ISSN 0280–5316
ISRN LUTFD2/TFRT--5597--SE

Analysis of Time Delays in Synchronous and Asynchronous Control Loops

Ben Bastian

Department of Automatic Control
Lund Institute of Technology
March 1998

Department of Automatic Control
Lund Institute of Technology
Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
98/3/26

Document Number
ISRN LUTFD2/TFRT--5597--SE

Author(s)
Ben Bastian

Supervisor
Björn Wittenmark & Brett Ninness(Univ. of Newcastle, Australia)

Sponsoring organisation

Title and subtitle
Analysis of Time Delays in Synchronous and Asynchronous Control Loops . (Analys av tidsfördröjningar i synkrona och asynkrona reglerkretsar).

Abstract

Interlinked control loops with asynchronous sampling can give rise to a time varying time delay which varies irregularly. Here we examine ways to model these effects for an arbitary number of control loops, and to build up a probability density function, and current delay, before implementing a controller

Key words
Distributed systems, Networked control, Layered systems, Time Delays,Random time delays

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
Engelska

Number of pages
61

Recipient's notes

Security classification

Contents

Chapter 1

Problem formulation

1.1 Introduction

This section is in the most part based on Nilsson (1998) and Nilsson, Bernhardsson and Wittenmark (1998).

One of the problems which occurs in commercial control systems is time variations due to asynchronous signals within the control loops. This means that due to the nature of the signals to and from an I/O device to the controller there may be a signal variation which affects the signal output.

Driving example This example was given in Nilsson (1998), and is the prime basis of the simulation of the system, and as examples for the modelling. The example system is shown in Figure 1.1, from that paper. This simple system consists of a control system and an I/O unit which are connected by a remote field-bus. A delay is caused in the system due to the asynchronous nature of the devices. This is shown in Figure 1.2.

This type of system is a simple version typical of commercial control systems with distributed I/O and control. Complications can include extra layers of control, for instance, a plant wide Ethernet with limited real time capabilities. This setup will give a system with varying delay from sensor to actuator. We will look at the variations of the control delay, which is the time from when the measurement signal is sampled to when it is used in the actuator.

1.2 The problem

When the input into the controller is compared with the output the problem can be seen more clearly. This is shown in Figures 1.3 and 1.4. In Figure 1.3 we can see the response of the system to a ramp input, having set the controller to be a simple unit gain. The times at the points indicated by circles are the ones of interest. In Figure 1.4 we have the system response to a delayed digital

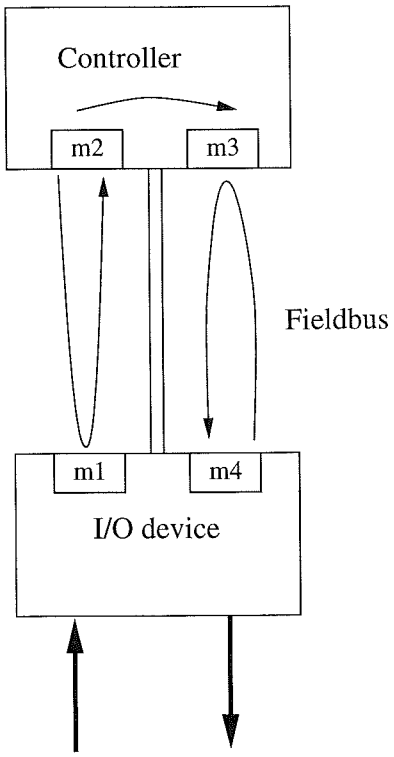

Figure 1.1: Control system with asynchronous loop.

system compared with the response of a digital signal without this delay, with the delay clearly marked.

1.3 Description of the delays

There are two possible description of the delays, the first of these is as a deterministic process, that is that the value of the process is simply a predictable function of time, for instance a sine curve. For a deterministic process the corresponding control structure is also deterministic, as the disturbance can be predicted, and usually to some extent compensated for.

A stochastic system on the other hand is one that is random with regards to time, so that the disturbance or delay in this case is not possible to predict exactly. In this case there is a whole theory of controller design based around stochastic disturbances. It has been shown that an optimal solution can be formed by using a system of quadratic cost functions. This is also known as LQG control, or *linear quadratic Gaussian.*

Here rather than examining *noises* in a system we are examining time delays,

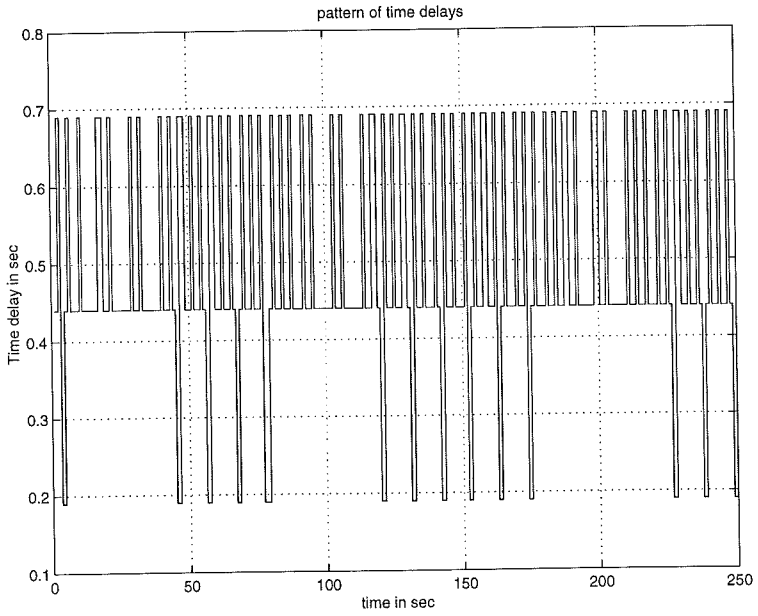

Figure 1.2: Total delay due to asynchronous loops of the system in Figure 1.1.

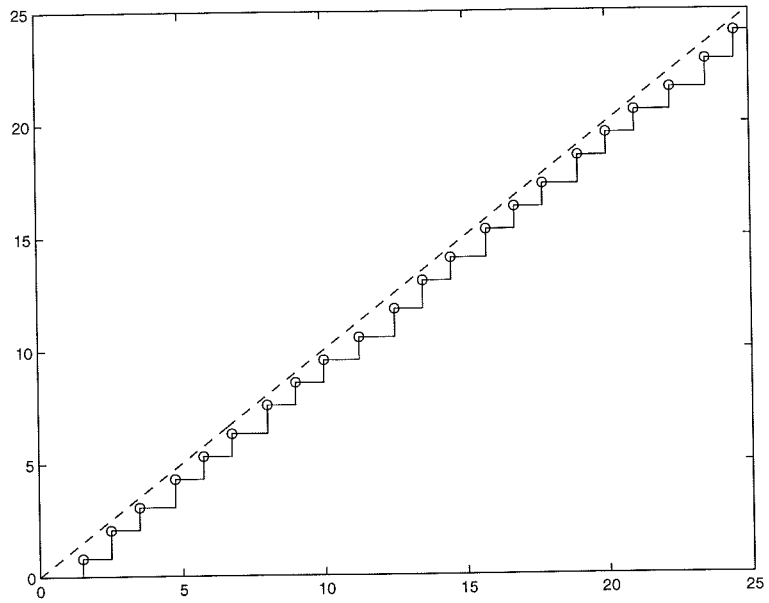

Figure 1.3: The system response signal to a ramp input.

however the basics remain similar. A deterministic delay can to some extent be compensated for, and a stochastic delay case can also have an optimal solution. The methods for finding this solution are related to LQG (Nilsson 1998).

The method pursued by Nilsson is a stochastic approach, modelling the delays using a sequence of Markov chains, however the approach finally taken in the modelling in this thesis is largely as a deterministic process, although one with random features. Thus this work looks at the delay in an entirely different fashion.

1.4 Driving example

The driving example is from Nilsson(1998) and is the circuit seen in Figure 1.1. This is the basic circuit used in both the simulations and the examples of modelling.

The control system in Figure 1.1 has two units, an I/O-module and a controller, which communicate using a field-bus. In the I/O module the output of the process is A/D converted, and the result is copied into m1. Synchronized with this is the D/A conversion which writes the content of m4 to the process. Both of these events occur every 250 ms, however they may not necessarily be completely synchronous, that is there may be a phase difference between the input and the output.

In a similar way, both the write bus and the read bus share the same timing,

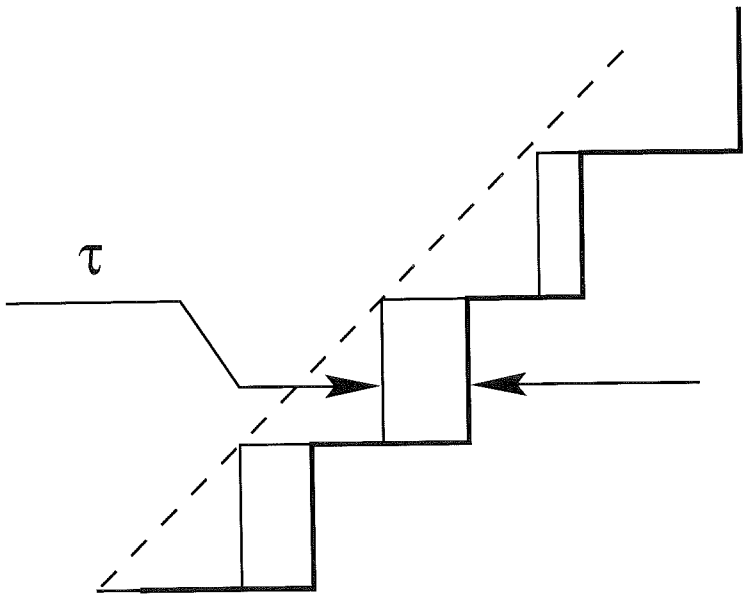

Figure 1.4: The time delay of interest, τ.

although there may be a phase difference between the two. The sampling is set to 256 ms.

Finally the controller is sampling and writing back to the bus with a sampling period of 1.1 s. The series of events of a typical cycle are shown in Figure 1.5

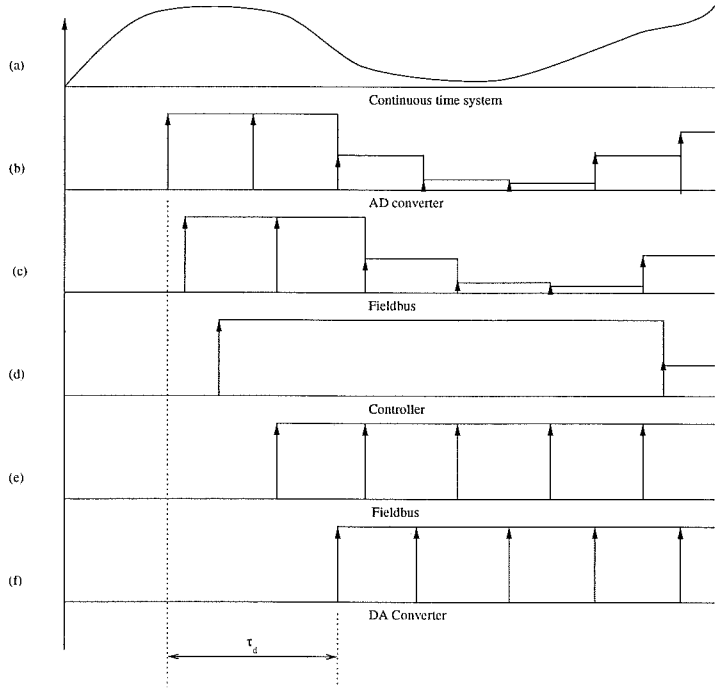

Figure 1.5: Timing of a typical cycle.

In Figure 1.5(a) we have the continuous time signal, which is the output of the process which is to be controlled. In Figure 1.5(b) the signal is first A/D converted. In our example this occurs once every 250 ms. This signal corresponds to the value of m1. In Figure 1.5(c) the signal is mirrored using the bus, thus this signal corresponds to the value of m2. In Figure 1.5(d) the signal is sampled and calculations are made, and a corresponding signal is written to the output of the controller, m3.It is assumed in this scenario that the process time of the controller is negligible. In Figure 1.5(e) the signal is mirrored by the bus once again. This occurs every 256 ms. and the signal obtained here corresponds to that found in m4. In Figure 1.5(f) the signal is A/D converted and sent to the actuators.

1.5 Event driven versus clock driven events

A large part of this thesis occurs due to the assumption that the events which occur are in the most part clock driven. Clock driven means that events take place at prespecified times according to parameters set by a timing device, usually periodic sampling in this case. It is also possible for events to be event driven, thus an event will trigger another event. In Nilsson (1998) it is assumed that while the input is clock driven, the output and controllers are both event driven, thus they start processing as soon as information is at hand. Here a different case is being examined, and all events are seen to be clock driven, creating random time delays due to the nested loops.

1.6 Thesis Outline

In chapter 2 the set of delays are modelled as a set of sliding intervals,it is shown how the delay can take on discrete values for some types of system, and a probability mass function was built up. The idea of a mass function is extended in chapter 3, to the idea of a density function, the continuous random variable equivalent to the mass function. It is shown here how a density function can be built up. Some example systems are simulated in chapter 5. Finally controller is implemented in chapter 6.

Chapter 2

Modelling

2.1 Model structure and assumptions

The basic model which will be examined is described by a series of "layers", like an onion, with the controller being encapsulated by both the bus and the I/O, thus an incoming process signal must pass through both layers to the controller and the outgoing control signal must again pass through both layers to get to the process. For the purposes of this model, the internal process is layer 0. For our driving example already defined this layer corresponds to the controller process, thus layer 1 corresponds to the bus layer, and layer 2 to the I/O layer.

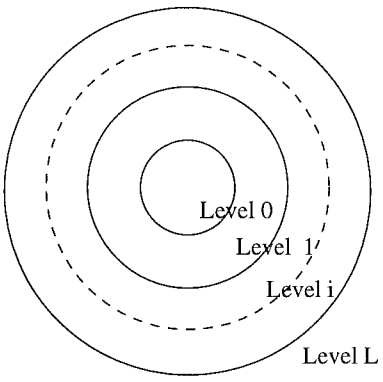

Figure 2.1: The various architecture layers can be seen as layers of an onion.

These layers will be examined with reference to the time delays caused by their interaction. This will be done both for an abstraction of this model, i.e. for any number of layers, and then as a case study for our driving example, to test the validity of the theories found. For the model, each layer is simulated by a separate timing diagram, which looks like the one shown in Figure 2.2.

Here a circle indicates the timing of the input and a cross indicated the timing of the output. A signal is defined as being input when it is moving towards layer zero, and output when moving away from layer zero, thus the DA converted signal is considered an output on layer 2.

Definition 1 *A layer is defined to be* definite synchronous *if the input and the output of that layer occur at exactly the same times.*

A layer where both input and output have the same sampling rate, but do not occur at the same time is defined as partially synchronous.

A layer where the input and the output have different sampling times is said to be asynchronous. *If the ratio of h_in/h_out for that level is a rational number then level can be said to be* partially asynchronous. *If the ratio is irrational, then the time between input and output are independent, and the level is defined as being* Definite asynchronous

Definition 2 *The* sampling frequency *of a layer h_i is defined to be the period of the inputs of that layer, measured in seconds.*

Definition 3 *The* phase *of a layer is defined to be the time difference between the input of a system and the next occurring output. For a definite synchronous system this is always zero, for a partially synchronous system this is always a constant, however for an asynchronous system this will be time varying. Phase is measured in seconds, and its symbol is θ.*

2.2 Properties of a definite synchronous system

2.2.1 Assumptions

Layer zero has a single action which is the core of the system. Information goes no further than layer zero. In the driving example layer zero is a controller. Initially layer zero is assumed to be almost instantaneous, and clock driven, with period h_0.

2.2.2 Delay in a single layer system

In this case there is only a single layer. This layer is the controller. In this trivial case the time delay is zero and there is instantaneous sampling every h_0 seconds.

2.2.3 Delay in a system with two layers.

Layer zero: Clock driven

Here there are two layers, layers zero and one. In this case layer one has two I/O events and a single event occurs on layer zero between them.

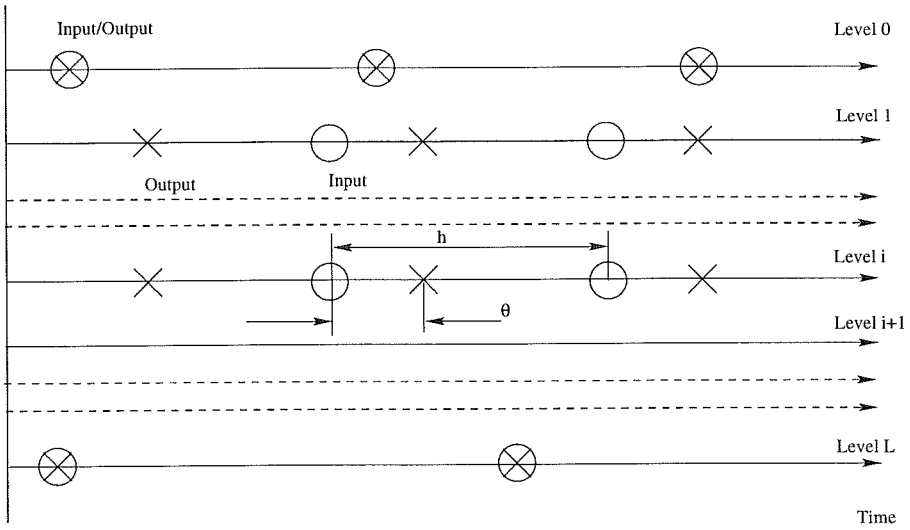

Figure 2.2: The format of the timing diagrams used for the models.

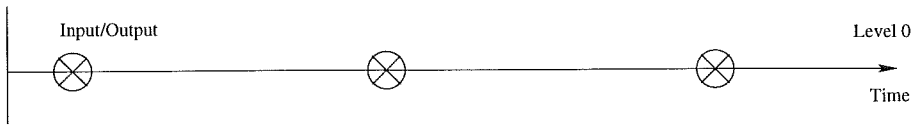

Figure 2.3: Delay with one layer.

Assumption 1 *It is assumed to be impossible for events to occur instantaneously. Thus, a signal cannot be input, processed and output at the same instant.*

Note 1 *This assumption is opposite to that made implicitly by Simulink, ut is defended on the grounds of common sense.*

With the previous assumptions in force it follows that the time delay for layer one is always exactly h_1 s, as the input is always at the first event on layer one and the output is always at the second event.

Layer zero: Event driven

Assumption 2 *Similarly to the clock driven case, the time events cannot occur totally simultaneously, but the event occurs an infinitesimal time after the bus input event - thus the signal cannot be input, processed ,and output at the same instant of time.*

Here again, the delay between the input and the output is exactly h_1 s. Thus for cases with no phase difference in layer one it makes no difference whether the events in layer zero are clock or event driven.

Relevance of layer zero sample time

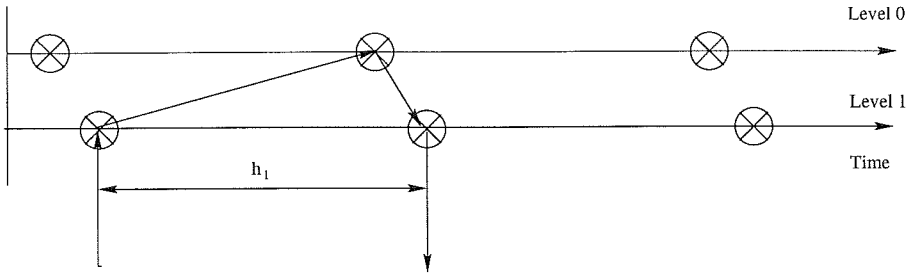

Figure 2.4: Both layers synchronized.

In the case where $h_0 = h_1$ there is a controller event for every interval of h_1 thus the total time delay is equal to h_1.

In the case where $h_0 > h_1$ Information i_1 is sampled at an initial sample, but no controller event occurs. The information is therefore discarded when the next sample occurs, with information i_2. This then is the information processed when the event on layer zero, the controller event, occurs. The corresponding information is then output at the next output of layer one. It follows from this that the total time delay is h_1 s.

In the case where $h_0 < h_1$ the information received by all events on layer zero is the same during each interval. Thus also in this case the total time delay is equal to h_1

2.2.4 Time delay on layer i

The system can now be generalized to accommodate for a system with more than two layers. Information can only pass through the layer at sampling times. Information goes in at time $n_i h_i$, and leaves at time $n_i h_i + k_i h_i$.

Thus the total time delay for sampling and returning information from process to controller and back to process is $k_i h_i$ where k_i is in $\mathbb{N}$ (plus 0).

Therefore on layer L (the bottom layer), for the basic model the total time delay is

$$\tau_d = k_L h_L \qquad \text{for } k_L \in \mathbb{N} \tag{2.1}$$

For the driving example the time delay of interest is the total delay at the I/O layer.

2.2.5 The time delay on layer i+1 given a constant time delay on layer i

In the case $\tau_i < h_{i+1}$. Assuming τ_i and h_{i+1} are constantly drifting in relation to each other, the possibility that any point p in the sample time of layer $i+1$ occurs within the interval of τ_i at any randomly chosen time is τ_i / h_{i+1}. From previous result (2.1)

$$\tau_{i+1} = k_{i+1} h_{i+1}$$

For this case it can be seen then that $\tau_{i+1} = h_{i+1}$ if and only if interval τ_i lies within h_{i+1}. For this situation then:

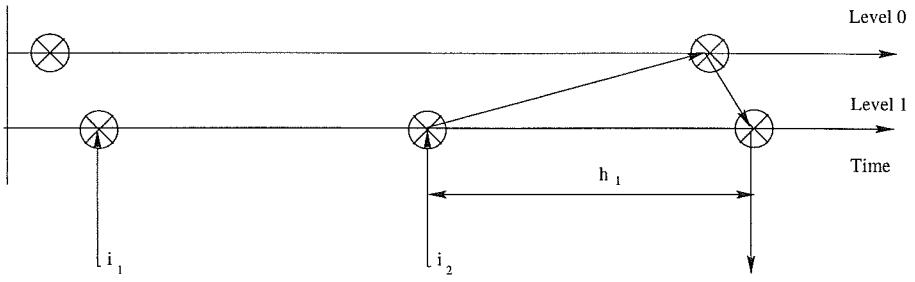

Figure 2.5: Two layers with $h_0 > h_1$.

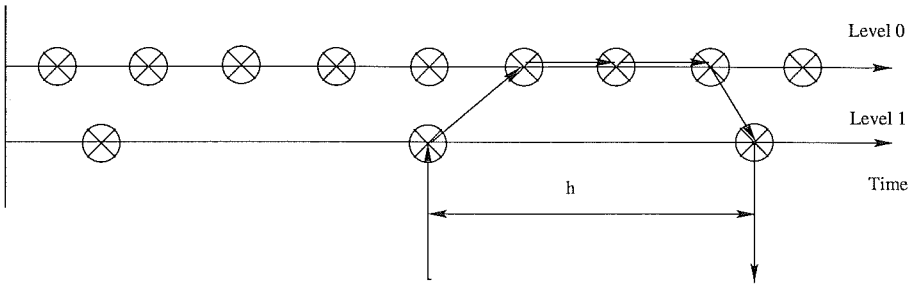

Figure 2.6: Two Layers with $h_0 < h_1$.

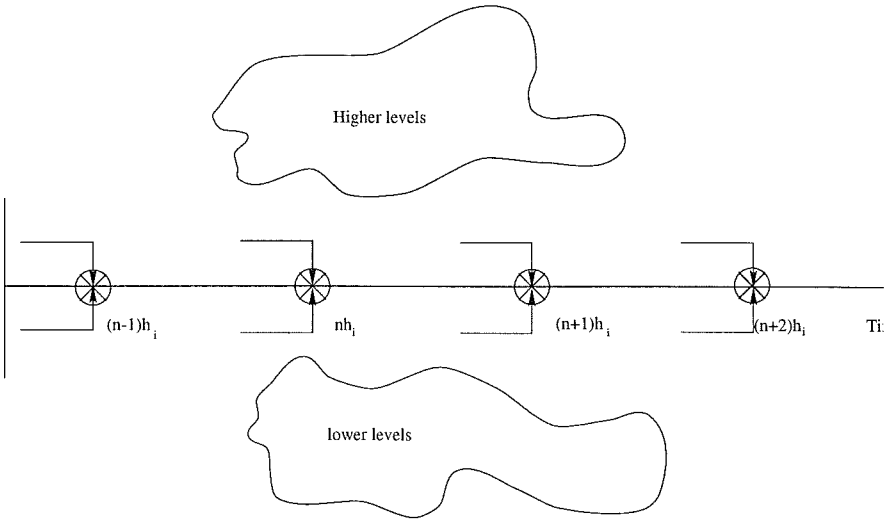

Figure 2.7: The possible delays for any layer.

$$P(k_{i+1} = 1) = \frac{h_{i+1} - \tau_i}{h_{i+1}} \tag{2.2}$$

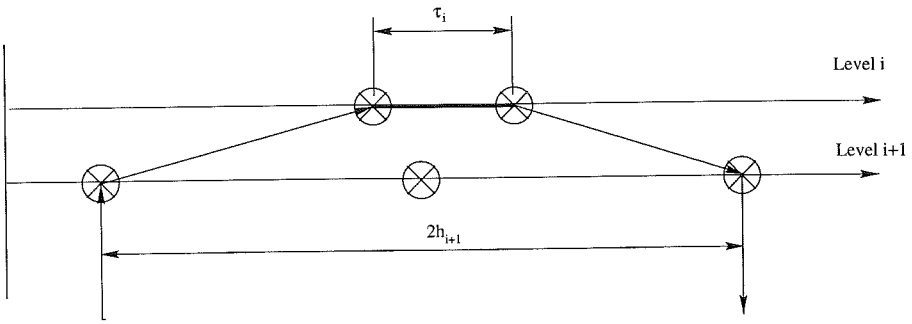

Figure 2.8: Delay on layer $i+1$ when $\tau_i < h_{i+1}$.

The other option is that $\tau_{i+1} = 2h_{i+1}$. This occurs if and only if interval τ_i crosses point p which corresponds to the beginning of an interval. Thus:

$$P(k_{i+1} = 2) = \frac{\tau_i}{h_{i+1}} \tag{2.3}$$

Now a longer delay time for layer i will be examined.

Consider $\tau_i > h_{i+1}$ If this is the case then

$$\tau_i = nh_{i+1} + \overline{\tau_i} \qquad \text{where } \overline{\tau_i} > h_{i+1}$$

here k is at least $= \tau_i/h$

From equations (2.2) and (2.3) then:

$$P(k_{i+1} = n+1) = \frac{h - \overline{\tau_i}}{h_{i+1}} = \frac{h_{i+1} - (\tau_i - nh_{i+1})}{h_{i+1}} \tag{2.4}$$

$$P(k_{i+1} = n+2) = \frac{\overline{\tau_i}}{h_{i+1}} = \frac{\tau_i - nh_{i+1}}{h_{i+1}} \tag{2.5}$$

Note 2 This result implies that every delay on level i there are two possible delays for level $i+1$.

2.2.6 Example

To illustrate the definite synchronous model, the numerical values of the driving example are put into the the given problem, for the basic case.

Because the layer above is the second layer and there is no phase difference between the input and the output of the second layer (the bus), then the delay τ_1 is equal to the sampling time of the layer, i.e. $tau_i = h_1 = 256$ ms. From (2.1)

$$\begin{aligned} \tau_d = \tau_2 &= k_2 h_2 \\ &= 250 k_2 \text{ ms} \qquad \text{where } k = 2, 3 \end{aligned}$$

The basic probability that the time delay is 500 and 750 ms, respectively, is worked out then from (2.4) and (2.5):

$$\mathrm{P}(\tau_d = 500) = \mathrm{P}(k = 2) = \frac{244}{250} = 0.9760$$

$$\mathrm{P}(\tau_d = 750) = \mathrm{P}(k = 3) = \frac{6}{250} = 0.0240$$

This is verified by the simulation data in Section 5.2

2.2.7 Conclusions

Here it is clear that given a synchronous layer one, the characteristics of layer zero can safely be ignored in respect to time delay (This however isn't necessarily true for the sample time, that is the controller time will affect the theoretical sample time of the system. This is shown in chapter 4). Because the time delay

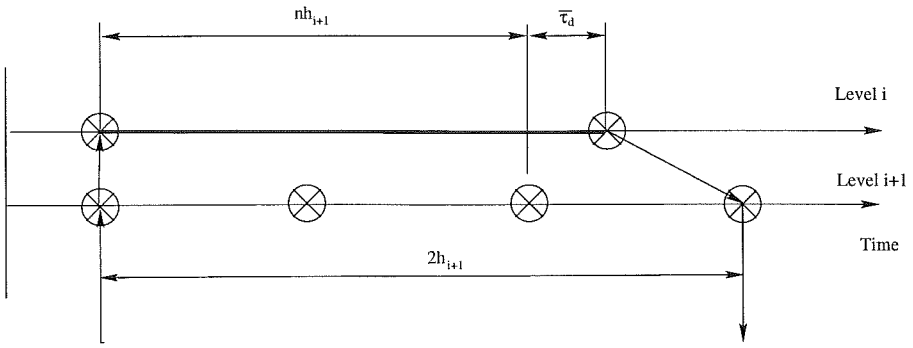

Figure 2.9: Time delay on layer $i+1$ when $\tau_i > h_{i+1}$.

on layer one is constant, and the two layers display a constant drift in relation to each other, then the behavior of layer two can be easily determined at any random time slice by using simple ratios of the two intervals. For our three layer system, there are only two possible time delays that can occur, and the appearance of each particular delay length appears to be strongly linked with these ratios. This is a very neat result, however this will change given a partially synchronous system.

2.3 Properties of partially synchronous systems

2.3.1 A more complex model

Here phase differences are introduced between the inputs and the outputs for all different levels. Recall that from definition 3 The *phase difference* for a layer i is the time difference between the input and the next output, for that layer, and its symbol is θ.

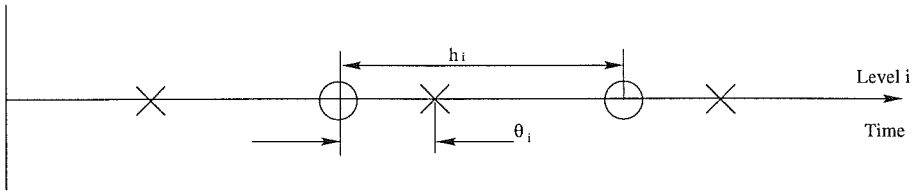

Figure 2.10: Definition of θ.

It can be seen intuitively from looking at layer i that the time delay at layer i can take the discrete values of $k_i h_i + \theta_i$. If a signal enters the layer at time t_{in}, and the corresponding signal exits the layer at t_{out} , then the signal must take a total time from sensor to actuator of $k_i h_i + \theta_i$ where $k_i = 0, 1, 2, \ldots$

$$\tau_{i+1} = k_{i+1} h_{i+1} + \theta_{i+1} \tag{2.6}$$

2.3.2 Delay in a two layered system

Layer zero: time driven

From Equation (2.6), the time delay at level 1 must be $k_1 h_1 + \theta_1$. From inspection, it can be seen that $k_1 = 0, 1$. If the controller event occurs at p_1 in Figure 2.11 then $k_1 = 0$, at p_2 then $k_1 = 1$. If a random time slice then the probability of either of the intervals occurring is:

$$\mathrm{P}(k = 0) = \frac{\theta_1}{h_1} \tag{2.7}$$

$$P(k=1) = \frac{h_1 - \theta_1}{h_1} \tag{2.8}$$

Note 3 *If the controller level (level 0) is event driven and there is no time delay then k_1 always is equal to zero. If level 0 is event driven but a time delay exists that is shorter than h_1, but longer than θ_1 then $k_1 = 1$*

2.3.3 Time delay at layer i+1

Consider the case where $\tau_i < h_{i+1}$.

From Equation (2.6) it was seen that

$$\tau_{i+1} = k_{i+1}h_{i+1} + \theta_{i+1}$$

for $\tau_i < h_{i+1}$

$$P(k_{i+1} = 0) = \max\left(\frac{\theta_{i+1} - \tau_i}{h_{i+1}}, 0\right) \tag{2.9}$$

$$P(k_{i+1} = 1) = \frac{h_{i+1} - |\theta_{i+1} - \tau_i|}{h_{i+1}} \tag{2.10}$$

$$P(k_{i+1} = 2) = \max\left(\frac{\tau_i - \theta_{i+1}}{h_{i+1}}, 0\right) \tag{2.11}$$

This can be seen from Figure 2.13, using similar techniques to those used in the definite synchronous model that for a random time slice, given that layer i and layer i+1 will have a drift in relation to each other.

Now consider the case where $\tau_i > h_{i+1}$ Here τ_i may be divided into $\tau_i = nh_{i+1} + \overline{\tau_i}$ where $\overline{\tau_i} < h_{i+1}$. Thus for $\tau_i > h_{i+1}$:

$$P(k_{i+1} = n) = \max\left(\frac{\theta_{i+1} - (\tau_i - nh_{i+1})}{h_{i+1}}, 0\right) \tag{2.12}$$

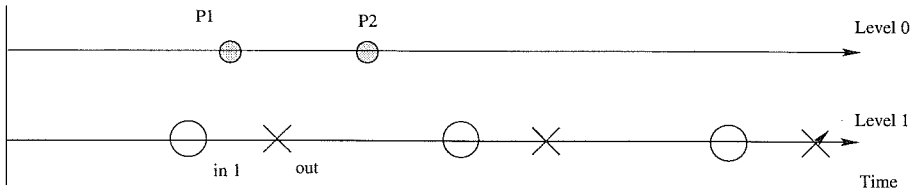

Figure 2.11: Levels 0 and 1 for a partially synchronous system.

$$\mathrm{P}(k_{i+1} = n+1) = \frac{h_{i+1} - |\theta_{i+1} - (\tau_i - nh_{i+1})|}{h_{i+1}} \tag{2.13}$$

$$\mathrm{P}(k_{i+1} = n+2) = \max\Big(\frac{(\tau_i - nh_{i+1}) - \theta_{i+1}}{h_{i+1}}, 0\Big) \tag{2.14}$$

Note 4 *Note that there are only two possible delays for a constant delay on level i, as* $k_{i+1} = h_{i+1}$ *and* $k_{i+1} = h_{i+1} + 2$ *are mutually exclusive.*

2.4 Joining levels

The possible delays on layer L, the bottom layer are given by $k_L h_L + \theta_L$. The probability for any one k is the additive conditional probabilities for all time delay lengths in the layer above.

$$\begin{aligned} \mathrm{P}(k_{i+1} = n) \quad = \quad & \mathrm{P}(k_{i+1} = n | \tau_i = \theta_{i+1}) \mathrm{P}(\tau_i = \theta_{i+1}) + \\ & \mathrm{P}(k_{i+1} = n | \tau_i = h_{i+1} + \theta_{i+1}) \mathrm{P}(\tau_i = h_{i+1} + \theta_{i+1}) + \\ & \mathrm{P}(k_{i+1} = n | \tau_i = 2h_{i+1} + \theta_{i+1}) \mathrm{P}(\tau_i = 2h_{i+1} + \theta_{i+1}) + \\ & \ldots \end{aligned} \tag{2.15}$$

2.4.1 Maximum possible number of different time delays

From this result it can be seen that there is a possible maximum number of time delays for both definite synchronous systems and partial synchronous systems:

For the definite synchronous systems: For layer zero there is no time delay. For level one there is one possible time delay. For level two, two possible delays. Now say for level i there D possible time delays. Then at layer $i+1$ there are $2D$ possible time delays because for each time delay on a layer there are 2 possible time delays on layer on the layer below, see Section 2.2.5.

Thus, for a definite synchronous system the number of possible time delay values is equal to 2^{L-1} where L is equal to the layer number of the last layer.

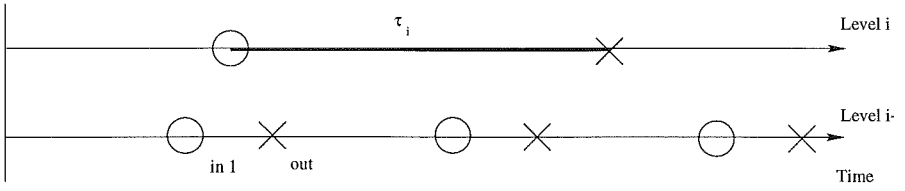

Figure 2.12: Time delay on layer i+1 given phase difference, and constant delay above.

For partially synchronous systems There is no time delay on layer zero, on layer one there are two possible time delays, and on on layer two there are four possible time delays.

Thus for each time delay on a layer there are 2 possible time delays on layer on the layer below, by extension of the results in Section 2.3.3. This shows that, for a system which is partially synchronous, there are 2^L possible time delay values.

2.4.2 Maximum possible time delay

The theoretical maximum occurs when the phase of the input layers and the output layers all just about line up.

Consider 2 layers i and $i+1$, shown in Figure 2.14.

Thus it can be seen that the maximum time is equal to double the added sample periods from all layers except layer 0. i.e

$$\tau_{max} = 2h_1 + 2h_2 + ... + 2h_L \tag{2.16}$$

2.5 Example

This example uses the driving example with three different layers, as well as two different phase differences for each layer. The numbers correspond to the omula simulation used in Nilsson (1998), where $h_1 = 256$ ms, $h_2 = 250$ ms, $\theta_1 = 166$ ms, $\theta_2 = 190$ ms.

Layer one First layer one is examined. Here it can be seen that there are two possible time delays, θ_1 and $\theta_1 + h_1$

$$\mathrm{P}(\tau_1 = \theta_1) = \mathrm{P}(\tau_1 = 166) = \frac{166}{256} = 0.6484$$

$$\mathrm{P}(\tau_1 = \theta_1 + h_1) = \mathrm{P}(\tau_1 = 422) = \frac{90}{256} = 0.3516$$

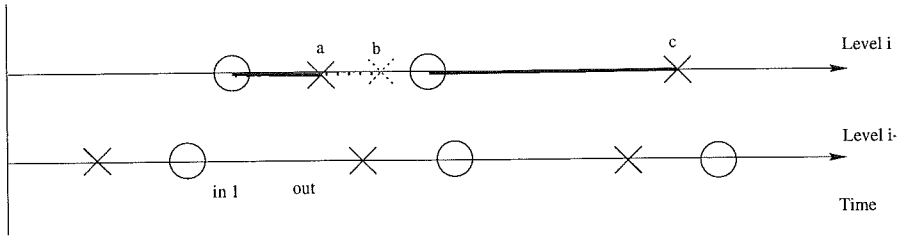

Figure 2.13: Length possibilities for $\tau_i < h_{i+1}$. (a)$k = 0$ (b)$k = 1$ (c)$k = 2$.

The second layer Now the values of the second layer are examined. possible time delays correspond to 190 ms and 440 ms due to the first value of τ_1 and 440 ms and 690 ms due to the second value of τ_1.

From equation (2.15)

$$\begin{aligned} \mathrm{P}(\tau_d = 440ms) &= \mathrm{P}(k_{i+1} = 1|\tau_i = 166 \text{ ms})\mathrm{P}(\tau_i = 166\text{ms}) + \\ &\quad \mathrm{P}(k_{i+1} = 1|\tau_i = 422 \text{ ms})\mathrm{P}(\tau_i = 422\text{ms}) \\ &= \frac{250 - |190 - 166|}{250}\frac{166}{256} + \\ &\quad \max\Big(\frac{190 - (422 - 250))}{250}, 0\Big)\frac{90}{256} \\ &= \frac{1223}{2000} = 0.6115 \end{aligned}$$

$$\begin{aligned} \mathrm{P}(\tau_d = 190ms) &= \mathrm{P}(k_{i+1} = 0|\tau_i = 166 \text{ ms})\mathrm{P}(\tau_i = 166\text{ms}) \\ &= \max\Big(\frac{190 - 166)}{250}, 0\Big)\frac{166}{256} \\ &= \frac{249}{4000} = 0.0622 \end{aligned}$$

$$\begin{aligned} \mathrm{P}(\tau_d = 440ms) &= \mathrm{P}(k_{i+1} = 2|\tau_i = 422 \text{ ms})\mathrm{P}(\tau_i = 422\text{ms}) \\ &\quad \frac{250 - |422 - 250 - 190|)}{250}\frac{90}{256} \\ &= \frac{261}{800} = 0.3263 \end{aligned}$$

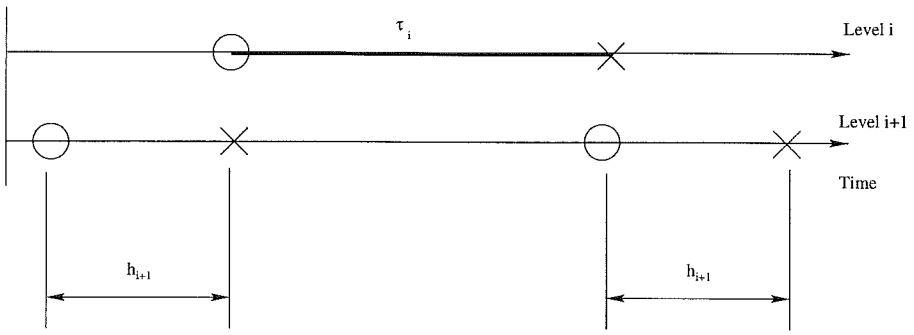

Figure 2.14: The maximum possible time delay for layer i+1 given a phase difference of almost h_{i+1}.

Chapter 3

Probability density modeling

3.1 Introduction

In the previous chapter the delay was modelled as a deterministic model. In this chapter, a different direction is taken. Here the delays are to be modelled as a set of probability density and mass functions.

If the sampling period of each level is known, but not the phase nor the exact timing of the sampling of each level, then the delay can be considered to be a random variable.

Consider a delay on level $i+1$ (Figure 3.1)

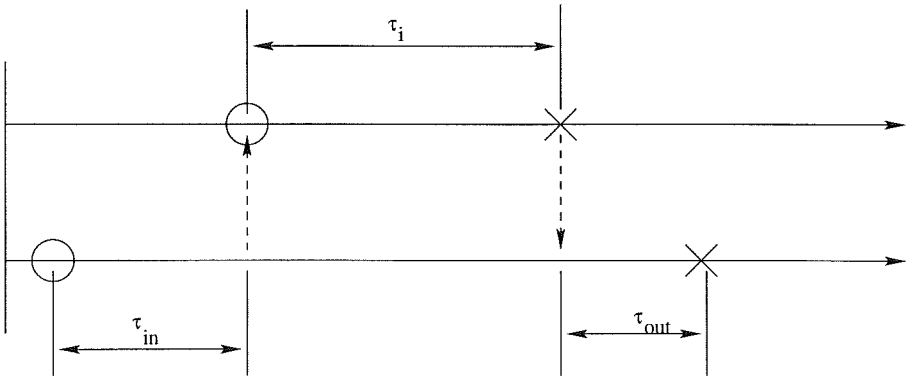

Figure 3.1: The total delay τ_{i+1} is the additive delay from τ_{in}, τ_i, and τ_{out}.

where the total delay can be considered to be.

$$\tau_{i+1} = \tau_{in} + \tau_i + \tau_{out} \tag{3.1}$$

3.2 Time delays as random variables

For a review of some results from probability theory, see appendix A

3.2.1 Random variables

Now if the delays in Figure 3.1 are considered to be instances of random variables, T_{i+1}, T_i, T_{in} and T_{out}, with probability density functions corresponding to $f_{T_{i+1}}, f_{T_i}, f_{T_{in}}$ and $f_{T_{out}}$, then if we consider equation (3.1) and result A.2 it can be seen that the delay τ_{i+1} has a probability density function given by

$$f_{T_{i+1}}(\tau_{i+1}) = f_{T_{in}} * f_{T_i} * f_{T_{out}} \tag{3.2}$$

This is true if and only if T_{in} and T_{out} are independent. I.e. for level $i+1$, h^{in}_{i+1} and h^{out}_{i+1} are considered to be asynchronous.

3.2.2 Delay density functions

T_{in} is a random variable describing the time between the input of layer $i+1$ and layer i, dependent on the input of the two layers. If h^{in}_i / h^{in}_{i+1} is irrational, then τ_{in} can take on any values in $[0, h^{in}_{i+1}]$ in a constant density function, as shown in Figure 3.2. Similarly if $h^{out}_i / h^{out}_{i+1}$ is irrational, then τ_{out} can take on values in $[0, h^{out}_{i+1}]$ in a constant density function.

Example For a simple case, consider τ_i to be constant. Thus it has a probability density function given by a Dirac delta function.

These three functions when convoluted will create the density function shown in Figure 3.3.

3.3 Partially asynchronous systems

Definition 4 *A level is defined here as* partially asynchronous *if h_{in} and h_{out} are rational with respect to each other, that is h_{in} and h_{out} have a highest common factor(h.c.f).*

Let the highest common factor of h_{in} and h_{out} be equal to λ, then we can break equation (3.1) into a multiple of λ, and a residue θ which is the smallest possible phase difference between an input event and an output event, This must be less than λ. This equation is given as:

$$T_{i+1} = \lfloor T_i + T_{in} \rfloor_{n\lambda} + \lceil T_{out} \rceil_{n\lambda+\theta} \tag{3.3}$$

. Now using the results shown in A.4 and A.5 the resulting probability density function is given by

$$f_{T_{i+1}}(\tau_{i+1}) = \sum_{n=0}^{\infty} \delta_{n\lambda} \int_{n\lambda}^{(n+1)\lambda} f_{T_i+T_{in}}(y)dy * \sum_{n=0}^{\infty} \delta_{(n+1)\lambda+\theta} \int_{n\lambda}^{(n+1)\lambda} f_{T_i+T_{in}}(y)dy \tag{3.4}$$

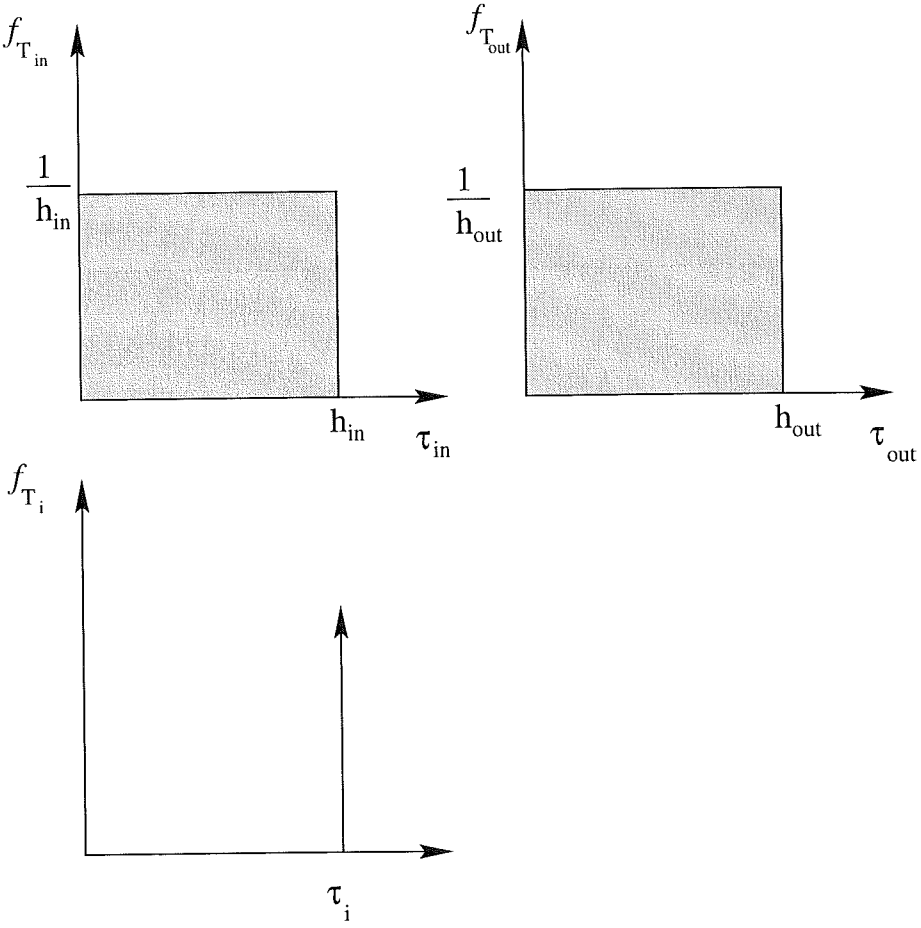

Figure 3.2: The density functions for τ_{in}, τ_{out}, and τ_i, respectively.

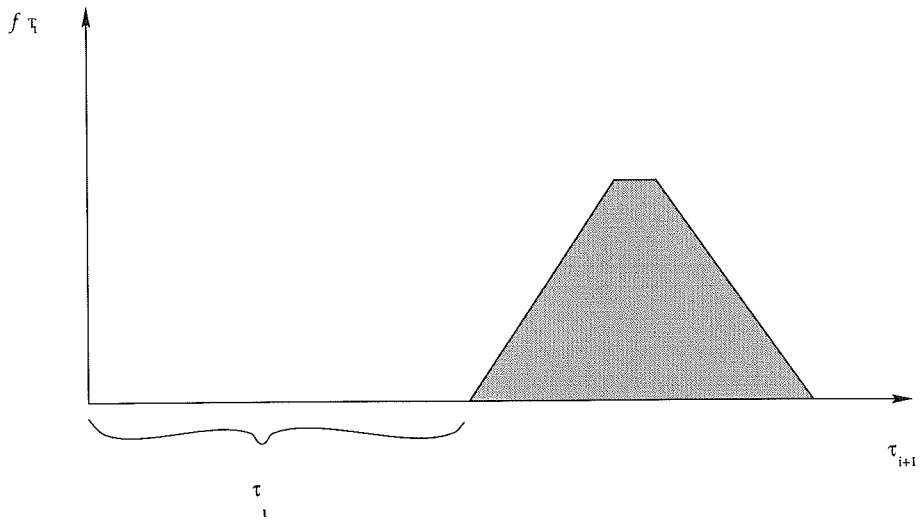

Figure 3.3: The probability density function of τ_{i+1}.

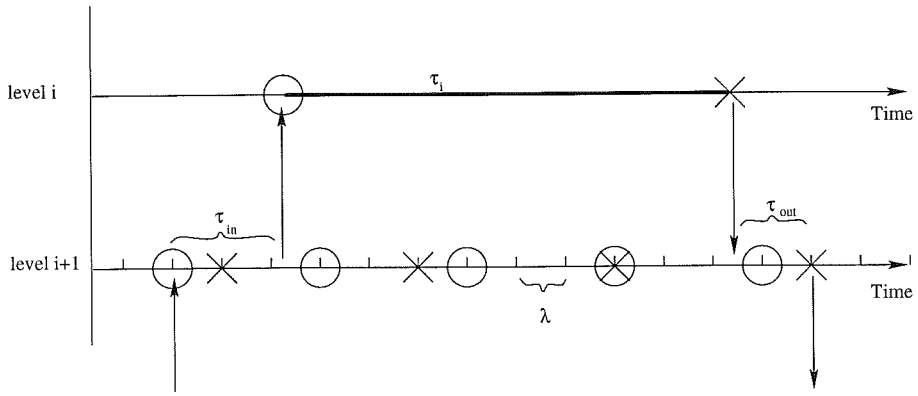

Figure 3.4: A partially asynchronous system

3.4 Partially synchronous systems

Here we will use the previous analysis to reanalyze partially synchronous systems. For a partially synchronous system, h_{in} and h_{out} are both the same, thus the highest common factor can be described as $h_{i+1} = h_{in} = h_{out}$. From equation (3.3) then tout can be described as

$$\begin{aligned} T_{i+1} &= \lfloor T_i + T_{in} \rfloor_{nh_i} + \lceil T_{out} \rceil_{nh_i+\theta} \\ &= \lceil T_{in} + T_i \rceil_{nh_i+\theta} \end{aligned}$$

thus the probability density function can be described as

$$f_{T_{i+1}}(\tau_{i+1}) = \sum_{n=0}^{\infty} \delta_{(n+1)h_{i+1}+\theta} \int_{nh_{i+1}+\theta}^{(n+1)h+\theta} f_{T_i+T_{in}}(y)dy \tag{3.5}$$

3.5 Example

This is a reworking of the example in section 2.5. We take the relative values at each level in turn, then we convolve these results until we gain the suitable f_τ for the lowest layer.

level 1

$$\begin{aligned} f_{T_1}(\tau_1) &= \lceil f_{in} \rceil_{nh+\theta} \\ &= \sum_{n=-1}^{\infty} \delta_{(n+1)h+\theta} \int_{nh+\theta}^{(n+1)h+\theta} \frac{1}{h}.dy \\ &= \delta_{166} \int_0^{166} \frac{1}{256} dy + \delta_{422} \int_{166}^{256} \frac{1}{256} dy \\ &= \delta_{166} \frac{166}{256} + \delta_{422} \frac{90}{256} \end{aligned}$$

level 2

$$\begin{aligned} f_{T_2}(\tau_2) &= \lceil f_{T_1} * f_{T_{in}} \rceil \\ f_{T_1} * f_{T_{in}} = f_{T_1+T_{in}} &= \begin{cases} \frac{156}{(256)(250)} & ; \quad 166 \leq h < 416 \\ \frac{90}{(256)(250)} & ; \quad 422 \leq h < 672 \\ 0 & ; \quad \text{otherwise.} \end{cases} \\ \therefore f_{T_2}(\tau_2) &= \delta_{190} \int_{166}^{190} \frac{166}{(256)(250)} + \delta_{440} \int_{190}^{416} \frac{166}{(256)(250)} + \\ & \quad \delta_{440} \int_{422}^{440} \frac{90}{(256)(250)} + \delta_{690} \int_{440}^{672} \frac{90}{(256)(250)} \\ &= 0.0622\delta_{190} + 0.6115\delta_{440} + 0.3262\delta_{690} \end{aligned}$$

As can be seen this is the continuous random density function version of the discrete random density function gained in the example in section 2.5.

3.6 Other Delay Variations

Consider now a system where level i is modelled as having a time varying transport delay d which is modelled as a random variable D with a probability distribution function of f_D The total delay for the level is now

$$T_{i+1} = T_i + T_{in} + T_{out} + D \tag{3.6}$$

this then has the probability density function given by

$$f_{T_{i+1}}(\tau_{i+1}) = f_{T_{in}} * f_{T_i} * f_D * f_{T_{out}}. \tag{3.7}$$

If the level is partially asynchronous or partially synchronous then the total delay is found in the normal way, except that the random variable D is added.

Example. Consider adding a constant delay to level 0, the controller for our driving example. This delay is set to one second, making f_D a Dirac delta function at 1.

level 1

$$\begin{aligned} f_{T_1}(\tau_1) &= \lceil f_{T_{in}} * f_D \rceil \\ &= \delta_{1024} \int_{1000}^{1024} \frac{1}{256} dy + \delta_{1280} \int_{1024}^{1256} \frac{1}{256} dy \\ &= 0.0938\delta_{1024} + 0.9062\delta_{1280} \end{aligned}$$

level 2

$$\begin{aligned} f_{T_2}(\tau_2) &= \lceil f_{T_1} * fT_{in} \rceil \\ &= \delta_{1250} \int_{1024}^{1250} \frac{24}{(256)(250)} dy + \delta_{1500} \int_{1250}^{1274} \frac{24}{(256)(250)} dy + \\ &\quad \delta_{1500} \int_{1280}^{1500} \frac{232}{(256)(250)} dy + \delta_{1750} \int_{1500}^{1536} \frac{232}{(256)(250)} dy \\ &= 0.0848\delta_{1250} + 0.8065\delta_{1500} + 0.1305\delta_{1750} \end{aligned}$$

this is the same result that is experimentally derived in Figure 5.11

Chapter 4

Sampling time effects

4.1 Introduction

Looking closely at the steps in Figure 1.3 we can see that not only is the time delay varying, but the sampling rate of the system is also varying with time.

4.2 Modeling

By using the model already presented we may gain insight into sampling time variation. In our model we are concerned mainly with the sampling period of each level, as defined by the input, thus we are not concerned whether the system is synchronous, partially synchronous or asynchronous. The output in this case is irrelevant, however here the sampling time of level zero is important.

Definition 5 *The Sampling period of the system h_{system} or more simply just h is defined to be the period of the sampling of the plant, as seen at the controller*

Note 5 *h_{system} above is in actual fact a time varying value, and like the values of the time delays can, given any random controller event be treated like a random variable, which can take on discrete values.*

By examining a single layer i we can see that the only time a signal can enter the layers above are at discrete intervals of $k_i h_i$ where h_i is the sampling period of the layer and k_i takes values in $\mathbb{N}$

4.2.1 A single level

A single level only has a single sampling time, which is invariant. This is also true for a clock driven level 0. This is shown in Figure 4.1

4.2.2 Two layers

In this case there are two layers, layer i and layer $i+1$. Layer i is considered to have a constant sampling rate. There are two distinct cases worth examining, first layer i faster than layer $i+1$, and secondly layer i slower than layer $i+1$.

If layer i is faster than layer $i+1$ Then as can be seen from Figure 4.2, the sampling time for the system above will be constant at h_{i+1}

If layer i is slower than $i+1$ Figure 4.3 shows a system where h_i is greater than h_{i+1}, with the system sampling times above each input event in layer i.

By making $h_i = nh_{i+1} + \overline{h_i}$ and by using sliding intervals it can be seen that the probability of a particular sample time occurring on level i corresponds to:

$$\mathrm{P}(k_{i+1} = n) = \frac{(n+1)h_{i+1} - h_i}{h_i + 1} \tag{4.1}$$

$$\mathrm{P}(k_{i+1} = n+1) = \frac{h_i - nh_i}{h_{i+1}} \tag{4.2}$$

Note 6 *This only applies when $h_{i+1} < h_i$*

Using time stamping it is trivial to find the actual sampling rate of any one sample. The probabilities between levels can be calculated much the same way as for the time delays shown in section 2.4

4.2.3 General results

From the analysis above some general results can be seen.

- The number of different possible sampling rates for a system is 2^{L-1} where L is the number of different layers of the system. sampling period of all of the levels.

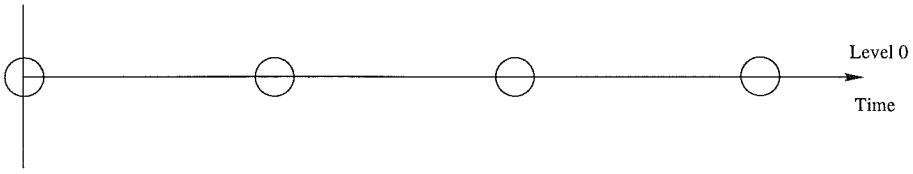

Figure 4.1: The sampling rate of a single level.

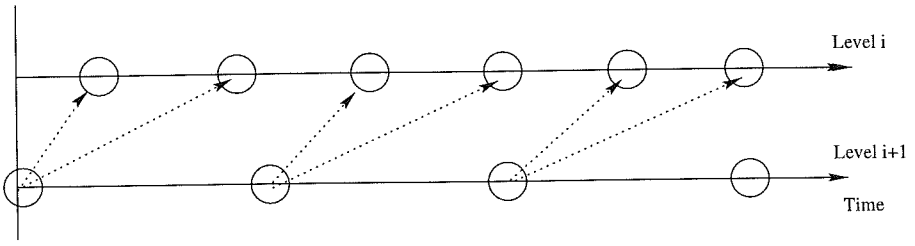

Figure 4.2: The system sampling rate for a system with h_i faster than h_{i+1} is constant at h_{i+1}.

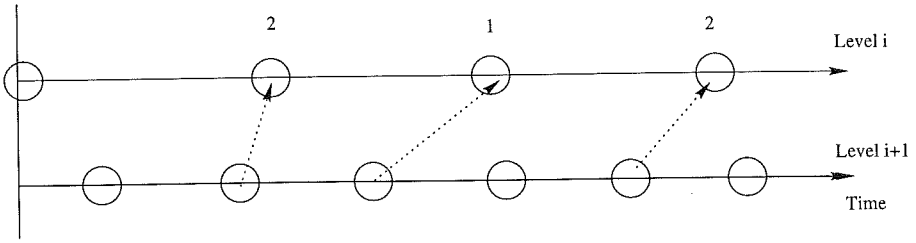

Figure 4.3: The system sampling rate for a system with h_i slower than h_{i+1} has varying sample time.

4.3 Examples

4.3.1 Driving example

$$\begin{aligned} \mathrm{P}(h_2 = 1000\mathrm{ms}) &= \mathrm{P}(h_2 = 1000|h_1 = 1024\ \mathrm{ms})\mathrm{P}(h_1 = 1024\ \mathrm{ms}) \\ &= \frac{226}{250}.\frac{180}{256} \\ &= \frac{1017}{1600} = 0.6356 \end{aligned}$$

$$\begin{aligned} \mathrm{P}(h_2 = 1250\mathrm{ms}) &= \mathrm{P}(h_2 = 1250|h_1 = 1024\ \mathrm{ms}).\mathrm{P}(h_1 = 1024\ \mathrm{ms}) + \\ &\quad \mathrm{P}(h_2 = 1250|h_1 = 1280\ \mathrm{ms}).\mathrm{P}(h_1 = 1280\ \mathrm{ms}) \\ &= \frac{24}{250}.\frac{180}{256} + \frac{220}{250}.\frac{76}{256} \\ &= \frac{263}{800} = 0.3287 \end{aligned}$$

$$\begin{aligned} \mathrm{P}(h_2 = 1500\mathrm{ms}) &= \mathrm{P}(h_2 = 1500|h_1 = 1280\ \mathrm{ms})\mathrm{P}(h_1 = 1280\mathrm{ms} \\ &= \frac{30}{250}.\frac{76}{256} \\ &= \frac{57}{1600} = 0.0356 \end{aligned}$$

4.4 One possible practical example

The sample time variation in Figure 4.4 is taken from the PhD of Johan Nilsson(Nilsson 1998), and is associated with the sapling time of Windows NT, using one of its real time modes. Notice the symmetric sample time variation in a bell curve pattern, at discrete intervals.

Windows NT like most operating systems consists of many "network layers", from the physical layer, up to the application layer. The bottom layer in this case and the one which will determine the sampling times is probably the physical level or the level above. An investigation of the layered architecture of windows NT would reveal more, but it could be possible that the modeling could be based on the research of this paper.

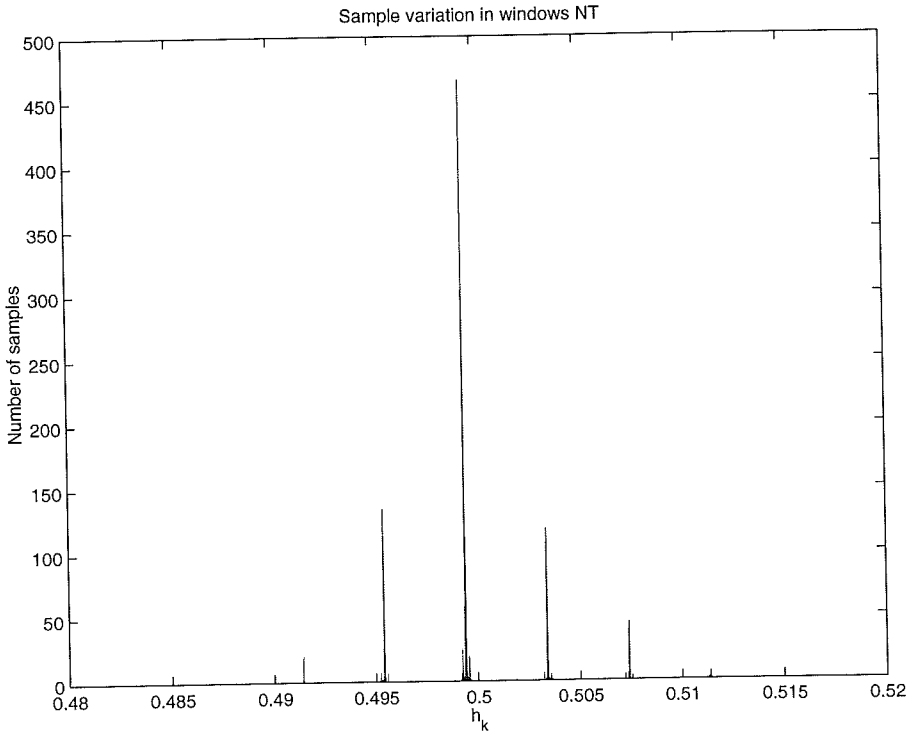

Figure 4.4: The sampling interval variation for windows NT. The sampling time was programmed to be 500 ms.

Chapter 5

Simulations

5.1 Introduction

This chapter deals with a set of simulations done using Simulink, and a simple model using zero order hold structures. It follows the course of the author's experiments and the steps in assumptions and conclusions made about the system models. The experiments were done using the Simulink representation of the system shown in Figure 5.1. The following system parameters were the basic setup, and are the ones which were used if not specifically stated otherwise in the simulation documentation:

$h_s = 250$ ms The sampling frequency of the of the AD and DA converters.

$h_b = 256$ ms The rate of the mirroring of the field-bus.

$h = 1.1$ s The rate of the controller.

These values, while not entirely typical of modern controlling hardware nonetheless provide an accurate enough qualitative description of what occurs with the time delay, and are the values given in the driving example mentioned in the introduction.

When this basic system is simulated, the resulting delay shown in Figure 1.2. was produced. As can be seen this noise on initial inspection appears to have a certain amount of pattern, and thus is not truly random in nature.

The system shown returned a discrete output signal, which was filtered to produce the time delay. This time delay was analyzed with a number of functions, most commonly in these experiments by the use of histograms, which reflect proportionally the probability mass function of the delay, when the histogram is fine enough. The histograms used here were made up of 1250 separate columns.

5.2 A definite synchronous system

5.2.1 Basic model

First the basic response of the system is examined, given the values which are shown above. The relative values of the time delay that were output is shown in Figure 5.2

Here there can be seen a maximum at 0.5 s, with a very small minimum at 0.75 s. This reflects the results found in the section on modeling (Section 2.2).

5.2.2 Control layer sample time

Three different values of controller sample time were used and effects on the total time delay of using each were noted. The values of controller time were 1.1, 2.0 (Figure 5.3), 0.2, $0.5 and 0.75$. None of the results showed much variation.

As can be seen there is very little difference here. It can thus be concluded that the controller sampling rate doesn't play a large part in the time delay, at least at the order of magnitude at which the system was simulated.

5.2.3 I/O device sample time

The sampling value of the I/O device was changed, however input and output were kept synchronous. The values of sampling time were set to 150, 200, 260 and 300 respectively.

The resulting histograms for the time delays were shown below, (Figures 5.4,5.5)

Here there is a certain amount of time delay variation, which seems to be somewhat related to the relation between the sample time of the bus and the sample time of the I/O device. More importantly note that the delay time remains a integral multiple of the sample time of the IO device. This verifies the result of equation 2.1.

5.3 A partially syncronous system

5.3.1 Input phase change

The phase of the AD converter was changed, in increments of .05, while the phase of the out DA converter remained the same. (Figures 5.6, 5.7)

Here the important result is the time values of the delays, which occur at equal intervals. These intervals are equal to $k\tau_s - \theta_{in}$, where θ_{in} is the time lag if the input, and $k = 1, 2, 3 \ldots$. The occurrence of these delays seems to reflect some probability mass function.

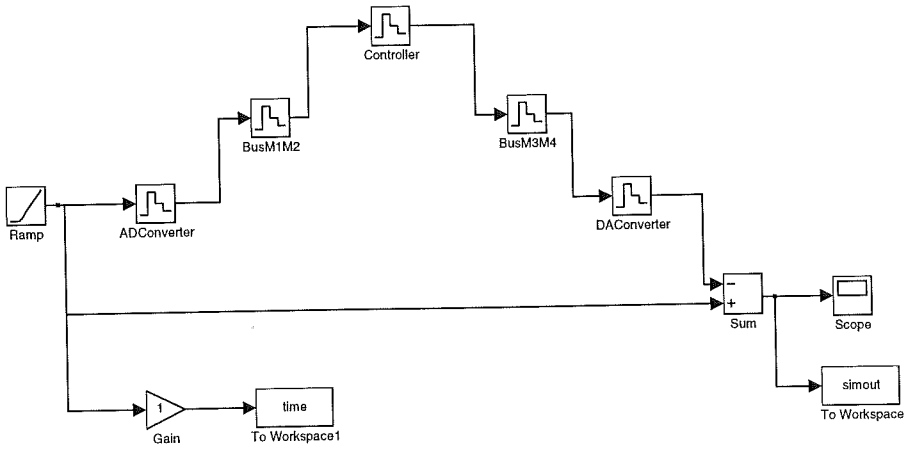

Figure 5.1: The Simulink simulation used for the experiments.

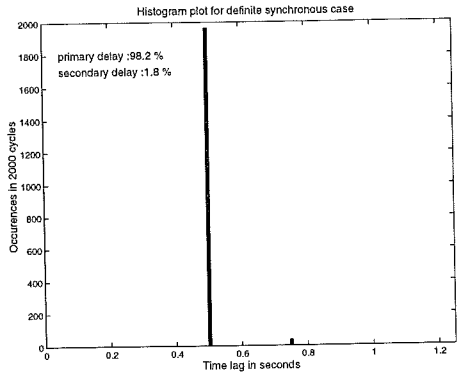

Figure 5.2: A histogram describing the time delays for the driving example simulated in the definite synchronous case.

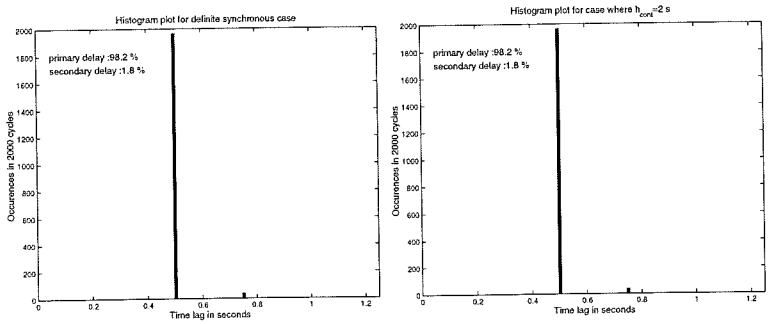

Figure 5.3: Control Sample time=1.1 and 2.0 respectively.

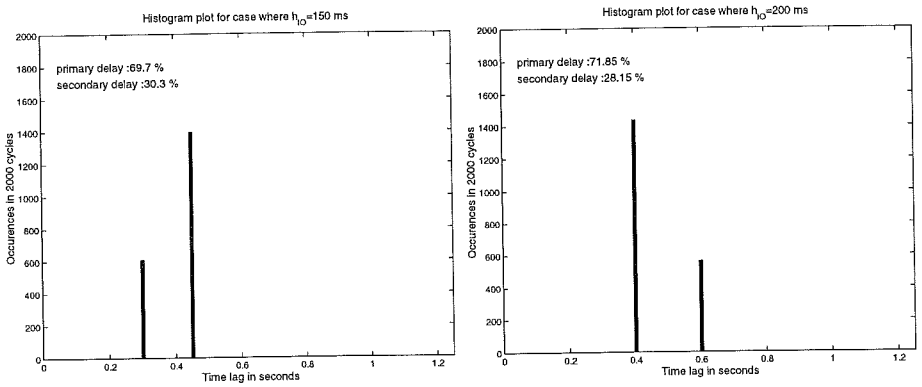

Figure 5.4: I/O sample time = 150, 200 ms respectively.

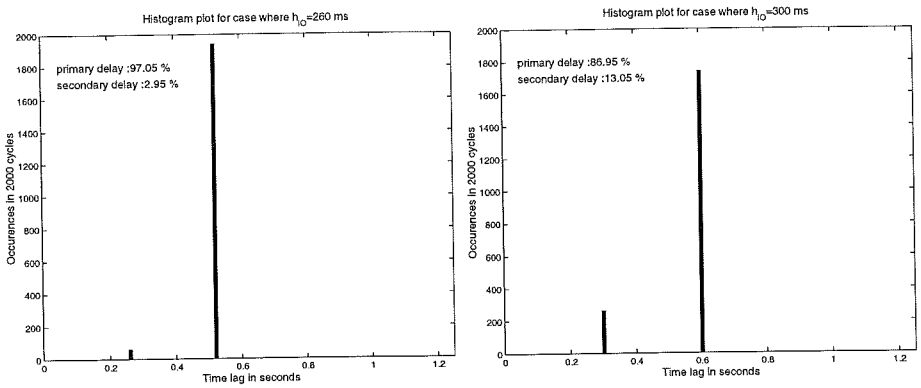

Figure 5.5: I/O sample time = 260, 300 ms respectively.

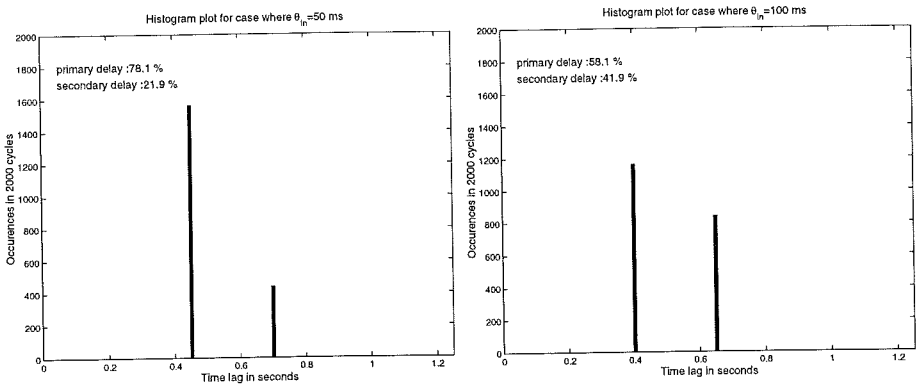

Figure 5.6: I/O sample time = 150, 200 ms respectively.

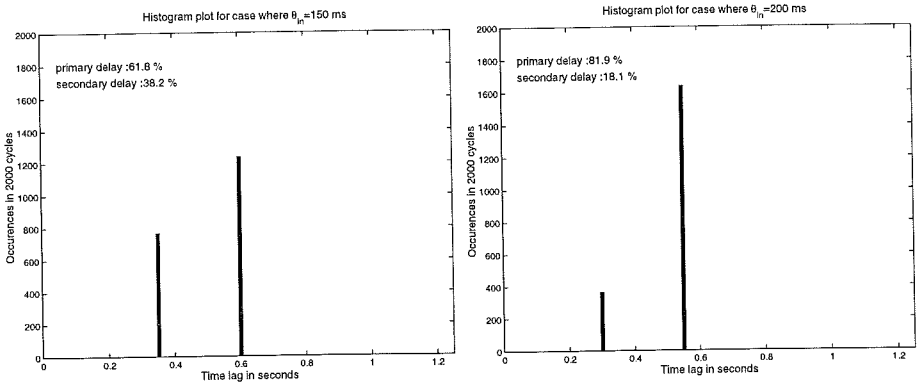

Figure 5.7: I/O sample time = 260, 300 ms respectively.

5.3.2 Output phase change

This Experiment is similar to the last except that the phase of the output was changed, rather than that of the input.

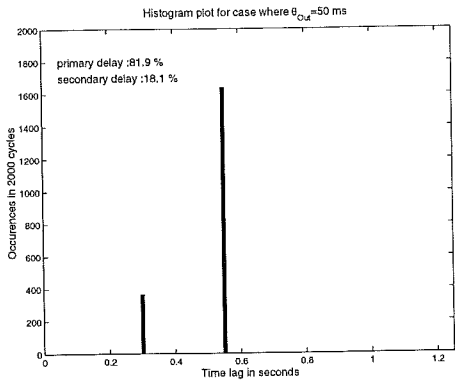

Figure 5.8: Input lag=200 ms.

This caused similar results to those in Section 5.3.1. However here the values of the Time delay take values of $k\tau_s + \theta_{out}$ where θ_{out} is the output lag, and $k = 1, 2, 3, \ldots$

5.3.3 I/O phase change

This produced no change to the standard model. What this means is that the timing of the I/O itself is irrelevant to the time delay but that the the phase *difference* between the two delays is very important.

5.4 An asychronous system

5.4.1 Output device sample time

The sample speed of the output only was set to 260 ms, while the input was left at the normal level of 250 ms. It was expected here for the input and the output delays to now be out of step, thus producing a more random time delay pattern. The simulation was run and the results were shown in Figure 5.9

Here the histogram shows some more interesting results, a time delay which varies as a set discrete values in a triangular distribution. Notice that the values of the time delay are separated into quanta separated by 0.01 s; that is, the difference between the input and output sampling rates. This result now can be analyzed similarly to a randomly varying time delay.

5.4.2 Input device sample time

Similarly to the previous simulation,the sample speed of the input was set to 260 ms, while the output sample speed was set to 250 ms. Again it was expected that the result would be a more randomly varying time delay. The results are shown in Figure 5.10.

Again the histogram shows a distinct triangular form, indicating that the process is again a more random process.

Both of the previous simulations have some importance, although this importance is system dependent. Many controller systems have single subsystems which contain both input and output. These subsystems will (usually) have a single clock. This being the case there is only a small amount of randomness associated with the time delay, as shown in the earlier simulations.

However, in some systems it is possible to have separate subsystems for both the input and the output. In this case there is no guarantee of a uniform time delay, but rather a random delay as indicated by these last two simulations.

5.5 Other complications

5.5.1 Sampling delay

The control process shown in Figure 5.1 seems quite unrealistic in some ways, the main one of these is the instantaneous nature of the sampling of the controller, and other components which are modeled here as zero order holds. To examine the effects of a time delay on the total time delay of the system, an invariant time delay was added to the model just after the zero order hold of the controller. The resulting histogram is shown in Figure 5.11.

Here the histogram shows a simple shift of the delay with a small amount of scattering to the I/O clock times to either side of the main delay. This makes sense as the delay simply shifts the overall delay of the system.

5.6 Test simulation

In sections 2.5 and 3.5 a particular system of delays was modelled. These had the following values $h_1 = 256$ ms, $h_2 = 250$ ms, $\theta_1 = 166$ ms, $\theta_2 = 190$ ms.

This system was simulated and the results are shown in figure 5.12. As can be seen the results confirm both the results in section 2.5 and section 3.5

5.7 Conclusions

In this series of simulations we examined the effects on the system of varying different system parameters and the effects that these had on the time delay of the system. In the first section the effects on the system of changing the sampling time of the controller were examined. This was found to have almost no effect on the resulting time delay of the basic system. Similarly in large part,

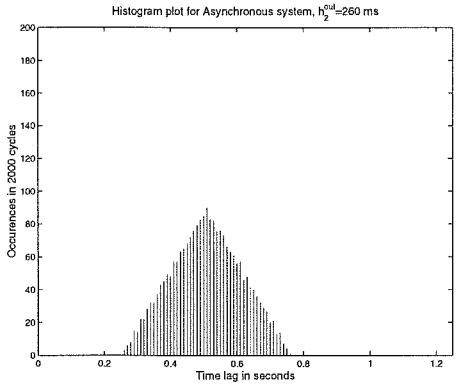

Figure 5.9: Output sampling rate=260ms.

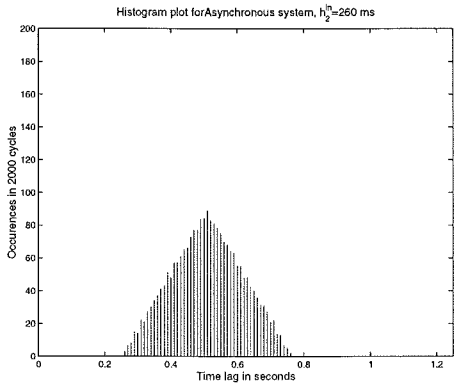

Figure 5.10: Input sampling rate=260ms.

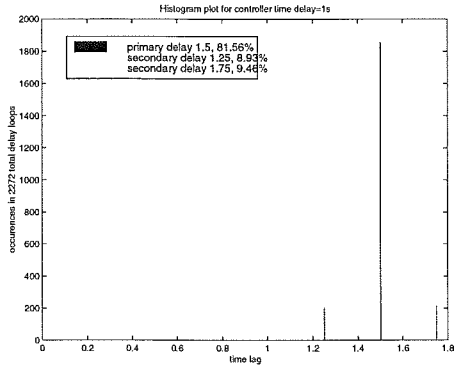

Figure 5.11: Controller delay=1 s.

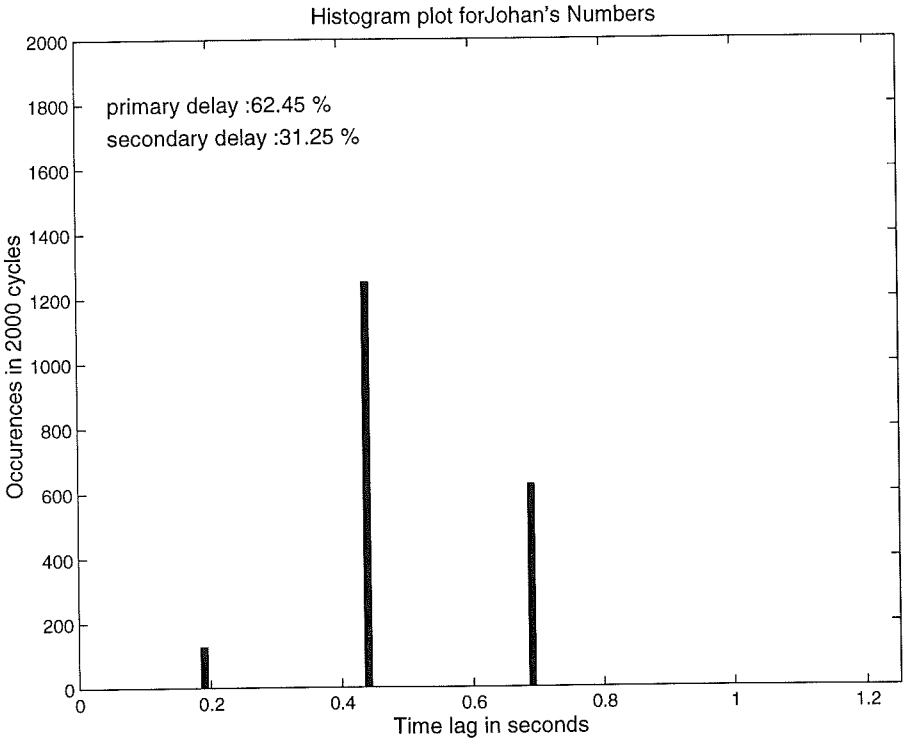

Figure 5.12: Simulation verification of theoretical results.

was the sample time of the field bus which was seen to have little effect on the system. What did play a very large role in the placement of the time delay of the system was the characteristics of the IO device used to read and write the functions. There were two distinct patterns of time delay placement, depending on whether or not the I/O blocks of the system were synchronous or not.

In the case where the I/O blocks were synchronous to one another then the total delay took on discrete values at levels corresponding to the I/O sampling rate which was modified by the difference in lag between the input and the output thus the time lag took on quantized values corresponding to $k\tau_s + (\theta_{out} - \theta_{in})$.

In the case where the I/O blocks were not synchronous the total delay took on quantized values corresponding to the difference between the input sampling rate and the output sampling rate. Again the occurrence of each time delay level was governed by an unknown probability function. It is hypothesized that this function is not a function of the sampling frequency of the I/O unit, or is at least, not strongly affected by it.

Chapter 6

Controller design

6.1 Introduction

In this section we use the model that has been developed for the partially synchronous case for a control design, using LQG methods. The first section covers a description of the plant and controller design model, and the requirements for design, then follows a review of results from previous work on such controller design, mostly from the work of Nilsson (1998). Three separate controller paradigms are explored, with increasing complexity, and finally these three paradigms are compared in simulation.

6.2 Time stamping

Using synchronized clocks at the network nodes, information about the time delays between sampling can be extracted. The overhead caused by this can mostly be seen as being negligible in comparison to message and network overhead (Nilsson 1998). By adding time stamping to the system the model can be reduced from a discrete random function, to being completely deterministic.

Consider the case of a partially synchronous system. The information needed for the given discrete random model is:

- Sampling rate h_i of every layer.
- Phase difference θ_i of every layer.

By adding time-stamp information (such information will in actual fact be needed for the clock synchronization anyway), we can reduce this to a deterministic system.

Example Consider the three layer system of the driving example, using the numbers suggested by Nilsson; $h_1 = 256, h_2 = 250, \theta_1 = 166$ and $\theta_2 = 190$.

For this cycle, the information given by the time stamps is that the signal was input by the AD converter at time 0 ms, the information was mirrored by the bus at time 85 ms, and the controller sampled at time 100 ms. Figure 6.1 shows how the time delay may be determined from time stamp information. Here there is 58 ms between the input on the I/O unit and the time delay above. By knowing the time delay of the bus we can know the total time delay. The bus time delay can be determined from the time of the controller event after the bus event, which is in Figure 6.1 42 ms. From this it can be seen that the delay between the two bus mirroring events will be 166 ms.

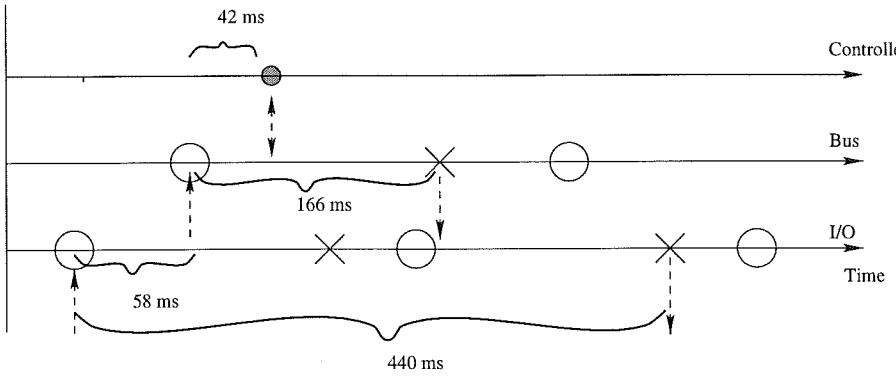

Figure 6.1: By using time stamping the total delay can be easily found.

By adding this delay on level one to the input delay on level two, the delay between the AD-conversion of the plant signal and the DA-conversion of the Actuator signal can be calculated to be $h_1 + \theta_1 = 440$ ms.

Thus determining the delay can be done with a simple recursive algorithm:

```
Totaltime(i)
        if i := 0
                return 0
        else
                abovedelay := Totaltime(i − 1) + τ'
                integer n_i := floor(abovedelay/h_i)
                if abovedelay ≤ n_i h_i + θ
                        return n_i h_i + θ
                else
                        return (n + 1)h_i + θ
                end
        end
end.
```

6.3 Design requirements

6.3.1 Plant analysis

A continuous time plant, when linearized is represented by

$$\dot{x} = Ax + Bu + v \tag{6.1}$$
$$y = Cx + D \tag{6.2}$$

where $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ and $v(t) \in \mathbb{R}^n$. A and B are the appropriate matrices, $x(t)$ is the state vector, $u(t)$ is the input vector and $v(t)$ is white noise with zero mean and incremental variance R. The system is also known to have a time varying time delay $\tau(t)$. This time delay for a continuous time system is an infinite dimensional system, however, when sampled this time delay becomes a finite dimensional system. When (6.1) is integrated over a sampling interval the system can then be represented by

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k)u_k + \Gamma_1(\tau_k)u_{k-1} + v_k \tag{6.3}$$

where

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma_0(\tau) &= \int_0^{h-\tau} e^{As}dsB \\ \Gamma_1(\tau) &= e^{A(h-\tau)}\int_0^{\tau} e^{As}dsB \end{aligned}$$

the output is given by

$$y_{k+1} = Cx_k + w_k$$

This is a standard result from digital control theory, for instance, see Åströ m and Wittenmark.

6.3.2 Controller analysis

A linear controller which addresses the regulation problem for this system could be described as

$$x^c_{k+1} = \Phi^c(\tau_k)x_k + \Gamma^c(\tau_k)y_k \tag{6.4}$$
$$u_k = C^c(\tau_k)x_k + D^c(\tau_k)y_k \tag{6.5}$$

The use of τ_k in the controller hare means that the controller knows either partially, or totally the delays of the network.

Thus the closed loop system can be written as

$$z_{k+1} = \Phi(\tau_k)z_k + \Gamma(\tau_k)e_k \tag{6.6}$$

where

$$\begin{aligned} z_k &= \begin{bmatrix} x_k \\ x_k^c \\ u_{k-1} \end{bmatrix} \\ \Phi(\tau_k) &= \begin{bmatrix} \Phi + \Gamma_0(\tau_k)D^c(\tau_k)C & \Gamma_0(\tau_k) & \Gamma_1(\tau_k) \\ \Gamma^c(\tau_k)C & \Phi^c(\tau) & 0 \\ D^c(\tau_k)C & C^c(\tau_k) & 0 \end{bmatrix} \\ e_k &= \begin{bmatrix} v_k \\ w_k \end{bmatrix} \end{aligned}$$

and

$$\Gamma = \begin{bmatrix} I & \Gamma_0(\tau_k)D^c(\tau_k) \\ 0 & \Gamma^c(\tau_k) \\ 0 & D^c(\tau_k) \end{bmatrix} \tag{6.7}$$

The variance R of e_k is given by

$$R = E(e_k e_k^T) = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

6.3.3 Control aims

the aim is to design a controller which minimizes the cost function

$$J = E\{\lim_{T\to\infty} \frac{1}{T}\int_0^T (x^T H^T H x + u^T G^T G u)dt\}$$

This is a common aim for LQG controllers.

6.4 Results from optimal stochastic control

Most of this section is taken from Nilsson(1998), and the reader is referred to this reference for proofs.

The theorems in Nilsson (1998) assume that the network delays are stochastically independent, that is that every delay is independent of the previous time. This is not entirely true for this case, however it is a viable enough approximation. An assumption is also made that the $\tau \leq h_{cont}$ a valid enough assumption given choice of h_{cont}.

There are three important results to see here. First is the optimal state feedback given full state information, then the optimal state estimate, and finally the LQG problem with output feedback is examined.

6.4.1 Optimal state feedback

Given the plant with noise free measurement of the state vector x_k i.e, $y_k = x_k$. the control law that minimizes the cost function

$$J_N = E(x_N^T Q_N x_N + \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T Q \begin{bmatrix} x_k \\ u_k \end{bmatrix}) \tag{6.8}$$

where Q is symmetric with the structure

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \tag{6.9}$$

is given by

$$u_k = -L_k(\tau) \begin{bmatrix} x_k \\ u_{k+1} \end{bmatrix} \tag{6.10}$$

where

$$\begin{aligned}
L_k(\tau) &= (Q_{22} + \tilde{S}_{k+1}^{22})^{-1}[Q_{12}^T + \tilde{S}_{k+1}^{23} \quad \tilde{S}_{k+1}^{23}] \\
\tilde{S}_{k+1}(\tau) &= E_{\tau_k}\{G^T(\tau_k) S_{k+1} G(\tau_k)\} \\
G(\tau_k) &= \begin{bmatrix} \Phi & \Gamma_0(\tau_k) & \Gamma_1(\tau_k) \\ 0 & I & 0 \end{bmatrix} \\
S_k &= E\{F_1^T(\tau_k) Q F_1(\tau_k) + F_2^T(\tau_k) \tilde{S}_{k+1}(\tau_k) F_2(\tau_k)\} \\
F_1(\tau_k) &= (Q_{22} + \tilde{S}_{k+1}^{22})^{-1} \begin{bmatrix} (Q_{22} + \tilde{S}_k^{22} + 1)I & 0 \\ -(Q_{12}^T + \tilde{S}_{k+1}^{21}) & -\tilde{S}_{k+1}^{23} \end{bmatrix} \\
&= \begin{bmatrix} I & 0 \\ -L_k(\tau_k) \end{bmatrix} \\
F_2(\tau_k) &= \begin{bmatrix} I & 0 \\ -L_k(\tau_k) & \\ 0 & I \end{bmatrix} \\
S_N &= \begin{bmatrix} Q_N & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

Here $\tilde{S}_{k+1}^{ij}$ is block (i,j) of the symmetric matrix $\tilde{S}_{k+1}(\tau)$ and $Q_{i,j}$ is block (i,j) of Q.

6.4.2 Optimal state estimates

Sometimes it is difficult or impossible, whether for practical or for theoretical reasons to form a full state feedback of a system, so a system of state estimation is used instead. Here there is a problem with random time delays entering the system. However, because we know the characteristics of the stochastic time delay, and all of the past time delays, then a standard Kalman filter is optimal, due to the fact that x_k only depends on previous time delays.

The state estimator which minimizes the error covariance given the plant model specified in equation (6.1)

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{K}_k(y_k - C\hat{x}_{k|k-1}) \tag{6.11}$$

$$\begin{aligned}
\hat{x}_{k+1|k} &= \Phi\hat{x}_{k|k-1} + \Gamma_0(\tau_k)u_k + \Gamma_1(\tau_k)u_{k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \\
\hat{x}_{0|-1} &= E(x_0) \\
P_{k+1} &= \Phi P_k|\Phi^T + R_1 - \Phi P_k C^T[CP_kC^T + R_2]^{-1}CP_k\Phi \\
K_k &= \Phi P_k C^T[CP_kC^T + R_2]^{-1} \\
\bar{K}_k &= P_k C^T[CP_kC^T + R_2]
\end{aligned}$$

minimizes the error variance $E\{[x_k - \hat{x}_{k|k}]^T[x_k - \hat{x}_{k|k}]\}$. The estimation error is Gaussian with zero mean and covariance $P_{k|k} = P_k - P_kC^T[CP_kC^T + R_2]^{-1}CP_k$

Note 7 *The filter gains K_k and $\bar{K}_k$ do not depend on τ_k. This follows from that τ_k doesn't enter the matrix Φ.*

6.4.3 Optimal output feedback

Given the plant, with all past delays known then the control signal is calculated, the controller that minimizes the cost function is given by

$$u_k = -L_k(\tau)\begin{bmatrix} \hat{x_k} \\ u_{k+1} \end{bmatrix} \tag{6.12}$$

with

$$L_k(\tau) = (Q_{12}^T + \tilde{S}_{k+1}^{22})^{-1}[Q_{12}^T + \tilde{S}_{k+1}^{21} \quad \tilde{S}_{k+1}^{23}]$$

Where $\hat{x}_{k|k}$ is the minimum variance estimate. This and $\tilde{S}_{k+1}$ are calculated from sections 6.4.2 and 6.4.1 respectively.

6.5 Controller paradigms

6.5.1 LQG design - No time delay

In this case there is no compensation for the time delay whatsoever, an its effect on the closed loop system is ignored in the design. This is the basic LQG model.

The problem with this paradigm is that it will become unstable for a large enough time delay, as shown in simulation in section 6.6.2

6.5.2 LQG design - Average time delay

Here the average time delay is used to design the controller, first the delayed plant is sampled, then the Kalman filter is calculated from that value. This result will produce a stable system for a much greater range of time delay than the previous system, but the control isn't stable for all $\tau \leq h$ and the control is conservative, as seen in simulations in Sections 6.6.2 and 6.6.3.

6.5.3 LQG design - Time varying time delay with known probability density function.

Here we use the results from Sections 6.4.1, 6.4.2 and 6.4.3 to form an optimal controller based on the probability density function and the current time delay formulated in previous chapters.

6.6 Simulation results

6.6.1 Basic system.

Consider the following plant, where both plant and design specifications are taken from Doyle and Stein (1979),

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \begin{bmatrix} 35 \\ -61 \end{bmatrix} \xi \\ y &= [2 \quad 1]x + \eta \end{aligned} \tag{6.13}$$

where ξ and η have mean zero and unit incremental variance. The control objective is to minimize the cost function

$$J = E\left(\lim_{T\to\infty} \frac{1}{T} \int_0^T (x^T H^T H x + u^2) dt\right) \tag{6.14}$$

where $H = 4\sqrt{5}[\sqrt{35} \quad 1]$. The sampling period for the controller is chosen as $h = 0.05$. This is in accordance with the rule of thumb that is given in Åström and Wittemark(1997). The sampling periods of the bus and I/O are chosen according to each experiment. This will effect the time delay of the system. The system being controlled has input and output sample periods the same, and is thus a partially synchronous system.

There are three controllers used in this experiment. The first of these is the controller design using only normal LQG methods without any change of time delay, the second uses an approximation of the average time delay, and the third uses the exact theoretic nature of the time delay density function and the current time delay to find an optimal solution.

The first controller is done without taking any time delays into account. The process and cost function are sampled to get discrete time equivalents, and the standard LQG controller is calculated. This gives the design

$$L = \begin{bmatrix} 38.911 \\ 8.094 \end{bmatrix}^T \quad K = \begin{bmatrix} 2.690 \\ -4.484 \end{bmatrix} \quad \bar{K} = \begin{bmatrix} 2.927 \\ -5.012 \end{bmatrix} \tag{6.15}$$

The second design uses an estimate of the mean delay, $h_{I/O} + h_{bus}$, by discretizing the plant and the cost function, and then designing an appropriate Kalman filter and estimated state feedback.

The third design was gained by using Nilsson's probability density function and time delay LQG tool-boxes in MATLAB to find appropriate values for L. The loss function was calculated using a Monte Carlo simulation.

The basic system simulated is shown in Figure (6.2).

There are two separate experiments done, one on the effects of changing the average time delay, which is done by changing the values of $h_{I/O}$ and h_{bus}, and the other of the effect of changing the phase of the I/O layer which allows us to see the differences between the constant delay controller and the gain scheduled controller.

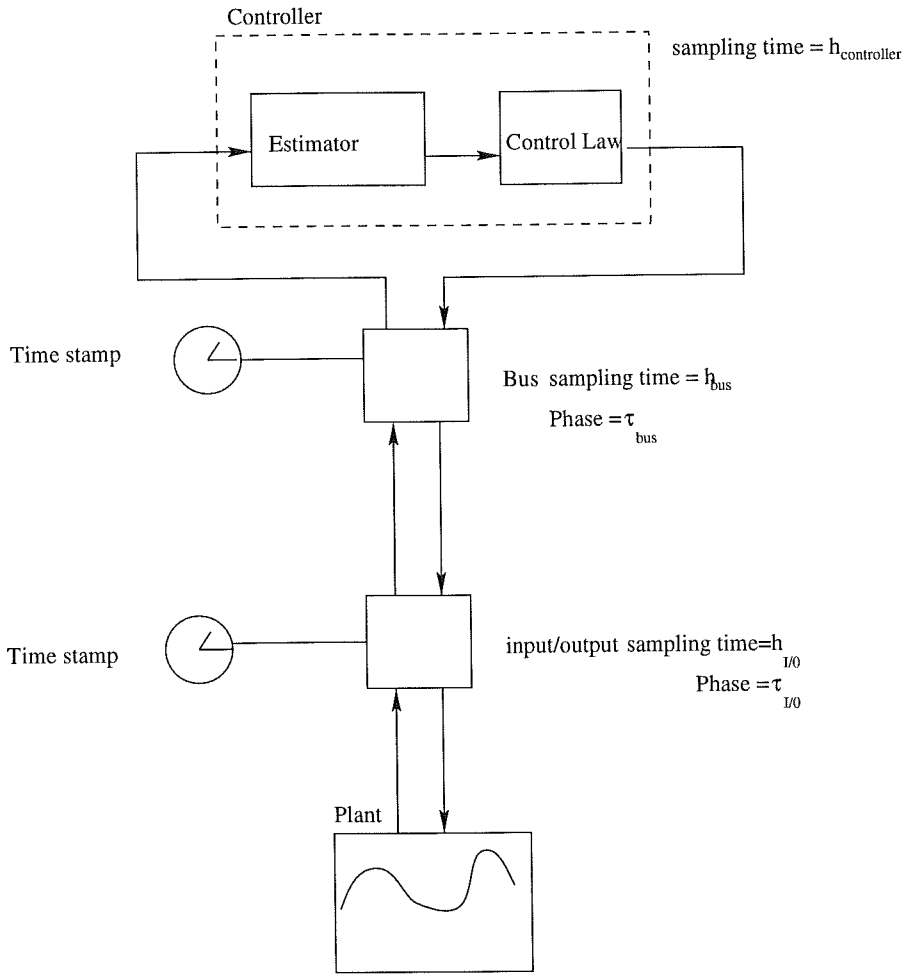

Figure 6.2: The control system used in simulation.

6.6.2 Delay variation

In this experiment the average time delay was changed by changing $h_{I/O}$ and h_{bus} simultaneously, giving a varying average time delay. Using this the loss function was plotted against average $h_{I/O}$, the value that was varied. h_I/O was varied between 0 and 0.025, h_{bus} was set to $1.06h_{I/O}$, and τ_{bus} to $0.5h_{I/O}$. The cost function was then plotted for the simple LQG controller the controller with delay, and the gain sheduled controller, versus the of the controller.

As can be seen in Figure 6.3, the basic LQG controller quickly becomes unstable, the constant time delay controller, reacts well, but not so well as the gain scheduled controller, because the gain scheduled controlles takes the exact time delay into account.

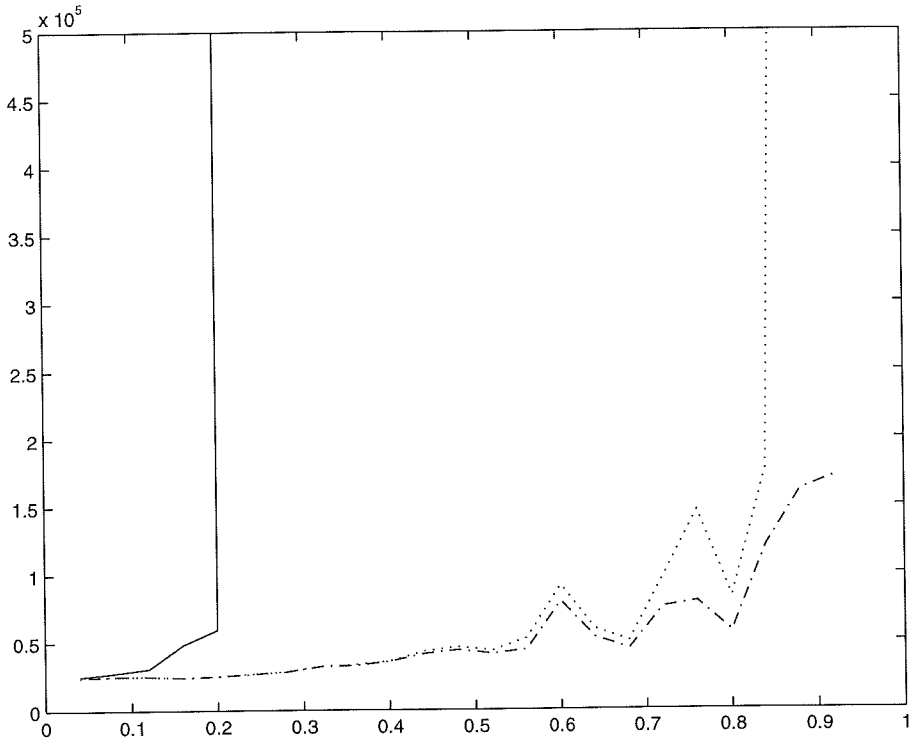

Figure 6.3: The reactions to the three different controllers to an increase in the lower level sampling frequencies. Here solid line is controller designed for no time delay, dotted is for controller with mean constant time delay, and the dash-dotted is for the gain scheduled controller.

6.6.3 Phase variation

In this experiment $h_{I/O}$ was set to 0.011 and h_{bus} was set to 0.0113. At the level of time delay caused by the interaction between these sampling times the simple LQG controller was unstable, however both the average time delay, and the gain scheduled controller were both stable.

The phase of the bus, θ_{bus} was set to 0.005, and the phase of the I/O level, $\theta_{I/O}$, was varied, from 0 to 0.01, comparing the cost function of both for these values.

These are shown in figure 6.4

Here it can be seen that the gain scheduled controller has results about two thirds of the average loss function of the mean delay controller.

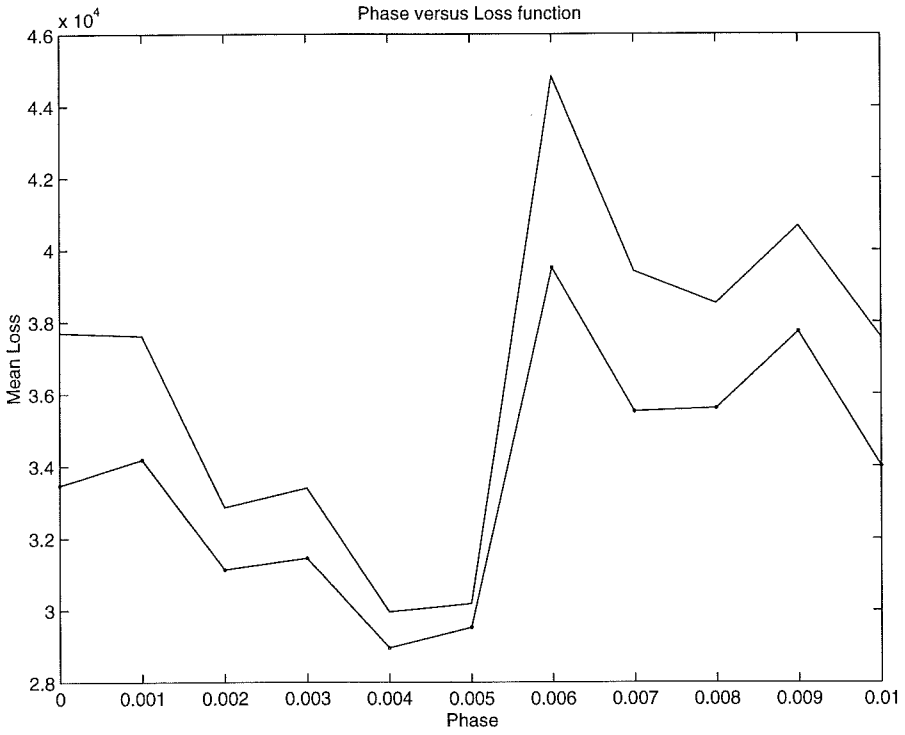

Figure 6.4: Loss function versus phase for the constant delay controller(above) and gain scheduled controller(below)

Chapter 7

Conclusions

In this thesis a delay caused by a series of asynchronous loops has been examined, especially for the example of a controller and a process both on different sampling times and separated by a field bus, again with a set of sampling times.

In chapter 2 it was shown that given a certain amount of synchronicity on the lowest level that the time delay can only take on certain values, and that the relative frequency of these values can be determined by using simple ratios between two levels. In this way a discrete probability mass function can be built up.

A slightly alternative slant was taken in chapter 3 where it was shown that if a layer has no synchronicity that it can be represented by a continuous random variable, which has a set probability density function. This probability density function can be found by simple convolution of three other probability density functions, two of which, the functions for input and output time, always have the same function. It was also shown in this chapter how, given a certain amount of synchronicity, that the density function reduced to an equivalent of the results found for the synchronous system.

In chapter 4 the problem of sampling interval variation was looked at, and it was shown that this variation, like the delay itself can also be modelled. However, because this variation is equivalent to a feedback delay from sensor to controller, it was ignored, for the controller implementation.

The results from the modelling were tested extensively by simulation in chapter 5, and it was shown that the simulations give very similar results to the modelling.

Finally, in chapter 6, it was shown that for a synchronous system, and using time stamps it is possible to find an exact value for the time delay. Using this, in addition to the previous modelling, and using the control laws formulated in Nilsson (1998), a gain scheduled controller was designed and simulated. This simulation was compared with results given by an LQG controller based on no time delay, and an LQG controller based upon an estimate of the average time delay. Here it was shown that the gain scheduled controller displayed acceptable results.

Appendix A

Mathematical results

A.1 Sliding intervals.

The mathematics of sliding intervals seems trivial, but because of its heavy use in this thesis, some simple results are worth mentioning.

Consider two intervals denoted λ_0 and λ_1. They have length h_0 and h_1 respectively.

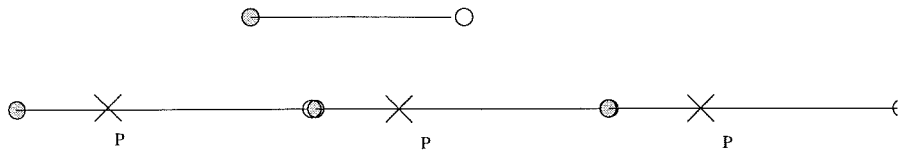

Figure A.1: Two sliding intervals.

λ_1 is a finite, constantly repeating interval. λ_0 is a finite interval which is moving constantly with respect to λ_1. There is an arbitrary point p on λ_1. It is then trivial to see that given a random time slice, the basic probability that λ_0 lies over a stretch of λ_1 that contains p is given by:

$$\mathrm{P} = \frac{\lambda_0}{\lambda_1} \tag{A.1}$$

A.2 Definition of Dirac δ function

If $x \in \mathbb{R}$, the symbol δ_x represents a notional function with the properties

(a) $\delta_x(y) = 0 \quad$ if $\quad y \neq x$

(b) $\displaystyle\int_{-\infty}^{\infty} g(y)\delta_x(y)\mathrm{d}y = g(x) \quad$ for all integrable $g : \mathbb{R} \to \mathbb{R}$

A.3 Continuous random variable results

A random variable X is *continuous* if its distribution function $F(x) = P(X \leq x)$ can be written as

$$F(x) = \int_{-\infty}^{x} f(u)du$$

for some integrable $f\ \mathbb{R} \rightarrow [0, \infty)$.

Definition 6 *f is called the* (probability) density function *of x*

Definition 7 *Two random variables X and Y are considered to be* independent *if $\{X \leq x\}$ and $\{Y \leq y\}$ are independent events for all $x, y \in \mathbb{R}$.*

To define the density function of a sum of random variables we use the convolution of the variables.

Theorem 1 *if X and Y are random variables, which are independent, and $Z = X + Y$ then the density function f_{X+Y} is given by*

$$f_{X+Y}(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z-x)dx = \int_{-\infty}^{\infty} f_X(z-y) f_Y(z)dy \qquad \text{(A.2)}$$

f_{X+Y} is known as the convolution of f_X and f_Y and is represented by

$$f_{X+Y} = f_X * f_Y(z)$$

These are standard results from probability theory([4])

A.4 Rounding down a random variable

Theorem 2 *Consider random variables X and K where*

$$K = \lfloor X \rfloor_u$$

where $\lfloor ... \rfloor_u$ means that X is rounded down to the nearest whole integral multiple of $u \in \mathbb{R}$. Then

$$f_K(k) = \sum_{i=-\infty}^{\infty} \delta_{iu} \int_{iu}^{(i+1)u} f_X(y)dy$$

proof

$$\begin{aligned} P(nu^+ \leq k \leq nu^-) &= P(nh \leq x < (n+1)h) \\ \therefore \quad \int_{nu_-}^{nu^+} f_K(y)dy &= \int_{nu}^{(n+1)u} f_X(y)dy \end{aligned}$$

now by definition of Dirac delta function

$$f_K(nu) = \delta_{nu} \int_{nu}^{(n+1)u} f_X(y)dy \qquad for \qquad nu^- \leq k \leq nu^+$$

thus for all k

$$f_K(nu) = \sum_{i=-\infty}^{\infty} \delta_{nu} \int_{iu}^{(i+1)u} f_X(y)dy$$

A.5 Rounding up a random variable

Theorem 3 *Consider random variables X and K where*

$$K = \lceil X \rceil_u$$

where $\lceil ... \rceil_u$ *means that X is rounded up to the nearest whole integral multiple of* $u \in \mathbb{R}$. *Then*

$$f_K(k) = \sum_{i=-\infty}^{\infty} \delta_{iu} \int_{(i-1)u}^{iu} f_X(y)dy \tag{A.3}$$

proof *follows previous proof*

Bibliography

[1] Nicholas Andreff. Robustness to jitter in real time systems. Master's thesis, Lund University, 1994.

[2] Karl Johan Åström and Björn Wittenmark. *Computer Controlled Systems.* Prentice Hall, 1997.

[3] Norman L. Biggs. *Discrete Mathematics.* Oxford University Press, 1989.

[4] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes.* Oxford University Press, 1992.

[5] Johan Nilsson. *Real-Time Control Systems with Delays.* PhD thesis, Lund Institute of Technology, 1998.