer optimization of the network.

For a complete description for WINTeR refer to Slipp et al. [21].

Chapter 4

WNCSCA strategy and Communication Scheme

Sampling time and time delay have a huge direct impact on the performance of the closed loop control over a WSN. Those two factors must be optimized in such way that maintains acceptable performance for the closed loop control system without violating the most important principal in the WSN, which is reduce the nodes' energy consumption and gives the WSN the freedom to reconfigure at any time without losses in the control loop(s) performance. We begin this chapter by studying the effects of sampling time and delay on WSN control loops performance, and understanding the type of relationship between the sampling time and the maximum delay time which the control packets encounter through the network. Then the overall WNCSCA strategy is developed and presented in a way that coordinates with the Gateway for reconfiguration of the WSN. Finally, the communication scheme linking the Gateway, WNCSCA and ICAM is described; by three consecutive phases, Booting and initialization phase, Normal Operation phase, and finally Abnormal Operation phase.

4.1 Sampling time and maximum delay time relationship

In order to show the relationship between the sampling time and the maximum time delay that degrades the closed control loop performance over the networked control system, a case study has been performed on the loops in the JCSTR model. By fixing the sampling time and the delay time of the temperature sensor (TC) at three operating points, as shown in figure 4.1, and varying the sampling time of the level sensor, and plotting the curves of τ_{dL-max} versus τ_{s-LC} for each case, we see a substantial inverse relationship between the sampling time and the maximum time delay has been discovered, as shown in that figure. These plots also emphasize the substantial interaction in these two loops.

The previous study was repeated in reverse, by fixing the sampling time and the delay time of the level sensor (LC) at three operating points and varying the sampling time of the temperature sensor, and plotting the curves of τ_{dT-max} versus τ_{s-TC} for each case. The same substantial inverse relationship between the sampling time and the maximum time delay has been discovered, as shown in figure 4.2. By these experiments the relationship between different sampling times and their corresponding maximum time delay which makes the closed control loop acceptable and the inverse relationship between them have been demonstrated. The temperature control loop is more sensitive than level control loop in terms of delay time, by the nature of the JCSTR plant.

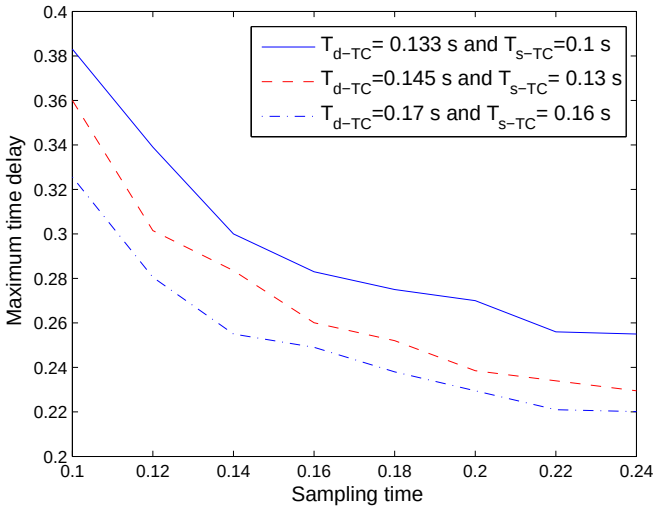

Figure 4.1: Maximum delay time and sampling time relationship for the level control loop.

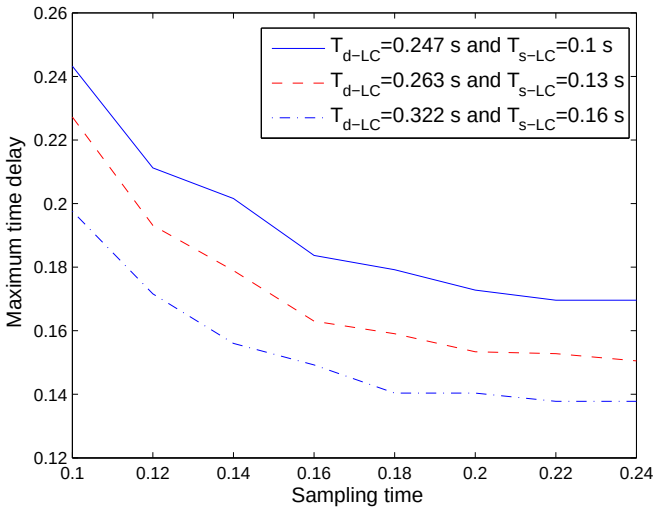

Figure 4.2: Maximum delay time and sampling time relationship for the temperature control loop.

4.2 WNCSCA strategy

Based on the observations presented above, we have created the following scheme for determining the minimal sampling rates for the WNCS; there are two parts:

Part 1: Determining τ_{s-max}

In order to determine the maximum sampling times for sensors and actuators that preserve an acceptable control system performance (to reduce node's energy consumption) an efficient (in terms of simulation time and number of retries) iterative approach has been developed. This approach successively doubles the ideal sampling time τ_{si} and runs the simulator until the % OS limit is violated (% OS > the upper limit of the acceptable % OS band, which is 27% in this example, as shown in figure 4.3), halves the last interval and checks again to see if acceptable % OS is achieved; continues halving the interval and increasing/decreasing τ_s until the % OS is within a specified tolerance ϵ_{OS} of the limit (in this illustration the trial values of τ_{sT} were $2\tau_{si}$, $4\tau_{si}$, $8\tau_{si}$ and finally $7\tau_{si}$). By using the above approach, the maximum sampling time for each control closed loop can be determined in an efficient way .

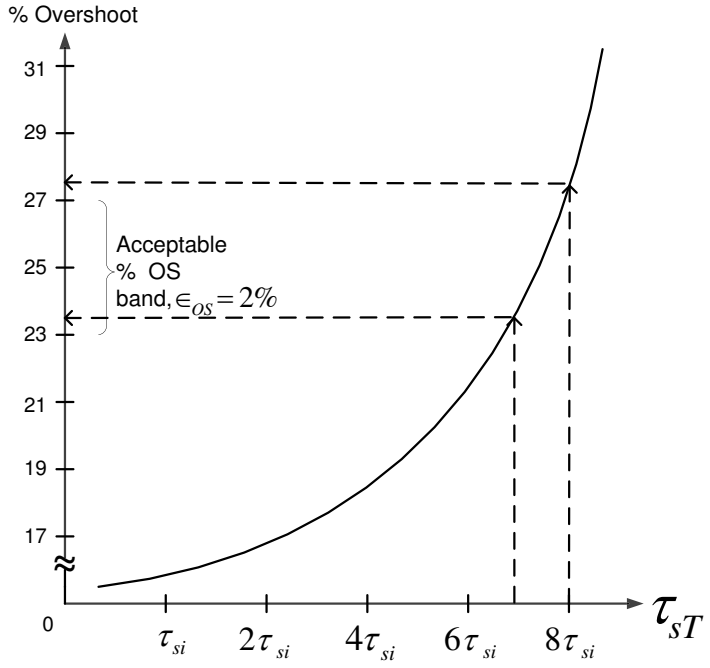

Figure 4.3: Determining the maximum sampling time for the TC loop.

Part 2: Determining both acceptable sampling times

- The Gateway proposes a configuration for the WSN and determines the control path delays, by either using time stamps, ping experiments, or simply multiplying the actual number of hops by an estimated delay-per-hop.

- The WNCSCA accepts the proposed configuration conditionally if its control path delays are less than the τ_{d-max} determined for the benchmark sampling rates, otherwise it rejects it and requests a new proposal.

- The WNCSCA uses the method of successive approximation[1] [2, 20] to find the minimum sampling rate (i.e., τ_{s-max}) for the two loops, as follows:

  - a) Fix the sampling time for the level sensor (τ_{sL}) at its benchmark value and its time delay (τ_{dL}) corresponding to that determined for the proposed configuration; find $\tau_{sT(1)}$ such that acceptable % OS is achieved for that loop;

  - b) Fix the sampling time of the temperature sensor $\tau_{sT} = \tau_{sT(1)}$ and its time delay τ_{dT} corresponding to the proposed configuration; find $\tau_{sL(1)}$ such that acceptable % OS is achieved for that loop;

  - c) Fix $\tau_{sL} = \tau_{sL(1)}$ and τ_{dL} corresponding to the proposed configuration; find $\tau_{sT(2)}$ such that acceptable % OS is achieved for that loop;

  - d) Iterate in this way until the % OS for $\tau_{sL(k)}$ and $\tau_{sL(k-1)}$ differ from the target (25% OS) by less than a tolerance ϵ_L; stop. (We can otherwise apply the stopping criterion to the temperature loop.)

We can illustrate applying the successive approximation method to the level and temperature control loops of our JCSTR simulator as follows: Assume that the Gateway determined the control path delays to 0.17 and 0.1 s for the level and temperature control loops, respectively. Assume that point A in figure 4.4 corresponds

[1]The method of successive approximation (also called relaxation) has been widely used in optimization [20] and solution of boundary value problems [2].

to $\tau_{sT(1)}$ for the temperature control loop which yields 25% OS resulting from fixing the level loop at its benchmark values for the sampling time and the time delay.

By fixing the temperature loop control system to operate at point A, which has $\tau_{sT(1)} = 0.185$ s and $\tau_{dT} = 0.10$ s, and varying the sampling time of the level sensor, τ_{sL}, we can plot the curve of time delay vs. sampling time (the "point A conditions" curve, as shown in figure 4.5). The "Point A conditions" curve cuts the time delay line $\tau_{dL} = 0.17$ s at point B. The temperature % OS for point B has been determined to be 27.2%, which is too high, so we need iterate again.

By repeating the above method, fixing point B which has $\tau_{sL(1)} = 0.108$ s and $\tau_{dL} = 0.17$ s and varying the sampling time of the temperature sensor, and plotting the curves of sampling time and time delay (the "point B conditions" curve), as shown in figure 4.4, we see that the "point B conditions" curve cuts the time delay line $\tau_{dT} = 0.1$ s at point C. By fixing the temperature loop control system to operate at the operating point C which has $\tau_{sT(2)} = 0.16$ s and $\tau_{dT} = 0.10$ s and varying the sampling time of the level sensor, and plotting the curves of sampling time and time delay (the "point C conditions" curve), as shown in figure 4.5, we determine that setting $\tau_{sL(2)} = 0.071$ (point D) the % OS of the two loops are 24.3% and 25%, respectively, which are quite acceptable performance (according to our criterion) and we can stop the iterations.

By using the above strategy the WNCSCA determines τ_{s-max} for both the level loop and the temperature loop in an efficient and fast way. The above strategy will achieve very good performance in terms of simulation time when the WNCSCA controls a process with a higher number of interacting loops, such as the pilot plant simulator that represents an oil production facility, which separates oil well fluids into crude oil, sales gas, and water which has five loops. The plant itself is at the College of the North Atlantic (CNA) [16], and may be used in future PAWS research.

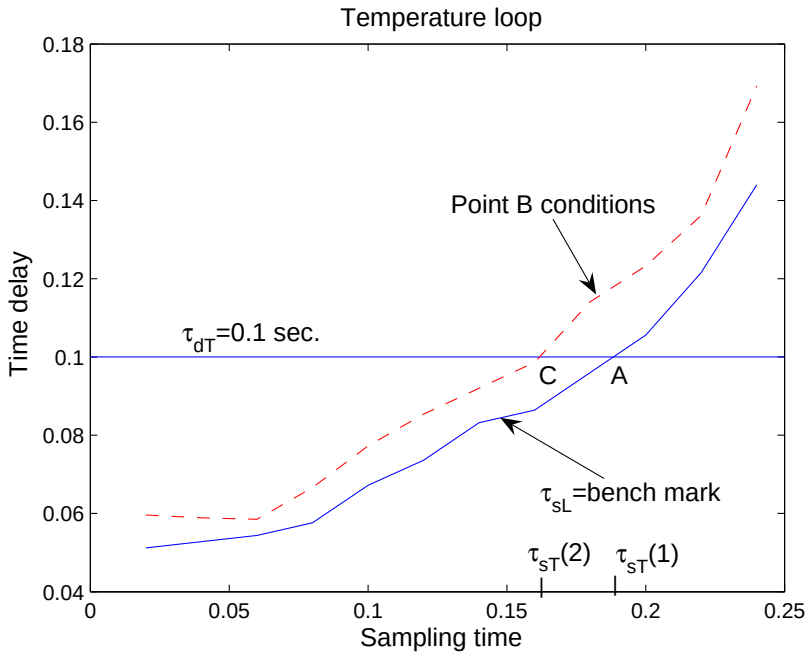

Figure 4.4: Successive approximation for level control.

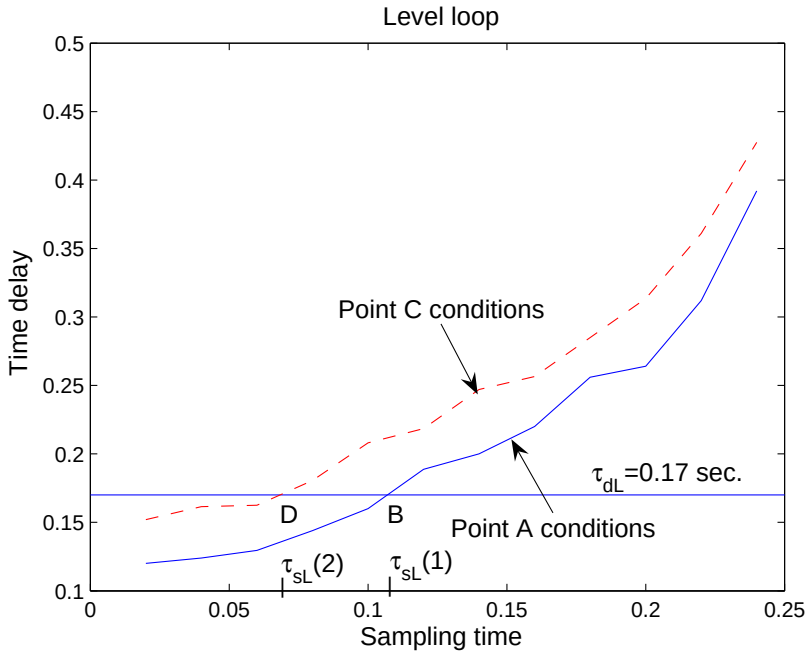

Figure 4.5: Successive approximation for temperature control.

4.3 WNCSCA communication scheme

The WNCSCA communication scheme is the protocol which interfaces between the ICAM system and the Gateway of WINTeR. The interface diagram is portrayed in figure 4.6. The ICAM system is typically installed in the same location as the WSN Gateway, and connected by a local network; it manages and controls the plant. The WSNCSA communication scheme provides the interface, which will be explained here.

In order to simulate a typical process control application and to enable us to study the performance of the closed-loop control system working over the WSN in terms of sampling time and time delay issues, a process simulator for a JCSTR (Jacketed Continuous Stirred-Tank Reactor) has been implemented as described in chapter 3. The two controlled process variables are the mixture level in the reactor and the temperature of the mixture inside the reactor. A more detailed process simulator and WSN interface diagram is shown in figure 4.7. A centralized controller technique has been considered, in which the controller is implemented in a DCS or PC near the Gateway of the wireless sensor network, as depicted. The process simulator JCSTR will run as the Data Generator in the Execution Layer, as shown in figure 3.23 which shows the software architecture of WINTeR.

4.3.1 TCP/IP communication technique

The Transmission Control Protocol (TCP) was used in order to implement the communications between the three entities ICAM, the WNCSCA and the Gateway of WINTeR through a local network and the internet based on figure 4.6. TCP is one of the main protocols used in TCP/IP networks. TCP enables two hosts to create a reliable connection and exchange streams of data through this connection. TCP creates an end-to-end connection (connection oriented), which means that TCP opens a connection between the two end points (two hosts) before actually transmitting data. The data transfer cannot take place before both ends have agreed upon the connection. Data transmitted through the network is often corrupted by errors, such

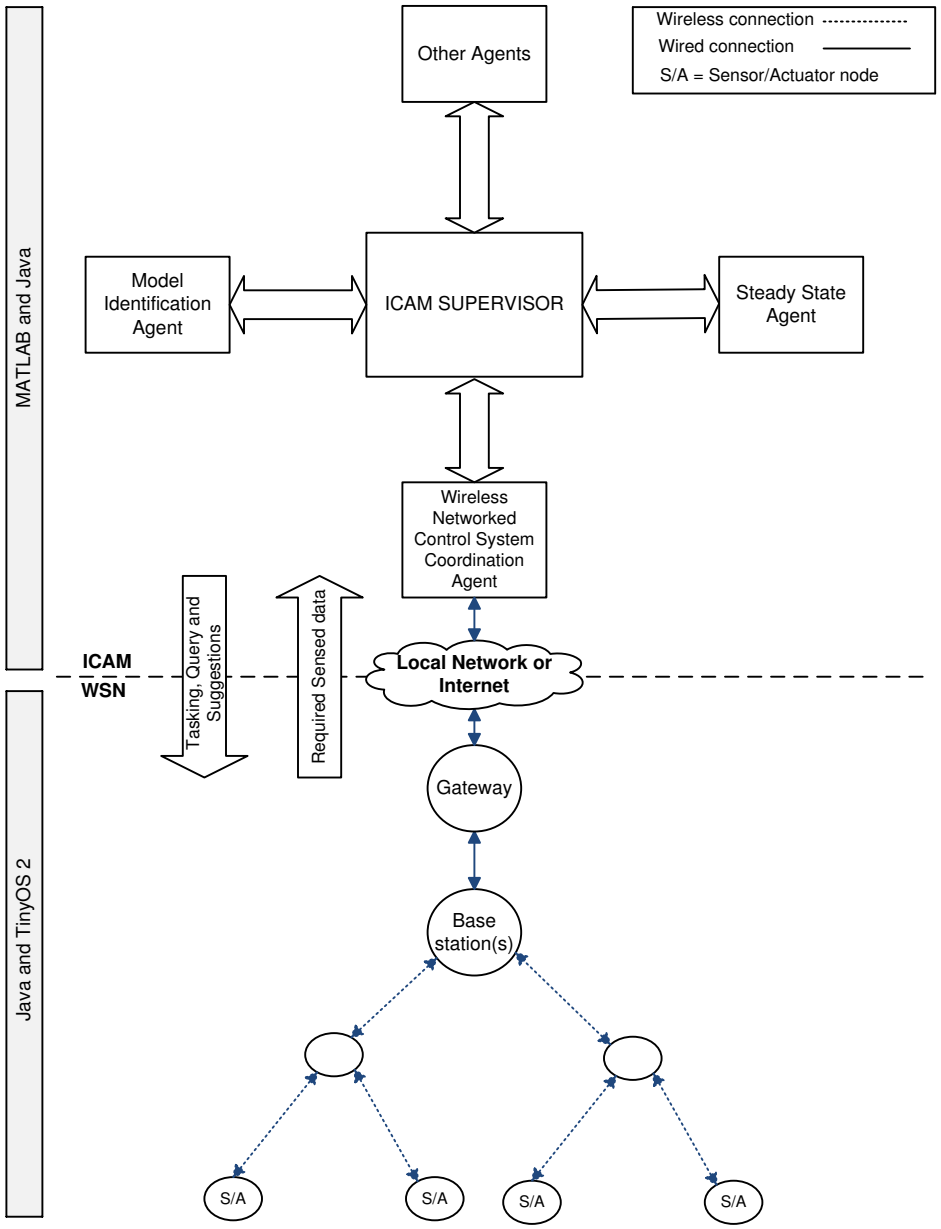

Figure 4.6: Block diagram of WNCSCA interface with ICAM and WINTeR.

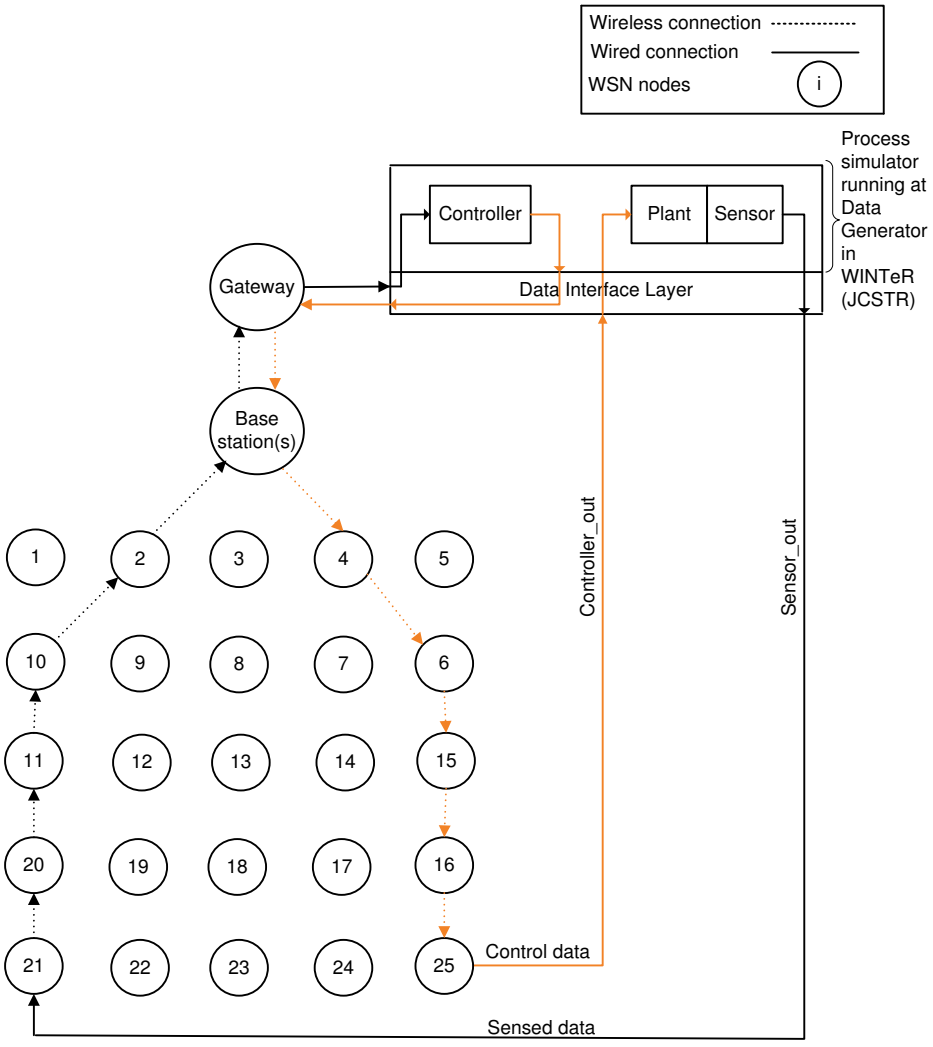

Figure 4.7: Process simulator/WSN interface diagram (one loop).

as missing and duplicated bits. That results in discrepancies between the transmitted data and the data received. However, TCP guarantees data delivery and also guarantees that bytes will be delivered in the same order in which they were sent. TCP uses checksums to protect against errors such as duplicated or missing bits. The transmitter calculates a checksum of the data and transmits the data with the checksum. The receiver will calculate the checksum of the received data with the same algorithm as the transmitter; if errors are found, packets are retransmitted. The reliability of TCP is provided by its acknowledgement property; the receiver machine transmits to the sender machine an acknowledgement that the sent packet was received.

There is another way to send data packets, by using the User Datagram Protocol (UDP). UDP offers a much faster packet transmission rate than TCP, but it is unreliable: It does not guarantee delivery.

The WNCSCA communication scheme was implemented by invoking TCP/IP Java sockets from MATLAB®. Two MATLAB® functions were developed during this research for sending and receiving data. The first function is called Receive_Server, which sets up a TCP receive server. This function will listen to a specific port on the working machine and receive any TCP messages that are sent to that server from a client. After the message is received it will be forwarded to a Check function that will handle the message and execute the required task. The second function is called Send_Client that sets up a TCP sending client. This function will connect to a server and send any TCP messages. These functions were specially implemented based on the nature of this work.

4.3.2 WNCSCA communication scheme

The WNCSCA interfaces with the ICAM Supervisor and the Gateway of the WSN at Cape Breton University (CBU) by using a communication scheme developed during this research. The scenario of that communication scheme is shown in figures 4.8, 4.9, and 4.11. The communication approach is being implemented by invoking TCP/IP

Java sockets in MATLAB®. The communication scheme can be divided into three phases, as follows:

1. **Booting and initialization phase**:

   - **Booting**: during the booting section the three entities (the ICAM Supervisor, the WNCSCA and the WSN Gateway) start working for the first time. The operator starts the operation of the ICAM Supervisor by sending a `Start` message, as shown in figure 4.8. Each entity tries to make sure that the other partners exist and are alive. Therefore, the ICAM Supervisor sends a `Wake Up`[2] message to the WNCSCA to start its operation. Once the WNCSCA gets the `Wake Up` message from the ICAM Supervisor, it sends another `Wake Up` message to the Gateway to start its operation. The Gateway responds by a `Hello` message to the WNCSCA (saying "I am ready") which responds also to the ICAM Supervisor with another `Hello` message (saying "We are ready"). Once the booting process is done, each system knows that the others are ready to start functioning and communicating properly.

   - **Initialization**: in general terms, during the initialization section the WNCSCA commands the Gateway to assess the health of the WSN (see which nodes are operational, check battery levels) and generate a configuration for the WSN, then the WNCSCA will check the proposed configuration based on the performance of the closed-loop control system over the wireless sensor network taking into consideration the sensor/actuator sampling time and the time delays for the packets in the wireless sensor network. Based on that analysis the WNCSCA will accept or reject the proposed configuration from the Gateway; if rejected, the Gateway must try again, decreasing the path delays (number of hops), as suggested by the WNCSCA. The WNCSCA determines τ_{s-max} for each S/A node involved in

[2]Messages are given in text for clarity; in the WNCSCA implementation they are passed as integers.

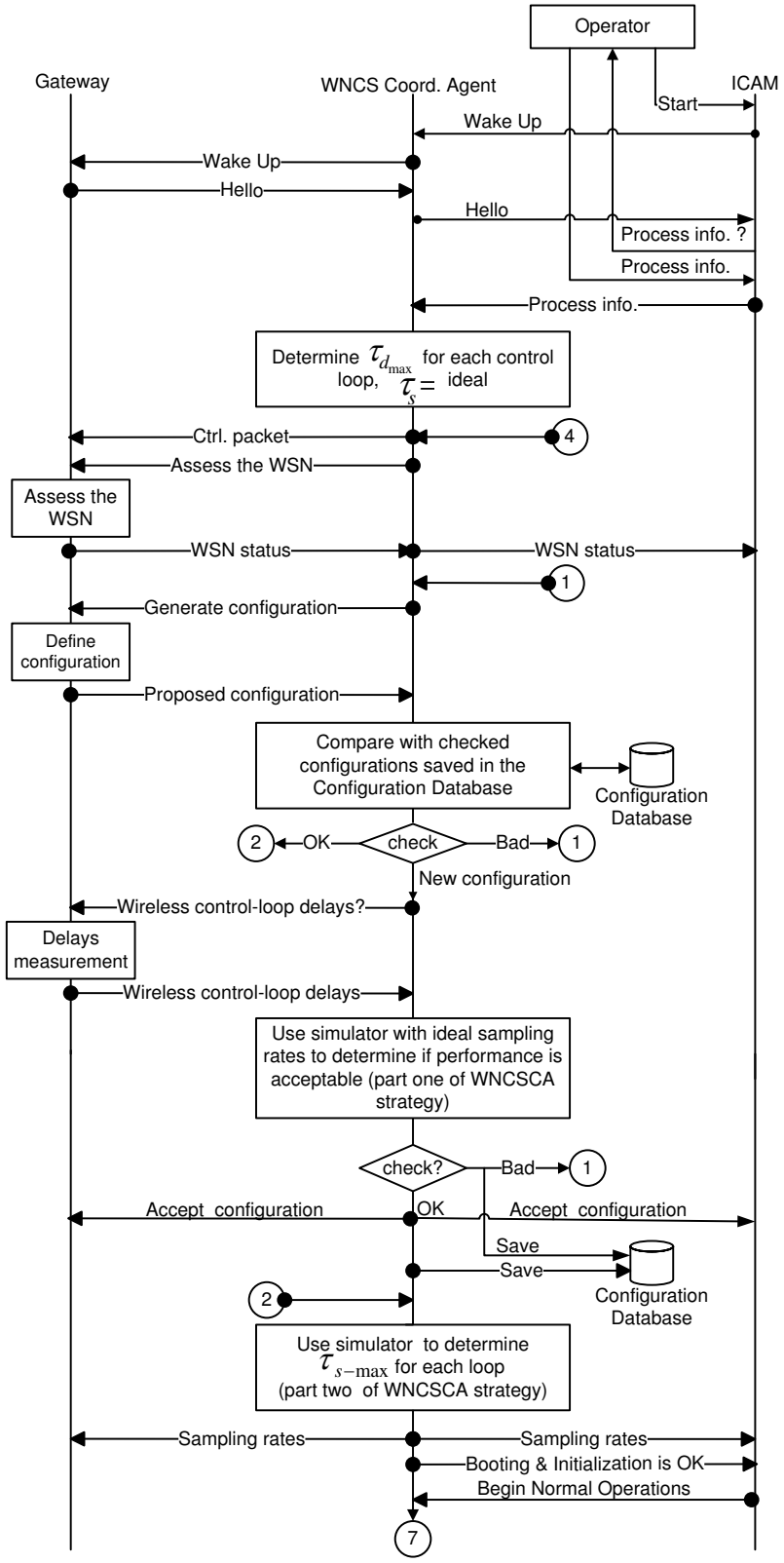

Figure 4.8: Communication scheme for WNCSCA interface with ICAM and the Gateway, Booting and Initialization.

closed loop control. The details of this process are discussed below.

Specifically, the initialization phase can be divided into two stages, as follows:

- **Collecting information**: The WNCSCA gathers information about the WSN nodes' health, battery levels, the WSN configuration, the process simulator to use, sensor and actuator node IDs, and the process set points. After finishing the booting phase the ICAM Supervisor requests from the operator the process information, as represented in the communication scenario figure 4.8 by `Process info.?`. The process information consists of the process simulator (e.g, JCSTR), S/A node information (S/A node IDs distinguish between the sensor node and actuator for each loop), and set points to be used in the simulator, as shown in figure 4.8. At this time, the WNCSCA does simulation experiments to determine the maximum time delay ($\tau_{d_{max}}$) which will cause the wireless control loop(s) performance to become unacceptable by using the percentage overshoot as a criterion for that determination with ideal sampling time τ_s, as illustrated in chapter 3. Then, the WNCSCA sends these maximum time delays to the WSN Gateway along with the S/A node information, in a `Ctrl. packet` which contains the node ID of each sensor and actuator (involved in closed-loop control) and the ideal data rate for each sensor/actuator node, as shown in table 4.1, where the index indicates loop number.

Table 4.1: Control signal packet from WNCSCA to WSN Gateway

Node ID	21	22	25	24
Ctrl nodes	S_1	A_2	A_1	S_2
Data Rate	0.3 s	0.15 s	0.3 s	0.15 s
$\tau_{d_{max}}$	$\tau_{d_{max_1}}$	0	0	$\tau_{d_{max_2}}$

Then the WNCSCA sends a request to the Gateway which asks the Base Station to assess the WSN by using its Energy-Efficient Protocol (checking

the connectivity between each node, the shortest and most reliable paths for the packets and the general health of the WSN). That is represented in the communication scenario as `Assess the WSN`, as shown in figure 4.8. Once the Gateway gets that request, it starts to assess the WSN by running its Energy-Efficient Protocol. Then, the Gateway of the WSN sends the results of that assessment to the WNCSCA as represented in the communication scheme figure 4.8 by `WSN status`. The WNCSCA also updates the ICAM Supervisor with the status of the WSN by relaying this information, as shown. After the Gateway finishes the WSN assessment process, the WNCSCA requests the Gateway to generate a configuration. This is represented in the communication scheme by `Generate configuration`. For example, the configuration in the figure 4.7 can be represented as C_{S_1}=[21 20 11 10 2 0] and C_{A_1}=[0 4 6 15 16 25]. This figure depicts only the first loop for simplicity, but the same (C_{S_2} and C_{A_2}) should be included (detail omitted for clarity).

Once the Gateway gets the WSN configuration from the Base Station, it forwards that configuration to the WNCSCA to check it regarding control loop performance. This is represented in the communication scenario as `Proposed configuration`. But, before carrying out that operation, the WNCSCA compares the proposed configuration with the existing checked configurations saved in the Configuration Database. This operation will save much time: If the proposed configuration already exists in the Configuration Database the WNCSCA can check whether that configuration is `OK Configuration` (meets the required performance of the closed loop system over the WSN) or `Bad Configuration` (violates the performance requirements), and accepts or rejects it accordingly, as shown in the communication scenario in figure 4.8. If it is a `New Configuration` then it needs to be checked, as described below. If the WNCSCA finds that the proposed WSN configuration is `OK Configuration`, the WNCSCA will go to execute point 2, and if the WNCSCA finds that the proposed config-

uration is `Bad Configuration`, the WNCSCA will go to execute point 1. If it is a new configuration, the WNCSCA will check that configuration to see if it is valid with respect to the performance of the control system over the wireless sensor network as shown below in the calculation section. The WNCSCA saves all OK and Bad configurations in the Configuration Database. This process saves much time and effort in checking any proposed configuration.

- **Checking a new configuration:** In the case of a `New configuration`, the WNCSCA needs to know the delays which the packets will encounter over the chain of nodes involved in the closed-loop control system during their journey from the sensor nodes to the Gateway and Gateway to actuators (chain delay) to be able to decide whether to accept that configuration or not according to the performance requirements. The WNCSCA sends a request to the Gateway asking for the wireless control-loop delays (hop-by-hop packet delays from each sensor node to the Gateway and back to its actuator node), as represented in the communication scenario a `Wireless Control-loop Delays?`.

  Once the Gateway gets that request, the Gateway will determine the control-loop delays by running "delay measurement" experiments using time stamps through the assigned paths of the proposed configuration, by performing a "ping" test, or by multiplying the number of hops times the per-hop delay. Then, the Gateway responds to that request by sending the control-loop delays for each control-loop chain as represented by `Wireless Control-loop Delays`. Then the WNCSCA uses the simulator with ideal sampling rates to determine if the performance is acceptable based on the $\%OS$ or not (perform Part 1 of WNCSCA strategy), as described in section 4.2. If the WNCSCA accepts that configuration based on the control loop performance (the $\%OS$ of that configuration is in the acceptable $\%OS$ band), it will send that acceptance to the Gateway and ICAM system. This

is represented in the communication scenario as `Accept configuration`; then the WNCSCA will go to the next step normally. If it rejects that configuration it will go to point 1, as shown in the communication scheme, where it will repeat this part of the scenario by requesting another proposed configuration by the Gateway until finding an acceptable configuration which will not violate the performance requirements of the closed-loop control system. After that the WNCSCA uses the simulator to determine τ_{s-max} for each control loop (perform Part 2 of WNCSCA strategy). Then the WNCSCA sends the τ_{s-max} to the Gateway and the ICAM system, and the WNCSCA starts Normal Operation phase.

2. **Normal Operation phase**:

   This phase is the normal working mode of the WNCSCA, which is focused on monitoring the wireless sensor network and maintaining control loop performance requirements. The logic in this phase continues to loop, as long as the situation remains normal. In this loop, sensed data is sent and the ICAM Supervisor performs a Steady State test, to check whether the controlled variables operate in steady state or in a transient state. The results of that test are sent to the WNCSCA as `Steady state/Transient state`. the WNCSCA checks if this is the first time for the controlled variables to operate in that state (Steady or Transient) or not, in other words, it detects the beginning of the steady state period or the beginning of the transient state period for that controlled variable. This is represented in figure 4.9 by `First time?`. If the answer of that question `No`, the WNCSCA goes to execute point 5, as shown in figure 4.9, but if the answer is `Yes`, the WNCSCA checks the state of the the controlled variables, to determine whether they are now operating in steady or transient state. If the controlled variables are operating in steady state, the WNCSCA determines a **R**educed **S**ampling **R**ate (RSR). At this time the WNCSCA sends an `Open the loops` message to the ICAM Supervisor, which is necessary so that the data rates can be decreased, to reduce the energy consumption. The ICAM Supervi-

sor system responds by an `OK` message to the WNCSCA to inform it that it has already opened the loops, so the Gateway can set the lower data rates accordingly. If the controlled variables are operating in the Transient state for the first time the WNCSCA sets the **N**ormal **S**ampling **R**ate (NSR), and the WNCSCA sends a `Close the loops` message to the ICAM Supervisor, so it can resume closed-loop control of the control variables. Then, the ICAM Supervisor System sends back an `OK` message to inform the WNCSCA that all the loops are closed. Immediately after that step, the WNCSCA sends a massage to the Gateway requesting it to set the proper sampling time, NSR. Once the Gateway finishes this operation (applying the S/A node sampling rates), it sends a `Sampling rate applied` message to the WNCSCA to confirm the ending of that operation; then, the WNCSCA goes to execute point 5. The ICAM Supervisor also invokes the NDDR (Nonlinear Dynamic Data Reconciliation) Agent to improve the quality of the collected signals and the FDIA (Fault Detection, Isolation, and Accommodation) Agent to perform its task during normal operation. For more information about NDDR and FDIA see [8] and [13].

It may be necessary to reconfigure the WSN during normal operation. There are two cases: (1) node death, or (2) new sensor and actuator nodes are installed (control loop(s) added). The Base Station tells the Gateway of the WSN which tells the WNCSCA if there are newly installed nodes or deleted ones, this is represented in the communication scenario by `New nodes installed or removed?` as shown in the bottom of figure 4.9. If the answer is `YES`, the Gateway sends the information about the newly installed or deleted nodes; this is represented in the communication scenario by `Information about newly installed nodes or removed nodes`, as shown in figure 4.9. Once the WNCSCA gets the information about new installed or deleted nodes, it sends that information to the ICAM supervisor, as shown in figure 4.9. Then, if new S/A nodes are installed the Operator must send the information which distinguishes between sensor nodes and actuator nodes to the ICAM system which sends this

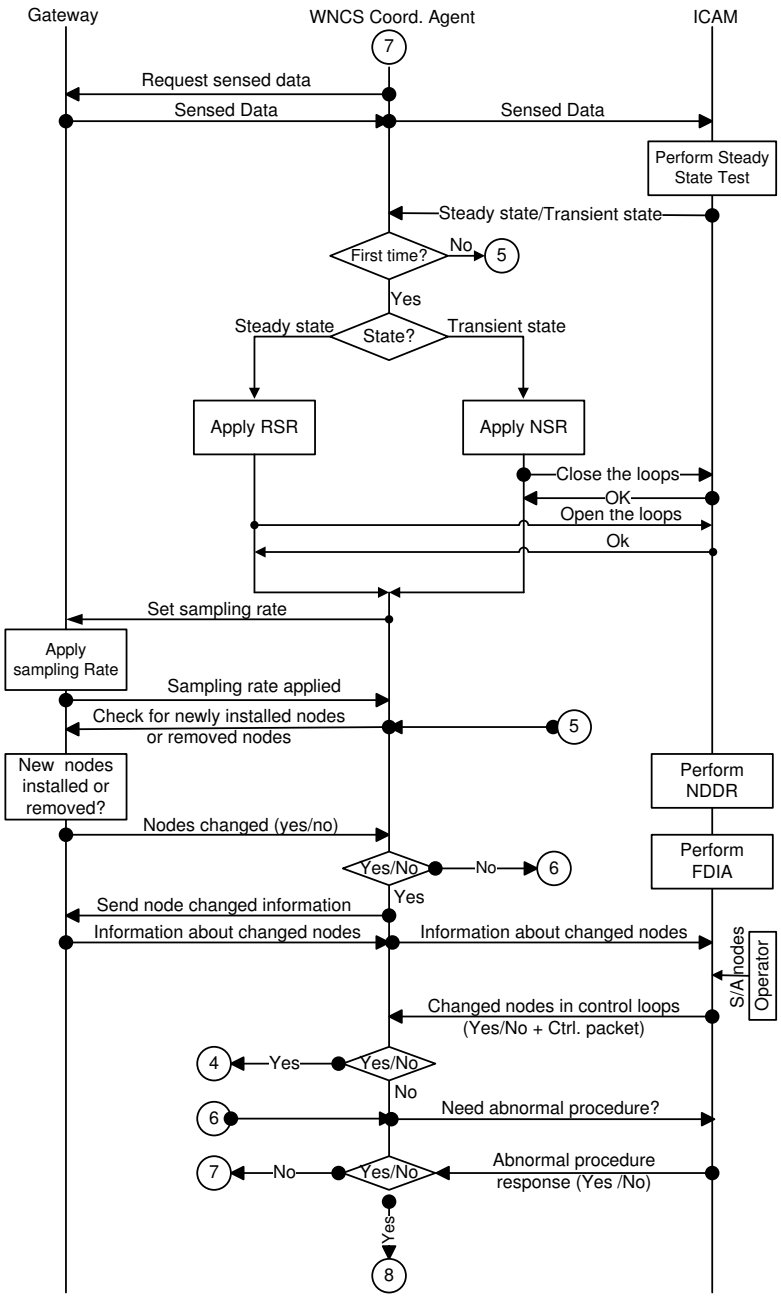

Figure 4.9: Communication scheme for WNCSCA interface with ICAM and the Gateway, Normal Operations phase.

information to the WNCSCA, this is presented in the communication scheme by `S/A nodes` as shown in figure 4.9. As an example of node removal, if node 9 and 10 died or their batteries are too weak to send messages, as shown in figure 4.10, then sensor data must be rerouted, as shown.

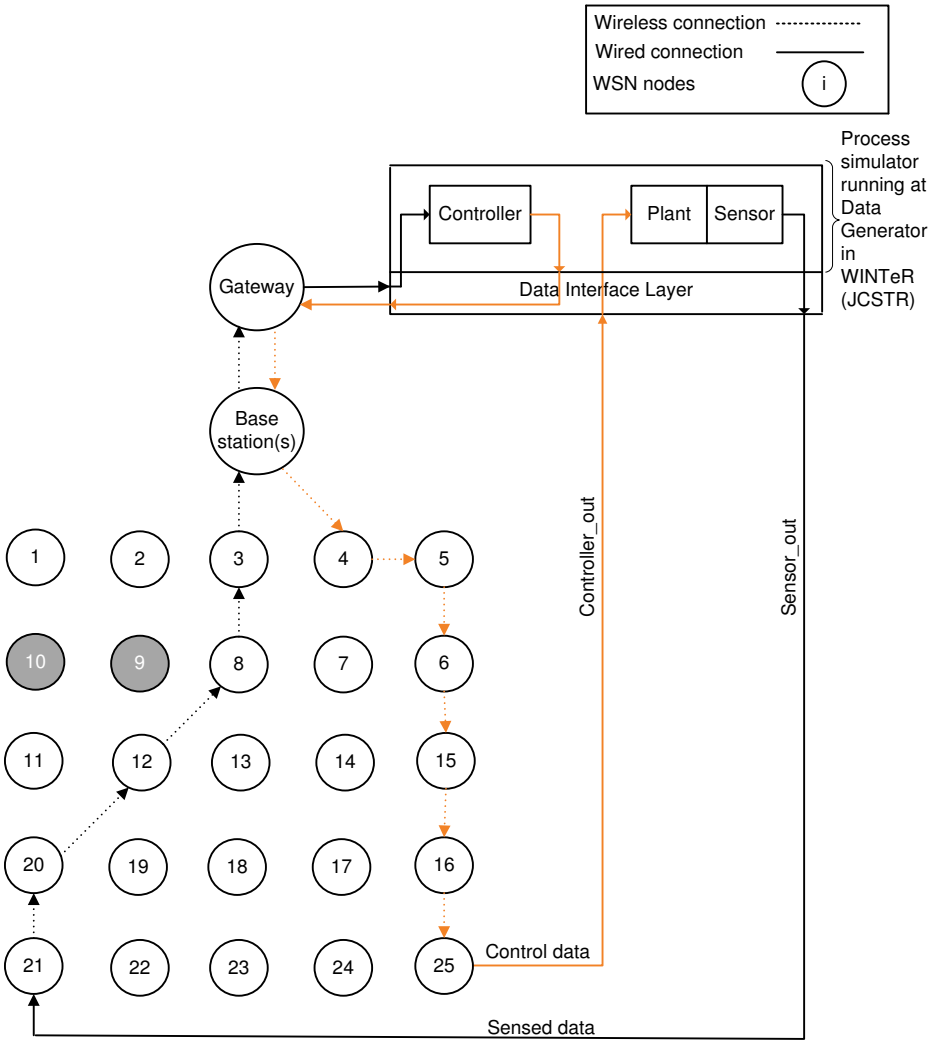

Figure 4.10: Process simulator/WSN interface diagram, two node death case.

If there are installed nodes or removed nodes in the control loops, the WNCSCA goes to execute point 4 in the booting and initialization phase to check the configuration due to the newly installed nodes or removed ones. Once this process is ended, ICAM supervisor decides whether it needs to perform an abnormal procedure or not. If the decision is that the ICAM Supervisor wants

to perform an abnormal procedure, it goes to execution point 8 in the Abnormal Operation phase; figure 4.11; if ICAM decides that it does not need to perform an abnormal operation, it goes to point 7 to repeat the Normal Operation phase again. By "Abnormal Operation" we mean a procedure that may require new S/A node data rates.

3. **Abnormal Operation phase**:

   This phase starts when the ICAM Supervisor decides that it needs to perform an abnormal procedure. The ICAM Supervisor sends a new control signal packet

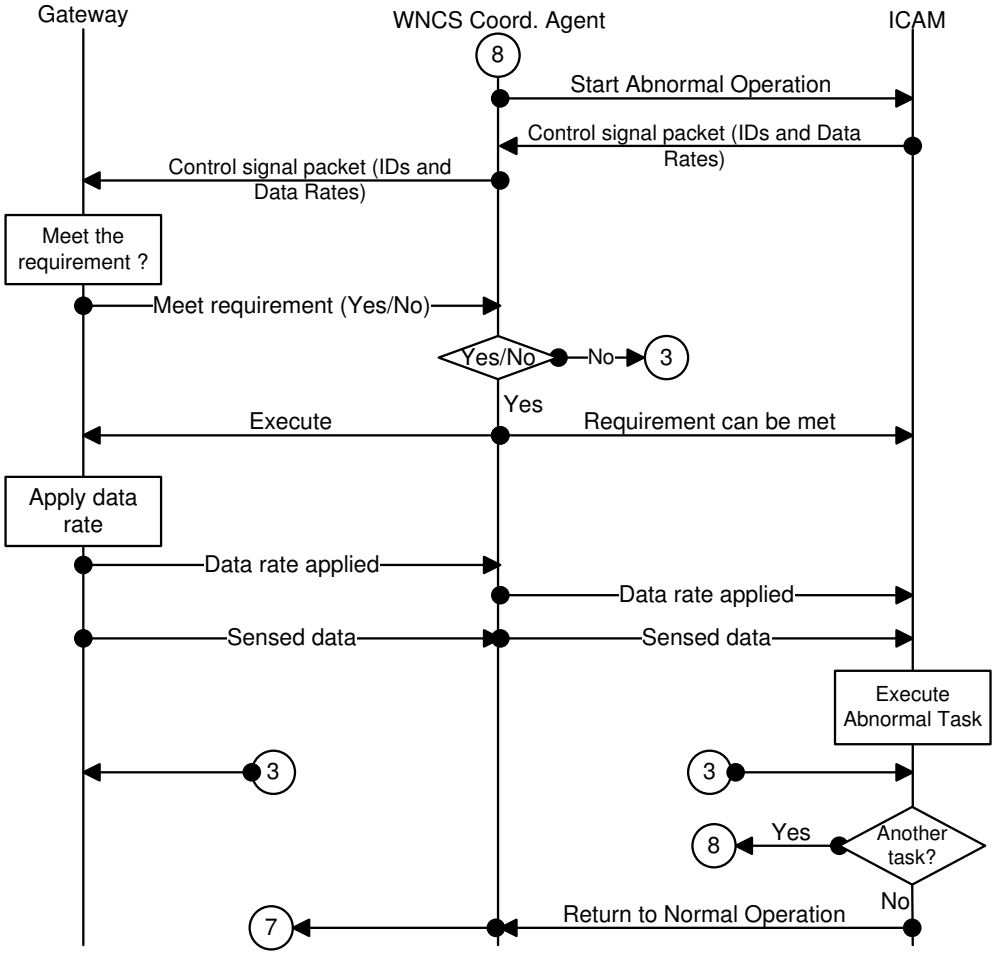

Figure 4.11: Communication scheme for WNCSCA interface with ICAM and the Gateway, Abnormal Operations phase.

to the WNCSCA which contains the address of each S/A node involved in the control loops and a revised data rate for each S/A node, as shown in table 4.2.

Table 4.2: Control signal packet from ICAM system to WNCSCA

S/A Node ID	21	22	25	24
Data Rate	0.3	0.15	0.3	0.15
Ctrl nodes	S_1	A_2	A_1	S_2

This is represented in the communication scenario by `Control signal packet (IDs and Data Rates)`, as shown in figure 4.11. Then, the WNCSCA sends this control signal packet to the Gateway, to apply these sampling rates for each S/A node in the WSN. This operation is done to provide certain higher or lower sensed data rates to support the ICAM Supervisor during performance of an abnormal task such as model identification, before applying the abnormal procedure commands such as providing perturbation signals to the plant. However, the Gateway will decide whether the WSN can support this (by providing the sensed data at an especially high sampling data rate, for example). If the decision of the Gateway is `YES`, the WNCSCA sends a `Requirement can be met` message to the ICAM Supervisor, informing that it can start performing the abnormal procedure, and also an `Execute` massage to the Gateway to apply those sampling rates to the WSN S/A nodes. After the Gateway applies those data rates for each sensor/actuator node, it sends an `Data rate applied` message to the WNCSCA, saying that the WSN nodes are sensing data at the required data rate, as requested, and the WNCSCA relays that message to the ICAM Supervisor. At this time, the Gateway of the WSN collects the sensed data and sends it to the WNCSCA, represented in the communication scheme as `Sensed Data` at figure 4.11. Once the WNCSCA receives the sensed data, it sends this sensed data to the ICAM Supervisor, to support executing the abnormal operation. After that, the ICAM Supervisor decides whether it is

done with abnormal operations or needs to perform another unusual task. If it reports that it is done, the WNCSCA will execute point 7 (execute Normal Operation phase), but if the decision is `NO`, the WNCSCA goes to execute point 8 (repeat the Abnormal Operation phase). If the WSN cannot support ICAM in performing an abnormal operation for any reason, the Gateway sends `NO` (an abnormal operation cannot be executed right now), so WNCSCA will go to execute point 3, as shown in the communication scenario. After that the WNCSCA goes to execute point 7 (repeat the whole Normal Operation phase loop again).

Chapter 5

Contributions, Future Work, and Conclusion

5.1 Contributions

- The main contribution of this thesis is the design and implementation of the WNCSCA which interfaces between the ICAM Supervisory system and the Gateway of WINTeR in such a way as to give the Gateway of the WSN the freedom to manage the WSN without degrading the performance of the closed loop control system by adjusting the data rates, which have a direct impact on the energy consumption of the WSN. This type of interface between an intelligent control system and a WSN to implement a distributed control system is totally **new**.

- The nonlinear simulator model JCSTR was designed and implemented in such a way as to be suitable for embedding in the WINTeR testbed at Cape Breton University.

- The communication functions Receive_Server and Send_Client were implemented and used in communication between WNCSCA, ICAM, and the Gateway. Those functions will be used by other members in ICAM project.

- The communication scheme for the Wireless Networked Control System Coordination Agent between the ICAM system and the Gateway of WINTeR has been developed[1].

- A mocked-up Gateway was implemented in MATLAB® to verify, test, and simulate the WNCSCA.

- WNCSCA forms important enabling technology for using WSNs in process control applications by coordinating between ICAM and the WSN Gateway to allow as much network optimization and flexibility as possible under constraints imposed by the requirements of the WNCS.

5.2 Future work

Although this project software is fully functional, it is important to note that it is in a relatively early stage of development. The WNCSCA has been successfully simulated on a 5×5 mesh wireless sensor network, with a mocked-up WSN Gateway. The WNCSCA will be installed and tested on WINTeR during the next few months.

The strategy of the WNCSCA deals with basic issues in WSNs, path delays and data rates for control data, both of which affect the performance of closed-loop control over a WSN, but that strategy does not consider variation of path delays due to the variation of the WSN load, and irregular arrival of control data (jitter). These phenomena must be dealt with to ensure satisfactory safe control-loop performance in the future.

Due to the departure of some staff members from the CBU PAWS project, there were some delays in developing WINTeR functionality that prevented us from running WNCSCA on WINTeR currently. After the WNCSCA is tested on an actual wireless sensor network with the energy efficient protocol (which is still under development by another PAWS partner), beneficial modifications and additional functions

[1] A U. S. Patent disclosure is being created for the Communication Scheme for Wireless Networked Control System Coordination Agent.

will be identified and incorporated in the WNCSCA.

5.3 Conclusion

In this thesis a unique synthesis of technologies, information and software was developed to construct a WNCSCA for the mission of interfacing between the ICAM system and the Gateway of WINTeR. The WNCSCA has been designed and implemented for coordinating and managing the conflicts between maintaining the performance of control loops, which can be badly degraded by slow data rates and delays in a wireless path, and the usual objectives in managing a wireless sensor network, namely freedom to configure the network to maximize efficiency and to adjust data rates in the network to conserve energy consumption in the network nodes, which are very often battery powered.

The WNCSCA determines the acceptable sampling time for each loop involved in the networked control system based on data delay and the process operating mode, i.e., steady state or transient state, and handles normal and abnormal operation in a way that guarantees the performance of the closed loop control system in different situations for the industrial process.

Although this WNCSCA is fully functional, it is still in a developmental stage. As the ICAM supervisory system project grows and matures with all its other agents, and the energy-efficient routing protocol (which still under development and enhancement) is done this agent will also evolve, eventually offering the Gateway of the wireless sensor network the maximum freedom to configure the network to maximize efficiency and to adjust data rates in the network to conserve energy consumption in the network nodes without violating the performance of the closed loop control over the networked control system.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarsubramaniam and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*. Vol. 38, pp. 393-422, June, 1998.

[2] D. Bertsekas, D. Castañon. Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control*. Vol. 34, No. 6, June 1989.

[3] C. Dorf and R. Bishop. *Modern control systems*. Addison-Wesley Longman Publishing Co., Inc. Boston, Mass., USA. 1994.

[4] C. Fischione, K. H. Johansson, F. Graziosi, and F. Santucci. Distributed cooperative processing and control over wireless sensor networks. *Proc. International Conference on Wireless Communications and Mobile Computing*. Vancouver, Canada, July 2006.

[5] B. Friedland. *Advanced control system design*. Prentice-Hall, Inc., NY, USA. 1995.

[6] P. Kawka and A. Alleyne. Stability and performance of packet-based feedback control over a markov channel. *Proc. American Control Conference*. Minneapolis, Minnesota, USA, June, 2006.

[7] X. Liu and A. Goldsmith. Wireless network design for distributed control. *IEEE/ACM Transactions on Networking*. 2004.

[8] Rocio del Pilar Moreno. Steady state detection. Data reconciliation, and gross error detection: Development for industrial processes. *Master's thesis, University of New Brunswick*. January 2010.

[9] M. Omana and J. Taylor. Fault detection and isolation using the generalized parity vector technique in the absence of an a priori mathematical model. *Proc. IEEE Conference on Control Applications*. Singapore, October 2007.

[10] M. Omana and J. Taylor. Fault detection, isolation and accommodation using the generalized parity vector technique. *Proc. IFAC World Congress*. Seoul, Korea, July 2008.

[11] M. Omana and J. Taylor. Enhanced sensor/actuator resolution and robustness analysis for fdi using the extended generalized parity vector technique. *Proc. American Control Conference*. Minneapolis, Minn., June 2006.

[12] M. Omana. Robust fault detection and isolation using a parity equation implementation of directional residuals. *Master's thesis, University of New Brunswick*. December 2005.

[13] M. Omana. Fault detection, isolation and accommodation using the generalized parity vector technique. *PhD thesis, University of New Brunswick*. October 2009.

[14] http://paws.cbu.ca/ and http://www.ee.unb.ca/jtaylor/PAWS.html.

[15] N. Ploplys, P. Kawka and A. Alleyne. Closed-loop control over wireless network. *IEEE Control Systems Magazine*. Vol. 24, pp. 58-71, June, 2004.

[16] A. Sayda and J. H. Taylor. A Multi-agent system for integrated control and asset management of petroleum production facilites - Part 1: Prototype design and develpoment. *IEEE International Symposium on Intelligent Control*. San Antonio, Texas, September 2008.

[17] A. Sayda and J. H. Taylor. A Multi-agent system for integrated control and asset management of petroleum production facilites - Part 2: Prototype design verification. *IEEE International Symposium on Intelligent Control*. San Antonio, Texas, September 2008.

[18] A. Sayda and J. H. Taylor. Modeling and control of three-phase gravity separators in oil production facilities. *Proc. American Control Conference*. New York, NY, July 2007.

[19] Atalla Sayda. Intelligent control and asset management of oil and gas production facilities. *PhD thesis, University of New Brunswick*. May 2008.

[20] A. Shapiro. On optimality conditions in quasidifferentiable optimization. *SIAM Journal on Control and Optimization*. Vol. 29, pp. 56-68, June, 1986.

[21] J. Slipp, M. Changning, N. Polu, J. Nicholson, M. Murillo and S. Hussain. WINTeR: architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. *Annual Conference on Communication Networks and Services Research*. Halifax, Nova Scotia, Canada, May, 2008.

[22] W. Su and T. Lim. Cross-layer design and optimization for wireless sensor networks. *Proc. of Seventh ACIS International Conference on Software Engineering, Artificial Intellenge, Networking, and Parallel/Distributed Computing*. Geneva, Switzerland, June 2006.

[23] J. Taylor and A. Sayda. Prototype design of a multi-agent system for integrated control and asset management of petroleum production facilities. *Proc. American Control Conference*. Seattle, Washington, June 2008.

CURRICULUM VITAE

- Candidates full name: Hazem Mohamed Saad Ibrahim.
- University attended: Zagazig University, Egypt. Bachelor in Electrical and Computer Engineering, 2006.
- **Patent**: J. Taylor, Hazem M. Saad Ibrahim and J. Slipp. Communication Scheme for a Wireless Networked Control System Coordination Agent. *Patent pending* .
- **Publications**:

  1. J. Taylor and Hazem M. S. Ibrahim . A New, Practical Approach to Maintaining an Efficient Yet Acceptably-performing Wireless Networked Control System. *International Conference of System Science and Engineering, Taipei, Taiwan* (submitted).
  2. J. Taylor, Hazem M. S. Ibrahim and J. Slipp. A Safe Communication Scheme for an Intelligent Wireless Networked Control System Coordination Agent. *2010 IEEE International Conference on Systems, Man, and Cybernetics, Istanbul, Turkey* (submitted).