Stochastic Estimation and Control over WirelessHART Networks: Theory and Implementation

JOONAS PESONEN

Masters' Degree Project
Stockholm, Sweden January 2010

XR-EE-RT 2010:001
Abstract

There is currently a high interest of replacing traditional wired networks with wireless technology. Wireless communications can provide several advantages for process industries with aspect to flexibility, maintenance and installation. The WirelessHART protocol provides a standardized wireless technology for large automation networks that explore wireless communication. However, wireless networks introduce time delays and losses in the communication system, which defines requirements for designing estimators and controllers that can tolerate and compensate for the losses and delays.

This thesis consists of several contributions. First, we develop tools for analyzing the delay and loss probabilities in WirelessHART networks with unreliable transmission links. For given network topology, routing and transmission schedule the developed tools can be used to determine the latency distributions of individual packets and quantify that a packet will arrive within a prescribed deadline. Secondly, we consider estimation and control when sensor and control messages are sent over WirelessHART networks. The network losses and latencies are modelled and compensated for by time-varying Kalman filters and LQG controllers. Both optimal controllers, of high implementation complexity, and simple suboptimal schemes are considered. The control strategies are evaluated on a simulation model of a flotation process in a Boliden mine where the wired sensors of the existent solution are replaced by a WirelessHART network scheduled for time-optimal data collection. Finally, we implement a WirelessHART-compliant sensor on a Tmote sky device and perform real experiments of wireless control on a water tank process.
Keywords: WirelessHART, Tmote, Wireless, Control, LQG, WSN, Kalman, Latency, Markov
Acknowledgements

This master’s thesis was performed in the Automatic Control department at KTH. I would like to thank my examiner Professor Mikael Johansson for his support and advices during this thesis.

I would also like to thank António Oliveira Gonga with all the help with the implementation of Tmote devices and contiki.
Contents

1 Introduction 1

1.1 Problem definition ........................................... 1

1.2 Background .................................................. 2

1.2.1 Control over Wireless Networks ......................... 2

1.2.2 WirelessHART ............................................ 3

1.2.3 The Boliden Ore Refining Process ....................... 3

1.2.4 Related work .............................................. 4

1.3 Outline ....................................................... 6

1.3.1 WirelessHART ............................................ 6

1.3.2 Scheduling ................................................ 6

1.3.3 Estimation and Control .................................... 6

1.3.4 Implementation .......................................... 7

1.3.5 Evaluation ............................................... 7

2 WirelessHART 9

2.1 Time Division Multiple Access (TDMA) ..................... 10

2.1.1 Superframes ............................................. 10

2.1.2 Time synchronization ................................... 11

2.2 WirelessHART Burst Mode Field Devices .................. 11

2.2.1 Burst Trigger Modes ................................... 12

2.2.2 Configuring a Device to Burst Mode .................... 14

2.2.3 Burst Mode Commands .................................. 15

2.2.4 Response Commands .................................... 17
CONTENTS

3  Scheduling

3.1  Unicast  .................................................. 19
3.2  Convergecast Routing .................................. 19
3.3  Modelling of Link Losses .............................. 20
   3.3.1  Bernoulli Random Variables ....................... 21
   3.3.2  The Gilbert-Elliot model .......................... 21
3.4  Latency Analysis ....................................... 22
   3.4.1  Convergecast of a Single Packet to Sink Node .... 24
   3.4.2  Simulation of the end-to-end connectivity ......... 28
   3.4.3  Conclusions ................................... 31
3.5  Scheduling of the Boliden Mining Process .......... 31
   3.5.1  Model of the process with control and latency .... 32
   3.5.2  Latency distribution for the Boliden process ...... 33

4  Estimation and Control  .................................. 39

4.1  Measurement Collection ............................... 39
4.2  System model ........................................... 41
   4.2.1  Sampling of the system ............................. 41
4.3  Time-varying Kalman Filters .......................... 42
   4.3.1  Estimator Design for a Time-varying Kalman Filter ... 42
   4.3.2  Kalman Estimation with Measurement Buffer ....... 43
4.4  LQG Design ............................................ 43
   4.4.1  Controller with Latency Compensation .............. 43
   4.4.2  Static Loop Gain Controller ........................ 45
4.5  Process Model for The Boliden Process ............. 46
   4.5.1  Nonlinear model .................................... 46
   4.5.2  Linearized state-space model ....................... 46
   4.5.3  Sampled system .................................... 47
   4.5.4  Sampled system with loop delay $\tau_k$ .............. 48
4.6  Simulations ............................................. 48
   4.6.1  System response at disturbances .................... 49
   4.6.2  System response for a test sequence ............... 52
4.6.3 System response comparisons for different link qualities
for the test sequence ........................................ 58
4.6.4 The LQG Cost as a function of the link probability ... 61
4.7 Conclusions .................................................. 61

5 Implementation ................................................. 63
5.1 Tmote sky .................................................... 63
5.2 Contiki ......................................................... 63
5.3 NI-DAQ Device ............................................... 64
5.4 Wireless Sensor Extended With WirelessHART Burst Mode
 Commands ......................................................... 64
5.5 Configuration .................................................. 65
  5.5.1 The Water Tank Model ................................... 65
  5.5.2 Reading the sensors ...................................... 67
  5.5.3 The wireless communication ............................ 67

6 Evaluation ....................................................... 69
6.1 Results ....................................................... 69
  6.1.1 Minimal losses ........................................... 70
  6.1.2 \( p_l = 0.5 \) .............................................. 74
  6.1.3 Conclusions .............................................. 77

7 Conclusions and Future Work .................................. 79
7.1 Conclusions .................................................. 79
7.2 Future Work ................................................ 80

A Transition Probability Matrices .................................. 81

B Implementation diagrams ........................................ 83
List of Abbreviations

HART  Highway Addressable Remote Transducer
LQG   Linear Quadratic Gaussian
TDMA  Time Division Multiple Access
ADC   Analog to Digital Converter
DAC   Digital to Analog Converter
Chapter 1

Introduction

Wireless mesh networks can be implemented on systems with a high number of sensors and actuators without the need of long cable installations to each component. WirelessHART provides a secure and robust wireless mesh networking technology to replace existing control technology with wireless technology for systems with high demands of reliable communication. Challenges brought by wireless mesh networking is to compensate for the packet losses and latency caused by disturbances, packet queuing and re-transmissions and still be able to uphold a robust system. In this thesis the WirelessHART standard, latency analysis and control design will be combined to design tools for a wireless networked control system.

1.1 Problem definition

The aim of this thesis is to develop tools for the standard WirelessHART using the operating system contiki. Furthermore, theory on estimation and LQG-control design from related work will be used to create a robust controller for a wireless control system. All parts in the thesis will be combined for implementation on the Quanser water tank process\cite{25}.

This thesis can be divided to the following steps:

1. Latency Analysis: Calculations of end-to-end latency for packet trans-
missions.

2. Control Design: Design of a LQG controller with Latency compensation.

3. Embedded systems programming: Setup devices to communicate with the WirelessHART standard.

4. Implement the steps above for the water tank process

1.2 Background

1.2.1 Control over Wireless Networks

A networked control system can be modelled as in Figure 1.1, where the process has a wireless connection with the controller.

![Figure 1.1: Model of a networked control system](image)

The wireless connections cause network latencies, $\tau^{sc}$ between the sensors to controller connection and $\tau^{ca}$ between the controller to actuators connection. The impact of the delays is that when the control signal is received at the actuator, the control signal has been subjected to a loop delay $\tau = \tau^{sc} + \tau^{ca}$. To compensate for the delay the process model can be modelled with the loop delay.
A general sampled state-space system according Eq. (1.1) - (1.2) is affected with a loop delay \( \tau \) and is modelled with the delay in Eq. (1.3)

\[
x_{k+1} = Ax_k + Bu_k \quad \text{(1.1)}
\]
\[
y_k = Cx_k \quad \text{(1.2)}
\]
\[
x_{k+1} = Ax_k + \Gamma_0(\tau)u_k + \Gamma_1(\tau)u_{k-1} \quad \text{(1.3)}
\]

By modelling the states \( x_{k+1} \) with Eq. (1.3), the system is extended with a state for the previous control signal \( u_{k-1} \).

### 1.2.2 WirelessHART

WirelessHART is an extension to traditional wired HART developed by the HART Communication Foundation and is intended for use in automation processes. With WirelessHART the field devices that consists of sensor nodes, actuator nodes and relay nodes are formed to a mesh network where the communication is performed time slotted with TDMA. WirelessHART uses the 2.4 Gz ISM band with IEEE 802.15.4 phisical layer. The WirelessHART devices are backward compatible with wired HART devices and can be implemented in a traditional HART network.

In this thesis the TDMA superframes are used for scheduling wireless networks and a wireless Tmote sky device is configured to work as a WirelessHART burst mode device for publishing measurement data for a process.

### 1.2.3 The Boliden Ore Refining Process

The Boliden ore refining process is a tank flotation process used to extract minerals from ore. The control problem is modelled as a tank process where the level of the tanks are controlled. Figure 1.2 shows a model of the process, which consists of five cascade coupled tanks. In the figure the tank levels for each tank are given by \( h_1, h_2, h_3, h_4 \) and \( h_5 \). There is a variable inflow
q_{in} to the first tank and the outflows q_1, q_2, q_3, q_4 and q_5 from each tank are regulated with the valve settings u_1, u_2, u_3, u_4 and u_5.

The flotation process can be implemented to a wireless mesh network using five wireless sensors and actuators, two relay nodes and a wireless gateway according to Figure 1.3 where the sensors and actuators are connected to a gateway through two relay nodes. The gateway can include a controller or it can be connected to an automation network. The wireless nodes connected to the tanks have sensors measuring the tank levels parallel with actuators for changing the valves for each tank. The relay nodes pass packets between the tanks and the gateway. All possible transmission links with directions are marked with the dashed arrows in the figure.

In the previous work on the Boliden ore refining process presented in [3,4], the control design has included optimal LQG controllers. As is the aim to design for the networked control systems in this thesis, the control will be designed as time-varying LQG controllers for wireless mesh networks with independent latency for each measurement and control signal.

1.2.4 Related work

Similar work has been done in the WISA research project, where tool-chains for wireless mesh networks has been created, but the work has not been on applications for the WirelessHART standard or with the contiki operating
1.2. BACKGROUND

Figure 1.3: Topology of the suggested mesh network for the mining process system.

The theory part of this thesis included scheduling of networks and estimation and control over wireless networks. Similar latency analysis has been completed in [1] where the same principle has been used as in this work, but as a difference our network contains multiple packets. In [5] - [15] work with estimation and lost packets for LQG control systems has been done, and similar methods are used in this thesis to estimate states and control the process. However, the control design for compensating the delays in this thesis is mostly inspired from the work presented in [16] about real-time control systems with delays.

In this thesis simulations of the Boliden mining process in Aitik [3] are made by combining the latency analysis, estimation and LQG control. Control of the Boliden mining process has been presented in [4] where different control strategies has been evaluated and LQG control has been found as the best solution for the process.

Also in previous projects within the department simulations on wireless mesh networks has been completed and in a paralell thesis a first part of the implementation of the WirelessHART protocol has been performed in [23]. However, within the time frame for this thesis the complete layers for the WirelessHART protocol were not available and therefore the communications
in the implementation was done with simple unicast transmissions without timeslots but the application layer was WirelessHART.

1.3 Outline

This thesis includes both theory and implementation. The work is divided into the following chapters.

1.3.1 WirelessHART

In this chapter the WirelessHART protocol is shortly described with emphasis on the time division multiple access (TDMA) and the wireless devices. TDMA and timeslots are firmly described since it is of big interest in several parts of this thesis.

1.3.2 Scheduling

In this chapter the latency probabilities are calculated analytically for mesh networks with one packet. Latency probabilities are also simulated for networks with multiple packets, and the link probabilities are modelled both with a constant successful transmission probability and with the Gilbert-Elliot model. Also scheduling for the Boliden process is suggested and analyzed.

1.3.3 Estimation and Control

In this chapter two different estimators are derived to estimate the states from measurements for a given system. The first estimator is a buffered estimator with time-varying Kalman gain and the second is a simple Kalman estimator with static gain.

The system matrices for a given system are sampled with aspect to the time delays in the system. Two controllers are designed matching the two Kalman filters, one with static loop gain and one time-varying controller.
The two controllers with the Kalman estimators are then evaluated with simulations for different combinations of the link probabilities for the Boliden mining process. The results can also be compared with simulation results in [4] and the same test sequence is simulated.

1.3.4 Implementation

A wireless Tmote sky device is configured to work with WirelessHART burst mode commands for publishing measurements on a water tank process, which is controlled with the estimation and control from Chapter 4.

1.3.5 Evaluation

The implemented process is evaluated for a few cases with losses and disturbances. The architecture of the system is also evaluated.
Chapter 2

WirelessHART

Figure 2.1: WirelessHART mesh network with 7 field devices
2.1 Time Division Multiple Access (TDMA)

2.1.1 Superframes

TDMA and channel hopping is used in WirelessHART to control transmissions between the devices included in a WirelessHART network. All devices in a WirelessHART supported network must be able to source, sink and route packets. TDMA uses time slots of 10 ms each and a series of time slots form a repeating superframe. If a device is scheduled for the present time slot, the device can access the network for either transmitting or listening for data packets. All devices in WirelessHART supports multiple superframes, which makes it possible to use different routing schemes at different times for the mesh network (one for data collection and one for publishing data, the active superframe is given by the network manager). Multiple transmission links can be active on the same time slot by using different frequency channels (channel hopping). Each transmission link is defined by a sender and a receiver, the active transmission links are controlled by the network manager, which controls the routing scheme for the field devices.

Each time slot in a TDMA superframe consist of specific time interval when a scheduled source node transmits the data and when the transmission is completed the device waits for a response from the destination device for a response message. Failed transmissions are responded with an error code and the packet is re-transmitted at the next scheduled time slot for the transmission link (Usually the next superframe).

Figure 2.2 shows a TDMA superframe with scheduled links for each time slot. Different channels can be used for scheduling multiple links for the same time slot. The superframes for a WirelessHART network are repeated continuously (Figure 2.3) and failed transmissions are re-transmitted by the repeating superframe instead of repeating a time slot if the transmission failed.
2.2 WIRELESSHART BURST MODE FIELD DEVICES

2.2.1 Time Synchronization

For TDMA transmissions to work properly it is required that the devices are synchronized. The timeslots for receiving devices consists of time intervals to receive a message and there is an ideal starting time to start receiving. The starting time is observed and by this way the time can be synchronized by calculating the time error between the actual starting time and the ideal starting time. The network manager selects neighbors to each device, which the devices synchronize their time with.

The Medium Access Control (MAC) layer keeps the slot synchronization and is also responsible for managing tables of neighbors, superframes and transmission links all provided from the network manager. The MAC layer also sets the device to listen for or transmit data according to the superframe.

2.2 WirelessHART Burst Mode Field Devices

A WirelessHART field device can be set to burst mode. While a field device is in burst mode it will publish data/messages with the burst mode configuration until it receives a command to stop. As an addition to the required
commands for general WirelessHART field devices, a burst mode field device has to support the burst mode commands. In figure 2.4 a device is first set to burst mode with the configuration sequence after which the device publishes data with the burst configuration.

Figure 2.4: Function of the burst device

### 2.2.1 Burst Trigger Modes

The burst trigger mode provides the WirelessHART burst mode device with its basic functionality. The device will publish messages while in burst mode according to the burst trigger mode with the parameters from the burst mode configuration. The burst trigger modes are defined in the WirelessHART standard document[18]
Continuous

In continuous burst trigger mode the field device transmits burst messages with the maximum update period, until it receives a command to stop.

Windowed

In windowed burst trigger mode a trigger level defines a symmetric window around the previous variable value. If the variable value exceeds the lower or upper window limit after that the minimum update time has exceeded, then a burst message is sent. If the variable value is within the window after the maximum update time has exceeded, a burst message is sent also.

![Windowed trigger mode diagram](image)

Figure 2.5: Windowed trigger mode

Rising

In rising burst trigger mode the device transmits with the maximum update period while the variable value is below a specified trigger level. When the variable value exceeds the trigger level the device transmits with the minimum update period.
Figure 2.6: Rising trigger mode

Falling

Falling burst trigger mode works as the rising mode, but transmits with minimum update period when below the trigger level and with the maximum update period when above the trigger level.

2.2.2 Configuring a Device to Burst Mode

A master device configures a connected field device to burst mode by sending the following commands:

1. Command 108 Write Burst Mode Command Number, selects the burst mode response command. The response command is selected between commands: 1, 2, 3, 9 and 33.

2. If command 9 or 33 is chosen in step 1, the device variables and unit codes are written to the data slots using command 107, Write Burst Device Variables. For the other commands the variables are assigned to predefined as the primary, secondary, tertiary and quaternary variables.

3. Command 103 Write Burst Period, sets the minimum and maximum update period for the burst messages sent by the device.
4. Command 104 *Write Burst Mode Configuration*, selects the burst trigger mode.

5. Command 109 *Burst Mode Control*, activates the field devices to burst mode

After the configuration the parameters can be changed by the master device by sending the corresponding commands.

![Figure 2.7: Configuration sequence for a gateway setting up a sensor device for continuous burst mode with sampling time 1.0 s and response command number 33](image)

**2.2.3 Burst Mode Commands**

According to the specifications documents [18][19] a WirelessHART burst mode field device must support the commands 103-105 and 107-109.
CHAPTER 2. WIRELESSHART

Command 103 Write Burst Period

Sets the minimum and maximum update period for the burst messages. The minimum period must be set to less or equal to the maximum and allowed values for both update periods are 0.100, 0.250, 0.500, 1.000, 2.000, 4.000, 8.000, 16.000, 32.000 and any value between 60-3600.\[18\]

Command 104 Write Burst Trigger

Sets the burst trigger mode, the possible modes are Continuous, Windowed, Rising and Falling. See section 2.2.1

Command 105 Read Burst Mode Configuration

Reads the whole configuration of the burst mode device, which includes the burst period, burst trigger mode, burst mode command number and device variable information.

Command 107 Write Burst Device Variables

Writes the variables and unit codes to each data slot for the variables. This command is only used when command 9 or 33 is used as the burst response command for command 108.

Command 108 Write Burst Mode Command Number

Selects the response command the field device will transmit during burst mode. The possible response commands are 1, 2, 3, 9 and 33. Each command differ in variable setup.

Command 109 Burst mode Control

Toggles the burst mode on/off and defines the type of connection.
2.2.4 Response Commands

In addition to the burst mode commands the field device must support the following commands for reading the variables. Suitable command is selected depending on the sensor configuration.

**Command 1 Read Primary Variable**

Reads the defined primary variable and unit code.

**Command 2 Read Loop Current and Percentage of Range**

Reads the loop current and its percentage of range.

**Command 3 Read Dynamic Variables and Loop Current**

Reads the loop current and the predefined variables with unit codes. There can be up to four variables, each variable are defined as primary, secondary, tertiary and quaternary by the device and cannot be changed by the master device (as with command 107 when command 9 or 33 is used as response command).

**Command 9 Read Device Variables with Status**

Reads the value and status of up to eight device variables. The variables and unit codes are assigned to data slots with command 107.

**Command 33 Read Device Variables**

Reads the value of up to four device variables. The variables and unit codes are assigned to data slots with command 107.

In this thesis command 33 is used as the response command, since the sensor variables can be configured to selected data slots and the command supports up to four variables and makes it suitable to use with a Tmote sky device.
Chapter 3

Scheduling

In this chapter given topologies and routing schemes for wireless networks are analyzed. The wireless links transmissions are modelled by random variables using two different models: Bernoulli and the Gilbert-Elliot model. The analysis is done numerically and by Monte-Carlo simulations. The results for the analysis shows the probability for the latencies for two operations: collecting data from field devices to a gateway and sending data from the gateway to field devices.

3.1 Unicast

In unicast, the transmissions are made between only two devices, one sender and one receiver. However, there can be several active transmission links as shown in Figure 3.1 where a unicast transmission is made between the nodes seven to five and between nodes six to three.

3.2 Convergecast Routing

Convergecast routing uses unicast transmission between the nodes to route packets to a destined receiver in a meshed network. All packets are routed towards a specific node. This is typically used when a gateway node collects
sensor data from a wireless sensor network. Figure 3.2 shows the routing
direction, all routing is made towards node seven.

Figure 3.2: Convergecast routing, node 7 collects data from the network

3.3 Modelling of Link Losses

The link losses are modelled by random variables. The losses are modelled
both by Bernoulli random variables and the Gilbert-Elliot model.
3.3. MODELLING OF LINK LOSSES

3.3.1 Bernoulli Random Variables

Using the Bernoulli random variables a successful transmission probability for a link sending from node $i$ to $j$ is given by (3.1) and the probability for a failed transmission is given by (3.2).

\begin{align*}
p_{ij} &= p \quad \text{(3.1)} \\
\bar{p}_{ij} &= 1 - p \quad \text{(3.2)}
\end{align*}

3.3.2 The Gilbert-Elliot model

With the Gilbert-Elliot model the state of each node are modelled by two possible states, a good state and a bad state. In the good state a scheduled transmission of a packet is successful with no loss and in a bad state there is a packet loss. When the node is in a good state it will remain in the good state with the probability $p$ or it can go to the bad state with the probability $1 - p$. If the node is in the bad state it will remain in the bad state with the probability $q$ or it can go to the good state with the probability $1 - q$.

\begin{figure}[h]
\centering
\begin{tikzpicture}
  \node [circle,draw] (no_loss) at (0,0) {No loss};
  \node [circle,draw] (loss) at (2,0) {Loss};
  \draw [->] (no_loss) edge [bend right] node [above] {1-p} (loss);
  \draw [<-] (loss) edge [bend right] node [below] {1-q} (no_loss);
  \draw [->] (no_loss) edge [loop above] node [above] {p} （无损失）;
  \draw [<-] (loss) edge [loop above] node [above] {q} （丢失）;
\end{tikzpicture}
\caption{The Gilbert-Elliot model for each node}
\end{figure}
3.4 Latency Analysis

In a wireless mesh network packets are lost in the wireless transmissions, which creates delays in the end-to-end packet delivery. The end-to-end latency is the time for an operation to complete, as collecting data from sensors to the gateway (Convergecast). The end-to-end connectivity for a packet can be modelled as a probability function depending on the latency. In this section latency calculations are made for a wireless mesh network with given topology and routing scheme. The transmissions are modelled by constant successful transmission probabilities and by the Gilbert-Elliot model. The probability function for the end-to-end connectivity is calculated by using Markov chain models for single packet transmissions and simulated for multiple packet transmissions. By simulating for multiple packets, different packet priorities can be modelled and the specific packets can be tracked and identified from which node the packets originated. Simulating the end-to-end latency also makes it possible to use more complex protocols for the transmissions. The topology and routing scheme used is shown in Figure 3.4 and Figure 3.5 - 3.7.

Figure 3.4: Topology of the mesh network
3.4. LATENCY ANALYSIS

Figure 3.5: Routing scheme timeslot 1

Figure 3.6: Routing scheme timeslot 2

Figure 3.7: Routing scheme timeslot 3
3.4.1 Convergecast of a Single Packet to Sink Node

The methodologies used in this section are the same as presented in [1]. The nodes in the network can be described as: \( N = 1, \ldots, 8 \) with transmission links: \( L = (i, j) \mid i, j \in N \) (transmissions from node \( i \) to \( j \)). The timeslots represent WirelessHART timeslots, each timeslot are of length 10 (\( ms \)) and repeating superframes consists of several timeslots. The transmissions between the nodes are made by TDMA routing as in the WirelessHART protocol. For the failed transmissions, re-transmissions are made by repeating the superframe.

Bernoulli model

The packet occurrence in each node can be described by eight states (3.3) that describe if the packet is present at a specific node at timeslot \( t \).

\[
S_1(t), S_2(t), \ldots, S_8(t)
\]  

(3.3)

The probabilities for the states are then described by the probability for the packet being present at the nodes at timeslot \( t \):

\[
p(t) = [P(S_1(t)) \ P(S_2(t)) \ ... \ P(S_8(t))]^T
\]

(3.4)

The probability for a successful transmission is given by \( p_{ij}(t) \) and the probability for a loss is given by a failed transmission: \( \bar{p}_{ij}(t) = 1 - p_{ij}(t) \). The probability for the packet to remain in the same node for the next timeslot is equal to the probability of a failed transmission and is given by: \( p_{ii}(t) = \bar{p}_{ij} = 1 - p_{ij}(t) \) (one node transmits to only one node at each timeslot for the given routing scheme). The link probability is modelled as \( p_{ij}(t) = 0 \) for non scheduled links for the current timeslot \( t \).

The transition probability matrices for the Markov chain at timeslot \( t \) can then be expressed by the \( N \times N \) matrix \( P(t) = [p_{ij}(t)]^T \). 

3.4. LATENCY ANALYSIS

Only one packet is sent in the network and the probability for the packet being at the node k \((k \in N)\) at the time \(t_d\) is given by the k:th element in vector \(p(t_d)\), which is given by the matrix equations:

\[
p(t_d) = P(t_d)...P(T)P(T - 1)...P(1)p(0)
\] (3.5)

where \(P(t)\) is the transition probability matrix at timeslot \(t\) and the transition probability matrix for the repeating superframe with the given routing scheme can be expressed as:

\[
P(T) = P(T)P(T - 1)...P(1) = P(3)P(2)P(1)
\] (3.6)

And because of a repeating superframe with the WirelessHART standard we have:

\[
P(T+h) = P(2T+h) = P(3T+h)... \Rightarrow P(3) = P(3+3) = P(6+3)...
\] (3.7)

\(p(0)\) is a vector with the a:th element equal to one and all the others zero and represents the initial condition for the network and describes from which node the packet in the network originated from at time \(t_0\) (In this case the packet originates in node \(a\)).

Implementation in MATLAB with starting node \(a = 1\) and sink node \(k = 8\) and all scheduled link probabilities \(p_i,j = 0.8\) gives the probability function shown in Figure 3.8.
CHAPTER 3. SCHEDULING

Figure 3.8: End-to-end probability function for the given routing scheme for a single packet originating in node 1 and being transmitted to node 8

The Gilbert-Elliot model

If a packet is at a node at time $t$ at a good or bad state, the current state of the node is denoted by:

$$S_1^G(t), S_2^G(t), ..., S_8^G(t), S_1^B(t), S_2^B(t), ..., S_8^B(t)$$  \(3.8\)

The probability of the states at timeslot $t$ is then given by:

$$p(t) = [P(S_1^G(t)) \ P(S_2^G(t)) \ ... \ P(S_8^G(t)) \ P(S_1^B(t)) \ P(S_2^B(t)) \ ... \ P(S_8^B(t))]^T$$  \(3.9\)

For each scheduled transmission from node $k$ at time $t$ the probability for a successful transmission is $p_k(t)$ if the node is currently in a good state and
3.4. LATENCY ANALYSIS

1−q_k(t) if the node is in a bad state. The probability for the packet to remain in node \( k \) is \( 1−p_k(t) \) if the node goes from a good state to bad and \( q_k(t) \) if the node is at a bad state and remains in that state. For non scheduled links at time \( t \) the node can switch state with the probabilities \( 1−p_k(t) \) or \( 1−q_k(t) \) and if there is a packet in the node it will remain in the node.

When the node \( i \) in a given state is scheduled to transmit to node \( j \), the probabilities for the transmissions in the different cases are:

- The node is in a good state: \( p_{ij}(t) = p_i(t) \) for a successful transmission and \( \bar{p}_{ij} = 1−p_i(t) \) for the transmission to fail due to the node going to a bad state.

- The node is in a bad state: \( p_{ij}(t) = 1−q_i(t) \) for the node to go to the good state and successfully transmit and \( \bar{p}_{ij} = q_i(t) \) for the transmission to fail due to the node staying in a bad state.

The probability for the states at time \( t \) is then given by Eq. \((3.5)\), but with the additional states \( p(0) \) is now a 16 × 1 vector with the a:th element equal to one and the rest is zero for the packet originating in the a:th state. The transition probability matrix is of size 16×16 because of the additional states.

The end-to-end probability for the packet to reach node \( b \) at time \( t \) is given by the union of the probabilities of the packet being in a good or bad state in node \( b \):

\[
P(S^G_b(t) \cup S^B_b(t)) = p_b(t) + p_{b+8}(t) \tag{3.10}
\]

Implementation in matlab from starting node 1 and sink node 8 with all \( p_k = 0.8 \) and \( q_k = 0.4 \) gives the following \( P(S^G_8(t) \cup S^B_8(t)) \):
3.4.2 Simulation of the end-to-end connectivity

Simulations in MATLAB are used to acquire the end-to-end latency for several attempts by randomizing losses. The latency values are then gathered for a distribution function for the end-to-end latency, which is used to get the statistic percentage of each latency value and integrated to acquire the probability function.

**Simulation algorithm**

The simulation of the end-to-end latency is done by repeating steps 1-3 $N$ times, $N$ should be chosen to a large number to increase the accuracy of the simulation, however, an increased $N$ requires more processing power. For each simulation the routing scheme is iterated until the packet reaches its final destination or the maximum timeslot is reached and the packet is
3.4. LATENCY ANALYSIS

considered as lost.

1. At each timeslot for the routing scheme, for each active transmission link and if there is a packet in the sending node specified by the active transmission link a random number between 0 and 1 is generated, and if it is greater than the specified successful link probability for the simulation then the transmission is considered as failed. If the transmission is successful then the packet is moved to the next node specified in the routing scheme.

2. If there are more then one packets on a node at a time, the packet that first arrived to the node is sent out first.

3. When the state of the network satisfies the final state, the total number of timeslots is stored and the next simulation is begun.

The statistics of the simulations are then put in to a distribution function. By integrating the distribution function, the probability function for the end-to-end connectivity is acquired.
CHAPTER 3. SCHEDULING

Single packet simulations

Figure 3.10: Simulation of the end-to-end probability with constant link probabilities

Figure 3.11: Simulation of the end-to-end probability with the Gilbert-Elliot model
3.4.3 Conclusions

The probability using the Gilbert-Elliot model is lower because if a node goes to a bad state then the probability that it will remain in the bad state is higher than the probability for a loss for the constant link probability. While for the markov chain model only depends on the link probabilities and one failed transmission does not affect the probability of the next one. The simulated probabilities match well with the calculated and by increasing the number of simulations the results will be even closer to the calculated results.

3.5 Scheduling of the Boliden Mining Process

The proposed routing scheme for the mining process is shown in Figure 3.12 and 3.13. The sensors-controller routing makes one superframe and the controller-actuators routing makes another superframe. The topology for the network is guessed solution and includes two relay nodes that increase reliability for an implementation if some of the field devices are blocked by physical structures. The routing scheme is designed time optimal for completing an operation (completing convergecast). Re-transmissions are made by repeating superframes as shown in Figure 3.14.

The latency analysis is simulated to be able to use complex routing and queues. The simulated latency distribution with $N = 3000$, $p_{\text{link}} = 0.9$ and $t_{\text{max}} = 20$ (timeslots) for the network in Figure 1.3 is shown in Figure 3.16 and integrating the latency distribution gives the probability function and is shown in Figure 3.17.
3.5.1 Model of the process with control and latency

In Figure 3.13, S and A represent the sensors and actuators at nodes 1, 2, 3, 4, and 5:

\[
S = \{S_1, S_2, S_3, S_4, S_5\} \quad (3.11)
\]
\[
A = \{A_1, A_2, A_3, A_4, A_5\} \quad (3.12)
\]

And the different latencies from the sensors to gateway and gateway to actuators:

\[
\tau_{sc_k}^k = \{\tau_{sc_1}^k, \tau_{sc_2}^k, \tau_{sc_3}^k, \tau_{sc_4}^k, \tau_{sc_5}^k\} \quad (3.13)
\]
\[
\tau_{ca_k}^k = \{\tau_{ca_1}^k, \tau_{ca_2}^k, \tau_{ca_3}^k, \tau_{ca_4}^k, \tau_{ca_5}^k\} \quad (3.14)
\]
3.5. **SCHEDULING OF THE BOLIDEN MINING PROCESS**

3.5.2 **Latency distribution for the Boliden process**

From the distributions it is possible to determine the possible latencies for the sensors-controller and controller-actuators connection. This can be used to determine the possible combinations for the latencies $\tau_i^{sc}$ and $\tau_i^{ca}$ for calculations of the latency dependent system matrices in the next chapter. The amount of the combinations has to be reduced in order to reduce memory usage and processing power. The latency distribution also shows the minimum and maximum end-to-end latency which can be taken in account when deciding the buffer length for a buffered controller.

The latency analysis is simulated by using WirelessHART timeslots and with constant link probability $p_l = 0.9$ for successful transmissions for each
Figure 3.14: The repeating superframes for the Booliden process

Figure 3.15: Model of the control system

link. At the relay nodes multiple packets can occur at a time and are put in queue, first packets received in the node are sent out before the later arriving packets.

End-to-end Probability for Sensors-Controller

The probability of the end to end latency is integrated from the latency distribution shown in Figure 3.16 and the end-to-end probability is shown in Figure 3.17.

End-to-end Probability for Controller-Actuators

The packets are set to be routed through a specific relay node, packets to node 1,2 and 3 are routed through relay node 6 and packets to node 4 and 5 are routed through node 7. The probability of the end to end latency is integrated from the latency distribution shown in Figure 3.18 and the end-to-end probability is shown in Figure 3.19.
3.5. SCHEDULING OF THE BOLIDEN MINING PROCESS

Figure 3.16: The simulated distribution of the latency for sensors to controller routing of the packets
Figure 3.17: The probability for the end to end latency for sensors to controller routing of the packets
3.5. SCHEDULING OF THE BOLIDEN MINING PROCESS

Figure 3.18: The simulated distribution of the latency for controller to actuators routing of the packets
Figure 3.19: The probability for the end to end latency for controller to actuators routing of the packets


Chapter 4

Estimation and Control

4.1 Measurement Collection

In a wireless networked control system all communication is made through a wireless link. Disturbances and communication interference can cause lost and delayed measurement packets. To compensate for the lost packets a Kalman estimator can be used to calculate an estimate of the observed states in the system in order to calculate a more accurate control signal. If the packets are delayed by retransmissions due to losses or busy devices, then a buffer of old measurements can be used to correct the estimation error from the old measurement combined with the delayed messages that are received afterwards, but are included in the measurement buffer.

The received measurements at the controller are delayed by $\tau_{sc}$ and the received control signals at the actuators are then delayed by the total loop delay $\tau = \tau_{sc} + \tau_{ca}$. The different signals for the different nodes can have different time delays and the effect of the delays are shown in Figure 4.1 for a system with two sensors and two actuators and the process transmits samples from both sensors at time $t_0$ and $t_5$. $\tilde{y}_1$ and $\tilde{y}_2$ in the figure corresponds to the received measurements at the controller. From the figure it is shown that the control signal at a time $t$ is delayed with the time $\tau = \tau_{sc} + \tau_{ca}$ and the separate control signals $u_1$ and $u_2$ can have different latencies. However, the control signal is only calculated after the data collection superframe is
compelled, after which the superframe for data publishing is started. If any of the measurements are still delayed after the data collection superframe, then the measurements are considered lost and a pure prediction is made with the Kalman filter. Basically the figure shows that the process gets the input by the old control signal in the time interval $t \rightarrow t + \tau$, this is compensated by adding the previous control signal as a state to the model.

Figure 4.1: Communication description for the wireless network with samples from a process at time $t_0$ and $t_5$ marked with red dashed lines
4.2 System model

A general linearized state-space system with process noise $w(t)$ and measurement noise $v(t)$ is given by:

$$\Delta \dot{x}(t) = A\Delta x(t) + B\Delta u(t) + w(t) \quad (4.1)$$
$$\Delta y(t) = C\Delta x(t) + v(t) \quad (4.2)$$

Where the number of states is $n_x$ and the number of control signals is $n_u$. It is assumed that there is one sensor and one actuator to measure and control each state.

4.2.1 Sampling of the system

The sampled process system with sampling time $h$ is described by the discrete-time system:

$$\Delta x_{k+1} = \Phi \Delta x_k + \nu_k \Gamma_0 \Delta u_k + w_k \quad (4.3)$$
$$\Delta y_k = \gamma_k C \Delta x_k + v_k \quad (4.4)$$
$$\Phi = e^{Ah} \quad (4.5)$$
$$\Gamma_0 = \int_0^h e^{As} ds B \quad (4.6)$$
$$Q = E[w_k w_k^T] \quad (4.7)$$
$$R = E[v_k v_k^T] \quad (4.8)$$
$$E[w_k] = 0 \quad (4.9)$$
$$E[v_k] = 0 \quad (4.10)$$

$$i = 1...n_x, \quad j = 1...n_u$$

$$\nu_k^i = \begin{cases} 
1 & \text{No loss at time } t_k \\
0 & \text{Loss at time } t_k 
\end{cases} \quad (4.11)$$

$$\gamma_k^j = \begin{cases} 
1 & \text{No loss at time } t_k \\
0 & \text{Loss at time } t_k 
\end{cases} \quad (4.12)$$
4.3 Time-varying Kalman Filters

The received measurements from a process are not only subjected with delays and losses, but also include noise. To estimate the states a time-varying Kalman filter increases the accuracy by computing new Kalman gain and uncertainty matrix each iteration with the received measurement data. The methodology used for the estimation is the same as in [5], but each $\nu$ and $\gamma$ are considered independent for each signal.

4.3.1 Estimator Design for a Time-varying Kalman Filter

The first step for the Kalman filter is to predict the next value of the states with the knowledge of the previous state and the system model. Optimal Kalman filter equations according to [5] are given by the equations (4.13) - (4.17)

Prediction step:

\[
\Delta \hat{x}_{k+1|k} = \Phi \Delta \hat{x}_{k|k} + \nu_k \Gamma_0 \Delta u_k \tag{4.13}
\]

\[
P_{k+1|k} = \Phi P_{k|k} \Phi^T + Q \tag{4.14}
\]

When a measurement is received the predicted state can be corrected by calculating the new Kalman gain. The uncertainty matrix is also corrected with the received measurement data. If there is a loss of measurement data, the correction of the states can not be made.

Correction step:

\[
K_{k+1} = P_{k+1|k} C^T (CP_{k+1|k} C^T + R)^{-1} \tag{4.15}
\]

\[
\Delta \hat{x}_{k+1|k+1} = \Delta \hat{x}_{k+1|k} + \gamma_{k+1} K_{k+1} (\Delta y_{k+1} - C \Delta \hat{x}_{k+1|k}) \tag{4.16}
\]

\[
P_{k+1|k+1} = P_{k+1|k} - \gamma_{k+1} K_{k+1} C P_{k+1|k} \tag{4.17}
\]
4.3.2 Kalman Estimation with Measurement Buffer

The buffer of measurement data at time $t_k$ can be filled as in Figure 4.2:

\[
\begin{array}{cccc}
  y_{k-4} & y_{k-3} & * & * \quad y_k \\
\end{array}
\]

Figure 4.2: The buffer at time $t_k$

At this point the data for the timeslots $t_{k-2}$ and $t_{k-1}$ are lost, but can be retransmitted. The lost data is retransmitted and can arrive to the buffer in a future timeslot, for example in the measurement buffer for timeslots $t_{k+1}$ and $t_{k+3}$ according to Figure 4.3. At time $t_{k+1}$ the measurement data $y_{k+1}$ was delayed, but the earlier data $y_{k-2}$ was acquired. At time $t_{k+3}$ all delayed data were acquired. The delayed data added to the buffer is then used to recalculate the state estimates to improve the estimation accuracy of present state.

4.4 LQG Design

4.4.1 Controller with Latency Compensation

Sampling of the system

The model for a networked control system in Figure 1.1 is used and the linearized system is modelled with independent loop delays $\tau_k^i = \tau_k^{sci} + \tau_k^{cai}$. 
for each signal $i$. The loop delays are assumed to be less than the sampling time $h$, if no packet has been received at the controller then an estimation by the prediction step for the Kalman filter is made. The sampled system can then be described by Eq. (4.18) - (4.20)

$$
\Delta x_{k+1} = \Phi \Delta x_k + \Gamma_0(\tau_k) \Delta u_k + \Gamma_1(\tau_k) \Delta u_{k-1} + w_k \quad (4.18)
$$

$$
\Gamma_0(\tau_k) = \sum_{j=1}^{n_u} \Gamma_{j0}^0(\tau_k^j) \quad (4.19)
$$

$$
\Gamma_1(\tau_k) = \sum_{j=1}^{n_u} \Gamma_{j1}^1(\tau_k^j) \quad (4.20)
$$

Where for each $j$, $B_j$ is the independent B-matrix for each control signal $j$

$$
\Gamma_{j0}^0(\tau_k^j) = \int_0^{h-\tau_k^j} e^{As} ds B_j \quad (4.21)
$$

$$
\Gamma_{j1}^1(\tau_k^j) = \int_{h-\tau_k^j}^{h} e^{As} ds B_j \quad (4.22)
$$

The state-space model is extended to include the past control signal $\Delta u_{k-1}$ as an additional state. The extended state-space model is given by:

$$
\begin{bmatrix}
\Delta x_k \\
\Delta u_{k-1}
\end{bmatrix}
$$

$$
\begin{bmatrix}
A & \Gamma_1 \\
0_{n_xn_u} & 0_{n_xn_x}
\end{bmatrix}
\begin{bmatrix}
z_k \\
\Delta u_k + w_k^e
\end{bmatrix} = \Phi^e z_k + \Gamma_{0}^e \Delta u_k + w_k^e \quad (4.24)
$$

$$
\begin{bmatrix}
C & 0_{n_u n_u}
\end{bmatrix}
\begin{bmatrix}
z_k \\
v_k^e
\end{bmatrix} = C^v z_k + v_k^e \quad (4.25)
$$

The loss function

The continuous time loss function is given by:

$$
J = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} (x_t^T Q x_t + u_t^T R u_t) dt \quad (4.26)
$$
4.4. LQG DESIGN

Where \( Q \) and \( R \) are the weighting matrices. The time continuous loss function can be expressed with the equivalent discrete loss function:

\[
J = \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} (x_k^T Q_1 x_k + x_k^T Q_{12} u_k + u_k^T Q_2 u_k)
\]

(4.27)

Where \( Q_1, Q_{12} \) and \( Q_2 \) are defined as:

\[
Q_1 = \int_0^h e^{A^T \tau} Q e^{A \tau} d\tau
\]

(4.28)

\[
Q_{12} = \int_0^h \Gamma_0^T (\tau) Q e^{A \tau} d\tau
\]

(4.29)

\[
Q_2 = \int_0^h \Gamma_0^T (\tau) Q e^{A \tau} \Gamma_0 (\tau) d\tau + h \cdot R
\]

(4.30)

The optimal solution for minimizing the loss function is solved with the Riccati equations and gives the feedback gain \( L(\tau_k) \). The control law is then given by:

\[
\Delta u_k = -L(\tau_k)(\Delta x_{\text{ref}} - \Delta \hat{x}_k)
\]

(4.31)

4.4.2 Static Loop Gain Controller

The time-varying Kalman filter and the latency compensating controller demands much performance and cannot be implemented in devices with limited processing power and memory. As an option an LQG controller with static loop gain and using a static gain Kalman filter is designed. The Kalman filter then estimates the states according to Eq. (4.32) with the static Kalman gain \( K_s \).

\[
\Delta \hat{x}_{k+1} = \Phi \Delta \hat{x}_k + \Gamma \Delta u_k + \gamma_k K_s (\Delta y_k - \Delta \hat{y}_k), \quad \Delta \hat{y}_k = C \Delta \hat{x}_k
\]

(4.32)

The control signal is calculated according to Eq. (4.33) with the static loop gain \( L_s \).

\[
\Delta u_k = -L_s (\Delta x_{\text{ref}} - \Delta \hat{x}_k)
\]

(4.33)
The optimal solutions to the static Kalman gain $K_s$ and the loop gain $L_s$ are derived from the Riccati equations. The optimal solution for the Kalman filter also gives the uncertainty matrix, which is constant for the static Kalman filter.

### 4.5 Process Model for The Boliden Process

#### 4.5.1 Nonlinear model

The non linear model for the tank level process is given by Eq. (4.34) - (4.36) as presented in [4]

$$\dot{x}_1(t) = \frac{1}{L_1} \left( \frac{1}{60} q(t) - K_1 u_1(t) \sqrt{x_1(t) - x_2(t) + \lambda_1} \right)$$

$$\dot{x}_i(t) = \frac{1}{L_i} \left( K_{i-1} u_{i-1}(t) \sqrt{x_{i-1}(t) - x_i(t) + \lambda_{i-1} - K_i u_i(t) \sqrt{x_i(t) - x_{i+1}(t) + \lambda_i}} \right)$$

$$\dot{x}_5(t) = \frac{1}{L_5} \left( K_4 u_4(t) \sqrt{x_4(t) - x_5(t) + \lambda_4 - K_5 u_5(t) \sqrt{x_5(t) + \lambda_5}} \right)$$

Where, for $i = 1, ..., 5$ the tank level for each tank is represented by $x_i(t) (m)$, $L_i (m^2)$ is the area of the surface in each tank, $K_i$ is the proportionality constant between the valve opening $u_i(t)$ and flow rate for each valve $i$. $\lambda_i$ is the height difference between tank $i$ and tank $i + 1$. $q_{in}(t)(m^3/min)$ is the flow rate in to tank 1.

#### 4.5.2 Linearized state-space model

The system is described by the linearized time continuous state-space model:

$$\Delta \dot{x}(t) = A \Delta x(t) + B \Delta u(t) + S \Delta q(t) + w(t) \quad (4.37)$$

$$\Delta y(t) = C \Delta x(t) + v(t) \quad (4.38)$$
Where,

\[ i = 1, \ldots, 5 \quad j = 1, \ldots, 5 \quad (4.39) \]

\[ A = \left[ \frac{\partial \dot{x}_i(t)}{\partial x_j} \right]_{x=x_0, u=u_0} \quad (4.40) \]

\[ B = \left[ \frac{\partial \dot{x}_i(t)}{\partial u_j} \right]_{x=x_0, u=u_0} \quad (4.41) \]

\[ S = \begin{bmatrix} \frac{1}{60L_1} & 0 & 0 & 0 \end{bmatrix}^T \quad (4.42) \]

\[ C = I_{5 \times 5} \quad (4.43) \]

\[ \Delta x = x - x_0, \quad \Delta u = u - u_0 \quad (4.44) \]

And \( w(t) \) is process noise and \( v(t) \) is measurement noise and \( E[w(t)w^T(t)] = R, E[v(t)v^T(t)] = Q, E[w(t)] = 0 \) and \( E[v(t)] = 0 \). The system is linearized around an equilibrium point \([x(0), u(0), q(0)]\).

### 4.5.3 Sampled system

The sampled process system with sampling time \( h \) is described by the discrete-time system:

\[ \Delta x_{k+1} = \Phi \Delta x_k + \nu_k \Gamma \Delta u_k + S_d \Delta q_k + w_k \quad (4.45) \]

\[ \Delta y_k = \gamma_k C \Delta x_k + v_k \quad (4.46) \]

\[ \Phi = e^{Ah} \quad (4.47) \]

\[ \Gamma = \int_0^h e^{As} ds B \quad (4.48) \]

\[ S_d = \frac{1}{T_s} S = T_s S \quad (4.49) \]

\[ i = 1\ldots n_x, \quad j = 1\ldots n_u \]

\[ \nu^i_k = \begin{cases} 1 & \text{No loss at time } t_k \\ 0 & \text{Loss at time } t_k \end{cases} \quad (4.50) \]

\[ \gamma^j_k = \begin{cases} 1 & \text{No loss at time } t_k \\ 0 & \text{Loss at time } t_k \end{cases} \quad (4.51) \]
The aim of the estimator design is to be able to estimate the process states $x_k$ from the transmitted measurement signal $y_k$, which includes measurement noise and the measurements can also be either lost or delayed. If a delayed packet has not arrived to its destination within the current superframe, it is discarded and considered as a lost packet. However, the estimator must be able to handle the losses and delays and give an accurate estimation.

### 4.5.4 Sampled system with loop delay $\tau_k$

The system is extended with additional states for the previous control signal $u_{t-1}$ and is sampled according to section 4.4.1. The sampled process system is then given by Eq. (4.52) - (4.53).

\[
\begin{align*}
\Delta x_{k+1} &= \Phi \Delta x_k + \nu_k \Gamma_0(\tau_k) \Delta u_k + \Gamma_1(\tau_k) \Delta u_{k-1} + S_d \Delta q_k + w_k \\
\Delta y_k &= \gamma_k C \Delta x_k + v_k
\end{align*}
\]

### 4.6 Simulations

As in [3] the weighting matrices for the LQG controllers are designed so that the tightest control in the last tank is obtained. The weighting matrices from [4] are modified due to the tank level errors being expressed as deviation in meters instead of percentage. The buffer length is $M = 10$ for the buffered controller and the successful transmission probability for each link is $p_l = 0.9$. The standard deviation for the process noise is $\sigma_p = 5.2^{-4}$ and $\sigma_m = 2.2^{-4}$ for the measurement noise. The sampling time is chosen to $h = 0.1$ (s), which includes 10 timeslots. The size of the noise was chosen to see a clear impact from the noise in the simulations. In the simulations the controller to actuators connection was assumed to be wired to reduce the number of combinations for the loop latency, which reduces the amount of matrix combinations to be stored.
4.6. SIMULATIONS

4.6.1 System response at disturbances

The tanks are affected by disturbances as an increase of the tank levels with $\Delta x = 0.1(m)$. Tank 1 is affected at time $t = 10(s)$, tank 2 at time $t = 510(s)$, tank 3 at time $t = 1010(s)$, tank 4 at time $t = 1510(s)$ and tank 5 at time $t = 2010(s)$. 
Buffered controller

Figure 4.4: System response to the disturbances with the buffered controller (estimate of the states (red) and the real process value (blue))
4.6. SIMULATIONS

Simplified controller

Figure 4.5: System response to the disturbances with the simple controller (estimate of the states (red) and the real process value (blue))

The figures show that the disturbances are suppressed efficiently, however the buffered control appears as faster and more independent from disturbances in the other tanks.
4.6.2 System response for a test sequence

The inflow $q_{in}$ to the first tank and the set value for the last tank will vary according to Figure 4.6.

Figure 4.6: Test sequence: Change in the inflow $q_{in}$ (left) and the set value for tank 5 (right)
4.6. SIMULATIONS

Buffered controller

Figure 4.7: System response to the test sequence with the buffered controller (estimate of the states (red) and the real process value (blue))
Figure 4.8: The response for the tank level (blue) on tank 5 for the test sequence with the set value (red)
4.6. SIMULATIONS

Simplified controller

Figure 4.9: System response to the test sequence with the simple controller (estimate of the states (red) and the real process value (blue))
Figure 4.10: The response for the tank level (blue) on tank 5 for the test sequence with the set value (red)

From the results it is shown that both controllers are able to remain stable and follow the set value for the test sequence. However, the buffered controller is more accurate and suppresses the disturbances and changes in the other tanks more efficiently. The estimation error for both controllers are calculated with Eq. (4.54), which shows that the estimation error for the buffered controller (4.55) is much smaller than the estimation error for the static controller (4.56).

\[ e = \sum_{k=1}^{N} (x_k - \hat{x}_k)^2 \]  
\[ \text{(4.54)} \]
Estimation error for the buffered controller

\[
e = \sum_{k=1}^{N} (x_k - \hat{x}_k)^2 = \begin{bmatrix}
0.0015 \\
0.0015 \\
0.0025 \\
0.0015 \\
0.0020
\end{bmatrix}
\] (4.55)

Estimation error for the simple controller

\[
e = \sum_{k=1}^{N} (x_k - \hat{x}_k)^2 = \begin{bmatrix}
0.6321 \\
0.7631 \\
0.8028 \\
0.6595 \\
0.7875
\end{bmatrix}
\] (4.56)
4.6.3 System response comparisons for different link qualities for the test sequence

\( p_l = 1.0 \) "Wired"

![Figure 4.11: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability \( p_l = 1.0 \).](image)

The simplified controller works fairly for this configuration, but the buffered controller gives a more accurate control.

\( p_l = 0.9 \)

![Figure 4.12: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability \( p_l = 0.9 \).](image)
The control still works fairly for the simplified controller and the buffered control appears almost as above.

\[ p_l = 0.8 \]

Figure 4.13: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability \( p_l = 0.8 \)

It can be clearly seen that the control starts to deviate more from the set values for the simplified controller.

\[ p_l = 0.7 \]

Figure 4.14: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability \( p_l = 0.7 \)
$p_l = 0.6$

Figure 4.15: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability $p_l = 0.6$

The simplified control starts to deviate much and the buffered controller still offers stable control close to the set value.

$p_l = 0.5$

Figure 4.16: Comparison between the simplified controller (left) and the buffered controller (right) with successful link probability $p_l = 0.5$

The simplified controller becomes unstable due to too many lost measurement packets.
4.6.4 The LQG Cost as a function of the link probability

For the same test sequence the cost function with Eq. (4.27) is calculated for the different link probabilities.

![Figure 4.17: Loss function for the buffered (left) and static (right) controller for different link qualities](image)

The figures are shown with standard deviation error bars and only the stable regions for the loss probability are included. The buffered goes unstable with $p_{\text{loss}} = 1$ and the simple goes unstable at $p_{\text{loss}} = 0.5$. The cost for the buffered controller does not increase very much for the increased loss probabilities. However, for the static controller the cost increases significantly for each increase of 0.1 in the loss probability.

4.7 Conclusions

The buffered LQG controller works well for all simulated link probabilities and for further notice it is also stable for link probabilities as low as $p_i = 0.1$. The simplified controller is stable for the system as long as the wireless network satisfies sufficient link probabilities. It can also be noted that the static gain controller gets high estimation errors with lower link probabilities and it cannot correct the old values as the buffered controller. However, the
simplified controller with static gains is much easier to implement and does not require the amount of processing power as for the buffered controller. The big disadvantage for the buffered controller is that it requires very much memory and processing power for implementation in a real-time system.
Chapter 5

Implementation

5.1 Tmote sky

The Tmote sky device is an ultra low power module and can be used for wireless sensor networks. It features a 250kbps 2.4 GHz IEEE 802.15.4 Chipcon wireless transceiver and is operable with other IEEE 802.15.4 devices. The module includes ADC and DAC ports and has an 8 MHz MSP430 microcontroller with 10k RAM and 48k Flash. The module can be connected to a PC through a USB port for programming and for transferring data between the module and the PC. The device can be supplied with power through the USB port when connected to a PC or it can use batteries or a wired external power supply.

5.2 Contiki

Contiki is a memory efficient open source operating system designed for microcontrollers with small amounts of memory. The Tmote sky module can be programmed with the contiki operating system. The devices for this implementation use the RIME communication stack, which is supported by contiki.
5.3 NI-DAQ Device

In the implementation a National Instruments data acquisition device of model NI USB-6221 is used to set the voltage for the water pump. The NI-DAQ device can be used to measure voltage and to generate voltage. The device is supported in MATLAB and in this implementation the device is used to generate the calculated control signal for the water pump.

5.4 Wireless Sensor Extended With WirelessHART Burst Mode Commands

The wireless Tmote sky device is programmed with the WirelessHART burst mode commands described in section 2.2.3 and a unicast connection supported by contiki. The communication layers are not WirelessHART, but in a parallel thesis in [23] the first part of the implementation of the WirelessHART layers was made. The purpose is to implement the burst mode commands and can later be combined with the complete WirelessHART layers in future projects. The Tmote device is configured as a wireless sensor, and supports two connected sensors to ports ADC0 and ADC1. It is possi-
ble to use more sensors to read data and also to write data from the DAC ports, but is not implemented in this project. The burst mode configuration assigns the variables to data slots 0 and 1, with WirelessHART command 33 as the response command for burst messages and the device is configured for continuous burst trigger mode.

5.5 Configuration

The process to be controlled is the Quanser water tank process [25]. A Tmote sky wireless device with the operating system contiki 2.3 is programmed with commands for WirelessHART burst mode according to section 2.2.3. The wireless device can then be set to burst mode by a gateway device, which is connected to a PC. While in burst mode the device then transmits sensor data from the two water tank sensors connected to two ADC-ports ADC0 and ADC1 on the Tmote sky device with the given update rate according to the current burst mode configuration. The PC reads the received data packets from the serial port(USB) and the data is processed in MATLAB, the calculated control signal is then transmitted with the NI-DAQ device to the water pump connected to the process. The control is applied with LQG control with different Kalman estimators. First a simple with static Kalman gain and a time-varying Kalman estimator where the Kalman gain and uncertainty is recalculated for each sample. An overview of the process configuration is shown in Figure 5.2

5.5.1 The Water Tank Model

The water level in the water tank process shown in Figure 5.3 is given by Eq. (5.1) - (5.2) where \( x_1(t) \) and \( x_2(t) \) are the water levels in the upper and lower tank, and \( u(t) \) is the applied voltage to the pump.

\[
\dot{x}_1(t) = -\frac{a_1}{A_1}\sqrt{2g \cdot x_1(t)} + k \cdot u(t) \\
\dot{x}_2(t) = \frac{a_1}{A_1}\sqrt{2g \cdot x_1(t)} + \frac{a_2}{A_2}\sqrt{2g \cdot x_2(t)}
\]
Figure 5.2: The process configuration

Figure 5.3: Quanser Water tank process
The outlet holes from both tanks and the diameter of the tanks are equal: $a_1 = a_2$ and $A_1 = A_2$. Which gives that the defined equilibrium point is given by $x_1^0, x_2^0, u^0$ and $x_1^0 = x_2^0$. The linearized system is then given by Eq. (5.4) where $\Delta x = x(t) - x_0$, $\Delta u = u(t) - u^0$ and $\Delta y = y(t) - y^0$.

$$\begin{bmatrix} \Delta \dot{x}_1 \\ \Delta \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} \\ \frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} \end{bmatrix} x=x_0, u=u_0 \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} k \\ 0 \end{bmatrix} \Delta u$$

$$\Delta y = \begin{bmatrix} k_{s_1} \\ 0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

(5.3)

(5.4)

The output constants $k_{s_1}$ and $k_{s_2}$ define the scaling on the sensor signal for conversion from the voltage to the tank level in (cm). The model parameters are given in the data sheet [25]. However, the constant $k$ for the inflow to the upper tank from the pump was acquired by manually adjusting the voltage to reach the steady state levels defined by $x_0$ for both tanks. The constant can then be calculated by Eq. (5.1) with $\Delta \dot{x}_1 = 0$ at steady state.

### 5.5.2 Reading the sensors

The pressure sensors connected to the water tanks give a reading of 0.2 V/cm, which gives a voltage of 6 V with the water level at 30 cm. Therefore, a voltage divider is used to scale the sensor output by the factor three to fit the voltage range on the ADC ports which is within the range 0 - 2.5 V. The voltage read from the ADC is an unsigned integer $ADC_{Count}$ in the range 0 - 4095 and needs to be converted back to recover the original sensor voltage according to Eq. (5.5)

$$U_{Sensor} = 3 \cdot \frac{ADC_{Count}}{4096} \cdot U_{Ref}$$

$U_{Ref}$ is set to 2.5 V in the ADC configuration for the Tmote for full range readings on the ADC ports.

### 5.5.3 The wireless communication

The configuration for controlling the water tank process includes two Tmote sky devices. One is configured as a the wireless sensor and connected to the
tank sensors and the other device is configured as a gateway and connected to a PC through the USB-port. At the start of the control application the gateway sets the wireless sensor to burst mode according to section 2.2.2. After the configuration the wireless sensor transmits sensor readings according to the burst mode configuration and the user can send additional commands to the wireless sensor to change the configuration. The sensor and gateway device are configured to send with unicast and the PC can read the received packets from the gateway device from the serial port. Lost packets are not re-transmitted and therefore a buffered controller does not give a better performance and is not used in this implementation.
Chapter 6

Evaluation

6.1 Results

Two different controllers are used on the process for a step response and system response for a large disturbance. Tests are made for two different scenarios, the first with optimal wireless connection and the second scenario where approximately 50% of the packets are considered lost. The losses are randomized using random variables in the software to throw away packets since the wireless connection itself did not introduce losses. The uncertainty of the time varying Kalman filter is evaluated with the trace of the uncertainty matrix for each sample. The trace is compared with trace of the uncertainty matrix for the Kalman filter with static gain, which is constant for all samples.
6.1.1 Minimal losses

Step response

Figure 6.1: Step response with the static Kalman gain

Figure 6.2: Step response with the single time-varying Kalman filter
Figure 6.3: Uncertainty for the single time-varying Kalman filter
CHAPTER 6. EVALUATION

Disturbance

Figure 6.4: System response to a disturbance with the static Kalman gain

Figure 6.5: System response to a disturbance with the single time-varying Kalman filter
Figure 6.6: Uncertainty for the time varying Kalman filter with the disturbance in the system

**Uncertainty Evaluation**

The uncertainty for the time-varying Kalman filter is larger at the beginning. However, after a few samples the uncertainty for the time-varying Kalman filter for both the step response and the disturbance goes clearly below the uncertainty for the static Kalman gain filter.
6.1.2 $p_l = 0.5$

Step response

Figure 6.7: Step response with the static Kalman gain

Figure 6.8: Step response with the single time-varying Kalman filter
Figure 6.9: Step response with the single time-varying Kalman filter
Disturbance

Figure 6.10: Step response caused by a disturbance with the static Kalman gain

Figure 6.11: Step response caused by a disturbance with the single time-varying Kalman filter
6.1. RESULTS

77

Figure 6.12: Step response caused by a disturbance with the single time-varying Kalman filter

Uncertainty Evaluation

The uncertainty for the time-varying Kalman filter for both the step response and the disturbance, deviates mostly below the uncertainty for the static Kalman filter. Long sequences of losses increase the uncertainty significantly, as can be seen in Figure 6.9 for the step response. Losses makes some negative impact on the uncertainty for the time-varying Kalman filter, but is improved by received measurements after a lossy sequence.

6.1.3 Conclusions

From the results it is clear that both types of Kalman estimation works well for this system. However, it is clear from the results that the two methods give different performance on different sequences. The static Kalman gain estimator works well when the system is in a static state and gives a smooth control where the levels of the tanks do not deviate very much, since the control is quite slow. Furthermore, the time-varying Kalman gain estimation
works better when the system is in a dynamic state caused by a changed set value or a disturbance. The time-varying estimation gives a faster control since the estimation is more accurate, which, is the reason that it gives better control in a dynamic state.

Especially in the step responses for the static gain estimation it can be noted that the estimated value rises before the measurements. This is because the hose from the pump to the upper tank being empty when the process starts and it causes an initial time delay. However, when the process is running and the hose is filled with water the time delay is not the same as the initial, which, makes it difficult to add to the model.

It should also be taken in consideration that these experiments were made for a real physical system and the sequence for the different controllers and estimators are different. If the system is for instance in a bad state at the time for the disturbance for one solution then it will give a poor performance compared to the other solution.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

The aim of this thesis was to develop tools for latency analysis for wireless mesh networks and to design a latency compensating Kalman filter and LQG controller. Furthermore, the latency analysis and the control design along with embedded systems programming was to be used for implementation and evaluation on a real system.

The latency analysis was made both for single and multiple-packet transmissions (i.e. one-to-one and one-to-many/many-to-one communication patterns). Furthermore, Monte-Carlo simulations allow to evaluate more complex protocols with packet queuing and priorities. The latency analysis gives us the probability distributions for latencies and the loss probabilities caused by packets not arriving before their intended deadline. The latency analysis can be used to determine the buffer length if a buffered controller is used and it also verifies if a given routing scheme is appropriate for the network.

Two solutions were considered for the estimation and control, an optimal solution that rely on buffering old data and solving Riccati-equations on-line from the latency depending system matrices, and a simple solution wit stationary Kalman gain and control loop gain without considering the effects of the latencies in the model. For the simulate and implemented networks in this thesis the time-varying matrices for calculating the Kalman gain and the
loop gain for control can be pre-computed, at the expense of high memory usage.

In the simulations and the implementation the time-varying Kalman filter gave a more accurate estimation. Also the control was faster for the time-varying LQG controller. In the implementations, the static estimator and controller worked well for a step response, but when the system was in transient state the time-varying solution displayed better performance. The analytical part of this thesis is implemented as MATLAB-functions and can easily be used in future projects.

7.2 Future Work

There is much work that can be continued from this thesis, mainly for the implementation. In the implementation in this thesis the wireless sensor did not use the complete WirelessHART communication stack, but it used the WirelessHART burst mode commands using the unicast communication stack provided by Contiki. This should be a first step to develop the wireless sensor further, also the DAC should be configured to remove the need of the NI-DAQ device and make the network completely wireless.

In this thesis all calculations for the estimation and the calculations for the control signals were made in MATLAB. If the wireless system would be implemented to full wireless system without connection to a PC, then the time-varying estimators and controllers would be impossible to implement and the static gain estimator and controller would be the only option due to processing power and memory limitations.
Appendix A

Transition Probability Matrices

The transition probability matrices for the Markov chain calculations of the latency probabilities are given by (A.1) - (A.3). In the calculated result with MATLAB all link probabilities are $p_{ij} = 0.8$.

\[
\begin{pmatrix}
\bar{p}_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
p_{12} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \bar{p}_{35} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bar{p}_{46} & 0 & 0 & 0 & 0 \\
0 & 0 & p_{35} & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & p_{46} & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \bar{p}_{78} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \bar{p}_{78} & 1
\end{pmatrix}
\]  

(A.1)
\[ P(2) = \begin{bmatrix} \tilde{p}_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{p}_{25} & 0 & 0 & 0 & 0 & 0 \\ p_{13} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{p}_{47} & 0 & 0 & 0 \\ 0 & p_{25} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \tilde{p}_{68} & 0 \\ 0 & 0 & 0 & p_{47} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & p_{68} & 0 & 1 \end{bmatrix} \] (A.2)

\[ P(3) = \begin{bmatrix} \tilde{p}_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{p}_{26} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{p}_{37} & 0 & 0 & 0 & 0 \\ p_{14} & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{p}_{58} & 0 & 0 \\ 0 & p_{26} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & p_{37} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & p_{58} & 0 & 0 & 1 \end{bmatrix} \] (A.3)
Appendix B

Implementation diagrams

Figure B.1: Connection of the sensor side for the implementation process

Figure B.2: Connection of the actuator side for the implementation process
Bibliography


