ON THE USE OF A DIGITAL COMMUNICATION CHANNEL FOR FEEDBACK IN A POSITION CONTROL SYSTEM

A Thesis
Presented to
The Graduate Faculty of The University of Akron

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

Kripesh Bhattarai
December, 2012
ABSTRACT

This thesis presents a study to understand how the features of the communication protocol used to transmit feedback information on a discrete-time position control system feedback channel affect the overall quality of control. The features of several industrial standards for the transmission of feedback position information by absolute position encoders in a position control system have been identified. The communication channel used to feed back digital position information and the identified parameters are modeled and simulated to determine how the communications affects the performance of the control system, especially while communicating via long cables in the presence of noise. So, a framework has been developed in order to simulate combined operation of the communication channel and the control system. In this work, a series of simulations are performed for an example scenario of a blade pitch position control in a wind turbine. The quality of control is studied for this application when various error checking strategies are used, in systems with different cable lengths and different levels of noise. The results show that for systems communicating via longer cables, error checking is essential to achieve a high quality of control. For control systems that require fast sampling, a trade-off must be made between the bit rate, which determines the sample rate, and the degree of
error checking. Recommendations for choosing which features to use are made based on the expected level of noise and cable length.
ACKNOWLEDGEMENTS

I am grateful to my advisors Dr. Joan E. Carletta and Dr. Kye-Shin Lee for their support and guidance throughout the work for this thesis. I thank Dr. Robert J. Veillette for his help. Special thanks to Dr. Alex De Abreu Garcia and the whole team of the Department of Electrical and Computer Engineering. I would like to thank all of my friends here at The University of Akron, especially Joseph P. Davis, Krishnaram Budhathoki, Pongpachara Limpisathian, Purushottam Parajuli, Sarita Bhandari, Sudip Adhikari and Vinod Upadhyay. I would also like to thank my family members without whom none of this would have been possible.
3.1 Application scenario: blade pitch control for a wind turbine

3.2 Continuous-time motor model and PID controller
   3.2.1 Continuous-time model of the DC Motor
   3.2.2 Continuous-time model of the PID controller

3.3 Discrete-time model with ideal feedback channel
   3.3.1 Discrete-time model of the DC motor
   3.3.2 Discrete-time model of the PID controller

3.4 Discrete-time model with realistic communications
   3.4.1 Quantizer
   3.4.2 Feedback channel
   3.4.3 Noise modeling
   3.4.4 Error checking code generator and checker
   3.4.5 Packet pass/fail decision
   3.4.6 Sampling rate selection for the system

3.5 Summary

IV. SIMULATION RESULTS FOR AN EXAMPLE SYSTEM

4.1 Selection of simulation parameters for the control system
   4.1.1 Parameters for continuous-time DC motor model
   4.1.2 Parameters for continuous-time PID controller
   4.1.3 Parameters for discrete-time DC motor model
   4.1.4 Parameters for discrete-time PID controller
4.2 Selection of simulation parameters for the communication channel

4.2.1 Parameters for quantizer

4.2.2 Parameters for feedback channel

4.2.3 Parameters for noise modeling

4.2.4 Parameters for error checking strategy

4.2.5 Sampling rates selection for different simulation cases

4.3 Example time domain simulations

4.4 Full simulation study

V. CONCLUSIONS

5.1 Summary

5.2 General recommendations

5.3 Future work

BIBLIOGRAPHY
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Phase comparison of reference signal with sine signal and cosine signal.</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>Parameters for the DC motor model.</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Parameters for the PID controller.</td>
<td>48</td>
</tr>
<tr>
<td>4.3</td>
<td>Values of maximum bit rate and total attenuation based on cable length.</td>
<td>59</td>
</tr>
<tr>
<td>4.4</td>
<td>Total number of bits needed to communicate one measured position for each of the four simulated error checking strategies.</td>
<td>62</td>
</tr>
<tr>
<td>4.5</td>
<td>Sampling rate selection and gain for the discrete-time PID controller for various cases in the simulation study.</td>
<td>64</td>
</tr>
<tr>
<td>4.6</td>
<td>Estimated theoretical values of bit error probabilities for systems with different cable lengths.</td>
<td>75</td>
</tr>
<tr>
<td>4.7</td>
<td>Estimated theoretical values of percentage of corrupted position data packets for the twelve systems simulated, for noise power spectral densities of 1 Watt/Hz, 2 Watts/Hz, 3 Watts/Hz and 4 Watts/Hz.</td>
<td>78</td>
</tr>
<tr>
<td>4.8</td>
<td>Estimated theoretical values of percentage of false positives for systems using CRC and retransmission for error checking.</td>
<td>79</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Closed-loop control system</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>Arrangement of holes on the circular disk for an incremental encoder</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Pulse stream produced by an incremental encoder</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Resolver reference signal and output signals for a motor shaft performing one complete rotation at a constant speed of 1 revolution per second</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Arrangement of holes on the circular disk for an absolute position encoder</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Typical maximum bit rate and attenuation curves for a twisted pair cable</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Various locations for control on a wind power generation plant</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>High level block diagram for the simulation model</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>DC motor electric model</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Block diagram of continuous-time feedback control system for position control of blade pitch, showing details of the DC motor model</td>
<td>26</td>
</tr>
<tr>
<td>3.5</td>
<td>Block diagram of continuous-time feedback control system for position control of blade pitch, showing details of the PID controller</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>Block diagram of discrete-time feedback control system for position control of blade pitch</td>
<td>31</td>
</tr>
<tr>
<td>3.7</td>
<td>Overview of the simulation setup</td>
<td>32</td>
</tr>
</tbody>
</table>
3.8 Data reconstruction by threshold detection method .......................... 40
4.1 Unit step response and pole-zero plot of the continuous-time DC motor model ......................................................... 47
4.2 Root locus plot of the continuous-time DC motor position control with PID controller .................................................... 49
4.3 Pole-zero plot of the closed-loop continuous-time DC motor position control system with PID controller with a gain of 0.255 .......... 50
4.4 Unit step response of the continuous-time closed-loop DC motor position control system using a PID controller with a gain of 0.255 ... 52
4.5 Unit step response of the discrete-time open-loop DC motor model from voltage input to speed output with a sample time of 0.4 milliseconds 53
4.6 Pole-zero plot of the discrete-time DC motor model with a sample time of 0.4 milliseconds ....................................................... 54
4.7 Root locus plot of the discrete-time DC motor position control system with PID controller with a sample time of 0.4 milliseconds ...... 55
4.8 Pole-zero plot of the closed-loop discrete-time DC motor position control system with PID controller with a gain of 0.242 and a sample time of 0.4 milliseconds ....................................................... 56
4.9 Unit step response of the discrete-time closed-loop DC motor position control system using a PID controller with a gain of 0.242 and a sample time of 0.4 milliseconds ....................................................... 57
4.10 Comparison of theoretical results with simulated results for noise modeling 61
4.11 Commanded and actual position of a system with no error checking and a cable length of 100 meters, for a noise power spectral density of 1 Watt/Hz ................................................................. 67
4.12 Commanded and actual position of a system with no error checking and a cable length of 100 meters, for a noise power spectral density of 1.2 Watts/Hz ................................................................. 67
4.13 Commanded and actual position of a system with CRC and a cable length of 100 meters, for a noise power spectral density of 1.2 Watts/Hz 68
4.14 Root mean square error for systems with cable length of 100 meters . . 73
4.15 Root mean square error for systems with cable length of 150 meters . . 73
4.16 Root mean square error for systems with cable length of 200 meters
CHAPTER I
INTRODUCTION

1.1 Motivation

An automatic closed-loop control system is comprised of a plant, a controller and feedback, as shown in Figure 1.1. A plant is a device to be controlled by the controller. A reference command signal is fed to the controller. Feedback from the output of the plant is required by the controller, so that the controller also has the measure of output of the plant. The controller compares the output of the plant with the commanded reference and commands the plant with the correction necessary to make the output of the plant and the reference command signal the same [1].

![Closed-loop control system](image)

Figure 1.1: Closed-loop control system

The feedback mechanism requires a sensing element, also called a sensor or a transducer, that produces some signal corresponding to the control parameter under consideration. In a digital control system, this signal is generally discrete in nature or
discretized with the help of an analog-to-digital converter. In a digital control system where speed control or position control of a rotating shaft of a motor is needed, several approaches are used to sense its speed or position.

An industrial position encoder is used to measure and transmit information about the present position of a rotating shaft. This position information is used as feedback information for a control system, to decide the next action based upon the measured position values. The quality of control in a system using a position encoder is affected by the accuracy of the position information and how often new position values can be sensed and fed back.

Traditionally, incremental encoders are used for the purpose of detecting speed of the rotating shaft, from which position is derived indirectly. Another device used for sensing position is the resolver, which produces analog outputs from which position can be derived directly using mathematical manipulation. An absolute position encoder is another type of device that is able to sense the position directly but without the need for mathematical manipulation. An absolute position encoder sends the absolute present position of the shaft periodically in the form of binary digits using serial communication.

Absolute position encoders are available using a variety of industrial standards for transmission of position information. Each of these individual standards defines a position data packet, which consists of position information along with various other parameters; each also defines a bus initiation scheme for starting the communication. All of the standards have different parameters, and each of them
offers different benefits. It requires significant engineering judgement or analysis to choose the standard that best meets the requirements of a particular application.

1.2 Goal and contribution of the thesis

The objective of this thesis is to study the quality of control in a control system that uses an absolute position encoder to measure the present position for feedback. The effects of the digital communication protocol used to communicate absolute position information on the feedback channel are considered. An important parameter for the feedback channel is the cable length. The cable length determines the maximum bit rate for the digital communications as well as the attenuation of the signal along the cable. The communication may take place in noisy environments, so the noise power spectral density of the received signal is also another parameter that should be considered. Noise on the received signal may cause the position information to be interpreted incorrectly, since some of the bits in the position data packet may be in error. Error checking may be used to detect some of these errors, and so strategies for error checking should also be considered. This thesis will help the reader to understand how a user with a particular application can determine what kind of digital protocol is required to achieve a required quality of control.

There exist many studies on position feedback control systems and communication systems separately; however, little work has studied the combination of the two, to understand how the limitations of the communication channel used to feedback digital position information affects the performance of a control system. This
thesis serves the purpose of analyzing the quality of control in a position feedback control system that uses a digital communication channel for feedback.

A series of simulations are performed to assess the quality of control given the number of bits used to represent position, the length of the cable, the noise power spectral density of the received signal and the type of error detection method used. A number of different industrial protocols have been developed for absolute position encoders, some available on commercial encoders and others proposed. No one protocol dominates the market. The protocols vary widely in features; for example, some include mechanisms for detecting communication errors; however, it is difficult to know which combinations of features are advantageous when communicating position over long cables in a noisy environment. This thesis provides insight into the value of various protocol features for such applications, to better recommend what sort of protocol to choose when cable length and noise conditions are known.

1.3 Organization of the thesis

This thesis is organized in the following way. Chapter II presents the theoretical background on position feedback and control. This chapter also describes the features of existing standards and protocols for the transmission of absolute position information. Chapter III describes the setup used for the simulations. Chapter IV presents the simulation results and discusses their significance. Chapter V draws conclusions, makes general recommendations, and scope for future work.
CHAPTER II

POSITION FEEDBACK AND CONTROL BACKGROUND

The position of a rotating shaft may be used as a feedback to a position control of a rotating motor. In this chapter, the types of devices used for speed or position detection of a rotating shaft are discussed. They include incremental encoders, resolvers, and absolute position encoders. The construction of each of these devices is described. Then, the advantages of using a digital communication protocol for transmission of position information are presented. Next, the features and attributes of digital protocols for transmission of position information are also discussed. Finally, the effects of cable length on a digital communication channel are discussed.

2.1 Devices for speed or position detection

Various devices are used for position detection in a control system that uses the position of a rotating shaft for feedback. Techniques for sensing the position information using these devices are discussed. The advantages and disadvantages of using a particular device for position feedback are also discussed. The devices used for sensing speed or position include incremental encoders, resolvers, and absolute position encoders.
2.1.1 Incremental encoder

A classical method of sensing speed of a moving shaft is implemented by constructing an assembly that consists of a light generator and a detector placed on opposite sides of a disk mounted on the motor shaft [2]. The disk contains several holes, so the light from the light generator does not reach the detector at all times; rather, light is detected as a continuous waveform of digital pulses as the shaft rotates. Therefore, a revolution of the motor shaft corresponds to \( N \) such pulses, where \( N \) is the number of holes in the disk.

The rotational speed of the motor shaft is determined by observing the pulses. The direction of rotation cannot be determined from the pulses. If the direction of rotation is to be known, another assembly is constructed in which a second pulse stream is created out of phase with the first pulse stream. These two pulses are known as quadrature pulses. Then, the direction of rotation can be determined by looking at the phase relation of the two pulse streams [3].

In a conventional incremental encoder, a third pulse is created to represent a home location of the motor shaft; this pulse appears once every revolution and is called the zero pulse. The home location is used to find the starting position of the shaft; this, along with the speed information, can be used to find the present position of the shaft. Thus, an incremental encoder has three outputs: quadrature pulses \( A \) and \( B \) that are out of phase with each other, and a zero pulse \( Z \) that appears once every revolution. The disk for an incremental encoder is constructed by making holes such that these three outputs are created simultaneously. Figure 2.1 shows the
arrangement of holes on the circular disk, where the holes are represented by the white segments. Each circular section corresponds to an output of the incremental encoder, and each section is equipped with its own light generator and detector on either side of the corresponding section of the disk. In the figure, a simplified disk with only eight sectors is shown in which the motion is only sensed after the disk makes one-eighth of a turn. In a practical encoder, the number of sectors in the disk is much larger, so that the motion is sensed with much higher precision.

Figure 2.2(a) shows the $A$, $B$ and $Z$ pulses produced by an incremental encoder for clockwise rotation of the motor shaft, whereas Figure 2.2(b) shows the $A$, $B$ and $Z$ pulses for counterclockwise rotation of the motor shaft. Speed can be determined from the pulses produced by an incremental encoder by dividing the
number of pulses per second by the number of pulses per revolution. These $A$, $B$ and $Z$ pulses may be communicated using differential signaling for better noise immunity.

The disadvantage of the incremental encoders is that determining the position of the motor shaft requires knowing the history since the home position (indicated by the $Z$ pulse) was last seen. The control system cannot recover normal operation after a power outage until it encounters the home location again. Thus, an incremental encoder senses the position only indirectly [4]; the history of the pulses is needed to infer position, rather than just the instantaneous values of the pulses.

2.1.2 Resolver

A resolver is an electromechanical device that allows direct sensing of both the speed and position of a rotating shaft. A resolver consists of a transformer with a rotating primary winding, which rotates with the rotation of the shaft, and two stationary secondary windings. A high frequency sinusoid reference signal is applied to the primary winding. Due to the rotation of the primary winding, the energy induced in the secondary windings varies sinusoidally. The two secondary windings are placed at orientations that are 90 degrees apart from each other so that the energies induced in the windings will be 90 degrees out of phase with each other. Hence, a resolver generates two output signals in the secondary windings: a sine signal $S$ and a cosine signal $C$, which are modulated by the reference signal. So, the envelopes of the output signals $S$ and $C$ gives the sine and cosine values, respectively, of the position angle of the shaft.
Figure 2.2: Pulse stream produced by an incremental encoder

(a) for clockwise rotation of the motor shaft

(b) for counterclockwise rotation of the motor shaft
Figure 2.3 illustrates the relationships among the signals used and produced by a resolver, when the motor shaft is rotating at a constant speed of 1 revolution per second, through one complete 360 degree rotation. In the figure, the reference signal’s frequency is 40 Hz, which is much lower than the typical frequency used for an industrial resolver. The selection of a lower frequency is only for illustration purposes, so that both the reference signal and the full cycle of sine and cosine envelope signal can be clearly seen.

The resolver reference signal is a function of time $t$ and is defined by

$$r(t) = A_{\text{ref}} \sin(2\pi f_{\text{ref}} t)$$

(2.1)
where $A_{\text{ref}}$ is the amplitude of the reference signal and $f_{\text{ref}}$ is the frequency of the reference signal. The resolver output signals $S$ and $C$ are then

$$S(\theta, t) = r(t) \sin(\theta)$$  \hspace{1cm} (2.2)

$$C(\theta, t) = r(t) \cos(\theta)$$ \hspace{1cm} (2.3)

where the shaft angle is $\theta$.

The first step in finding the position of the motor shaft is to demodulate the output signals with the reference signal $r(t)$ to yield the amplitudes of the sine of the shaft angle $|\sin(\theta)|$ and the cosine of the shaft angle $|\cos(\theta)|$, and to calculate

$$\theta_e = \tan^{-1}\left(\frac{|\sin(\theta)|}{|\cos(\theta)|}\right).$$ \hspace{1cm} (2.4)

Interpreting the output of a resolver to determine the position requires some circuitry to demodulate the sine and the cosine signals from the reference signal. The resolver demodulator usually consists of an envelope detector circuit [5]. This is known as arc-tan interpolation on the envelope of the resolver signals. To resolve the angle over the full 360 degrees of rotation, the comparison of the phase of the sine and cosine waves with the reference is needed [6]. The angle of the motor shaft $\theta$ is reconstructed from angle $\theta_e$ by knowing the quadrant in which the angle lies. This is done by comparing the sine and the cosine signals with the reference signal as shown in Table 2.1.

Figure 2.3 shows that the resolver can provide both speed sensing and direct position sensing. Speed information is determined by the frequency of the envelope of sine and cosine waves. Position information is also provided directly by the amplitude
Table 2.1: Phase comparison of reference signal with sine signal and cosine signal

<table>
<thead>
<tr>
<th>Phase of reference signal versus</th>
<th>Sign of $\theta_e$, $\theta$</th>
<th>Quadrant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase of sine signal</td>
<td>Phase of cosine signal</td>
<td>$1^{st}$</td>
</tr>
<tr>
<td>In phase</td>
<td>In phase</td>
<td>$2^{nd}$</td>
</tr>
<tr>
<td>In phase</td>
<td>Out of phase</td>
<td>$3^{rd}$</td>
</tr>
<tr>
<td>Out of phase</td>
<td>Out of phase</td>
<td>$4^{th}$</td>
</tr>
</tbody>
</table>

of the instantaneous waveforms, with some mathematical manipulation required to convert the analog output signals into position values. This gives the resolver an advantage over the incremental encoder, for which the history of the pulse streams is needed to determine position.

2.1.3 Absolute position encoder

The basic construction of an absolute position encoder differs from an incremental encoder by the arrangement of holes on the disk. The disks are fabricated such that the position can be found directly with the help of instantaneous signal values in the detectors without mathematical manipulation. The disk is divided into $n_{ST}$ segments and $2^{n_{ST}}$ sectors. Each sector in the disk represents a single position on the disk. Each segment is monitored with its own light transmitter-detector pair. As the disk rotates, the values seen at the $n_{ST}$ detectors together form a binary code that indicate the position of the shaft; the resolution of the sensed position is $360/2^n_{ST}$.
degrees. Figure 2.4 shows the location of holes for an absolute position encoder with $n_{ST} = 3$. In this figure, the bits $(b_2, b_1, b_0)$ represent the instantaneous position with resolution of $360/8 = 45$ degrees. Hence, the position information in an absolute position encoder is digital by nature of construction of the disk.

This technique for absolute position detection is able to resolve the position within a single turn of the disk. A count of these single turns is also made, by a microprocessor within the encoder. The position information, which consists of a single-turn position and a multi-turn count, may then be transmitted digitally using serial communication.

The direct transmission of absolute position information will help in instantaneous recovery from momentary loss of power. This fast recovery is not possible in
incremental encoder and resolvers [4]. The resolver also transmits position information directly using instantaneous signals but there is some amount of mathematical manipulation to determine the position, which will make the recovery slower.

2.2 Digital communication protocols for position feedback

Advancement in digital communications technology has made high-speed wireline serial data transfer possible. Digital transmission offers numerous advantages over analog transmission; these include reliable communication by addition of parity checks and other error coding, ease in multiplexing for communication with multiple devices, and less sensitivity towards electromagnetic and other environmental noise. This is achieved at the cost of increased bandwidth requirement, increased complexity of the system, and the need for precision timing across the system [7]. Digital communication technology enabled the development of the absolute position encoder, which sends absolute position information as binary digits using serial communication.

There are various digital communication protocols that are used for transmitting the position information serially in an absolute position encoder. All protocols transmit the position information, but the protocols differ according to the features they provide. The features include transmitting additional bits for error checking, transmitting additional bits for diagnostic messages, providing support for point-to-point and multi-point networks, and using bus initiation strategies for starting the communication. These features are described next.
Apart from single-turn position and multi-turn count, additional information may be used for error checking. Error checking helps in increasing the transmission reliability of the channel. This error checking mechanism is advantageous since digital communication may take place in noisy environments. Usually, error checking is done by introducing redundant bits in each data packet communicated. Some of the protocols use a cyclic redundancy check (CRC) for this purpose. The CRC bits are calculated using the bits of the position information. The transmitted data packet then includes both the position information and the appended CRC bits. On the receiver side of the communication channel, a CRC algorithm uses the position information bits and the CRC bits to check whether the CRC bits are still a correct match for the bits of position information. A number of CRC schemes are possible, with different numbers of CRC bits used. The CRC method for error checking will be able to detect whether one or more bits in the data packet have been corrupted in the course of transmission, provided that the number of bits affected is less than the number of bits used for the CRC. An alternative method of error checking is retransmission. This is a simple method in which each position data packet is transmitted twice. The receiver checks whether the two consecutive received packets are identical. Whether a CRC or retransmission is used, the usual strategy is to discard packets that fail the error checking test, using the most recent position that passed the error checking test in place of the erroneous position.

In a typical application, the motor drive is typically the device that initiates communication, also known as the master device. The absolute position encoder is the
device that responds to the master device, also known as the slave device. Additional information such as alarm, warning and diagnostic messages may be produced by the absolute position encoder to indicate the malfunction of various submodules within the encoder. Depending on the protocol, these messages may be transmitted upon the request of the master, or asynchronously when various conditions occur to trigger the slave to send an update. Protocols may include defined commands so that the associated electronics can determine the nature of the message being transmitted.

All together, a typical position data packet contains the single-turn position, the multi-turn count, and optional bits such as error checking codes, alarms, warnings, and diagnostic messages.

Some protocols allow for only point-to-point communication, whereas others may allow multi-point communication. In a point-to-point network, a single slave device communicates with a single master. The master requests data by sending out a clock signal and the slave responds by transmitting the position data packet, one bit per clock cycle. In a setup for devices supporting multi-point communication, there is usually a single master that communicates with a number of slaves. The multiple slaves are typically connected with the single master in a chain-like arrangement.

In some protocols, the number of bits in a position data packet is fixed in the slave device. The master knows the number of bits before the communication starts. In this case, no bus initiation schemes are needed. In other protocols, the number of bits in the position data packet is not fixed. In this case, bus initiation is
required. According to the bus initiation strategy, the master will configure itself to communicate with the slave.

Popular industrial protocols for absolute position encoders include: Synchronous Serial Interface (SSI) [8], Bidirectional Synchronous Serial Interface (BiSS) [9], Encoder Data (EnDat) [10], and High Performance Interface (Hiperface) [11]. SSI and EnDat support only point-to-point networks, whereas BiSS and Hiperface support communication of a master with multiple slaves. Bus initiation of one form or another is required for all protocols except SSI. Some sort of alarm messages, diagnostic messages, and warning messages are available on all protocols except SSI. EnDat also provides retransmission for error checking. The BiSS and Hiperface protocols do not allow for retransmission. For error checking purposes, SSI relies solely on retransmission. CRC is possible in BiSS, EnDat and Hiperface. EnDat and Hiperface also have an additional analog channel, from which position can be derived. Since EnDat and Hiperface provide two independent sources of position detection, the additional source can help in error checking without retransmission.

2.3 Effects of cable length on digital transmission of position information

Absolute position encoders generally use the RS-485 standard for serial transmission of digital data. The RS-485 cables generally use 24-AWG twisted pair cables for differential transmission of the data packet. Different impairments such as the interference from one bit to the next (inter-symbol interference), propagation delay, and timing synchronization error (jitter) are increased when the bit rate and cable
length is increased. The data sheets of cables adhering to the RS-485 standard give experimentally derived relationships between the maximum bit rate and the cable length; a typical relationship is plotted in Figure 2.5(a) [12, 13].

The attenuation of the signal at the receiver end of the cable is a function of the signal frequency and is given in terms of decibels per unit length. Attenuation increases as the cable length increases. Figure 2.5(b) shows the attenuation per unit length of cable versus frequency in a typical twisted pair cable meant for communication using the RS-485 standard.

2.4 Summary

In this chapter, the different techniques for position detection are presented. The advantages of using a digital protocol while communicating absolute position information are explained, and features of digital protocols are described. The industrial protocols that exist for absolute position encoders vary in features, and a summary of main features are discussed. The simulations done in this thesis do not use a particular protocol, but instead consider how the various protocol features influence the quality of control when the absolute position encoder is used in a feedback position control system.
Figure 2.5: Typical maximum bit rate and attenuation curves for a twisted pair cable.
CHAPTER III

SIMULATION SETUP

A series of simulations are performed to study the quality of control in a digital control system that uses an absolute position encoder for position feedback. This chapter describes the simulation setup and the parameters used in the simulation. First, the application scenario for the simulation is discussed. Next, the continuous-time models for the DC motor and the PID controller are given. Then, methods used to discretize the models are described. Finally, a model is set up for the communications channel, via which the feedback information is sent. A model for the noise in communication channel is included.

3.1 Application scenario: blade pitch control for a wind turbine

As an example scenario, a wind power generation system is used. A typical wind turbine requires three independent feedback control systems. These three feedback controls are depicted in Figure 3.1. The first is generator control, which is the speed control of the rotary motion of the wind turbine. The second is yaw control, which is to turn the turbine so that the turbine faces the direction of wind. The third is pitch control, which is to turn the individual blades of the turbine. In pitch control, the individual blades should be turned so that the attack angle of the wind
hitting the blade is optimized. Blade pitch control is important to ensure that the turbine is not subjected to loading forces larger than it can safely handle, and that the power generated by the turbine does not exceed the maximum ratings for which the electronics were designed [14]. The pitch control requires position control of the turbine blade, so that the blade can be turned to a certain angle. The simulation considers this position control required for the blade pitch control for a wind turbine.

The motor drive is located at the top of the wind turbine tower. The motor drive consists of the motor and the power electronics circuitry that control the motor. The absolute position encoder is mounted to the motor. The controller is located at the bottom of the wind turbine tower. The high level block diagram of the closed-loop control system is shown in Figure 3.2. Two types of wires run between the top of the turbine tower and the bottom of the turbine tower: the communication
Figure 3.2: High level block diagram for the simulation model

channel, which carries the feedback information from the encoder to the controller, and the control channel, which carries pulse width modulated gating signals from the controller to the motor. The signal in the control channel suffers only negligibly in the presence of external noise, since the pulse width modulated signals are not particularly prone to noise. In the communication channel, the communication takes place by transmitting bits serially. Some of these bits may be misinterpreted by the receiver due to external noise. Thus, it is important for the simulation setup to model the effects of noise on the digital communication channel.

3.2 Continuous-time motor model and PID controller

The full simulation model is built up, starting with a simple continuous-time model of a feedback control system that includes only a DC motor model to represent the
motor that moves the blade and a PID controller that controls the blade pitch by using position control. The continuous-time models for the DC motor and PID controller are given next.

3.2.1 Continuous-time model of the DC Motor

A standard circuit model is used for the DC motor. This circuit is shown in Figure 3.3. It is comprised of two sides, the armature side (electrical part) and the load side (mechanical part) [15]. The input to the motor is provided as an armature voltage $V_a$. The armature side also has some resistance $R_a$ and inductance $L_a$ associated with it. The back electromotive force $V_b$ developed on the motor is proportional with the angular velocity $\omega$,

$$V_b = K_b\omega$$  \hfill (3.1)

where $K_b$ is the back emf constant. Applying Kirchhoff’s current law on the armature side yields

$$V_a = iR_a + L_a \frac{di}{dt} + V_b$$  \hfill (3.2)

where $i$ is the current flowing in the armature side, also known as armature current, and $t$ is the time. Combining (3.1) and (3.2) gives

$$V_a = iR_a + L_a \frac{di}{dt} + K_b\omega.$$  \hfill (3.3)

At the load side, the torque depends upon the armature current,

$$T = K_t i$$  \hfill (3.4)

where $K_t$ is the torque constant.
The torque on the load side is represented as the rotor’s moment of inertia $J$ multiplied by the angular acceleration and the product of motor’s mechanical system damping $B$ with the angular velocity, that is,

\[ T = J \frac{d^2\theta_o}{dt^2} + B \frac{d\theta_o}{dt}. \]

(3.5)

Combining (3.4) and (3.5) gives

\[ J \frac{d^2\theta_o}{dt^2} + B \frac{d\theta_o}{dt} = K_i \omega \]

(3.6)

where $\theta_o$ is the angular position, $\frac{d\theta_o}{dt}$ is the angular velocity ($\omega$) and $\frac{d^2\theta_o}{dt^2}$ is the angular acceleration.

Taking the Laplace transforms of (3.3) and (3.6) yields

\[ V_a(s) = R_a I(s) + L_a s I(s) + K_i \omega(s) \]

(3.7)
and

\[ Js^2 \theta_o(s) + B s \theta_o(s) = K_t I(s). \]  

(3.8)

By solving (3.7) and (3.8) for the angular velocity of the DC motor in terms of the armature voltage, the transfer function is obtained. That is,

\[ \frac{\omega(s)}{V_a(s)} = \frac{K_t}{(J s + B)(L_a s + R_a) + K_t K_b}. \]  

(3.9)

The transfer function can be further integrated to obtain angular position \( \theta_o \) as a function of armature voltage:

\[ G(s) = \frac{\theta_o(s)}{V_a(s)} = \frac{K_t}{s[(J s + B)(L_a s + R_a) + K_t K_b]} \]  

(3.10)

A block diagram for the continuous-time DC motor position control system is shown in Figure 3.4, where the DC motor model is shown in detail. The voltage applied to the armature side \( V_a \) is controlled by setting the duty cycle of the pulse width modulated control signal. The voltage \( V_a \) must stay within \( \pm V \), where \( V \) is the rated voltage of the DC motor under consideration. This limit is applied by clipping the output of controller when it is out of range, a process referred to as actuator saturation. This clipping of voltage provides a method for preventing the motor from exceeding its maximum rated speed. Actuator saturation is shown as a block inside the DC motor model illustrated in Figure 3.4.

The step response of the closed-loop system of Figure 3.4 gives information on how quickly and how well the motor responds to a step change in the commanded input. Important parameters of the step response include the rise time, the percentage
overshoot, and the settling time. The rise time is the time the output takes to go from 10% to 90% of the final value. The percentage overshoot is the percentage by which the peak of the step response exceeds the steady-state value. The settling time is the time to reach and stay within 5% of the steady-state value [16]. These measures will be used to evaluate the unit step response of a system.

3.2.2 Continuous-time model of the PID controller

For the blade pitch control example developed in this chapter, a proportional-integral-derivative (PID) controller is used. The PID control operates on the error signal $e(t)$, which is the difference between the commanded angular position of the motor shaft that determines the blade pitch $\theta_i(t)$ and actual motor shaft position $\theta_o(t)$, both measured in degrees. The PID controller has three different constant parameters,
$K_p$, $K_i$ and $K_d$, which are the proportional constant, the integral constant and the derivative constant, respectively. The proportional term is dependent on present error, the integral term on the overall sum of past errors, and the derivative term on a prediction of future errors [16]. The general equation for the PID control is given by

$$f(t) = K \left[ K_p e(t) + K_i \int_0^t e(\tau) d(\tau) + K_d \frac{d}{dt} e(t) \right],$$  

(3.11)

where $f(t)$ is the output of the PID controller and $K$ is a gain that scales all three PID constants. Taking the Laplace transform of both sides yields

$$F(s) = K \left[ K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s) \right].$$  

(3.12)

Rearranging (3.12), the transfer function for the PID controller is

$$C(s) = \frac{F(s)}{E(s)} = K \left[ K_p + \frac{K_i}{s} + K_d s \right].$$  

(3.13)

A block diagram of the closed-loop control system that highlights the PID controller in a continuous-time DC motor position control system is shown in Figure 3.5.

### 3.3 Discrete-time model with ideal feedback channel

The next step in developing the simulation models is to replace each continuous-time block with its discrete-time equivalent. The controller will be implemented as a discrete-time system using digital hardware; it is convenient for the analysis to have a discrete-time version of the DC motor model, so that z-domain transfer functions can be used to describe the entire system. For now, in the interest of simplicity, an
ideal feedback channel is assumed; in other words, it is assumed that the controller knows the exact position of the motor at all times. Later, more realistic models for the communication channel that transmits feedback position data packets will be included.

3.3.1 Discrete-time model of the DC motor

The zero order hold (ZOH) method is used to convert the continuous-time DC motor model to a discrete-time DC motor model [17]. The ZOH is the preferred method for discretizing plant models, as it ensures that the discrete-time model produces an output that exactly matches the original continuous-time model at the sampling instants. The discrete-time model is given as

\[
G(z) = \frac{z - 1}{z} \mathcal{Z}\left[ L^{-1}\left(\frac{G(s)}{s}\right) \right]
\]

(3.14)
where $\mathcal{Z}$ is the Z-transform and $\mathcal{L}^{-1}$ is the inverse Laplace transform.

### 3.3.2 Discrete-time model of the PID controller

The PID controller, which was developed as a continuous-time controller, must also be discretized, so that it can be implemented on a digital computer. The discretization is performed by using a rectangular integrator in the PID transfer function [18], which is a usual method for commercial controllers. The discrete-time implementation of the PID controller is

$$C(z) = K \left[ K_p + K_i T_s \frac{z}{z-1} + K_d \frac{z-1}{T_s z} \right],$$

(3.15)

where $T_s$ is the sampling rate of the system.

The transfer function of the PID controller may also be considered in terms of contribution of three separate parts: the proportional part, the differentiation part, and the integral part. The transfer function can be represented as

$$C(z) = \frac{F(z)}{E(z)} = \frac{F_p(z)}{E(z)} + \frac{F_d(z)}{E(z)} + \frac{F_i(z)}{E(z)}.$$  

(3.16)

Comparing (3.15) and (3.16),

$$\frac{F_p(z)}{E(z)} = K [K_p],$$  

(3.17)

$$\frac{F_d(z)}{E(z)} = K \left[ \frac{K_d}{T_s} (1 - z^{-1}) \right] \text{ and}$$  

(3.18)

$$\frac{F_i(z)}{E(z)} = K \left[ K_i T_s \frac{1}{1 - z^{-1}} \right].$$  

(3.19)

In the time domain, (3.17), (3.18) and (3.19) are expressed as

$$f_p(n) = K [K_p e(n)],$$  

(3.20)
\[ f_D(n) = K \left[ \frac{K_d}{T_s} [e(n) - e(n - 1)] \right] \] and
\[ f_{I1}(n) = K [K_i T_s e(n)] + f_{I1}(n - 1). \]

The magnitude of the integral term \( f_{I1}(n) \) may increase with every time step if the error is either positive or negative in sign persistently. This means that the integral term may grow without bound if the control system is not able to control the plant as commanded due to limits such as those imposed by actuator saturation. This phenomenon is known as integral windup. The controller should include a method to prevent integral windup; commonly, the integral term is intentionally saturated to prevent its magnitude from becoming larger than a maximum value \( f_{I,max} \):

\[
 f_I(n) = \begin{cases} 
 +f_{I,max}(n) & \text{if } f_{I1}(n) > +f_{I,max} \\
 -f_{I,max}(n) & \text{if } f_{I1}(n) < -f_{I,max}.
\end{cases}
\]

The overall discrete-time closed-loop model for the DC motor position control system, with the discrete-time PID controller illustrated in detail, is shown in Figure 3.6. The system may behave differently depending upon the sample rate chosen for the discretization. In order to meaningfully compare discrete-time systems with different sample rates, a procedure is developed to choose the PID control parameters so as to ensure similar overall behaviors. The same values for the PID terms \( K_p, K_i \) and \( K_d \) that were used for the continuous-time PID controller are used for the discrete-time PID controller. For a given sample rate, the gain \( K \) is used to adjust the PID parameters while maintaining their corresponding ratios. For a particular sample rate, the \( K \) value is chosen with help of a root locus plot of the closed-loop system,
which shows system pole locations for a range of $K$ values. The value of $K$ is chosen from the root locus plot, so that the step response of the closed-loop system has a particular damping ratio.

3.4 Discrete-time model with realistic communications

When digital information is communicated via a long cable, the quality of the communication is an important consideration, since the individual bits of a digital packet may be corrupted due to noise in the environment. Thus, the simple discrete-time model developed so far is not sufficient; a more realistic model of the communication channel model must be incorporated. Figure 3.7 shows the full simulation setup, in which the communication channel model is included. The description of each block in the simulation setup is explained next.
3.4.1 Quantizer

The absolute position information of a moving shaft sent along the feedback channel consists of two parts: a single-turn position within 360 degrees of rotation and a multi-turn count. Both must be quantized for digital transmission. The single-turn position is quantized using $n_{ST}$ number of bits. As a result, the single-turn position has a resolution of $360/2^{n_{ST}}$ degrees. The multi-turn count holds the count of the number of times the shaft has undergone a complete rotation. The number of bits used to transmit the multi-turn count is $n_{MT}$. The multi-turn count rolls over after it reaches its maximum value of $2^{n_{MT}} - 1$. The quantized bits for both the single-turn position and the multi-turn count are transmitted serially, with the multi-turn count in the higher order bits and the single-turn position in the lower order bits.

Figure 3.7: Overview of the simulation setup
3.4.2 Feedback channel

The RS-485 feedback channel uses a 24-AWG twisted pair cable. The length of the cable determines the maximum bit rate at which data can be transmitted. The length of the feedback channel also determines the total attenuation of the signal along the cable. The analytical relationships for the maximum bit rate and the attenuation with respect to the cable length are established next; these relationships are based on empirical data provided by cable manufacturers.

A typical relationship between the cable length and maximum bit rate for a 24-AWG twisted pair was introduced in Figure 2.5(a). This figure is plotted on a log-log scale. It is observed that the logarithm of maximum bit rate decreases linearly with the logarithm of the cable lengths for cable lengths less than 1200 meters. In order to establish the relationship between maximum bit rate and cable length on the linear part of the plot, a line is fit to two known points read from Figure 2.5(a). The two known points selected are \((r_1, l_1) = (90 \text{ kHz, 1200 meters})\) and \((r_2, l_2) = (7 \text{ MHz, 20 meters})\), where \(r_1\) and \(r_2\) are maximum bit rates for cable lengths of \(l_1\) and \(l_2\), respectively. Because the points lie on a line in a log-log plot, it must be that

\[
\log_{10}(l_1) = m \log_{10}(r_1) + b \tag{3.24}
\]

and

\[
\log_{10}(l_2) = m \log_{10}(r_2) + b \tag{3.25}
\]

where \(m\) is the slope of the linear segment and \(b\) is the intercept on the axis that represents the cable length. Solving for the slope by subtracting (3.25) from (3.24)
yields

\[ m = \frac{\log_{10}(l_1/l_2)}{\log_{10}(r_1/r_2)}. \]  
(3.26)

With the slope known, it is possible to find the maximum data rate \( r \) for cable length \( l \). The pertinent relationship is

\[ \log_{10} \left( \frac{l}{l_2} \right) = m \log_{10} \left( \frac{r}{r_2} \right) = \log_{10} \left( \frac{r}{r_2} \right)^m. \]  
(3.27)

Taking the inverse logarithm on both sides, (3.27) becomes

\[ \frac{l}{l_2} = \left( \frac{r}{r_2} \right)^m. \]  
(3.28)

Solving for \( l \) yields

\[ l = \left( \frac{l_2}{r^{2m}} \right) r^m. \]  
(3.29)

Thus, the cable length can be written as a function of the maximum bit rate as

\[ l = kr^m, \]  
(3.30)

where \( k = \frac{l_2}{r^{2m}} \) is a constant. Rearranging (3.30) yields the equation to calculate the permissible maximum bit rate of the channel for a given cable length; it is

\[ r = \sqrt[2m]{\frac{l}{k}}. \]  
(3.31)

A similar procedure is used to find the relationship of the signal attenuation at the receiver end of a cable and the cable length. As shown in Figure 2.5(b), the logarithm of signal attenuation per unit length of the cable, measured in decibels per meter, grows linearly with the logarithm of operating frequency. In order to establish the relationship between operating frequency and attenuation per unit length on the
linear part of the plot, a line is fit to two known points read from Figure 2.5(b). The two known points selected are \((f_1, a_1) = (2 \text{ MHz}, 0.03 \text{ dB/m})\) and \((f_2, a_2) = (20 \text{ MHz}, 0.09 \text{ dB/m})\), where \(f_1\) and \(f_2\) are operating frequencies that correspond to signal attenuations per unit length of \(a_1\) and \(a_2\), respectively. Because the points lie on a line in a log-log plot, the relationship between the attenuation per unit length of cable \(a\) and operating frequency \(f\) is

\[ a = k_1 f^{m_1}, \quad (3.32) \]

where

\[ m_1 = \frac{\log_{10}(a_1/a_2)}{\log_{10}(f_1/f_2)} \quad (3.33) \]

and

\[ k_1 = \frac{a_2}{f_2^{m_1}}. \quad (3.34) \]

The total attenuation of the signal along the cable is found by multiplying the signal attenuation per unit length in (3.32) by the cable length.

The relationships in (3.31) and (3.32) are used to find the maximum data rate and the signal attenuation per unit length analytically for simulations in which the cable length is varied.

3.4.3 Noise modeling

The RS-485 feedback channel under consideration uses differential signaling. Differential signaling uses two complementary signals that are transmitted on a shielded twisted wire pair [19]. An antipodal transmission scheme is used, in which the binary symbols ‘1’ and ‘0’ are transmitted as the same voltage level but with opposite
polarity. The signaling scheme is:

\[ V_{t1} = \begin{cases} 
+V & \text{if symbol '1' is transmitted} \\
-V & \text{if symbol '0' is transmitted}
\end{cases} \quad (3.35) \]

\[ V_{t2} = \begin{cases} 
-V & \text{if symbol '1' is transmitted} \\
+V & \text{if symbol '0' is transmitted}
\end{cases} \quad (3.36) \]

where \(V_{t1}\) is the (true) signal transmitted on the first wire in the pair, and \(V_{t2}\) is the (complement) signal transmitted on the second wire in the pair. The signals received at the receiver, \(V_{r1}\) and \(V_{r2}\), are:

\[ V_{r1} = \begin{cases} 
+V + n_1(t) & \text{if symbol '1' is transmitted} \\
-V + n_2(t) & \text{if symbol '0' is transmitted}
\end{cases} \quad (3.37) \]

\[ V_{r2} = \begin{cases} 
-V + n_3(t) & \text{if symbol '1' is transmitted} \\
+V + n_4(t) & \text{if symbol '0' is transmitted}
\end{cases} \quad (3.38) \]

where \(n_1(t), n_2(t), n_3(t),\) and \(n_4(t)\) are the additive noise on the channel.

At the receiver end of the cable, the signal is passed through a differential receiver, which reconstructs the signal based on the difference between the voltages on the two wires. Thus, the received difference signal \(V_r\) is

\[ V_r = V_{r1} - V_{r2} = \begin{cases} 
+2V + [(n_1(t) - n_3(t))] & \text{if symbol '1' is transmitted} \\
-2V + [(n_2(t) - n_4(t))] & \text{if symbol '0' is transmitted}.
\end{cases} \quad (3.39) \]

Reconstruction is carried out by using threshold detection where the threshold (\(Th\)) is considered to be exactly at the midpoint of the low and high bit levels. Hence, for antipodal signaling, the threshold level is considered to be zero volts.
If the same amount of noise affects both the wires in the twisted pair, i.e., if \( n_1(t) = n_3(t) \) and \( n_2(t) = n_4(t) \) in (3.39), the differential signaling scheme cancels out the noise completely. On the other hand, noise from electromagnetic interference (EMI) will be different on a differential signaling system if the two wires in the wire pair are not well balanced [20]. Another type of noise that the differential signaling cannot eliminate is the thermal noise. Thermal noise is always present at temperatures greater than absolute zero, due to the mobility of charge carriers in the cable. The thermal noise also results in random jitter, which leads to sampling error on the channel [21]. Since there are multiple wire pairs in the same cable bundle carrying different information like clock and data, there is a possibility of near end crosstalk from other pairs [22]. All of these sources of noise can be accounted for together as an additive noise with a statistical normal or Gaussian distribution [21, 22].

For the simulations in this thesis, the noise introduced in the communication channel is quantified using a parameter called noise power spectral density \( N_p \), expressed in Watts per Hertz. The noise signal is generated by creating a Gaussian distributed random variable for which the mean is equal to zero and variance is equal to the desired average noise power spectral density. This noise signal is added to the signal to simulate an additive noise in the system.

The noise power spectral density determines the bit error probability in the digital communications channel. For an antipodal transmission scheme, the relationship between the probability of bit error \( P_b \) and the signal-to-noise ratio \( E_b/N_p \) is
given by [7]

\[ P_b = Q \left( \sqrt{\frac{2E_b}{N_p}} \right) \]  \hspace{1cm} (3.40)

where \( E_b \) is the energy per bit, and \( Q \) is the Q-function. While calculating the energy per bit, the total attenuation of the signal discussed in Section 3.4.2 should also be considered. If \( N_b \) is the total number of bits in a packet, the probability of packet error can be found as

\[ P_p = 1 - (1 - P_b)^{N_b}. \]  \hspace{1cm} (3.41)

Both (3.40) and (3.41) assumes that the bit errors are independent of each other and are uncorrelated.

3.4.4 Error checking code generator and checker

The differential signals, which may be corrupted by noise, are transmitted along the channel and are reconstructed by using threshold detection at the receiving side. The reconstructed bits are then passed through an error checking algorithm. Two common techniques for error checking are cyclic redundancy check (CRC) and retransmission. The two techniques are described next.

The CRC algorithm is based upon modulo-2 division. The bits of the position information are first divided by a divisor polynomial. The remainder of this division process is included as part of the position data packet that is transmitted through the communications channel. At the receiver side, the same divisor polynomial used in the transmitter is used to divide the bits of the position information in the received
packet, to check whether the CRC bits computed and sent by the transmitter match the CRC bits newly computed by the receiver [23].

Figures 3.8(a) and 3.8(b) shows example cases in which the position data packet passes a CRC, and fails a CRC, respectively. Vertical bars in transmitted and received position data packet shows the boundaries between the single-turn position field (13 bits), the multi-turn count field (12 bits) and the CRC field (4 bits). The threshold for threshold detection at the receiver is denoted by $T_h$. It can be seen in the figures that the received voltage values at sample instants do not match those sent; this is due to noise and signal attenuation. In the first case, the received values at sample instants are still on the proper side of the threshold; and so are received correctly. In the second case, the last bit for single-turn position field, which is a logic high, is received as a logic low. This will cause a mismatch between the transmitted CRC bits and the ones computed by the receiver.

When using a CRC algorithm for error checking in a noisy environment, there is a chance that some combination of bit errors may remain undetected. There is a possibility of noise corrupting multiple bits in a way that would result in calculating the same remainder when dividing the received data packet by the CRC polynomial, so that the CRC passes and the bit errors are undetected. This is known as a false positive. The more the number of CRC bits, the less the chance for false positives. In a CRC algorithm, the probability of getting a false positive $P_{f,CRC}$ is given by [24] as

$$P_{f,CRC} = \frac{P_p}{2^{n_{CRC}}}$$  \hspace{1cm} (3.42)
Figure 3.8: Data reconstruction by threshold detection method
where $n_{CRC}$ is the number of bits used for CRC. This relationship holds as long as the number of bits of position information is much larger than the number of bits of CRC.

Another technique for error detection is retransmission. The transmitter sends each position data packet twice so that the two versions can be compared at the receiver side to verify whether they are the same. When using the retransmission method, it is possible that noise will corrupt the same bits of both packets, so that the packets as received are in error, but still identical. The probability that this will happen is very low, so there is only a low chance for getting a false positive when retransmission is used. The approximated probability for getting a false positive in a retransmission algorithm may be calculated using combinatorics as

$$P_{f,RET} = P_b^2 \cdot \frac{(N_b)}{2N_b}$$  \hspace{1cm} (3.43)

where $N_b$ is the total number of bits in a position data packet.

3.4.5 Packet pass/fail decision

The position information from a position data packet that passes the CRC is presumed valid, and it is sent to the control system as a feedback signal. The packets that do not pass the CRC are discarded altogether. In this case, position information from the most recent position data packet that passed the CRC will be re-sent to the control system as the feedback signal, as it is presumed to be the last known valid position.
In the case of retransmission, a position data packet is presumed valid if it is one of two consecutive identical packets. If valid, the position information is sent to the control system for feedback. If there is a mismatch, it is certain that at least one of the packets has been corrupted. In this case, position information from the last presumed valid packet is sent as feedback.

For either error checking scheme, a position data packet presumed valid is sent to the control system for feedback. If the packet is not valid, that is, if it is the result of a false positive, the control system will act based on erroneous position data. This will cause the control system to act to change the position inappropriately, resulting in error in the position.

3.4.6 Sampling rate selection for the system

So far a complete discrete-time model for the control system, including a realistic model for the communication channel, has been discussed. However, the sampling rate for the discrete-time system has not yet been selected. The sampling rate for the discrete-time system is chosen based upon three different parameters: the bit rate of the communication channel, the total number of bits being transmitted for a single position measurement, and the silent time. On a digital bus, it is not reasonable to expect to be able to transmit data at the maximum bit rate all of the time; buses are designed so that they are silent between data packets. The silent time is the period of time when the digital channel has no activity. Silent time is expressed as a certain percentage of the position transmission time, where the position transmission time is
the total number of bits being transmitted for a single measured position divided by
the maximum bit rate for a given cable length.

The total number of bits $N_b$ to be sent corresponding to a single position
measurement includes the number of bits for single-turn position $n_{ST}$ as well as the
number of bits for multi-turn count $n_{MT}$. The number of bits used for CRC $n_{CRC}$
and retransmission factor $R$ also contribute to the total number of bits. The retrans-
mission factor is the number of times the bits are retransmitted. The total number
of bits $N_b$ associated with a single fed-back position is

$$N_b = R [n_{ST} + n_{MT} + n_{CRC}] .$$

(3.44)

The bit rate divided by the total number of bits gives the time taken for the
transmission of a single position. The silent time $S$ is then added to this transmission
time to find the minimum period with which position can be fed back. The sample
time for the discrete-time system is set to this value, with the presumption that
sampling as fast as communications will allow is advantageous. So, the sampling
time $T_s$ for the system, and the control loop time, is

$$T_s = \left( 1 + \frac{S}{100} \right) \frac{N_b}{r}$$

(3.45)

where $r$ is the maximum bit rate of communication. The reciprocal of the sampling
time in (3.45) gives the sampling rate. This sampling rate selection method is used
to determine the sampling rate of the control loop, to compare and contrast the
performance of the control loop under different cable lengths. As different cable
length yields different bit rates, the sampling rate varies with the length of the cable.
The communication channel is operated at the maximum possible bit rate, so that the control system is operated at the fastest sample rate that the communications will allow. This sampling rate selection method ensures that the focus is mostly on the communication system aspect of the model.

3.5 Summary

This chapter presented a general overview of the entire simulation setup without setting particular numeric values for the simulation parameters. The application scenario of a blade pitch position control in a wind turbine is discussed. At first, the continuous-time models for the blade pitch control system were discussed. Then, the models were discretized in order to obtain a discrete-time model, such that the controller may be implemented on a digital computer. Next, a realistic model for the feedback communication channel was developed; this model considers the bit rate for communicating via a long cable, and the attenuation of the signal along the cable. A noise model for the communication channel was also developed, and the error checking methods were discussed. In the next chapter, the selection of numeric values for the control parameters and the communication parameters are made, and simulations of the blade pitch control are conducted to understand the impact that the features of the digital communication protocol have on the quality of control.
CHAPTER IV

SIMULATION RESULTS FOR AN EXAMPLE SYSTEM

The previous chapter discussed the setup used to simulate a system for a DC motor position control using a PID controller. The system includes a digital communications channel via which position information is fed back using a serial protocol to the controller; the contents of the position data packet were also described, along with the issues involved in modeling the communication channel. This chapter fills in the details needed to apply the simulation model to the example application, pitch control of a wind turbine blade. Simulation results for the example application are given, and the quality of control is studied for systems using communications protocols with different features.

4.1 Selection of simulation parameters for the control system

In the previous chapter, the models for the various pieces in the simulation set-up were described in a general way, without considering a particular motor or a controller. In this section, the simulation parameters are selected for the continuous-time motor model, the continuous-time PID controller and the discrete-time versions of the motor and the PID controller.
4.1.1 Parameters for continuous-time DC motor model

The first step in filling in the details for the simulation is to choose the parameters for the DC motor. The parameters were chosen arbitrarily to closely match the parameters for Faulhaber Series 2657-024CR commercial DC motor [25]. Table 4.1 shows the values of the DC motor parameters.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armature resistance $R_a$</td>
<td>2.84 Ω</td>
</tr>
<tr>
<td>Armature inductance $L_a$</td>
<td>$380 \times 10^{-6}$ H</td>
</tr>
<tr>
<td>Back emf constant $K_b$</td>
<td>0.0348 $V \cdot s \cdot rad^{-1}$</td>
</tr>
<tr>
<td>Torque constant $K_t$</td>
<td>0.0348 $N \cdot m \cdot A^{-1}$</td>
</tr>
<tr>
<td>Moment of inertia of rotor $J$</td>
<td>$1.7 \times 10^{-6}$ kg $\cdot m^2 \cdot s^{-2}$</td>
</tr>
<tr>
<td>Damping ratio of mechanical system $B$</td>
<td>$3.0 \times 10^{-6}$ N $\cdot m \cdot s$</td>
</tr>
</tbody>
</table>

With the parameters given in the table, and using the DC motor model transfer function derived in (3.10) in the previous chapter, the DC motor model has the unit step response shown in Figure 4.1(a). The steady state speed of the motor reaches 28.53 radians/sec. The pole-zero plot for the DC motor model is shown in Figure 4.1(b). The motor model consists of three poles as expected. The first pole is from the integrator ($P_1 = 0$). The second pole is from the mechanical part of the motor model ($P_2 = -262$). The third pole is from the electrical part of the model ($P_3 = -7210$).
(a) Unit step response of the continuous-time open-loop DC motor model from voltage input to speed output

(b) Pole-zero plot of the continuous-time DC motor model

Figure 4.1: Unit step response and pole-zero plot of the continuous-time DC motor model
4.1.2 Parameters for continuous-time PID controller

Next, the details of the continuous-time PID controller used to control the position of the DC motor model are specified. These parameters include the proportional constant, the integration constant, and the differentiation constant; the parameters were selected arbitrarily so that the rise time and the settling time were kept within acceptable limits. The selected parameter values for the PID controller are shown in Table 4.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional constant $K_p$</td>
<td>20</td>
</tr>
<tr>
<td>Integral constant $K_i$</td>
<td>100</td>
</tr>
<tr>
<td>Derivative constant $K_d$</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Now, the PID controller is implemented in the control system as shown in Figure 3.5. The root locus plot of the continuous-time DC motor model along with the PID controller is plotted for a range of gain values as shown in Figure 4.2(a). The expanded view for the root locus plot which shows the root locus of the poles near the origin is shown in Figure 4.2(b). The value of gain chosen for the continuous-time PID controller determined from the root locus plot; the gain $K$ is chosen so that the step response of the closed-loop system has a damping ratio of 0.707. This value of $K$ is found to be 0.255.
Figure 4.2: Root locus plot of the continuous-time DC motor position control with PID controller
Figure 4.3: Pole-zero plot of the closed-loop continuous-time DC motor position control system with PID controller with a gain of 0.255
The pole-zero plot for the closed-loop system is shown in Figure 4.3(a). The expanded view showing the roots near the origin are shown in Figure 4.3(b). The PID controller contributes one extra pole \((P_4 = -5.20)\) and two zeros \((Z_1 = -5.01, Z_2 = -1990)\) to the system. Due to the addition of the controller, the two dominant motor poles \((P_1 \text{ and } P_2)\) move so that they become a pair of complex conjugate poles. The new values of the motor model poles after the inclusion of the controller are \(P_1 = -135 - j135, P_2 = -135 + j135, \text{ and } P_3 = -7200\).

The unit step response of the closed-loop system with the chosen PID parameters and gain is shown in Figure 4.4. The characteristics of the unit step response of the closed-loop system are observed. The closed-loop system response is observed to have a rise time of 10.5 milliseconds, a settling time of 14.1 milliseconds, and an overshoot of 7.9%.

4.1.3 Parameters for discrete-time DC motor model

The ZOH method is used to convert the continuous-time motor model to a discrete-time motor model, as described in section 3.3.1 of Chapter 3. Figure 4.5 shows the unit step response of the discrete-time model of the DC motor when an example sample time of 0.4 milliseconds is used. It is observed that the step responses for the continuous-time system and the discrete-time system match exactly at the sample points. The continuous-time to discrete-time conversion by the zero order hold method adds two zeros \((Z_3 = -0.115 \text{ and } Z_4 = -2.148)\) to the discrete-time model.
of the motor, as shown in Figure 4.6. The discrete time motor poles are at $P_1 = 0.9875$, $P_2 = 0.9163$ and $P_3 = 0.062$.

4.1.4 Parameters for discrete-time PID controller

A discrete-time equivalent of the continuous-time PID controller is needed. The discrete-time PID controller is implemented using Equations (3.20), (3.21) and (3.22), with the same values for $K_p$, $K_i$ and $K_d$ as were used for the continuous-time PID controller and given in Table 4.2. The PID controller discretization adds a pole (P5) to the system. The value of the gain $K$ for the discrete-time version of the PID controller is chosen with the help of a root locus plot of the discrete-time closed-loop system. The gain value chosen is the one that results in the discrete-time closed-loop
system having a step response with a damping ratio ($\zeta$) of 0.707; in this way, the discrete-time systems are similar in performance to the continuous-time system. The root locus plot for an example discrete-time system with a sampling rate is 2500 Hz, which corresponds to the example sample time of 0.4 milliseconds, is shown in Figure 4.7(a). An expanded view of the root locus plot which shows the root locus of the poles near the right-hand corner of the unit circle is shown in Figure 4.7(b). The gain is found from the root locus plot to be 0.242. It can be seen that the gains for the continuous-time and discrete-time controllers are almost the same; the difference is due to the approximation used in converting from continuous time to discrete time. The pole-zero plot for the discrete-time closed-loop system is shown in
Figure 4.6: Pole-zero plot of the discrete-time DC motor model with a sample time of 0.4 milliseconds

4.8(a), and an expanded view showing locations of pole P4 and zero Z1 is shown in Figure 4.8(b). The values of the roots of the overall system are: P1 = 0.949 − j0.048, P2 = 0.949 + j0.048, P3 = 0.074, P4 = 0.9980, P5 = −0.006, Z1 = 0.9981, Z2 = 0.567, Z3 = −0.116, and Z4 = −2.15.

For integral anti-windup as discussed in (3.23), the maximum magnitude of the integral term \( f_{t,\text{max}} \) for the integral term in the PID controller is arbitrarily chosen to be 24. The discrete-time controller is now implemented with the parameters selected. The unit step response of the discrete-time DC motor position control system is shown in Figure 4.9. The unit step response of the discrete-time system closely matches the unit step response of the continuous-time system.
Figure 4.7: Root locus plot of the discrete-time DC motor position control system with PID controller with a sample time of 0.4 milliseconds
Figure 4.8: Pole-zero plot of the closed-loop discrete-time DC motor position control system with PID controller with a gain of 0.242 and a sample time of 0.4 milliseconds.
4.2 Selection of simulation parameters for the communication channel

So far, the discrete-time model for the system has assumed an ideal feedback communication channel. Next, the parameter values for the corresponding blocks in the realistic model for the communication channel are sought. These blocks are quantizer, feedback channel, noise model and the error checking code generation and detection and pass/fail detection.

4.2.1 Parameters for quantizer

In the quantizer, each position data is quantized as 13 bits of single-turn position and 12 bits of multi-turn count. These values are selected because they are the typical...
values for industrial absolute position encoders. A thirteen-bit single-turn position gives position with a resolution of 0.0439 degrees. A twelve-bit multi-turn count can specify 4096 unique turns. The total number of bits for the transmission of a single-turn position field and a multi-turn count field is 25 bits.

4.2.2 Parameters for feedback channel

The quantized position data is transmitted serially on a differential wire pair with signal levels +5 Volts for logic high and −5 Volts for logic low. The differential pair is a 24-AWG twisted pair cable; the cable’s length determines the maximum bit rate possible and the amount of attenuation of the signal along the cable. The lengths used in the simulation study are 100, 150, and 200 meters to accommodate the typical heights of wind turbine towers, as the communication wires must run from the top of the tower to the bottom of the tower. The values of maximum bit rate and signal attenuation are chosen according to the experimental data presented in Figures 2.5(a), and 2.5(b). The maximum bit rate is a function of cable length, but the signal attenuation per unit length is the function of operating frequency. Although a digital pulse has an infinite bandwidth in theory, in practice it is not possible or necessary to transmit the full pulse, with its entire harmonics, to the receiver. For this work, a bandwidth of 10 MHz is considered sufficient for the fastest bit rate (shorter cables), and thus is a conservative selection for the slower bit rates (longer cables); therefore, an operating frequency of 10 MHz is used when calculating signal attenuation at the
receiver. The bit rates and total attenuation corresponding to the cable length in the simulations are shown in Table 4.3.

Table 4.3: Values of maximum bit rate and total attenuation based on cable length

<table>
<thead>
<tr>
<th>Length (m)</th>
<th>Maximum bit rate (Mbps)</th>
<th>Total attenuation (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.26</td>
<td>6.56</td>
</tr>
<tr>
<td>150</td>
<td>0.82</td>
<td>9.84</td>
</tr>
<tr>
<td>200</td>
<td>0.64</td>
<td>13.12</td>
</tr>
</tbody>
</table>

4.2.3 Parameters for noise modeling

The amount of noise introduced into the communication channel is measured by the noise power spectral density, in units of Watts/Hz. An appropriate amount of noise is produced in simulation by generating a random Gaussian sequence with zero mean and variance equal to half of the desired noise power spectral density. Each element of the sequence is a noise power to be introduced at a particular time sample. This sequence is converted from power to voltage by taking the square root of each element; this assumes that the resistance has been normalized to 1 Ω. The resulting noise voltages are added to the attenuated signal voltages at the receiver end of the cable [26].
A separate simulation was performed to check that the process used to add noise to the communication bit stream is valid, and corresponds to the desired noise power spectral density. For this simulation, noise is added to a stream of bits, and the reconstruction process at the receiver is modeled, using the threshold to determine whether each received bit is a zero or one. The reconstructed received bits are then compared to the transmitted bits to determine which bits were received in error. The number of bits in error divided by total number of bits gives an estimate of the probability of bit error. Also, these bits are grouped into packets and a similar simulation is repeated. The number of packets in error divided by the total number of packets gives an estimate of the packet error rate. The estimates of the probability of bit error and packet error found through simulation are compared to ones calculated using the theoretical relationships of (3.40) and (3.41). The comparisons are shown for a simulation of one million bits in Figures 4.10(a) and 4.10(b). It can be seen that the simulated probabilities are a good match for the theoretical ones; this gives confidence that the noise simulations are done correctly. For high ratios of bit energy to noise power spectral density, the match is not as good; this is due to the fact that the number of bits in the simulations is not large enough to produce a sufficient number of bit errors. Increasing the number of bits in the simulations would make the model more accurate.
(a) Probability of bit error versus signal energy to noise power spectral density ratio

(b) Probability of packet error versus signal energy to noise power spectral density ratio

Figure 4.10: Comparison of theoretical results with simulated results for noise modeling
4.2.4 Parameters for error checking strategy

In the simulation study, two error checking methods are considered: CRC and retransmission. The simulation uses CRC of length $n_{CRC} = 4$, for which the CRC polynomial is $x^4 + x + 1$. This length of CRC is commonly used in industrial absolute position encoders. Retransmission with retransmission factor $R = 2$ is also studied, because it can provide good error checking without increasing the total number of bits for transmission overwhelmingly. Altogether, four error checking strategies are simulated: no error checking, error checking by using CRC, error checking by using retransmission, and error checking by using a combination of CRC and retransmission. The total number of bits for the position data packet for each strategy is given in Table 4.4.

Table 4.4: Total number of bits needed to communicate one measured position for each of the four simulated error checking strategies

<table>
<thead>
<tr>
<th>CRC used</th>
<th>Retransmission used</th>
<th>Total bits $N_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>25</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>29</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>50</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>58</td>
</tr>
</tbody>
</table>
4.2.5 Sampling rates selection for different simulation cases

Given a combination of cable length and error checking strategy, an appropriate sample rate for the system must be found for each simulation case. As described previously in Section 3.4.6, the sample rate chosen presumes that sampling is done as quickly as possible, given the maximum bit rate of the cable, the total number of bits transmitted per measured position, and the required sampling time. The silent time is considered to be $S = 75\%$. Once the sample time is chosen, the gain $K$ of the discrete-time PID controller can also be chosen, following the procedure outlined in Section 3.3.2. Table 4.5 shows the different sampling rates and gain values for the various simulation cases studied.

The discrete-time motor model, the discrete-time PID controller and the model for realistic communication channel developed so far are combined together to form a simulation setup that replicates a control system used in position control of blade pitch in a wind turbine. This simulation setup models a complete discrete-time position control system using a serial digital communication protocol for the sensed position.

4.3 Example time domain simulations

In order to illustrate the effects of the digital communications protocol on the quality of the position control of blade pitch in a wind turbine, a small set of example time domain simulations is done and described; this will be followed by a full simulation
Table 4.5: Sampling rate selection and gain for the discrete-time PID controller for various cases in the simulation study

<table>
<thead>
<tr>
<th>Length (m)</th>
<th>CRC used</th>
<th>Retransmission used</th>
<th>Sample rate (Hz)</th>
<th>Gain K</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>No</td>
<td>No</td>
<td>12642.22</td>
<td>0.252</td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
<td>No</td>
<td>10898.85</td>
<td>0.251</td>
</tr>
<tr>
<td>100</td>
<td>No</td>
<td>Yes</td>
<td>6321.13</td>
<td>0.249</td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
<td>Yes</td>
<td>5449.25</td>
<td>0.248</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
<td>No</td>
<td>8214.34</td>
<td>0.251</td>
</tr>
<tr>
<td>150</td>
<td>Yes</td>
<td>No</td>
<td>7081.33</td>
<td>0.250</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
<td>Yes</td>
<td>4107.17</td>
<td>0.246</td>
</tr>
<tr>
<td>150</td>
<td>Yes</td>
<td>Yes</td>
<td>3540.66</td>
<td>0.245</td>
</tr>
<tr>
<td>200</td>
<td>No</td>
<td>No</td>
<td>6049.44</td>
<td>0.249</td>
</tr>
<tr>
<td>200</td>
<td>Yes</td>
<td>No</td>
<td>5215.03</td>
<td>0.248</td>
</tr>
<tr>
<td>200</td>
<td>No</td>
<td>Yes</td>
<td>3024.72</td>
<td>0.243</td>
</tr>
<tr>
<td>200</td>
<td>Yes</td>
<td>Yes</td>
<td>2607.51</td>
<td>0.242</td>
</tr>
</tbody>
</table>
study. For the illustration, three different cases are presented. The cable length for all three cases is 100 meters. The first case uses no error checking, and assumes a noise power spectral density of 1 Watt/Hz. The second case also uses no error checking, but assumes an increased noise power spectral density of 1.2 Watts/Hz. The third case, which assumes the increased noise power spectral density of 1.2 Watts/Hz, adds CRC capability.

Because the example application is position control of blade pitch in a wind turbine, inputs appropriate to that situation should be used. The blade pitch of a wind turbine does not change by more than five degrees per second [27]. To observe the response of the system to step changes in commanded blade pitch angle, the simulated commanded position is made to increase or decrease by one degree in every one second.

The output of the simulation is the actual position. The error between the actual position and the commanded position is used as a measure of the quality of control. Errors in both a root mean square error sense and a maximum absolute error sense are considered.

The error in position for a position control system is defined as

$$E_1[n] = \theta_i[n] - \theta_o[n]$$  \hspace{1cm} (4.1)

where $\theta_i[n]$ is the commanded position, $\theta_o[n]$ is the actual position, and $n$ is the sample index. The error must be adjusted for the wrap-around that occurs at 360
degrees; the error $E[n]$ after the wrap operation is

$$E[n] = \begin{cases} 
E_1[n] - 360 & \text{if } E_1[n] > +180 \\
E_1[n] + 360 & \text{if } E_1[n] < -180.
\end{cases} \tag{4.2}$$

Now, the root mean square (RMS) error is calculated as

$$E_{rms}(t) = \sqrt{\frac{\sum_{all \ n} (E[n])^2}{N_{total}}} \tag{4.3}$$

where $N_{total}$ is the total number of samples in the simulation. The maximum absolute error is calculated as

$$E_{max}(t) = \max_{all \ n} |E(n)|. \tag{4.4}$$

The root mean square error in position is used when the quality of control is to be assessed in an average sense. The maximum absolute error in position is used when the quality of control is to be assessed in a worst-case sense. The root mean square error gives the measure of error in position for the entire simulation, whereas the maximum absolute error gives the measure of error in position at the instant the actual position deviates most from the commanded position.

The commanded and actual positions seen for the three example time-domain simulations are given in Figures 4.11, 4.12 and 4.13. The figures show 10 seconds of real-time data so that the features can be clearly seen; the actual simulations used to compute the root mean square error were at least 500 seconds long.

In Figure 4.11, the noise power spectral density is 1 Watt/Hz and no error checking is used. It can be observed that the actual position of the shaft angle
Figure 4.11: Commanded and actual position of a system with no error checking and a cable length of 100 meters, for a noise power spectral density of 1 Watt/Hz.

Figure 4.12: Commanded and actual position of a system with no error checking and a cable length of 100 meters, for a noise power spectral density of 1.2 Watts/Hz.
Figure 4.13: Commanded and actual position of a system with CRC and a cable length of 100 meters, for a noise power spectral density of 1.2 Watts/Hz

between the times of 4 and 5 seconds, deviates away from the commanded position by about five degrees. This is because the feedback received is an erroneous position data possibly corrupted in one or more of the most significant bits. This will cause the error signal in the controller to be a large value on which the controller acts. The controller then turns the motor as fast as possible in the direction indicated by the sign of the error signal. However, due to the limits imposed by actuator saturation, the motor is able to change the shaft angle by only about five degrees. After this change of about five degrees, the controller now starts to receive good position data packets, so the system slowly recovers toward the commanded position value. Between the times of 5 and 6 seconds, there is only a small deviation in actual position from the
commanded position because there is an erroneous data packet, but it is corrupted in one or more of the least significant bits, and so the error on which the controller acts is relatively small. The other small deviations in actual position that occur immediately following the step change in commanded position are due to overshoot in the characteristics of the control system. Over the entire simulation run, the root mean square error is observed to be 0.1138 degrees and the maximum absolute error is observed to be 4.8274 degrees.

In Figure 4.12, the noise power spectral density is increased to 1.2 Watts/Hz. As a result, there are several instances where the error in position is large because a high-order bit in a position field was corrupted; there are also several instances where the error in position is small because the position field was corrupted in a lower-order bit. In the time intervals of 7 to 8 seconds and 9 to 10 seconds, there are also instances where multiple additional corrupted position data packets are received while the control system is still recovering from a previous error. The errors for this case are significantly larger than for the first case, which was for a lower noise power spectral density. The root mean square error is 0.5650 degrees and the maximum absolute error is 4.8298 degrees.

In Figure 4.13, a significant reduction in error is observed due to the addition of a 4-bit CRC despite the noise power spectral density of 1.2 Watts/Hz. The actual position tracks the commanded position well, even in the presence of bit errors. The root mean square error is 0.0203 degrees and the maximum absolute error is 1.0205 degrees.
As seen in all of the time domain simulations, an erroneous position data packet is mostly likely to drop below the commanded position. This is because the multi-turn count is transmitted as part of the position data packet, and the simulation started with a multi-turn count of zero; as a result, if a bit of the multi-turn count is corrupted, the erroneous position is much larger than the commanded position. This causes the error input to the controller to be large and negative, so the controller will command the motor to reduce the position as quickly as is physically possible. In comparison, a corrupted bit in the single-turn position causes a relatively small error that may be either positive or negative, so the controller will command the motor to move in the direction of the error signal.

The time domain simulation provides the following insight: increasing the noise power spectral density increases position error, but for a given level of noise power spectral density, the position error can be kept low by adding error checking capabilities.

The root mean square error can be considered as the most appropriate measure for the quality of control since it can indicate the average quality of control achieved using the particular combination of parameters. Different runs of the simulation produced more consistent results in terms of root mean square error whereas the maximum absolute error tends to vary more from run to run, depending on the particular noise sequence injected into the communication channel.
4.4 Full simulation study

A full simulation study was done to understand the quality of control for position control systems with different cable lengths and different error checking strategies. Cable lengths of 100 meters, 150 meters and 200 meters were used. For each cable length, systems using four error checking strategies were simulated, for a total of twelve systems. The four error checking strategies are: no error checking (CRC=0; RET=0), error checking with a four-bit CRC (CRC=1; RET=0), error checking via retransmission (CRC=0; RET=1), and error checking combining a four-bit CRC and retransmission (CRC=1; RET=1). The twelve cases simulated, along with their respective sample rates and the gains for the discrete-time PID controllers were shown in Table 4.5.

The simulation of each system is performed by injecting noise of a given noise power spectral density and measuring the resulting root mean square error in position. The length of each simulation was chosen so that similar numbers of packets, no less than 250,000 packets, were transmitted in all simulations. The bit error probability plays an important in how long a simulation must be; it must be long enough for a significant number of erroneous positions to be encountered. Noise power spectral densities from 0 to 4 Watts/Hz are simulated. For each system and noise power spectral density, three simulation runs are done; seeing that the three runs give similar root mean square error gives confidence that the simulations were sufficiently long.
The time domain simulations done in the previous section give insight into what level of root mean square error in position is tolerable; those simulations show root mean square errors up to about 0.5 degrees. It is assumed that root mean square error of no more than 0.1 degrees is tolerable; the simulations in the full study provide the relationship between noise power spectral density and root mean square error. It is observed that for noise power spectral densities corresponding to more than about 0.1 degrees of root mean square error, the quality of control of the control system begins to degrade. Due to the use of integrator anti-windup and actuator saturation, the error in position at any point in time can be no more than about five degrees if the system is tracking the position command at all; a system with a root mean square error nearing five degrees is as bad as it can be, all the time. Systems with root mean square error values above about five degrees are receiving so much bad position data that they are not tracking the input position command.

Figure 4.14 shows the root mean square error for systems with a cable length of 100 meters. The results shows that systems using no error checking can tolerate noise with noise power spectral density up to 1 Watt/Hz. If a CRC is used, the system can tolerate noise power spectral density up to 1.4 Watts/Hz. If retransmission is used, the system can tolerate noise power spectral density of 2 Watts/Hz. When both retransmission and CRC are used, noise power spectral density of more than 4 Watts/Hz can be tolerated.

Figure 4.15 shows the root mean square error for systems with a cable length of 150 meters. The system using no error checking can tolerate noise with noise power
Figure 4.14: Root mean square error for systems with cable length of 100 meters

Figure 4.15: Root mean square error for systems with cable length of 150 meters
spectral density up to 0.4 Watts/Hz. If a CRC is used, the system can tolerate noise power spectral density up to 0.6 Watts/Hz. If retransmission is used, the system can tolerate noise power spectral density of 1 Watt/Hz. When both retransmission and CRC are used, noise power spectral density of up to 1.4 Watts/Hz can be tolerated.

Figure 4.16 shows the root mean square error for systems with a cable length of 200 meters. The system using no error checking can tolerate noise with noise power spectral density up to 0.2 Watts/Hz. If a CRC is used, the system can tolerate noise power spectral density up to 0.3 Watts/Hz. If retransmission is used, the system can tolerate noise power spectral density of 0.4 Watts/Hz. When both retransmission and CRC are used, noise power spectral density of up to 0.6 Watts/Hz can be tolerated.

![Figure 4.16: Root mean square error for systems with cable length of 200 meters](image-url)
In Figures 4.14, 4.15 and 4.16, it can also be seen that when noise is not introduced to the system, the root mean square error is almost equal to zero degrees.

Insight into the results can be gained by considering how the theoretical probabilities of a bit error, a packet error, and a false positive are affected by noise power for the systems simulated. The theoretical values of the bit error probabilities for noise power spectral densities of 1 Watts/Hz to 4 Watts/Hz, calculated by using (3.40), are given in Table 4.6. It can be seen that the bit error probabilities increase when the cable length increases for a given noise power. This is due to the attenuation of the signal along the cable. For a given cable length, bit error probability grows with increasing noise power.

Table 4.6: Estimated theoretical values of bit error probabilities for systems with different cable lengths

<table>
<thead>
<tr>
<th>Cable Length (m)</th>
<th>Bit error probabilities for noise power spectral density of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Watt/Hz</td>
</tr>
<tr>
<td>100</td>
<td>1×10^{-6}</td>
</tr>
<tr>
<td>150</td>
<td>0.0006</td>
</tr>
<tr>
<td>200</td>
<td>0.0137</td>
</tr>
</tbody>
</table>

Table 4.7 shows the theoretical values of the percentage of corrupted position data packets, which are calculated using (3.41). The percentage of corrupted packets
gives insight as to whether a particular system will result in reasonable values for
the root mean square error in position. The root mean square error increases as the
percentage of packets corrupted increases, and the system will completely fail if the
percentage of corrupted packets is 90 percent or more, even if sophisticated error
checking strategies are used.

Table 4.8 shows the percentage of false positive packets for the systems that
use CRC and retransmission for error detection; the percentage is calculated using
\(3.42\) and \(3.43\). It can be seen that the percentage of false positive packets for
systems with retransmission is significantly lower than that for systems with CRC
for a given cable length. So, it can be deduced that for a given cable length, a larger
percentage of the packets are corrupted when a sophisticated error checking strategy
is used, simply because each packet contains more bits; however, the number of false
positive packets is significantly reduced. This explains the improvement in root mean
square error obtained when more sophisticated error checking is implemented in the
system.

In the simulation for position control of a blade pitch in a wind turbine, it can
be observed that the system having a cable length of 200 meters using combination
of both CRC and retransmission for error checking tolerates almost the same level
of noise as the system with a cable length of 100 meters without any error checking.
For a given error checking strategy, root mean square error grows with cable length.

In conclusion, given a level of noise and an error checking strategy, the quality
of control is better for systems with shorter cables. If an application requires a certain
cable length and must tolerate a certain level of noise, the quality of control can be improved by using some more sophisticated error checking. The systems with a 4-bit CRC yield a better quality of control than systems with no error checking, and systems with retransmission yield much better quality of control than those with 4-bit CRCs.

When choosing an industrial protocol for an absolute position encoder to be used in a particular application, the expected level of noise and cable length should be considered in deciding which protocol features are necessary. Protocols without error checking capabilities are clearly not sufficient for communicating over long distances in noisy environments.
Table 4.7: Estimated theoretical values of percentage of corrupted position data packets for the twelve systems simulated, for noise power spectral densities of 1 Watt/Hz, 2 Watts/Hz, 3 Watts/Hz and 4 Watts/Hz

<table>
<thead>
<tr>
<th>Cable Length (m)</th>
<th>CRC used</th>
<th>Retransmission used</th>
<th>Percentage of corrupted packets for noise power spectral density of 1 Watt/Hz</th>
<th>2 Watts/Hz</th>
<th>3 Watts/Hz</th>
<th>4 Watts/Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>No</td>
<td>No</td>
<td>0.0033</td>
<td>1.1112</td>
<td>8.0233</td>
<td>21.0537</td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
<td>No</td>
<td>0.0038</td>
<td>1.2878</td>
<td>9.2459</td>
<td>23.9841</td>
</tr>
<tr>
<td>100</td>
<td>No</td>
<td>Yes</td>
<td>0.0066</td>
<td>2.2100</td>
<td>15.4028</td>
<td>37.6749</td>
</tr>
<tr>
<td>100</td>
<td>Yes</td>
<td>Yes</td>
<td>0.0076</td>
<td>2.5590</td>
<td>17.6369</td>
<td>42.2158</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
<td>No</td>
<td>1.5894</td>
<td>24.9097</td>
<td>55.0785</td>
<td>74.8338</td>
</tr>
<tr>
<td>150</td>
<td>Yes</td>
<td>No</td>
<td>1.8413</td>
<td>28.2739</td>
<td>60.4773</td>
<td>79.8187</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
<td>Yes</td>
<td>3.1535</td>
<td>43.6145</td>
<td>79.8206</td>
<td>93.6666</td>
</tr>
<tr>
<td>150</td>
<td>Yes</td>
<td>Yes</td>
<td>3.6487</td>
<td>48.5536</td>
<td>84.3795</td>
<td>95.9272</td>
</tr>
<tr>
<td>200</td>
<td>No</td>
<td>No</td>
<td>29.0819</td>
<td>78.3081</td>
<td>93.0717</td>
<td>97.3282</td>
</tr>
<tr>
<td>200</td>
<td>Yes</td>
<td>No</td>
<td>32.8760</td>
<td>83.0134</td>
<td>95.4801</td>
<td>98.5035</td>
</tr>
<tr>
<td>200</td>
<td>No</td>
<td>Yes</td>
<td>49.7063</td>
<td>95.2946</td>
<td>99.5200</td>
<td>99.9286</td>
</tr>
<tr>
<td>200</td>
<td>Yes</td>
<td>Yes</td>
<td>54.9437</td>
<td>97.1146</td>
<td>99.7957</td>
<td>99.9776</td>
</tr>
</tbody>
</table>
Table 4.8: Estimated theoretical values of percentage of false positives for systems using CRC and retransmission for error checking

<table>
<thead>
<tr>
<th>Cable Length (m)</th>
<th>CRC used</th>
<th>Retransmission used</th>
<th>1 Watt/Hz</th>
<th>2 Watts/Hz</th>
<th>3 Watts/Hz</th>
<th>4 Watts/Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Yes</td>
<td>No</td>
<td>0.0002</td>
<td>0.0805</td>
<td>0.5779</td>
<td>1.4990</td>
</tr>
<tr>
<td>100</td>
<td>No</td>
<td>Yes</td>
<td>$3 \times 10^{-12}$</td>
<td>$4 \times 10^{-7}$</td>
<td>$2 \times 10^{-5}$</td>
<td>$1 \times 10^{-4}$</td>
</tr>
<tr>
<td>150</td>
<td>Yes</td>
<td>No</td>
<td>0.1151</td>
<td>1.7671</td>
<td>3.7798</td>
<td>4.9887</td>
</tr>
<tr>
<td>150</td>
<td>No</td>
<td>Yes</td>
<td>$8 \times 10^{-7}$</td>
<td>0.0003</td>
<td>0.0020</td>
<td>0.0059</td>
</tr>
<tr>
<td>200</td>
<td>Yes</td>
<td>No</td>
<td>2.0547</td>
<td>5.1883</td>
<td>5.9675</td>
<td>6.1565</td>
</tr>
<tr>
<td>200</td>
<td>No</td>
<td>Yes</td>
<td>0.0004</td>
<td>0.0072</td>
<td>0.0209</td>
<td>0.0371</td>
</tr>
</tbody>
</table>
CHAPTER V

CONCLUSIONS

5.1 Summary

The quality of control in a position feedback control system that uses a digital communications protocol for the position feedback is studied. Features of the various protocols used for the communication of the absolute position information in the feedback channel are studied. The performance of the communication channel depends on the length of the cable, since the length of cable determines the maximum bit rate and the signal attenuation at the receiver end of the cable. The performance also depends upon the total number of bits used for representing the position information. To enhance the quality of control, error checking measures such as CRC and retransmission are suggested.

A particular application scenario of a blade pitch of a wind turbine is simulated where the input is a commanded position value for the blade pitch, and the output is the present actual position of the blade pitch. The simulation model includes the motor controlling the blade pitch, the discrete-time PID controller, and an absolute position encoder communicating via a serial cable in a noisy environment. The quality of control of a particular system is measured in terms of the root mean
square error in blade pitch position. In the motor model, only the inertia of the motor is considered, while the inertia of the wind turbine blades is neglected. In an actual system, the inertia of the blades will play a big role in the system response; overall, the system would be slower. As a result, if the inertia of the blades were considered, the root mean square error would be significantly reduced, because when an erroneous position is received, the controller will not be able to move the system so quickly in response. Therefore, neglecting the inertia of the blades is a conservative approach.

The series of simulations are performed for systems communicating over cable of different lengths that match the heights of typical wind turbines. The respective root mean square error in position of the various systems is observed. For a given length of cable, it is found that root mean square error in position increases as the noise power spectral density is increased. Using CRC and retransmission for error checking increases a system’s tolerance of noise.

In the simulation, the framework was intentionally developed to use a fast sampling rate, even though such fast sampling is not strictly necessary for the wind turbine example. This was done to highlight the trade-off between the amount of error checking and the sampling rate that can be achieved. Some control systems, such as those that use a current control loop, require fast sampling, so that using additional error checking bits requires that sample rate be reduced. For many other control systems, in which slow sampling is sufficient, it will be generally better to use a communication protocol with a high number of error checking bits or with a sophisticated error checking scheme.
5.2 General recommendations

The transmission of absolute position values for feedback along long cables requires a sophisticated error checking scheme even when noise power spectral density is low; for short cables, the same quality of control can be found using a simpler error checking scheme. If the noise power spectral density is too high, a high percentage of position data packets are lost, and the control system does not track the position command.

Error checking using CRC is beneficial when transmitting the absolute position values are transmitted in an operating environment that includes significant noise. Retransmission allows for even more robustness in the presence of noise. The combination of CRC and retransmission is found to result in a more robust control than either CRC or retransmission alone.

Based on the results of the simulations done in this thesis, one can reasonably expect industrial protocols for absolute position encoders that include CRC error checking to perform significantly better in noisy environments than those that do not. The industrial protocols BiSS, EnDat or Hiperface all offer CRC for error detection. The SSI protocol does not have CRC error checking built in; if using this protocol in a noisy environment, it may be necessary to add error checking on top of the protocol by using retransmission. SSI also lacks key features such as alarms, warnings or diagnostic information to report inconsistencies about the internal circuitry in the encoder, and support for multi-slave networks.
This work highlights the trade-offs a user must make when choosing a communication protocols for an absolute position encoder for a particular application. Simulation of the application scenario, with corresponding parameters, can help in deciding what protocol features are important. The parameters to be identified include the bit resolution for the single-turn position field, the bit resolution for the multi-turn count field, the number of bits used for CRC, the amount of noise power spectral density in the operating environment, and whether retransmission is used.

5.3 Future work

The simulation study presented in this thesis uses a generalized noise model that assumes that the noise is Gaussian. The assumption may not hold true in a real world application. Future work could be done to develop a noise model particular to differential signaling. For the development of this noise model, the model for the twisted wire pair should be developed so that the impedance imbalance between the two wires in the pair is taken into consideration. Simulations done with such a noise model would allow for more accurate prediction of the values of the root mean square error of the position in real systems; however, the general recommendations made here are not likely to change.

In this work, a fixed number of bits are considered for the representation of the single-turn position data, the multi-turn count and the CRC. Another extension to this work is to analyze the effect of varying the number of bits for each of these fields in the position data packet. While the number of bits of single-turn position
is nearly always thirteen in industrial absolute position encoders, and comes from the resolution of the underlying sensing mechanism, it would be relatively easy to implement new digital communications protocols with increased length CRCs. A study like the one done here would give insight into whether such an increase is worthwhile.
BIBLIOGRAPHY


