Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2018

Modelling and Fault Detection of an Overhead Travelling Crane System

Ingrid Sjöberg

Master of Science Thesis in Electrical Engineering

Modelling and Fault Detection of an Overhead Travelling Crane System

Ingrid Sjöberg

LiTH-ISY-EX–18/5127–SE

Supervisor: **Shervin Parvini Ahmadi**
ISY, Linköpings universitet
Wolfgang Reinelt
Schneider Electric Automation GmbH

Examiner: **Martin Enqvist**
ISY, Linköpings universitet

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Hoists and cranes exist in many contexts around the world, often carrying very heavy loads. The safety for the user and bystanders is of utmost importance. This thesis investigates whether it is possible to perform fault detection on a system level, measuring the inputs and outputs of the system without introducing new sensors. The possibility of detecting dangerous faults while letting safe faults pass is also examined.

A mathematical greybox model is developed and the unknown parameters are estimated using data from a labscale test crane. Validation is then performed with other datasets to check the accuracy of the model. A linear observer of the system states is created using the model. Simulated fault injections are made, and different fault detection methods are applied to the residuals created with the observer. The results show that dangerous faults in the system or the sensors themselves are detectable, while safe faults are disregarded in many cases.

The idea of performing model-based fault detection from a system point of view shows potential, and continued investigation is recommended.

Acknowledgments

I want to start by thanking my supervisor at Schneider Electric, Wolfgang Reinelt. Thank you for your guidance and expertise, and always being there to answer my many questions.

Further I want to give thanks to my supervisor and examiner at Linköping University, Shervin Parvini Ahmadi and Martin Enqvist, for supporting me and giving me helpful comments.

My colleagues at Schneider were most welcoming. Thank you for intern breakfasts, teaching me interesting German words, introducing me to smelly Chinese tea, and not letting me forget the importance of face control.

Vielen Dank an Familie Zimmermann die dafür gesorgt haben dass ich mich in Deutschland wie zuhause fühlen dürfte.

Sollentuna, July 2018
Ingrid Sjöberg

Contents

Notation

Model

Symbol	Description
p_x	Position of trolley on bridge
$\dot{p}_x$	Velocity of trolley along bridge
r	Length of hoisting wire
$\dot{r}$	Hoisting speed
θ	Sway angle of the hoist wire to the vertical line
$\dot{\theta}$	Angular velocity of sway
x	State vector
u	Input vector
y	Measurement vector
F	Force of motor acting on trolley
C	Torque of hoist motor acting on hoist wheel
m	Mass of load
m_t	Mass of trolley
J	Hoist wheel inertia
b	Radius of hoist wheel
g	Acceleration of gravity
c_x	Friction related constant for trolley
c_r	Friction related constant for hoist wheel
c_θ	Friction related constant for swaying

Fault detection

Symbol	Description
y	Simulated output vector
$\hat{y}$	Observer output vector
ϵ	Residual
s	Symptom
g	Alarm
A,B, C	Matrices for linearised model
A_2, B_2, C_2	Matrices for linearised observer
L	Luenberger observer gain
θ	Variable in which a change is to be detected
θ_0	Variable before change
θ_1	Variable after change
p_{θ_0}, p_{θ_1}	Distribution of random variable under the assumption that $\theta = \theta_0$, $\theta = \theta_1$
$\mathbb{E}_{\theta_0}$, $\mathbb{E}_{\theta_1}$	Expectation of random variable subject to p_{θ_0}, p_{θ_1}
μ	Mean of normal distribution
σ	Standard deviation of normal distribution
ν	Magnitude of change in mean
b	Signal to noise ratio
$\mathbf{H}_0$, $\mathbf{H}_1$	Hypothesis
d	Decision function
h	Threshold
S	Cumulative sum
n	Allowed drift parameter, CUSUM
λ, α	Weights, forgetting factor GMA

Abbreviations

Abbreviation	Description
CUSUM	CUmulative SUM
FPE	(Akaike's) Final Prediction Error
GMA	Geometric Moving Average
IFAC	International Federation of Automatic Control
LS	Least Squares
NARX	Nonlinear Auto-Regressive model with eXogenous input
NRMSE	Normalised Root Mean Square Error
PDF	Probability Density Function
PI	Proportional, Integral (regulator)
PID	Proportional, Integral, Derivative (regulator)
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RLS	Recursive Least Squares
RPM	Revolutions Per Minute (unit)

1

Introduction

This thesis concerns the modelling and fault detection of an overhead travelling crane from a safety perspective. The work has been performed on request of Schneider Electric Automation GmbH, at their facilities in Marktheidenfeld, Germany.

In the first chapter an introduction is given, the purpose and method of the project is defined and the structure of the thesis presented.

1.1 Background

Different types of hoists and cranes are used in industry to lift and move heavy loads. The type of crane considered in this thesis, called *overhead travelling crane*, *bridge crane* or *girder crane*, is often used to carry loads of up to 100 tonnes distances reaching 50 meters. Transportation of goods of this size and weight can be dangerous. Failures or mistakes can lead to unsafe situations for the operator, bystanders or the construction itself. During 2016, 262 crane-related work-place accidents, leading to absence from work, were reported in Sweden alone. Of these, 90 accidents were reported to have been caused by overhead travelling cranes or gantry cranes [1].

In earlier days of industry, safe operation of the crane was the responsibility of the crane operator. Today, cranes come equipped with many advanced safety functions, such as overload control, anti-sway and speed-optimisation. Functions like these minimise the risk of human error when working a functional crane. However, accidents can also be caused by the crane malfunctioning. Some accidents caused by a failure in the crane can be avoided using fault detection. This method uses knowledge of the system's usual behaviour to detect faults, or changes in behaviour, before a failure can occur.

Functional safety is an area that is always looking for improvements, since even slight advances can improve the safety for the user.

1.2 Purpose and objectives

Many cranes today already come with some sort of fault detection system. Often each component in a crane comes with some sort of signal monitoring system, that raises an alarm when the signal deviates from the norm. This is an effective way of detecting imperfections in a system, but from a safety point of view it can be too precise. For a crane system, there is a large grey area between perfect functionality and possible danger. For example if the crane starts moving a little slower than expected, there is probably a fault within the system, but it is not generally a dangerous situation. However, if the crane starts moving in the opposite direction than intended, or it starts moving faster than intended, an unsafe situation may occur.

The purpose of this thesis is to investigate fault detection from a safety and system point of view. The question is whether it is possible to create a model-based fault detection system for an overhead travelling crane, observing the available system signals, that gives alarms only in case of faults that can lead to harmful conditions.

Based on the project, Schneider Electric can decide whether the idea is worth developing further or not.

1.3 Method

The problem is divided in two parts. Firstly, a physical model of a crane will be developed and validated using data from a test crane. This model will be used as the true system for the second part of the thesis, which is fault detection. During the fault detection, an observer will be developed for estimation of the states. The difference between the true system (simulation model), and the observed states will be used as residuals for the fault detection.

Relevant faults will be defined, and classified according to their effects on the system. Relevant means that they can result in a dangerous situation, and that they can be observed using the available output signals.

Different change detection techniques will be applied and evaluated.

1.4 Scope

A test crane available in the Schneider Electric Marktheidenfeld lab is used as a basis for the thesis. The crane has a fixed girder, which means that the load can move in one horizontal direction and vertically. Since it is only possible to validate movement in two dimensions, the model will also be two-dimensional.

The model is derived based on physical relations, so it should be possible to adapt the model to other similar cranes by changing the parameters.

As previously mentioned, the simulation model will be used as the *true system* during fault detection. This is because performing fault insertions on a real setup would take time and resources that do not reflect the extent of this thesis.

The sensors treated in this project provide information about the speed of the trolley, and the hoisting speed and position. Thus, only faults affecting these signals can be detected.

1.5 Related work

Many reports are written on modelling of cranes with the objective to use the model for developing control methods. Fliess et al. develop a generalised state space model of a two-dimensional crane. They use this model as a basis for developing a linearising feedback loop for position control. They also conclude that their results can be generalised to the three-dimensional case [4].

Garcia et al. treat the case of an three-dimensional crane lifting a load that is not placed directly underneath the trolley [6]. Lifting an off-centered load leads to significant sway. Garcia and his companions develop a model accurately predicting this movement, which will be used to develop a sway-minimising control system.

Anti-sway control of cranes is also handled by Terashima et al. [20]. They propose a load position feedback control, as opposed to the more conventional trolley position feedback. Using this control they manage to minimise the sway and the risk of load collision.

Regarding fault detection of crane systems less is to be found. In general the articles are focused on a single sensor or fault. Pohjola et al. look into a malfunctioning ultra sound sensor, measuring the load angle in an overhead travelling crane system [17]. They investigate the performance of different Kalman filters for detecting the faults.

Henao et al. take on the rather common issue of bird-caging of the hoisting wire [7]. This is a form of degrading of the hoisting wire, caused by regular wear and tear. The team investigate whether it is possible to detect this issue by analysing the motor torque and current signature.

A system approach to fault detection is used by Saif and Chen [19]. They develop an input/output relation for diagnosis of actuator faults in nonlinear systems with partially unknown inputs, to be used instead of nonlinear observers. Additionally, the results are applied to a 3D bridge crane.

A thorough examination of fault detection methods in general was performed in 2015 by Gao, et al. [5].

Model based change detection methods are well described in a series of articles written by Reinelt et al. [18], [15] and [16], and Malinen [14]. There, the methods are applied to the situation of active front steering, and different types of sensor failures.

1.6 Thesis outline

The chapters in this thesis are arranged in chronological order, since each task is based on the previous. First, Chapter 2 provides a system description, giving information about overhead travelling cranes in general and the test crane in particular. This is followed by modelling of the crane, and validation of the model in Chapter 3. Next, in Chapter 4, the fault detection methods are introduced, and the simulations and results are given in Chapter 5. Discussion of methods and results is found in Chapter 6, followed by conclusions and recommendations for future work.

2

System overview

Overhead travelling cranes, also called bridge cranes or girder cranes, are normally used indoors in industrial facilities. The main bearing unit is the bridge or girder. This bridge can travel horizontally on runways mounted in the ceiling, on the walls or sometimes on stands on the ground. A trolley carrying the hoist moves along the bridge, allowing for movement in three dimensions. See Figure 2.1 for a sketch.

The specific crane considered in this thesis is a mock-up crane used for training and development of safety functions. As previously mentioned it has a fixed bridge, which means that movement in the Y direction, defined in Figure 2.1, is not possible. The crane measures 292 cm (bridge length) by 216 cm (bridge height).

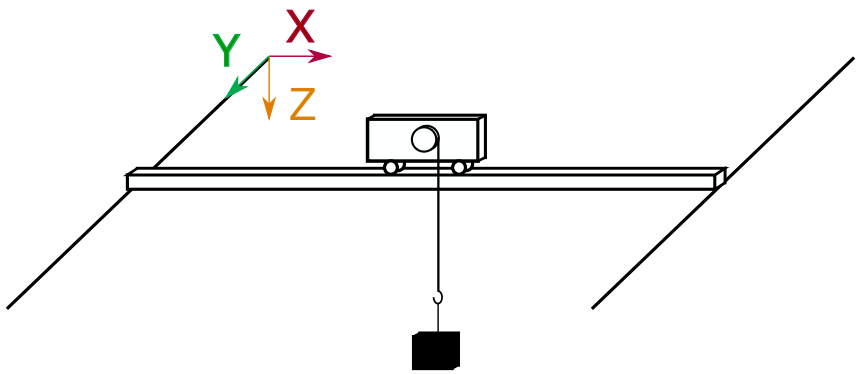

Figure 2.1: *Overhead travelling crane, with trolley moving in the X direction on a girder, the girder moving on rails in the Y direction, and a hoist mounted on the trolley allows movement of the load in the Z direction.*

The entire system is programmed and configured using SoMachine software and controlled by a Programmable Logic Controller (PLC). As the system is used for training and development it is equipped with many sensors and functions. Since the aim of this thesis is detecting faults in a normal setup under normal conditions, only the common sensors and basic functionality were enabled during tests. The choice of sensors was made based on a system description.

2.1 Inputs and sensors

Inputs to the test crane are given via a hand-held wireless remote control in the form of reference speeds for the trolley and the hoist. The remote buttons for crane movement are illustrated in Figure 2.2. The functionalities of the buttons are explained in Table 2.1. Button 4 is placed on the back of the remote, so the remote can be operated with one hand.

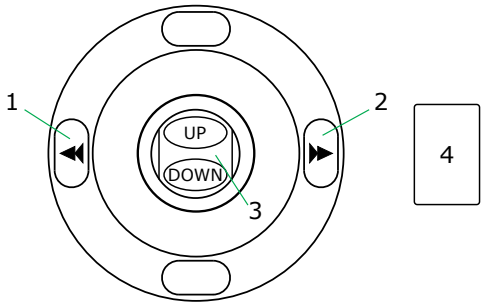

Figure 2.2: *Remote buttons providing the input to the system. The button commands are given in Table 2.1*

Table 2.1: *Remote button functionality.*

Button	Action	Instruction
1	Press halfway	Move trolley left, normal speed
	Press fully	Move trolley left, high speed
2	Press halfway	Move trolley right, normal speed
	Press fully	Move trolley right, high speed
3	Move downward	Lift load, normal speed
	Move upward	Lower load, normal speed
3 + 4	Move 3 downward, press 4	Lift load, high speed
	Move 3 upward, press 4	Lower load, high speed

The value for the *normal* and *high* speeds are set by the user through the SoMachine computer interface. The chosen speeds can differ between the trolley and the hoist.

Further information is given to the system by the sensors listed in Table 2.2, where the axes are defined as in Figure 2.1. To make sure the trolley does not

run with full speed into the end of the bridge, the bridge has two limit switches on each side. When the trolley reaches the first limit switch, the setpoint speed is set to the safety level, and when the second is reached the setpoint is set to zero. At either end of the bridge there is also a buffer absorbing the energy from any collision that may still occur. Since the load can have different heights, there are only two limit switches for the hoist, stopping the load at the highest allowed position.

Table 2.2: *List of sensors in the test crane.*

Axis	Sensor	Information
x	Limit switches	Endpoint reached, trolley
	Encoder	Trolley speed
z	Limit switches	Endpoint reached, hoist
	Encoder	Hoist speed
		Hoist position

2.2 Hardware and information flow

An overview of the information flow in the system is given in Figure 2.3.

The references are provided using a Schneider Electric Harmony eXLhoist wireless remote. The references are then collected at a receiver, and transmitted to the controller. The controller of the test crane is a Modicon M241 Logic Controller, with model number TM241CEC24T/U. This PLC controls both the trolley and the hoist. Next, the reference speeds are combined with information from the limit switches to provide setpoint speeds for the variable speed drives.

The drives, an Altivar ATV930 (model number ATV930U07N4) for the trolley, and an Altivar ATV340 (model number ATV340U07N4) for the hoist, then adapt the PWM duty cycles for the motors to reach the desired speeds. Both motors are produced by Sew-Usocome. The trolley motor is a helical gearmotor of type R17 DT71D4/BMG/TF/VR/ES1T, and the hoisting motor is a helical-worm gearmotor of type SAF37 DR63L4/BR.

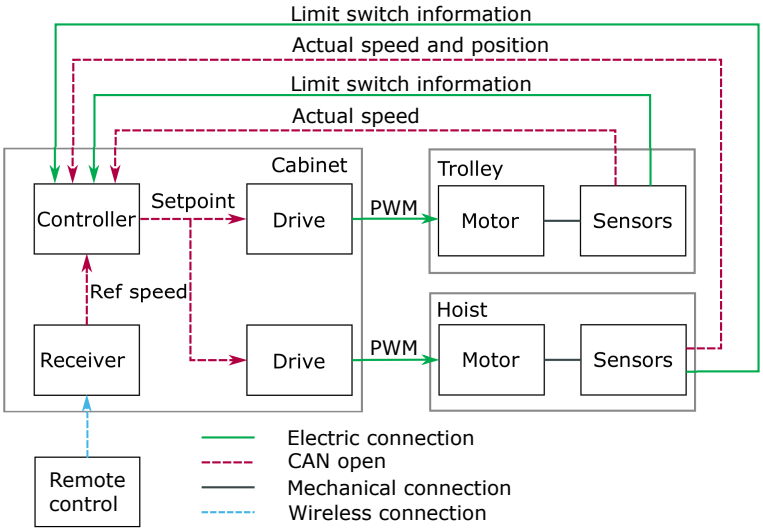

Figure 2.3: *Information flow in the test crane. The remote control provides the reference speeds wirelessly, the receiver, controller and drives are placed in an electrical cabinet, while the motors and sensors are placed on the crane.*

2.3 Test data

Data for parameter estimation and model validation are collected at the test crane. In SoMachine, parameters and variables can be monitored using a trace command. This functionality allows to register any signals accessible to the PLC. The signals in Table 2.3 were sampled with a sampling time of 40 ms.

Initial calibration tests were done to find the relations between the speeds in RPM and [m/s]. Table 2.4 shows the results for the preset speeds used in the calibration tests.

Table 2.3: *Measured signals from the test crane.*

Signal	Unit or span
Reference speed trolley	RPM
Actual speed trolley	RPM
Input, Forward trolley	[0,1]
Input, Reverse trolley	[0,1]
Input, High speed trolley	[0,1]
Reference speed hoist	RPM
Actual speed hoist	RPM
Relative position hoist	[0,2^16]
Input, Forward hoist	[0,1]
Input, Reverse hoist	[0,1]
Input, High speed hoist	[0,1]

Table 2.4: *Preset speeds for the test crane motors in RPM and the resulting translational speeds in [m/s]*

Name	**RPM**	**[m/s]**
Trolley safe speed	100	0.02
Trolley normal speed	500	0.12
Trolley high speed	1500	0.35
Hoist safe speed	100	0.03
Hoist normal speed, calibration	300	0.08
Hoist normal speed, tests	800	0.21
Hoist high speed	1500	0.39

3

Modelling and validation

When creating a fault detection system, data from the true system performing fault free is necessary to establish the nominal behaviour. One also needs data from the system when faults occur, to be able to test the system, but also to know for which deviations to look . Making a fault injection in a real overhead crane installation can lead to potentially dangerous situations. Thorough safety measures are needed, and it is both time and resource consuming. During this project simulations will thus be used instead of real life fault injections.

In this chapter, a physical model of the test crane is introduced. A physical model is a set of equations describing the dynamics of the system. The equations are derived from the laws of physics, and some level of detail is chosen.

There are other techniques for creating models. It is for example possible to fit a standard model type to the measured data, an approach known as black-box modelling. This method often gives a better fit to measurement data than a physical model, due to the level of abstraction in the physical model. However, in this case certain aspects of the system are not measurable, e.g. the swinging of the load, which means the black-box method cannot model that behaviour. The goal is also to have a general model of a crane, where it is possible to introduce faults in specific parts, rather than a model acting exactly like the test crane. Thus a physical modelling approach is chosen.

3.1 Crane model

A sketch of the test crane is given in Figure 3.1. The model treats the trolley as a box being pushed or pulled by the force of the motor. The hoist is described with a hoist wheel, on which the hoisting motor applies a torque. Frictions for trolley movement, hoisting and swinging are described as proportional to the speeds and angular velocity.

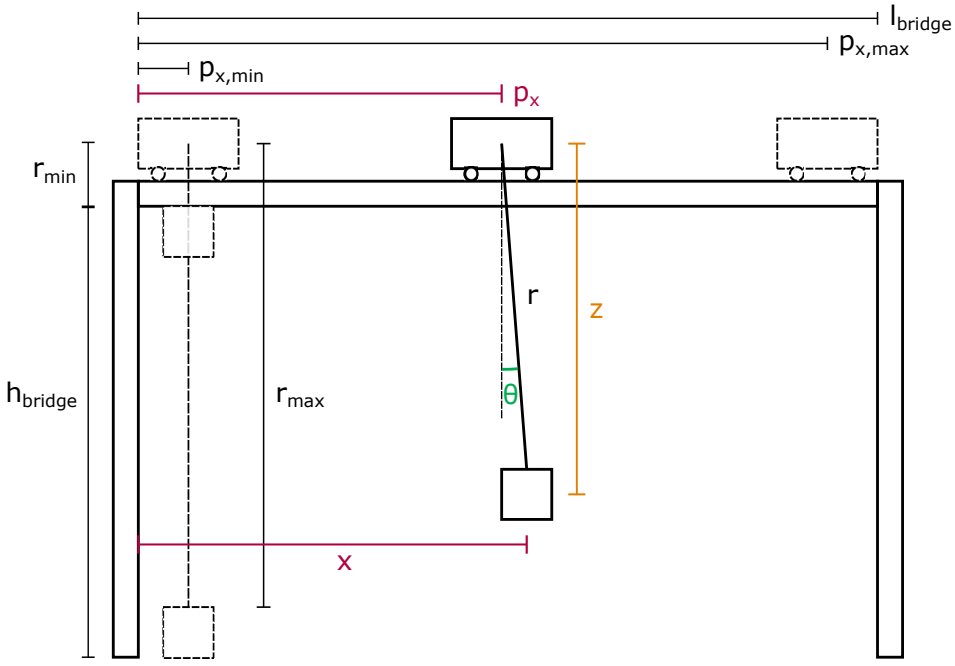

Figure 3.1: *Illustration of the test crane. The variable px is the position of the trolley on the girder, r is the hoist rope length, θ is the load swing angle, x is the position of the load along the girder, and z is the position of the load under the hoist. The minimal and maximal positions of both trolley and load are also indicated.*

The position of the load is given by

$$\begin{aligned} x &= p_x + r\sin\theta \\ z &= r\cos\theta, \end{aligned} \tag{3.1}$$

where x is the horisontal position of the load, p_x is the position of the trolley on the girder, r is the hoist rope length, θ is the sway angle, and z is the vertical position of the load, as illustrated in Figure 3.1.

The system can be described using the following states, where $\bar{x}$ is the state vector which is different from the position x above,

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} p_x \\ \dot{p}_x \\ r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{pmatrix}, \tag{3.2}$$

and the following state equations, found using Newton's second law of motion. For a full derivation, refer to Appendix A.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{(F - c_x x_2)(J + b^2 m) + m\sin x_5 (J(g\cos x_5 + x_3 x_6^2) + (c_r x_4 + bC))}{J(m_t + m\sin^2 x_5) + b^2 m m_t} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{-(c_r x_4 + bC)(m_t + m\sin^2 x_5) + b^2 m (m_t (x_3 x_6^2 + g\cos x_5) - \sin x_5 (F - c_x x_2))}{J(m_t + m\sin^2 x_5) + b^2 m m_t} \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{2(x_4 + c_{th})x_6 + g\sin x_5}{x_3} \\ &\quad - \frac{\cos x_5}{x_3} \frac{(F - c_x x_2)(J + b^2 m) + m\sin x_5 (J(g\cos x_5 + x_3 x_6^2) + (c_r x_4 + bC))}{J(m_t + m\sin^2 x_5) + b^2 m m_t}. \end{aligned} \tag{3.3}$$

The notation is defined in Table 3.1.

The initial conditions are

$$\bar{x}_0 = \begin{pmatrix} p_{x,0} \\ \dot{p}_{x,0} \\ r_0 \\ \dot{r}_0 \\ \theta_0 \\ \dot{\theta}_0 \end{pmatrix}. \tag{3.4}$$

Since the setpoints are provided as velocities, they need to be converted to linear force, F, and torque, C, respectively. Newton's second law again gives

Table 3.1: *Notation used in the crane model (3.3)*

Symbol	Unit	Explanation
$x_1 = p_x$	[m]	Position of the trolley on the girder
$x_2 = \dot{p}_x$	[m/s]	Velocity of the trolley along the girder
$x_3 = r$	[m]	Hoist rope length
$x_4 = \dot{r}$	[m/s]	Hoisting speed
$x_5 = \theta$	[rad]	Sway angle
$x_6 = \dot{\theta}$	[rad/s]	Angular velocity of sway
F	[N]	Linear force from motor acting on trolley
C	[Nm]	Motor torque acting on hoist wheel
m	[kg]	Mass of load
m_t	[kg]	Mass of trolley
g	[m/s^2]	Acceleration of gravity
J	[kg$\cdot$m^2]	Moment of inertia, hoist wheel
b	[m]	Hoist wheel radius
c_x	[kg/s]	Friction related constant, trolley movement
c_r	[kg$\cdot$m^2/s]	Friction related constant, hoisting
c_{th}	[m/s]	Friction related constant, sway

$$\begin{aligned} F &= m \cdot a \\ C &= r \times F. \end{aligned} \tag{3.5}$$

Introducing the parameters of the model, and using feedback from the actual speeds to calculate the desired accelerations, the following PI-regulators are found

$$\begin{aligned} F &= K_{px}(m + m_t)\frac{r_{px} - x_2}{T_{px}} \\ C &= -K_r bm\frac{r_r - x_4}{T_r} + bmg, \end{aligned} \tag{3.6}$$

where K_{px}, K_r, T_{px} and T_r are the regulator gains and the integral times. The term bmg is inserted in the calculation of C to counteract the force of gravity.

3.2 Parameter estimation

In the model, five parameters are unknown and unmeasurable: the friction related constants (c_x, c_r, and c_{th}), the hoist wheel inertia (J), and the mass of the trolley (m_t). These parameters together with the control parameters (K_{px}, K_r, T_{px}, and T_r) need to be estimated.

Firstly, the model is implemented as a nonlinear grey-box model in MATLAB [8], using the System Identification Toolbox [10] and the IDNLGREY object. This object is created by specifying the model structure; the number of inputs, outputs, and states of the model; and the parameters. The parameters can then be assigned certain properties. The applied properties are given in Table 3.2. Since the gains

K_i are directly multiplied with the integration times T_i, only the ratio $\frac{K_i}{T_i}$ must be optimised. Thus, the integral times T_i are given a fixed value.

Table 3.2: *Parameter properties used in estimation*

Par	Unit	Fixed	Min	Comment
b	[m]	True	-	Measurable
c_r	$[\mathrm{kg \cdot m^2/s}]$	False	0	Friction works against direction of movement
c_{th}	[m/s]	False	0	Friction works against direction of movement
c_x	[kg/s]	False	0	Friction works against direction of movement
g	$[\mathrm{m/s^2}]$	True	-	Known
J	$[\mathrm{kg \cdot m^2}]$	False	0	Moment of inertia always positive
m	[kg]	True	-	Measurable
m_t	[kg]	False	27	Weights of motors known $\approx$ 27 kg
K_{px}	[1]	False	-	
K_r	[1]	False	-	
T_{px}	[s]	True	-	Set to sample time of estimation data
T_r	[s]	True	-	Set to sample time of estimation data

A data set displaying much of the system dynamics was chosen for the parameter estimation. The model input signals, consisting of the remote control provided reference speeds, the measured setpoint speeds (i.e. the output from the controller) and the measured speeds of this test are shown in Figure 3.2. It is found that there is a delay of 3 samples (0.12 s) between the setpoint trolley speed and the speed of the trolley. For the hoist, the delay is measured to be on average 6 samples (0.24 s).

First a comparison is run between measured values and simulation of the greybox model using parameter values that were deemed reasonable using prior knowledge of the system. The result is shown in Figure 3.3. The fit of the trolley speed is very good already (97.47 %), but the hoisting fit needs improvement (66.56 %). The fit is calculated using the normalised root mean square error (NRMSE) between the simulated data and the measured data.

Next, the *parameter optimisation*, or *greybox estimation*, is done using the the trust-region-reflective algorithm, which is a method for numerical solution of complex non-linear problems. The algorithm is used to solve the nonlinear least squares problem defined as

$$\min_x \sum_i f_i^2(x) = \min_x \|F(x)\|_2^2, \tag{3.7}$$

where x is the vector of parameters and the i-th component of $F(x)$ is $f_i(x)$, which is the difference between the measured value and the value calculated using the model at sample i [9].

The algorithm is run until the first order optimality is 0, which means that a minimum is reached. At this possibly local minimum, the fit is increased to [97.42; 80.4]%. Akaike's final prediction error (FPE) is also given. This is an

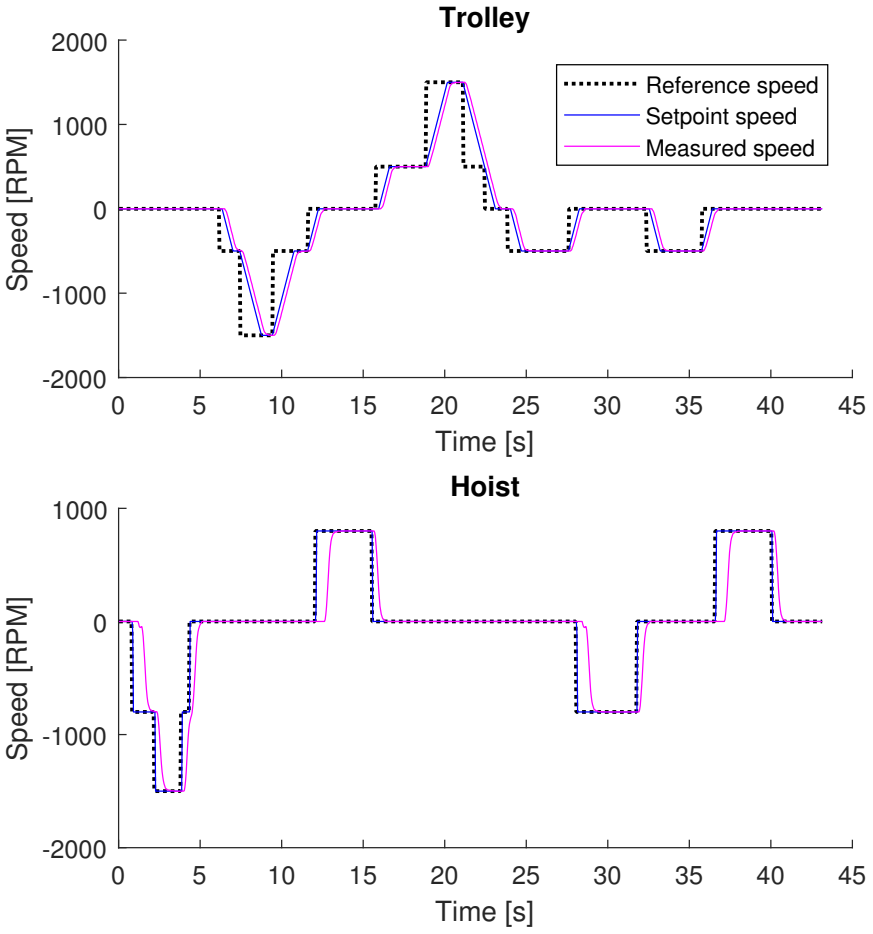

Figure 3.2: *Measured input reference speeds, setpoint speeds and actual speeds of the estimation data set, collected from the test crane.*

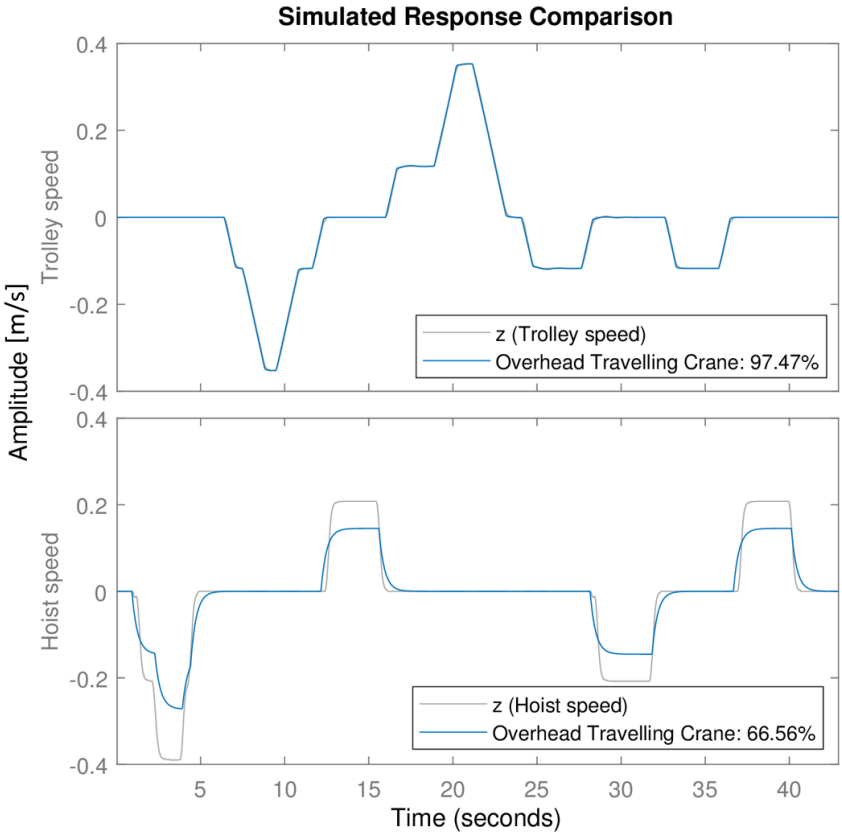

Figure 3.3: *Comparison between measured speeds and simulated speeds without optimised parameters, using the estimation data. The NRMSE fit of the trolley speed is 97.47%, and the NRMSE fit of the hoisting speed is 66.56 %.*

estimate of the prediction error variance that would be expected from using the model with a new data set [13, p.365]. At the minimum the FPE is 6.28242e-09.

The estimated values for the parameters can be found in Table 3.3. The friction related constants all have reasonable values. The hoist wheel inertia J is higher than expected. However, this can be explained by there being some other damping constant related to the hoisting acceleration or angular acceleration of the hoist wheel. The mass of the trolley matches initial guesses. The trolley control gain K_{px} can be set to 1 without any loss of fit, also the hoisting gain is acceptable.

Table 3.3: *Parameter values estimated using grey-box identification.*

Parameter	Value	Unit
c_r	0.3333	$[\mathrm{kg \cdot m^2/s}]$
c_{th}	0.3894	[m/s]
c_x	0.5223	kg/s]
J	1.7089	$\mathrm{kg \cdot m^2}$
m_t	40.0079	[kg]
K_{px}	1.0003	[1]
K_r	5.4990	[1]

The plot in Figure 3.4 shows the same comparison as in Figure 3.3 but with the optimised parameters. It can be noted that the fit of the trolley speed decreases by 0.05 percentage points, but this is insignificant compared to the large increase in fit of the hoisting speed.

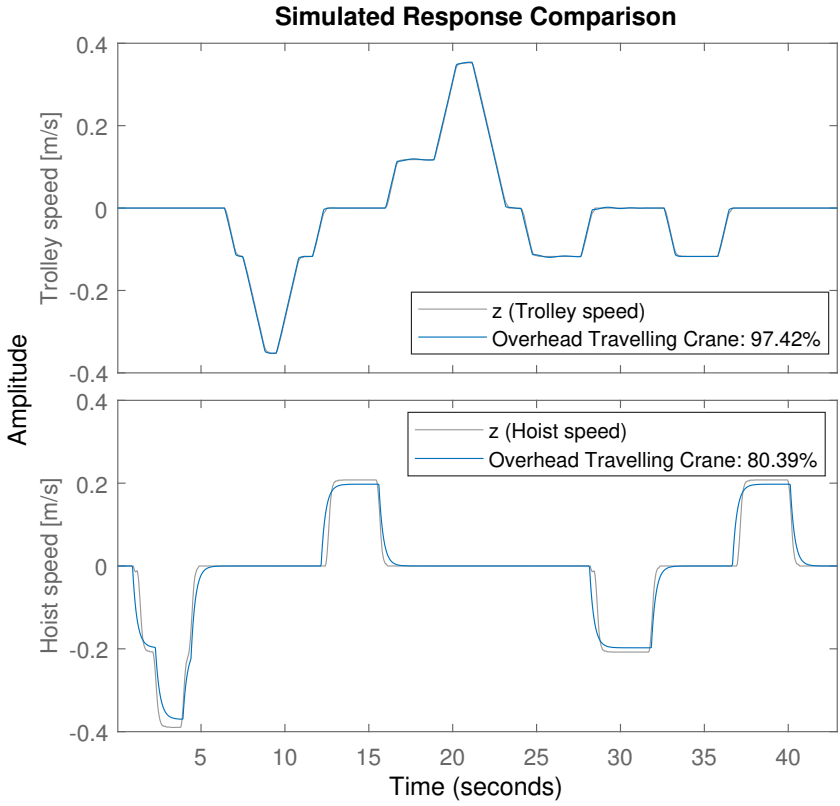

Figure 3.4: *Comparison between measured speeds and simulated speeds for the estimation data, using the optimised parameters found through grey-box estimation. The NRMSE fit of the trolley speed has decreased by 0.05 percentage points to 97.42%, and the fit of the hoisting speed has increased with 13.83 percentage points to 80.36%.*

3.3 Validation

To verify that the model describes the behaviour of the crane in general, and not just the estimation data, validation is performed on another data set. The input reference speeds, setpoint speeds and measured speeds of the validation data are shown in Figure 3.5.

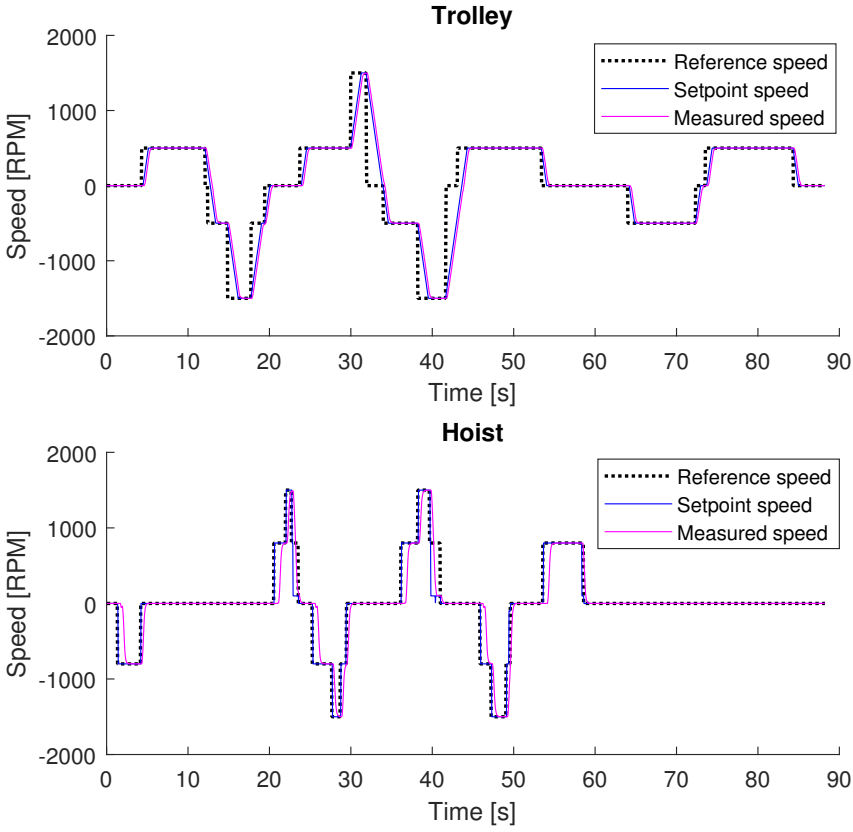

Figure 3.5: *Inputs, setpoint speeds and measured speed for the validation data set collected using the test crane.*

Fit to measured data

First, the simulated response is compared to the measured response, to find the NRMSE fit of the model to new data. The result is found in Figure 3.6. As can be seen, the model simulates the data well, with a fit that slightly exceeds the one of the estimation data.

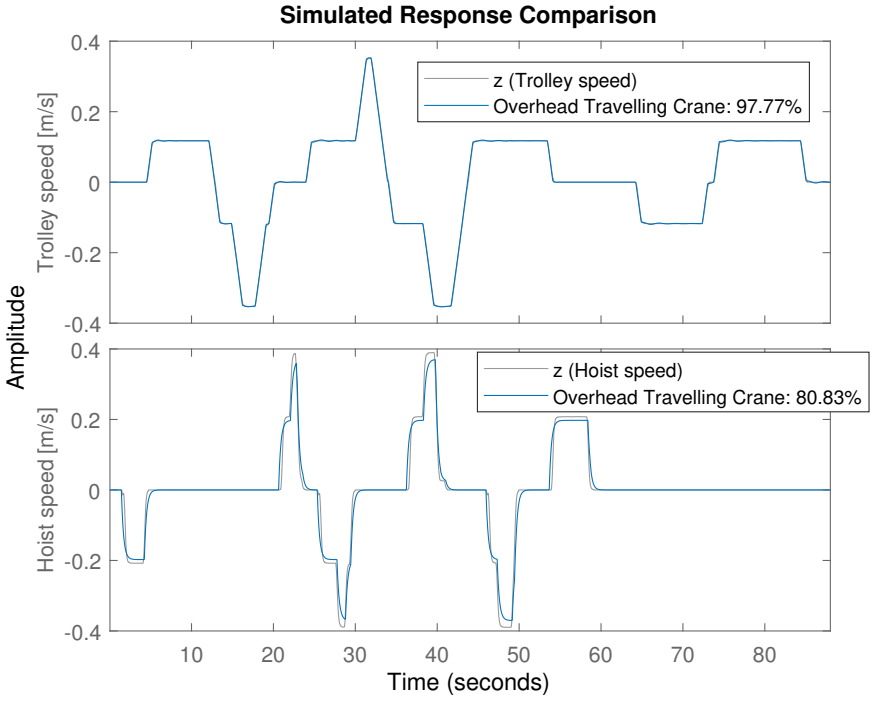

Figure 3.6: *Comparison between measured and simulated speeds using the validation data set and parameters optimised using greybox estimation. The* NRMSE *fit of both the trolley speed and the hoisting speed are still good.*

Residual analysis

Another validation method is analysis of the residuals. The residuals, also known as prediction errors, are the differences between the measured values and the simulated results. The residuals of the validation test are plotted in Figure 3.7. Having residuals resembling a white noise process indicates that the errors are caused by noise, and not a modelling error. As can be seen in the figure, the residuals do not have the shape of white noise.

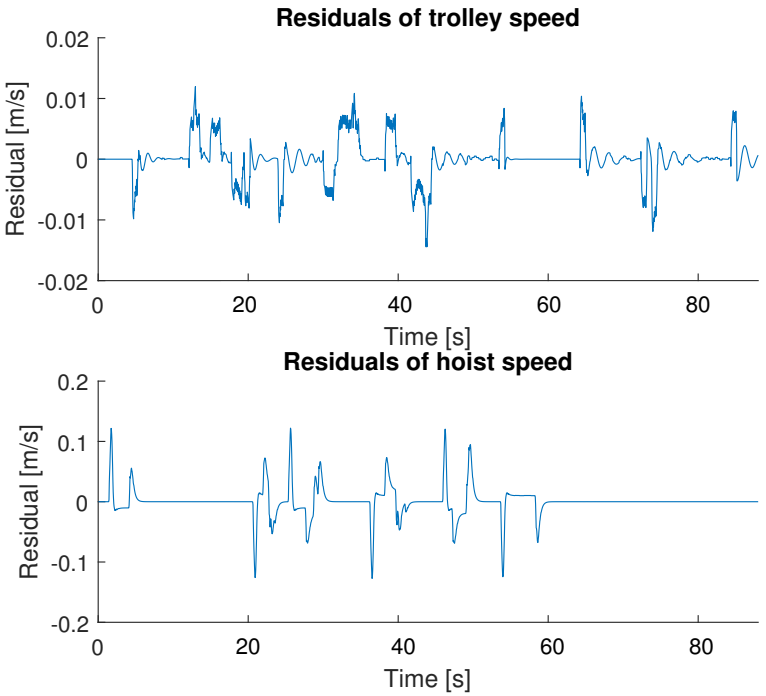

Figure 3.7: *Residuals of the validation data, showing the difference between the measured speeds and the speeds estimated using the model.*

To analyse the residuals further, a correlation analysis can be performed. Correlation is a measure of the association between two variable quantities. It is often expressed as a correlation coefficient between -1 and 1. Values close to 0 mean that the variables are uncorrelated, values close to 1 indicate that the variables vary in the same way, and values close to -1 imply that when one increases the other decreases and vice versa.

The plots in the first column in Figure 3.8 show the autocorrelation of the outputs. That is the correlation between the residuals sampled at different times. The x-axis shows the lag which is the number of samples between the indicated residuals. According to the definition, a white noise process has uncorrelated samples, making the autocorrelation plot a delta function. The plots in the figure

are clearly indicating that the residuals are not white noise.

The middle and right columns illustrate the correlation between the outputs and the inputs. A high correlation between input and output reveals that there might be system dynamics that are not modeled [13, p. 367]. The shaded area shows the 99% confidence interval marking statistically insignificant correlations. Almost all the values are within these regions. The larger correlation for negative lags between the trolley speed and input 1, which is the reference trolley speed, and the hoisting speed; and input 2, the reference hoisting speed, can be a sign of the feedback present in the system during measurements.

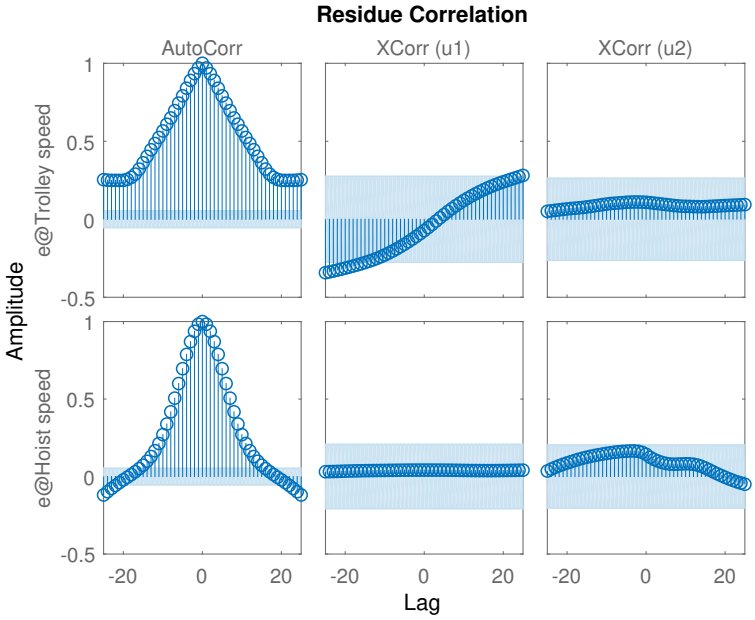

Figure 3.8: *The left column shows the autocorrelation between the residuals of the trolley speed (top) and the hoist speed (bottom). The plots indicate that the residuals are not white noise caused by measurement errors. The middle column contains the correlation between the output speeds and the trolley input, and the right column presents the correlation between the outputs and the hoist input. The middle and right columns' values are mostly within the 99% confidence region for non-significant residuals, but could be a sign of the feedback within the system or indicate the presence of unmodelled aspects of the system.*

Comparison to estimated polynomial model

A third way to validate a model is to compare it to another model. Since a physical model is always subject to some level of abstraction, it is interesting to compare the result with a black-box model.

The type of model chosen for this comparison is the non-linear auto-regressive model with exogenous output (NARX). This type of model predicts values of a time series based on previous values of the same series, as well as current and previous values of some other driving time series[13, p.318]. In our case the predicted time series contain the actual speeds, and the driving time series are the reference speeds:

$$\hat{y} = g(y(t-1), \ldots, y(t-n), u(t), \ldots, u(t-m)). \tag{3.8}$$

Models of different orders, i.e. using different numbers of earlier samples, are adapted to the estimation data. Their responses to validation data are then compared to that of the physical model. Figure 3.9 shows the best model found. It uses the most recent samples of both outputs and the current inputs for estimating the trolley speed, and using only the latest estimate of the hoisting speed together with the current and most recent reference speeds for hoisting to estimate the hoisting speed:

$$\begin{aligned} \hat{y_1} &= g_1(y_1(t-1), y_2(t-1), u_1(t-1), u_2(t-1)) \\ \hat{y_2} &= g_2(y_2(t-1), u_2(t-1), u_2(t-2)) \end{aligned} \tag{3.9}$$

The trolley movement is better described by this NARX model than by the physical model, and the hoisting movement is as good. The interesting thing is that the fit of the NARX model is not significantly better than the physical model, and does not get better with more information either.

Conclusion

The residual analysis shows that the model is not perfect. There are some aspects not covered, leading to higher correlation of the residuals shown in Figure 3.8. This can also be seen for the trolley speed in the comparison plot in Figure 3.6, where it is clear that the speeds are not the same. However, the purpose of the model is not to be an exact representation of the test crane. The goal is to have a model that behaves as an overhead travelling crane, so that it can be used to derive general fault detection algorithms. It will also be used as the true system later on.

The NRMSE fit is quite good, especially for the trolley speed. For the hoisting speed, the general shape of the curves are the same, meaning the model does not introduce any new dynamics and it does not miss any significant dynamics. The physical model is also as good at describing the hoisting dynamics as a general nonlinear model fitted to the data. In conclusion the model is good enough to act as the true system for the fault detection purposes.

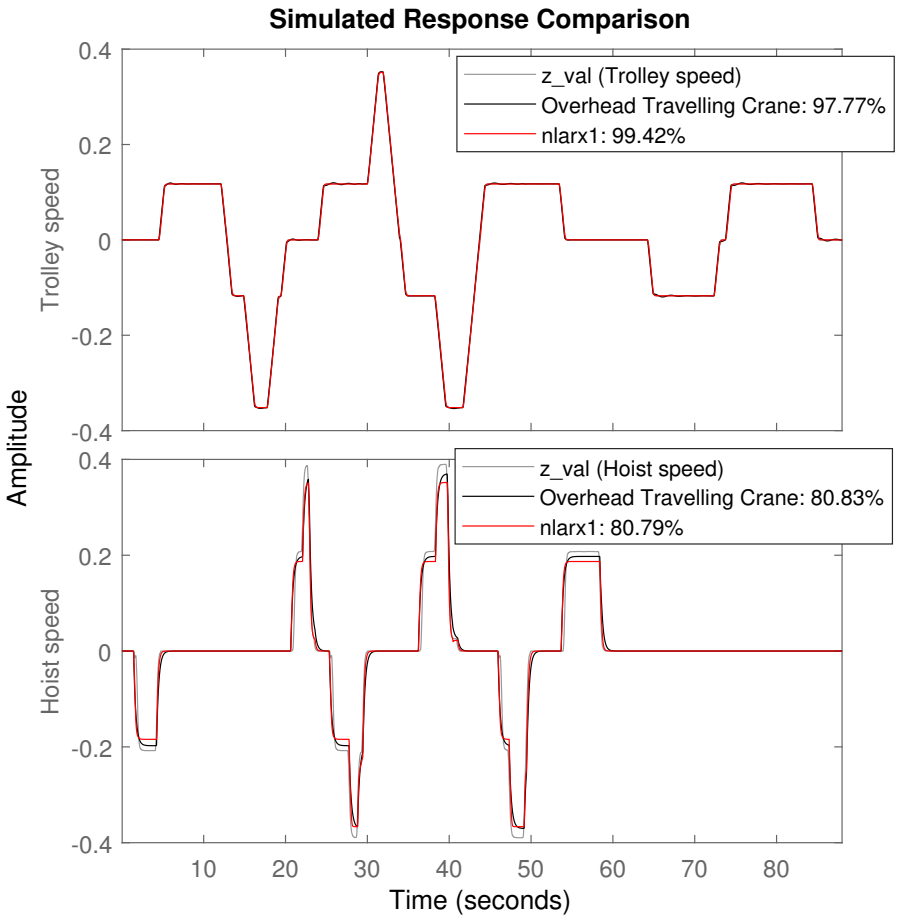

Figure 3.9: *Comparison between measured validation data, simulated response using the physical model, and the simulated response using a NARX model. The NARX produces a better fit for the trolley speed, but the physical model is better at describing the hoisting dynamics.*

4

Fault detection

The main goal of the fault detection part of this thesis is to investigate whether it is possible to detect faults within the system, whilst only monitoring the inputs and outputs. The second goal is to see whether one can distinguish between dangerous and non-dangerous faults.

There are different definitions of the terms used within fault detection. In this thesis, the definitions in Table 4.1, made by the International Federation of Automatic Control (IFAC) [3, p.559], will be used.

Table 4.1: *Definitions of fault detection terms, compiled by IFAC.*

Term	**Definition**
Error	Deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.
Failure	Permanent disruption of a system's ability to perform a required function under specified operating conditions.
Fault	Unpermitted deviation of at least one characteristic property or parameter of a system from its acceptable, usual or standard condition. A fault is the occurrence of a failure mode.
Fault detection	Determination of faults present in a system and time of detection.
Fault modelling	Determination of a mathematical model to describe a specific fault effect.
Residual	Fault information carrying signals, based on deviation between measurements and model based computations.

The techniques described in this chapter can all be classified as model-based change detection. When used in reality, a model of the system is used to predict the behaviour of the system. The model outputs are compared to measurements of the real system. If the model is good enough, the model outputs and the measurements will be very similar as long as the true system behaves normally. A fault in the system leads to a change in the measurements. Since the model observing the system maintains the normal behaviour, one can detect a change in the system by monitoring the difference between the model outputs and the measurements. As previously mentioned, a simulation model will be used instead of the true system in this project.

There are many ways of implementing a fault detection system, but most of them share the same general structure, which is shown in Figure 4.1.

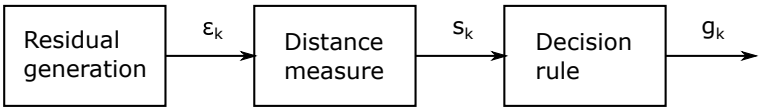

Figure 4.1: *General structure of a fault detection system, showing the different steps of fault detection, and the notation of the signals connecting the steps.*

In the first step, the residual generation, a measured signal from the system is compared with an estimate of the same signal. The difference between the two values becomes the residual, ε_k. Since the model is rarely perfect, and there are measurement errors etc, the residuals are often very noisy. To make the residuals more understandable, a distance measure is applied. The *distance* between the current situation and the no-change situation is measured, which results in a symptom, called s_k in Figure 4.1. Finally a decision rule produces an alarm if the symptom is severe enough.

4.1 Residual generation

In model-based fault detection, the residual is generated by comparing a measured signal, with that created by some sort of model or observer, as in Figure 4.2. There u_k is the input signal at time k, y_k is the signal containing the measurements from the true system, $\hat{y}_k$ contains the estimated outputs, and ε_k is the resulting residuals.

There is no general method of creating nonlinear observers, every situation is different. Optimal nonlinear observers contain partial differential equations to solve for each iteration, making them computationally heavy. Some simplification or linearisation is necessary. Linear observers do however often work sufficiently well for fault detection, around certain stationary points. In this case,

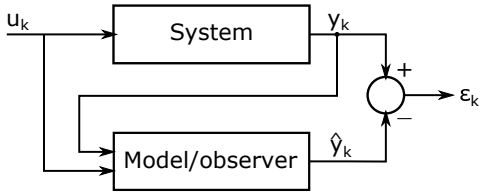

Figure 4.2: *Residual generation in model-based fault detection, where an observer is used to create estimates of the output signals from the system.*

there are stationary points defined by the load hanging straight down:

$$\begin{aligned} x_1 &= p_x = p_x \\ x_2 &= \dot{p}_x = 0 \\ x_3 &= r = r \\ x_4 &= \dot{r} = 0 \\ x_5 &= \theta = 2n\pi \quad n \in \mathbb{Z} \\ x_6 &= \dot{\theta} = 0 \end{aligned} \tag{4.1}$$

for any values of p_x and r. These points are good candidates for a linearisation of the system, which is the first step in creation of a linear observer.

The linearisation of (3.3) is found to be

$$\begin{aligned} \dot{x} &= \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{c_x}{m_t} & 0 & 0 & \frac{mg}{m_t} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{c_r}{J+b^2m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{c_x}{rm_t} & 0 & 0 & -\frac{g}{r} - \frac{mg}{rm_t} & -\frac{2c_{th}}{r} \end{pmatrix}}_{A_1} x + \underbrace{\begin{pmatrix} 0 & 0 \\ \frac{1}{m_t} & 0 \\ 0 & 0 \\ 0 & -\frac{b}{J+b^2m} \\ 0 & 0 \\ \frac{1}{rm_t} & 0 \end{pmatrix}}_{B_1} \begin{pmatrix} F \\ C \end{pmatrix} \\ y &= \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{C_1} x \end{aligned} \tag{4.2}$$

where r is chosen to be at the middle of its range, so that both hoisting and lowering is possible. Adding the treatment of the reference signals, the state matrix

becomes

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{c_x}{m_t} - \frac{m+m_t}{T_s m_t} & 0 & 0 & \frac{mg}{m_t} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{c_r}{J+b^2m} - \frac{K_r b^2 m}{T_s(J+b^2m)} & 0 & 0 & -\frac{b}{J+b^2m} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{c_x}{r m_t} - \frac{m+m_t}{r T_s m_t} & 0 & 0 & -\frac{g}{r} - \frac{mg}{r m_t} & -\frac{2c_{th}}{r} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.3}$$

where the new state represents the constant bmg which is added to the torque C to counteract the force of gravity.

The input matrix turns into

$$B_2 = \begin{pmatrix} 0 & 0 \\ \frac{m+m_t}{T_s m_t} & 0 \\ 0 & 0 \\ 0 & \frac{K_r b^2 m}{T_s(J+b^2m)} \\ 0 & 0 \\ \frac{m+m_t}{r T_s m_t} & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.4}$$

and the measurement matrix is extended to

$$C_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{4.5}$$

The linearised model is compared with the non-linear model in Figure 4.3, using validation data. It appears that the linear model works quite well for describing the system.

Luenberger observer

The general idea of a *Luenberger observer*, or *state observer* is to use the measured states to update the observer exactly, and estimate the rest.

Consider the continuous state space system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t). \end{aligned} \tag{4.6}$$

The state vector $x(t)$ can be estimated by the measured inputs and outputs as

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad \hat{x}(0) = \hat{x}_0 \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{4.7}$$

where L is the observer gain [11, p.15].

tee

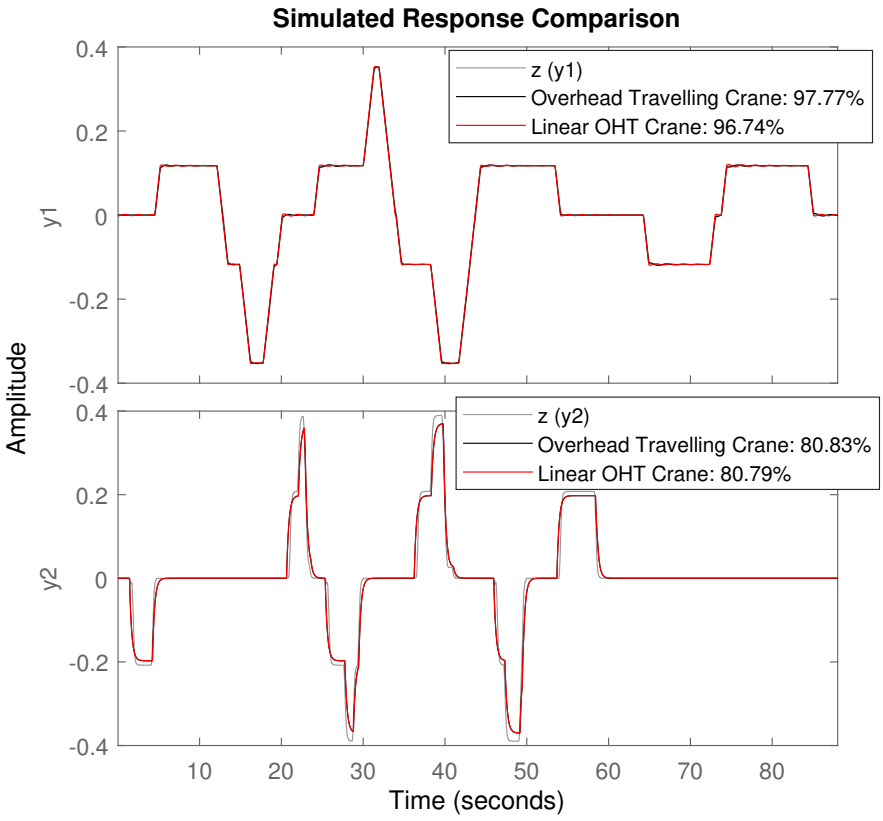

Figure 4.3: *NRMSE fit of the nonlinear and linear models to the measured validation data. The fit of the linear model is slightly lower than that of the non-linear model.*

The observer error $e = x - \hat{x}$, the error between the true and the estimated state, satisfies

$$\dot{e}(t) = (A - LC)e(t). \tag{4.8}$$

The error will tend to zero when the eigenvalues of $(A - LC)$ are placed in the open left half plane. When the system is observable, the eigenvalues can be placed arbitrarily by selection of L. The choice of poles affect the speed with which the error tends to zero. Placing the poles far into the left plane gives an observer that adapts quickly. However, a large L makes the observer sensitive to initial estimation errors and measurement errors since the difference $y - \hat{y}$ will be multiplied with a large gain.

For the situation in question, A, B and C are given as in (4.3)-(4.5).This choice of $[A, C]$ is not observable, since the first state, x_1, is not measurable and does not affect the output. Or, according to the definition: the observability matrix $\mathcal{O}$, defined as in (4.9), does not have full rank [12, p.45].

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix} \tag{4.9}$$

However, since the state x_1 does not affect the output, is not measurable, and is not measured in the mock-up crane, it is not something that has to be part of the observer. Removing state x_1 leads to the following reeduced linearised model

$$\begin{aligned} \dot{x} &= A_3 x_r + B_3 u \\ y &= C_3 x_r \end{aligned} \tag{4.10}$$

where

$$A_3 = \begin{pmatrix} -\frac{c_x}{m_t} - \frac{m+m_t}{T_s m_t} & 0 & 0 & \frac{mg}{m_t} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{c_r}{J+b^2 m} - \frac{K_r b^2 m}{T_s(J+b^2 m)} & 0 & 0 & -\frac{b}{J+b^2 m} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{c_x}{r m_t} - \frac{m+m_t}{r T_s m_t} & 0 & 0 & -\frac{g}{r} - \frac{mg}{r m_t} & -\frac{2c_{th}}{r} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B_3 = \begin{pmatrix} \frac{m+m_t}{T_s m_t} & 0 \\ 0 & 0 \\ 0 & \frac{K_r b^2 m}{T_s(J+b^2 m)} \\ 0 & 0 \\ \frac{m+m_t}{r T_s m_t} & 0 \\ 0 & 0 \end{pmatrix} \qquad C_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$
$$x_r = \begin{pmatrix} \dot{p}_x \\ r \\ \dot{r} \\ \theta \\ \dot{\theta} \\ bmg \end{pmatrix} \qquad u = \begin{pmatrix} u_{px} \\ u_r \end{pmatrix}. \tag{4.11}$$

This results in the following Luenberger observer:

$$\begin{aligned} \dot{\hat{x}}_r(t) &= A_3 \hat{x}_r(t) + B_3 u(t) + L(y_r(t) - \hat{y}_r(t)) \\ \hat{y}_r(t) &= C_3 \hat{x}_r(t). \end{aligned} \tag{4.12}$$

The oobserverer gain L can then be chosen by placing the poles of the error dynamics so that the observer gets the desired properties. Specifically if they are placed in the open left half plane, the error will tend towards zero. In this case L is chosen to be fast enough that residuals die out fast, but also slow enough that faults will still be visible. Figure 4.4 shows residuals and signals for a test where the input to the trolley makes a step at 10s, and the input to the hoist makes a step at 20s. No measurement noise is added to the simulation. Since the observer is based on a linearisation, the slight oscillation in the trolley speed, caused by the swinging of the load, is exaggerated. The residuals do however converge to zero.

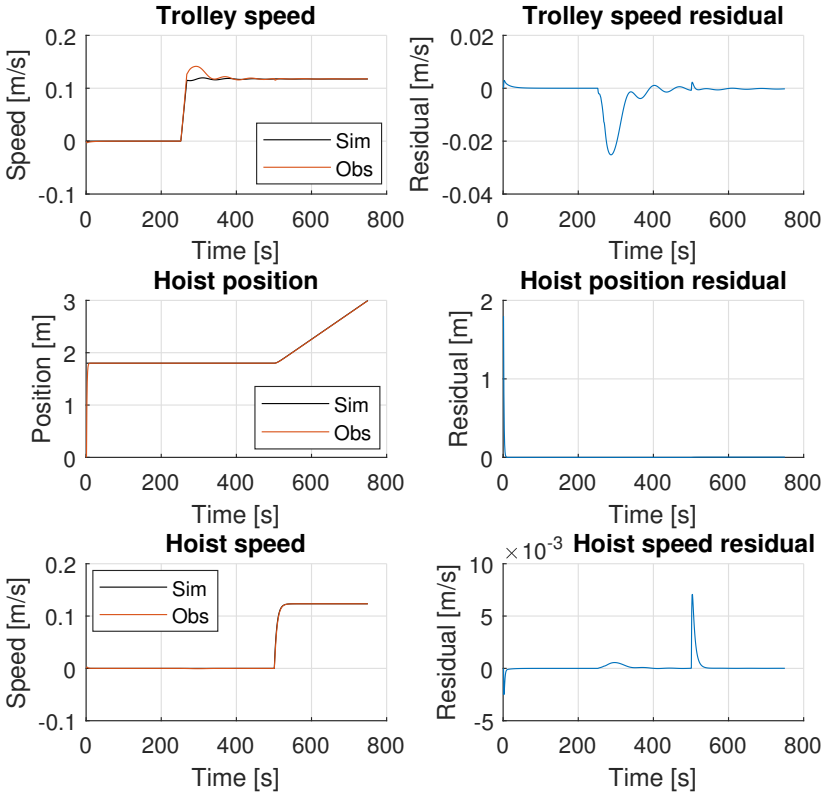

Figure 4.4: *Comparison between simulated output signals and the observer outputs, as well as the residuals, for a simulation where the trolley input makes a step to normal speed at $t = 10$ s, and the hoisting input makes a step to normal speed at $t = 20$ s.*

4.2 Distance measure

To make the residuals more understandable, to be able to apply a stopping rule, a distance measure is used. In this section three different distance measures are presented.

Least squares

Least squares (LS) is a method that has many applications in optimisation and data analysis. For this fault detection application, the idea is to calculate the mean of all previous residuals, and remove this mean from the current residual. A residual that is significantly larger than the others will thus stand out more clearly in the symptom signal s.

The mean of all residuals up until the the time $t = k$, $\hat{\theta}_k$, is found as

$$\hat{\theta}_k = \frac{1}{k} \sum_{i=1}^{k} \varepsilon_i \tag{4.13}$$

and the symptom can be created in two ways

$$s_k = (\varepsilon_k - \hat{\theta}_k)^2 \quad \text{or} \quad s_k = \varepsilon_k - \hat{\theta}_k. \tag{4.14}$$

Technically only the first one is true LS. However, squaring the residuals removes information regarding the sign of the residual, which can be of interest from a safety point of view. The second distance measure, where the mean is simply removed from the residuals, will also be applied.

Recursive least squares

The LS approach assumes that the residuals are time-invariant. To handle the case of time-variant residuals, the recursive least squares (RLS) method can be used instead. This technique uses a *forgetting factor*, λ, that gives more recent measurements a larger effect than older ones.

$$\hat{\theta}_k = \frac{1 - \lambda}{1 - \lambda^k} \sum_{i=1}^{k} \lambda^{k-i} \varepsilon_i. \tag{4.15}$$

Once again, both the squared and non-squared symptoms can be used.

Log-likelihood ratio

Assuming that the residuals are independent and following some probability density function (PDF), $p_\theta(\varepsilon)$, where θ is the scalar parameter of which change is to be detected. Initially $\theta = \theta_0$. After the unknown change time t, θ becomes $\theta_1 \neq \theta_0$.

Assuming next that θ_0 is known, the logarithm of the ratio of probability densities can be used as distance measure:

$$s(\varepsilon) = \ln \frac{p_{\theta_1}(\varepsilon)}{p_{\theta_0}(\varepsilon)}. \tag{4.16}$$

The important characteristic of this log-likelihood ratio is that the expected value of s will be smaller than 0 before the change, i.e. when θ_0 is true, and larger than 0 after the change, when θ_1 is true. Using $\mathbb{E}_{\theta_0}$ and $\mathbb{E}_{\theta_1}$ to describe the expectations of the random variables subject to the distributions $p_{\theta_0}(y)$ and $p_{\theta_1}(\varepsilon)$, respectively, then

$$\mathbb{E}_{\theta_0}\{s\} < 0 \quad \text{and} \quad \mathbb{E}_{\theta_1}\{s\} > 0. \tag{4.17}$$

That is, a change in the variable θ leads to a change in sign of the log-likelihood ratio [2].

Using the common assumption that the residuals are a Gaussian process, the PDF is

$$p_\theta(\varepsilon) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{-(\varepsilon - \mu)^2}{2\sigma^2}, \tag{4.18}$$

where μ is the mean of the white noise process, and σ is the standard deviation of the process.

A change in the distribution can either be a change in mean or change in variance. For a change in mean from μ_0 to μ_1, with a constant variance σ^2, the log-likelihood ratio becomes

$$s(\varepsilon) = \ln \frac{p_{\theta_1}(\varepsilon)}{p_{\theta_0}(\varepsilon)} = \ln \left(\frac{\frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(\varepsilon-\mu_1)^2}{2\sigma^2}}{\frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(\varepsilon-\mu_0)^2}{2\sigma^2}} \right) = \frac{\mu_1 - \mu_0}{\sigma^2} \left(\varepsilon - \frac{\mu_0 + \mu_1}{2} \right) \tag{4.19}$$

which is called the sufficient statistic for a change in the mean. If σ^2, μ_0, and μ_1 are known, $s = \varepsilon$ can be used as symptom.

Introducing the magnitude of the change

$$\nu = \mu_1 - \mu_0 \tag{4.20}$$

and the signal-to-noise ratio

$$b = \frac{\mu_1 - \mu_0}{\sigma} \tag{4.21}$$

(4.19) becomes

$$s = \frac{b}{\sigma} \left(\varepsilon - \mu_0 - \frac{\nu}{2} \right). \tag{4.22}$$

Similarly, for an increase in variance from σ_0 to $\sigma_1 > \sigma_0$, with the mean constant μ, the sufficient statistic is

$$s(\varepsilon) = \ln \frac{p_{\theta_1}(\varepsilon)}{p_{\theta_0}(\varepsilon)} = \ln \left(\frac{\frac{1}{\sigma_1\sqrt{2\pi}} \exp - \frac{\varepsilon^2}{2\sigma_1^2}}{\frac{1}{\sigma_0\sqrt{2\pi}} \exp - \frac{\varepsilon^2}{2\sigma_0^2}} \right) = \ln \frac{\sigma_0}{\sigma_1} + \left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2} \right) \frac{\varepsilon^2}{2}. \tag{4.23}$$

If σ_0 and σ_1 are known, then ε^2 can be used as a symptom.

In reality, while μ_0 and σ_0 can be measured, one usually does not know the parameters μ_1 and σ_1. There are three ways of handling the choice of ν, one can set ν as the minimum possible magnitude of jump, as the most likely magnitude of jump, or as a worst case magnitude, based on the cost of not detecting the jump. In this thesis ν and σ_1 are chosen to be minimal values for detection.

4.3 Decision rule

The decision rule is the last part of the fault detection system, and the part that raises the alarm. Samples of the symptom signal are taken with given intervals. For each sample, a hypothesis test is made for the parameter θ symbolising the change in the system:

$$\begin{cases} \mathbf{H_0}: & \theta = \theta_0 \\ \mathbf{H_1}: & \theta = \theta_1 \end{cases} \tag{4.24}$$

where $\mathbf{H_0}$ and $\mathbf{H_1}$ indicate the different hypotheses.

If the second hypothesis is chosen, meaning the parameter has changed value from θ_0 to θ_1, an alarm is raised. The hypothesis test consists of a decision function, and a stopping rule. The decision function d treats the *symptom* or *sufficient statistic* (s_k), usually resulting in a signal that deviates from the accepted interval only when a fault is present in the system. The fault detection alarm is raised if the result of the decision function reaches above or below given thresholds. When choosing the threshold, the trade-off lies between detection time and false alarm. Putting the accepted levels of the decision function too high leads to a large time from fault to detection, or even to missed alarms, which is not good from a safety perspective. On the other hand, putting the thresholds too low can lead to false alarms, which is bad from a productivity perspective, since the system is unnecessarily stopped.

Direct thresholding

The first method, direct thresholding, has the simplest decision function

$$d_k = s_k. \tag{4.25}$$

A threshold, h, is given, and if the symptom reaches above this level an alarm, g, is raised.

$$g_k = 1 \quad \text{if} \quad s_k \geq h \tag{4.26}$$

The threshold can be fixed, or adapted to the size of the input. Since the residuals can be noisy, a counter is often added to avoid false alarms. This counter delays the alarm until a fixed number of samples are above the threshold. To handle deviations in both positive and negative directions, either a squared symptom signal or the absolute values of the symptom can be used, or a separate negative threshold can be set. In this thesis the threshold is fixed, and the counter is set to one sample.

Recursive CUSUM

The cumulative sum (CUSUM) is based on the characteristic property of the log-likelihood ratio. It is negative before a change, and positive after a change, which means that the cumulative sum of the sufficient statistic s

$$d_k = S_k = \sum_{i=1}^{k} s_i \quad \Longleftrightarrow \quad S_k = S_{k-1} + s_k \tag{4.27}$$

has a negative drift before a change, and positive after a change has occurred [2]. The typical behaviour is shown in Figure 4.5.

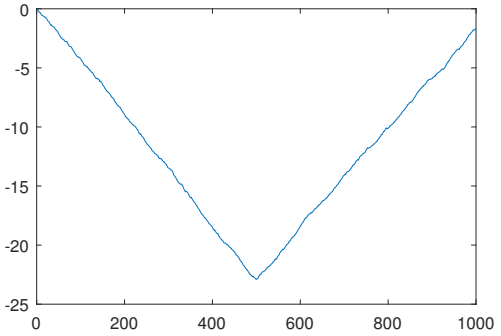

Figure 4.5: *Typical behaviour of an CUSUM decision function for a white noise residual where a change in mean has occured at* $t = 500$.

The stopping rule for CUSUM becomes

$$g_k = 1 \quad \text{if} \quad S_k - m_k \geq h \tag{4.28}$$

where m_k is the current lowest value of the cumulative sum S_k

$$m_k = \min_{1 \leq j \leq k} S_j. \tag{4.29}$$

The relative threshold can be avoided by using the recursive CUSUM

$$\begin{aligned} S_k &= \max\{0, S_{k-1} + s_k - n\}, \quad S_0 = 0 \\ g_k &= 1 \quad \text{if} \quad S_k \geq h \end{aligned} \tag{4.30}$$

where n is a term indicating the allowed drift. The typical behaviour of the recursive CUSUM is shown in Figure 4.6.

For a change in mean from μ_0 to μ_1 the decision function becomes

$$\begin{aligned} S_k &= \max\left\{0, S_{k-1} + \frac{\mu_1 - \mu_0}{\sigma^2}\left(\varepsilon - \frac{\mu_0 + \mu_1}{2}\right) - n\right\} \\ g_k &= 1 \quad \text{if} \quad S_k \geq h \end{aligned} \tag{4.31}$$

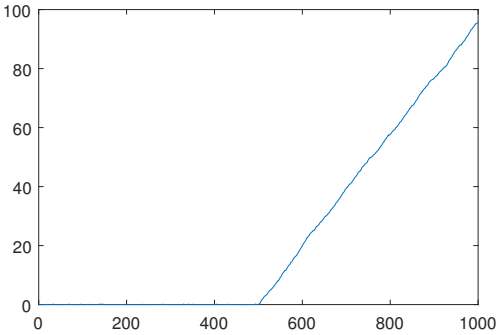

Figure 4.6: *Typical behaviour of a recursive CUSUM decision function for a white noise residual where a change in mean hass occured at* $t = 500$

which can be written as

$$\begin{aligned} \tilde{S}_k &= \max\{0, \tilde{S}_{k-1} + \varepsilon - \mu_0 - n\}, \quad \tilde{S}_0 = 0 \\ g_k &= 1 \quad \text{if} \quad \tilde{S}_k \geq h \end{aligned} \tag{4.32}$$

for an appropriate change of h. The sufficient statistic s_k can be replaced with $\varepsilon - \mu_0$, which is the same as the results of the non-squared versions of the distance measures LS and RLS.

The same rearrangement for a change in variance yields

$$\begin{aligned} \tilde{S}_k &= \max\{0, \tilde{S}_{k-1} + (\varepsilon - \mu)^2 - n\}, \quad \tilde{S}_0 = 0 \\ g_k &= 1 \quad \text{if} \quad \tilde{S}_k \geq h \end{aligned} \tag{4.33}$$

meaning that the squared versions of the LS and RLS can be used.

The two-sided CUSUM can be used to allow detection of change in both positive and negative directions, for example a change in mean from μ_0 to $\mu_1^+ = \mu_0 + \nu$ or to $\mu_1^- = \mu_0 - \nu$, where ν is the magnitude of the change. The two-sided CUSUM is achieved by performing two CUSUM tests simultaneously, one with negative symptoms, and one with positive, according to

$$\begin{aligned} S_k^+ &= \max\{0, S_{k-1} + s_k - n\}, \quad S_0^+ = 0 \\ S_k^- &= \max\{0, S_{k-1} - s_k - n\} \quad S_1^- = 0 \\ g_k &= 1 \quad \text{if} \quad S_k^+ \geq h \quad \text{or} \quad S_k^- \leq h. \end{aligned} \tag{4.34}$$

Geometric moving average

The negative drift of the cumulative sum before a change, can be handled by giving older samples less impact than newer ones. Using geometric moving average

(GMA) weights, λ, the CUSUM decision becomes

$$\begin{aligned} S_k &= \sum_{i=1}^{k} \lambda_i s_{k-i} \\ g_k &= 1 \quad \text{if} \quad S_k \geq h \end{aligned} \tag{4.35}$$

The weights are exponential,

$$\lambda_i = \alpha(1-\alpha)^i, \quad 0 < \alpha < 1, \tag{4.36}$$

making α act as a forgetting factor. (4.35) can be written recursively as

$$\begin{aligned} S_k &= (1-\alpha)S_{k-1} + \alpha s_k, \quad S_0 = 0 \\ g_k &= S_k \geq h. \end{aligned} \tag{4.37}$$

For a forgetting factor $\alpha = 0.4$, and the same white noise residuals as in Figure 4.5 and Figure 4.6, GMA provides the decision function shown in Figure 4.7.

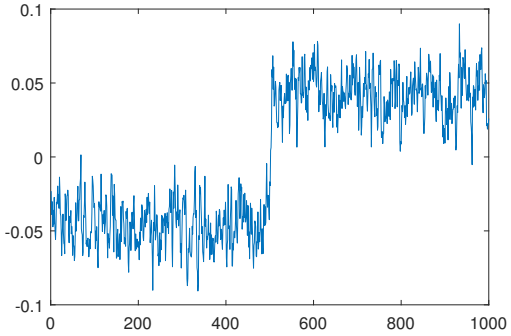

Figure 4.7: *Typical behaviour of a GMA decision function where a change in mean has occured at $t = 500$. The decision function has smoothed the residuals, and shifted them downwards.*

GMA is as CUSUM based on the behaviour of the log-likelihood ratio, and as for CUSUM it can still be used with the other distance measures. For a change in mean, (4.37) becomes

$$\begin{aligned} S_k &= (1-\alpha)S_{k-1} + \alpha s_k \\ &= (1-\alpha)S_{k-1} + \alpha \frac{\mu_1 - \mu_0}{\sigma^2}\left(\varepsilon - \frac{\mu_0 + \mu_1}{2}\right) \\ \implies \frac{\sigma^2}{\mu_1 - \mu_0} S_k &= \frac{\sigma^2}{\mu_1 - \mu_0}(1-\alpha)S_{k-1} + \alpha(\varepsilon - \mu_0) - \alpha\frac{\mu_1 - \mu_0}{2}. \end{aligned} \tag{4.38}$$

This gives a new GMA algorithm

$$\tilde{S}_k = (1-\alpha)\tilde{S}_{k-1} + \alpha(\varepsilon - \mu_0), \quad \tilde{S}_0 = 0 \tag{4.39}$$

where the decision functions are related as

$$\tilde{S}_k = \frac{\sigma^2}{\mu_1 - \mu_0} S_k - \frac{\mu_1 - \mu_0}{2}. \tag{4.40}$$

The result is that the residuals themselves, with the initial mean removed, can be used instead of the log-likelihood ratio, as long as the threshold is changed according to

$$\begin{aligned} g_k = 1 \quad \text{if} \quad S_k \geq h \\ = \tilde{S}_k \geq \frac{\sigma^2}{\mu_1 - \mu_0} h - \frac{\mu_1 - \mu_0}{2}. \end{aligned} \tag{4.41}$$

Similarily, for a change in variance the decision function becomes

$$\tilde{S}_k = (1 - \alpha)\tilde{S}_{k-1} + \alpha(\varepsilon - \mu)^2, \quad \tilde{S}_0 = 0 \tag{4.42}$$

which means that the non-squared versions of the LS and RLS can be used for detecting a change in mean, and the squared ones for detecting a change in variance [2].

To handle a negative change in θ, a lower threshold can be specified.

Combination of methods

Table 4.2 summarises the possible combinations of distance measures and decision rules, where LS ^{2}, and RLS ^{2} indicate that the squared, true least squares versions have been used, whilst LS and RLS mean that only the mean has been removed.

Table 4.2: *Possible combinations of distance measures and decision rules*

	LS	LS ^{2}	RLS	RLS ^{2}	Log-likelihood ratio
Direct thresholding	x	x	x	x	x
Recursive CUSUM, mean	x		x		x
Recursive CUSUM, variance		x		x	x
GMA, mean	x		x		x
GMA, variance		x		x	x

5

Simulations and results

In the simulations, faults were injected into the system, and the behaviour of the system reacting to these faults was noted by recording the true speeds and hoist position; the measured outputs; and the predicted speeds and hoist positions. The different fault detection methods were applied in the cases where unsafe situations could occur, to see if the faults were detected. The methods were also applied in the non-dangerous cases, to investigate whether false alarms would be raised. Since many simulations were made, not all can be represented in the thesis. This chapter contains a description of the simulations, and of the different faults that can occur. In the end, three examples are provided, one of a dangerous fault being detected, one of a safe fault being allowed, and one example of a fault in the load sensor.

5.1 Simulation method

For the fault injections, the model and observer were implemented in Matlab Simulink. The block diagram showing the architechture of the system is given in Figure 5.1. The `y_real` output contains the true speeds and hoist position of the system. Sensor faults are added after the `y_real`, but before the feedback to the speed controls. The measurement noise is added after the feedback, since the fed back signals are filtered in the true system. `y_meas` contains the sensor measurements, and `y_hat` is the estimated signals. The Simulink model was used for the simulation of the model and the observer. Creation of residuals and application of distance measures and decision rules were made offline using scripts.

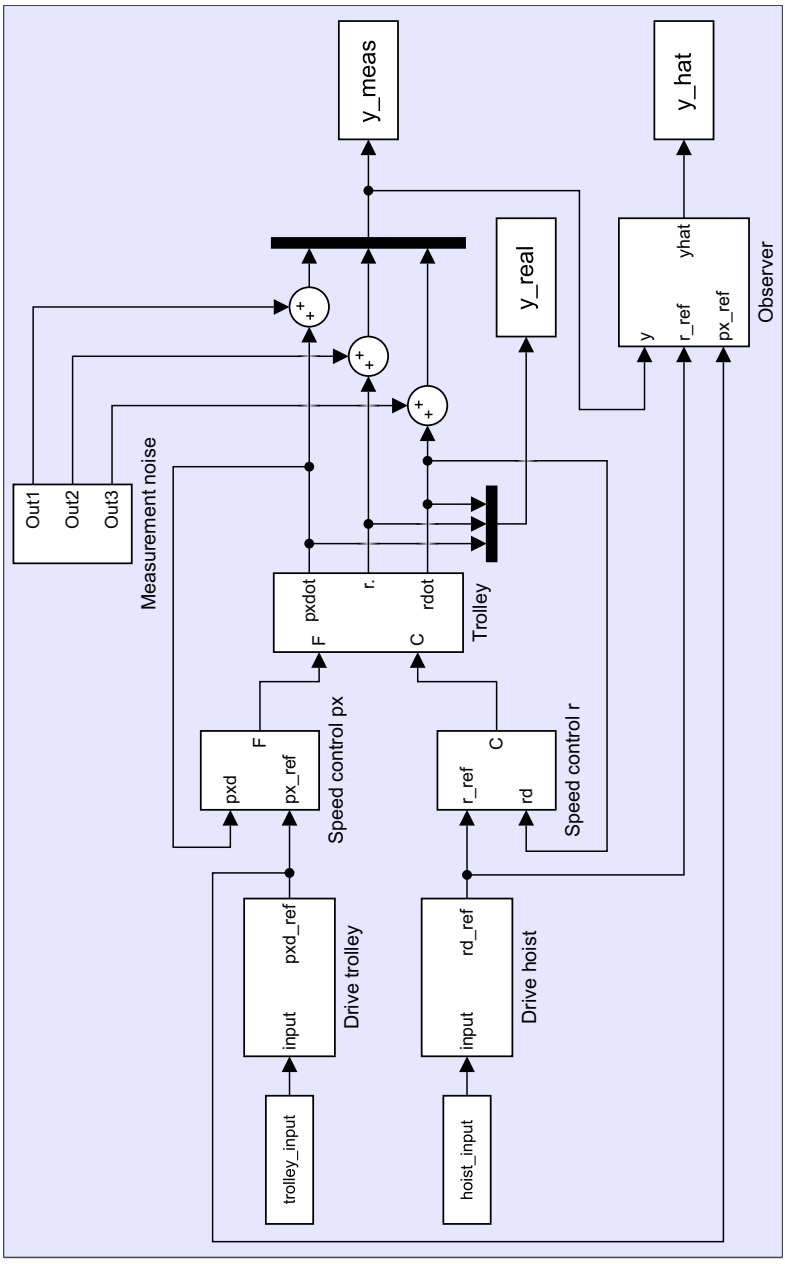

Figure 5.1: *Block diagram of the Simulink model used for fault simulations. The block* Trolley *contains the model created in Chapter 3. The blocks called* speed control *contain the PI regulators.* Observer *contains the linear observer developed in Section 4.1. The* Drive *blocks contain pre-treatment of the input reference signals.*

The input reference signals to the system are not exactly the same as the signals from the remote control in the mockup crane. The mockup crane gets three binary inputs each for the trolley and hoist. The inputs to the Simulink model consist of one signal each, with reference speeds in RPM. The drive blocks add the delay that is present in the true system, limits the acceleration for the trolley and converts the inputs from RPM to [m/s]. The speed control blocks contain the PI controllers, the trolley block includes the differential equations, and the observer block contains the Luenberger observer. The added measurement noise is white with a variance that was chosen to 0.003 m/s for the speeds, and 0.01 m for the hoist position.

To perform the fault injections, the normal behaviour has to be established. Input signals were created for the following crane movement:

1. System is still for 5 seconds
2. Lifting 5 seconds, normal speed
3. Moving trolley in positive direction 10 seconds, normal speed
4. Lowering the load 5 seconds, normal speed
5. Standing still 2 seconds
6. Lifting 4 seconds, normal speed
7. Moving trolley negative direction 10 seconds, normal speed
8. Lowering load 4 seconds, normal speed
9. System is still 5 seconds

This corresponds to a load being lifted 1.05 m, moved 1.2 m, lowered back down, lifted 0.84 m, moved back 1.2 m, then finally lowered back down. This is the basis for the fault simulations. The signals for the fault-free case are found in Figure 5.2.

Because the observer is linear, there are nonzero residuals also in the fault-free case. The residuals without noise can be found in Figure 5.3. The fault detection system must let these residuals pass, whilst still detecting faults. Before any other simulations were done, the thresholds and other parameters were set to allow these residuals with some margin. It is important to tune these parameters to let the fault-free case pass, without false alarms, but also not to miss any real faults.

It was found that the residuals for the fault free case depend on the mass of the load, such that a higher load mass leads to more overshoot for the linear observer. So as not to have to tune several sets of parameters and thresholds, all simulations were made for the same load, $m = 23$ kg, which is the load used during the data collection with the mockup crane.

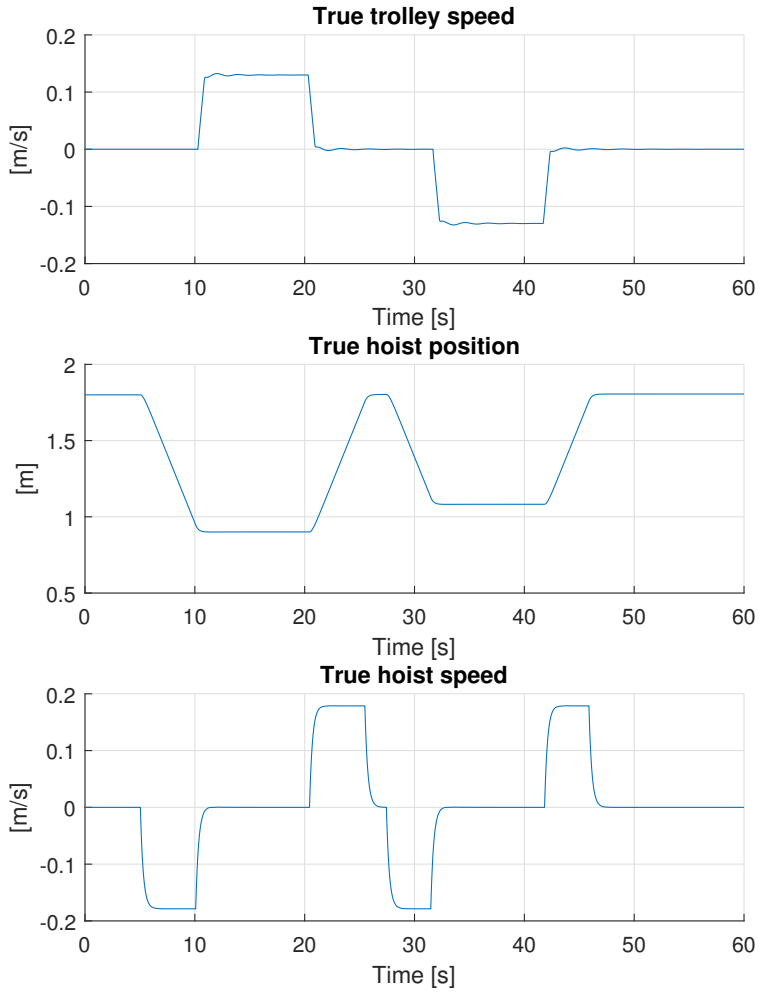

Figure 5.2: *Simulated true speeds and hoist position for the fault free case. This is the nominal behaviour used to set the parameters for the fault detection.*

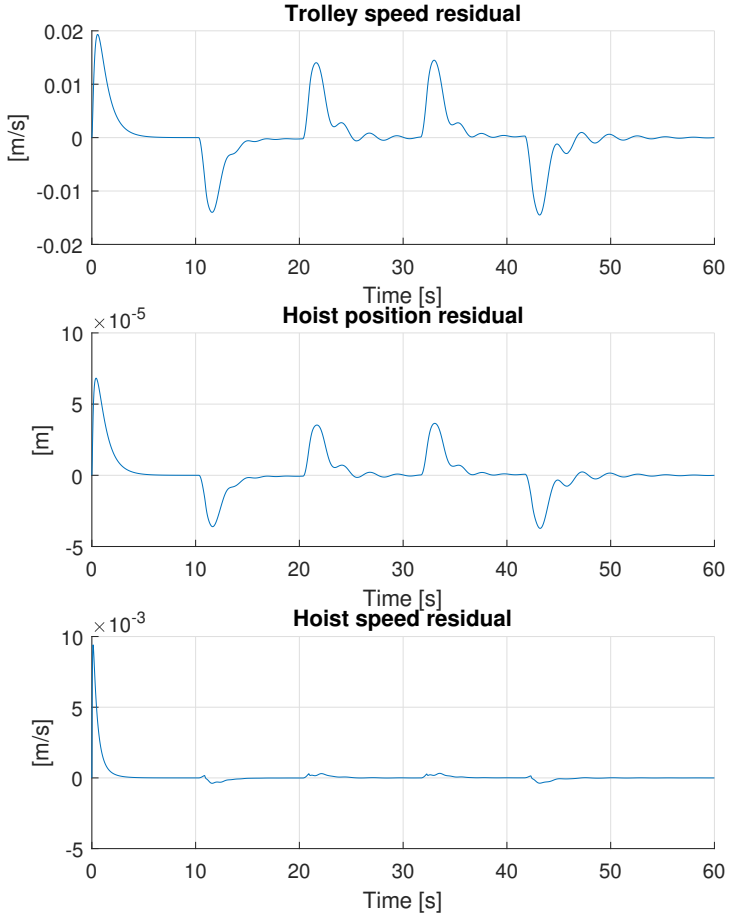

Figure 5.3: *Residuals for the fault free case without added measurement noise, that is the difference between the simulated signals and the observed signals. The observer is linear, leading to overshoots when the input signals change value. These residuals have to be accepted by the fault detection system.*

5.2 Fault modelling

To make a fault injection, the faults have to be modelled. The faults are inserted either by addition to a signal, or multiplication with a signal. In general all faults can be categorised as one of the following types, or some combination thereof:

Abrupt fault: A fault that occurs abruptly, and stays with a constant magnitude. For example a sensor getting a bias.

Drift fault: A fault that grows with time. Often the type of fault caused by deterioration of components over time.

Intermittent fault: A fault that occurs and disappears randomly. Can be caused by a loose contact.

The abrupt fault is modelled as a step being added to or multiplied with a signal. The drift fault is modelled with an added or multiplied ramp, and the intermittent fault is described with a random signal. Figure 5.4 illustrates the different types of fault signals. Intermittent faults are not treated in the remainder of the thesis; abrupt and drift faults were considered more relevant.

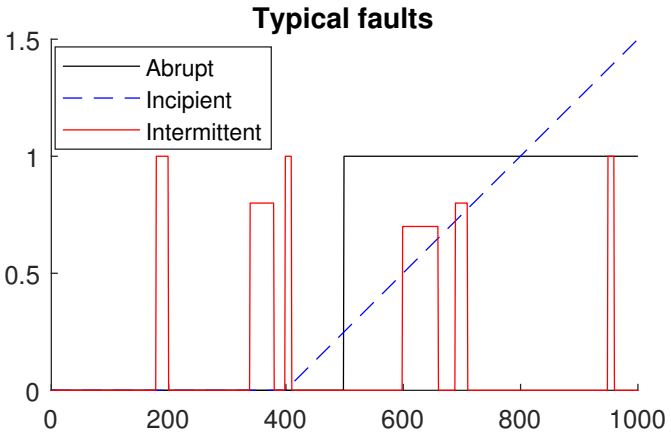

Figure 5.4: *Typical fault behaviours.*

5.3 Safe and unsafe faults

Before the simulations, a definition of what is a safe or an unsafe fault must be established. Given a certain reference speed and direction, the system can in general terms react in one of four different ways:

1. The system moves in the correct direction, at the specified speed. This is the normal, expected response.

2. The system moves in the correct direction, at a lower speed. This is an indication that there is something wrong in the system, but it is usually not dangerous.

3. The system moves in the correct direction, at a higher speed. This can lead to potentially dangerous situations.

4. The system moves in the wrong direction. This is also potentially dangerous.

Using this generalisation, the safety question can be reduced to detecting when the system moves faster than expected, or in the opposite direction.

Many systems also contain a load sensor, which is used to stop the user from trying to lift loads that are too heavy for the crane. A fault in this sensor can lead to breaking of the crane, dropping of load, or a complete stopping of the crane with the load hanging in mid-air. Being able to detect issues with the load sensor is thus also highly interesting.

5.4 Faults

Firstly, it is interesting to investigate whether the different fault detection methods can detect faults within the sensors on which they rely. Typical sensor faults are short cuts or cut-offs within the sensors, leading to a change in gain or bias. Other possible faults are that the sensors send out their maximum or minimum value.

The following faults are investigated for the trolley speed sensor and the hoisting position and speed:

- A change in bias, modelled for the abrupt case as an added step, and for the drift case as an added ramp.

- A change in gain, modelled by multiplying the signal with a step or ramp.

- The sensor sending out 0 or maximum values, modelled by replacing the measured signal with a constant.

A change in the bias of the sensor can always lead to unwanted situations, since it alters the supposedly stationary behaviour. As can be seen in Figure 5.5, where a bias of 0.03 m/s is added to the trolley speed sensor at $t = 15$ s, the trolley is moving with a speed of -0.03 m/s when it is supposed to stand still. Since this system uses predefined speed levels, it is also not possible to counteract the speed change manually. Thus, it is of interest to detect both negative and positive bias steps and ramps. Note that a positive bias leads to a negative change in speed, and vice versa, because of the feedback of the signal.

An error in gain is less important to detect for small faults, since it does not affect the direction of the speed or the stationary state. However, for larger decreases in gain, the speed can increase to dangerous levels.

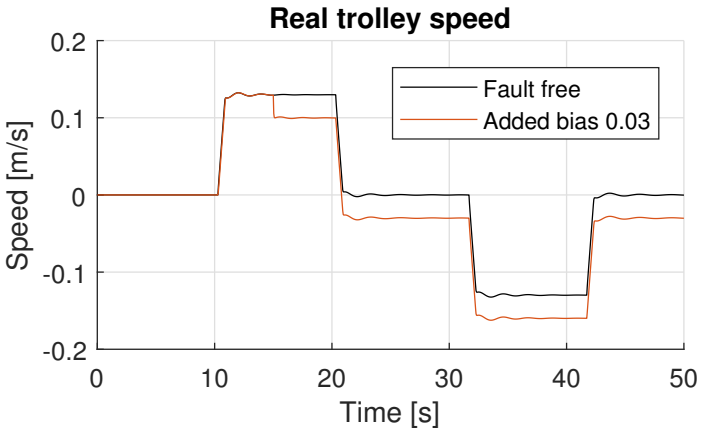

Figure 5.5: *Comparison between the real trolley speed in the fault-free case, and when a bias of +0.03 m/s is added at t = 15 s.*

The sensor sending out maximum or minimum values can be seen as an especially large change in bias, and leads to large changes in speed. The sensor value being stuck at 0 does not have any effect, as long as the input is zero. When the input is not zero, however, it quickly leads to a large increase in speed.

Values from the position are registered, but not fed back into the system. An error in the hoist position sensor will not affect the true speed of the system, but it might have an affect on the observer. The position sensor is also useful to detect certain changes in the hoisting speed, if there is an error in the hoist speed value.

Load sensor

Hoists and cranes are specified to work within a certain load mass range, and if the load weight is more than 10% above the limit, the crane can stop. Often an overload function is in use, warning the user if the load is too heavy before operation starts. However, if there is an issue with the load sensor, one can be left with a heavy load hanging mid-air, with the crane system not moving. It is thus of interest to investigate whether it is possible to detect a fault in the load sensor using the speed and position sensors. Since the observer is based on the value of the load sensor, a fault in the load sensor was modelled by giving the observer block, and the trolley block different values for the load mass.

System faults

Faults can occur in other parts of the system, not only in the sensors. Any faults occuring from the remote control up until the reference speed cannot be detected, since the faulty signals would be given to the observer as well. Faults that occur within the controllers or the mechanics can be detected.

5.5 Fault simulation

This section provides three examples selected among the many simulations made throughout the project. These examples have been chosen to display the properties of the fault detection system. The basis of the simulations is the movements described in Section 5.1, and thus the nominal behaviour of the crane is described by Figure 5.2. The parameters have been tuned to allow the fault-free case, and the same parameter sets have been used in all three examples.

Firstly, an example of a dangerous fault in the hoisting sensor being detected is presented. This is followed by an example of a safe fault within the same sensor being allowed. Finally a fault in the load sensor is investigated.

Detection of an abrupt bias change in the hoist speed sensor

As previously mentioned, a change in the bias of a sensor is always important to detect. In this section, the process of detecting an abrupt change in the bias of the hoist speed sensor is presented.

The fault injection consists of a step of size +0.02 m/s added to the hoist speed measurement at $t = 7$ s. This results in the system behaviour in Figure 5.6.

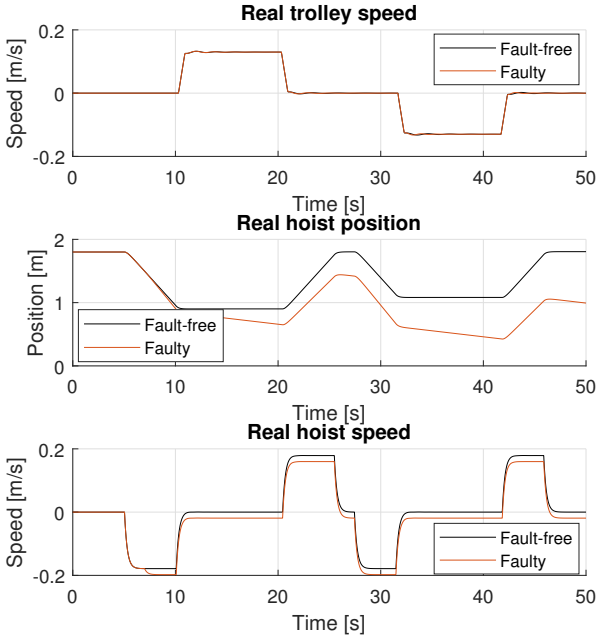

Figure 5.6: *Comparison of the nominal behaviour of the crane and the reaction to a fault injection consisting of an abrupt bias change +0.02 m/s in the hoisting speed sensor at $t = 7$ s.*

The hoist speed residuals from the simulation are found in Figure 5.7. The abrupt change causes a peak in the residuals, since the feedback adapts the speed, and the observer adapts to the new value.

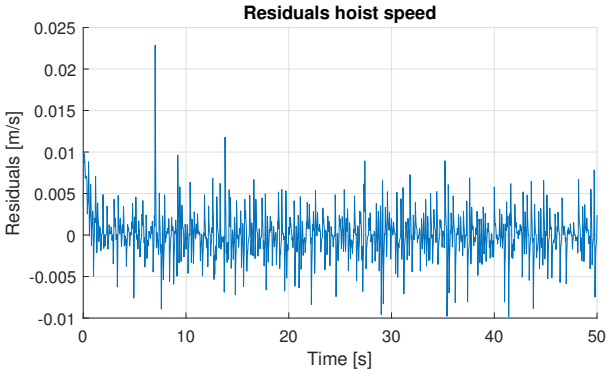

Figure 5.7: *Hoist speed residuals for a fault injection consisting of an abrupt bias change +0.02 m/s in the hoisting speed sensor at $t = 7$ s.*

The residuals are treated using the different distance measures. The results of the distance measures are shown in Figure 5.8, where the thresholds for direct thresholding and the alarm times are also indicated. Direct thresholding detects the fault after one sample for all the different distance measures.

Figure 5.9 and Figure 5.10 present the results of the CUSUM method, and the GMA method, respectively. The fault is detected after one sample for all the distance measures used in CUSUM. For the GMA method, the fault is detected after one sample for the LS and the RLS symptoms, as well as for the log-likelihood sufficient statistics for variance. The log-likelihood sufficient statistics for a change in mean do however not detect the fault.

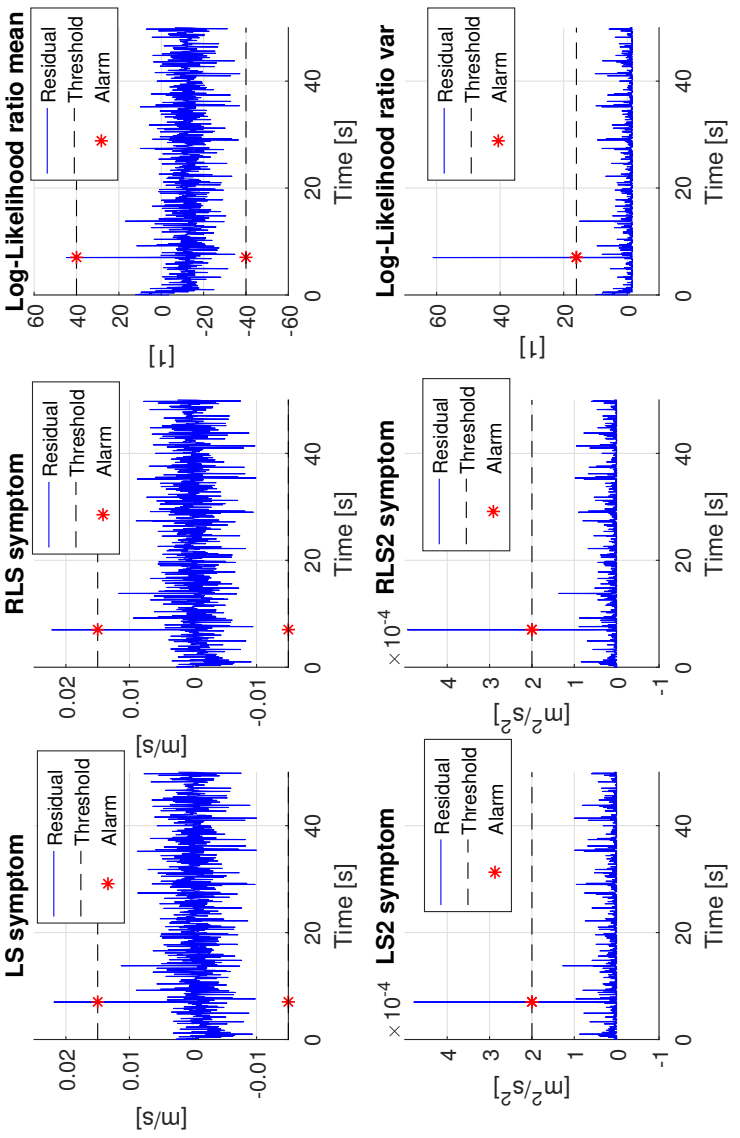

Figure 5.8: *Direct thresholding performed on residuals from a fault injection where an abrupt bias change of +0.02 is introduced at $t = 7s$. The fault is detected for all different distance measures.*

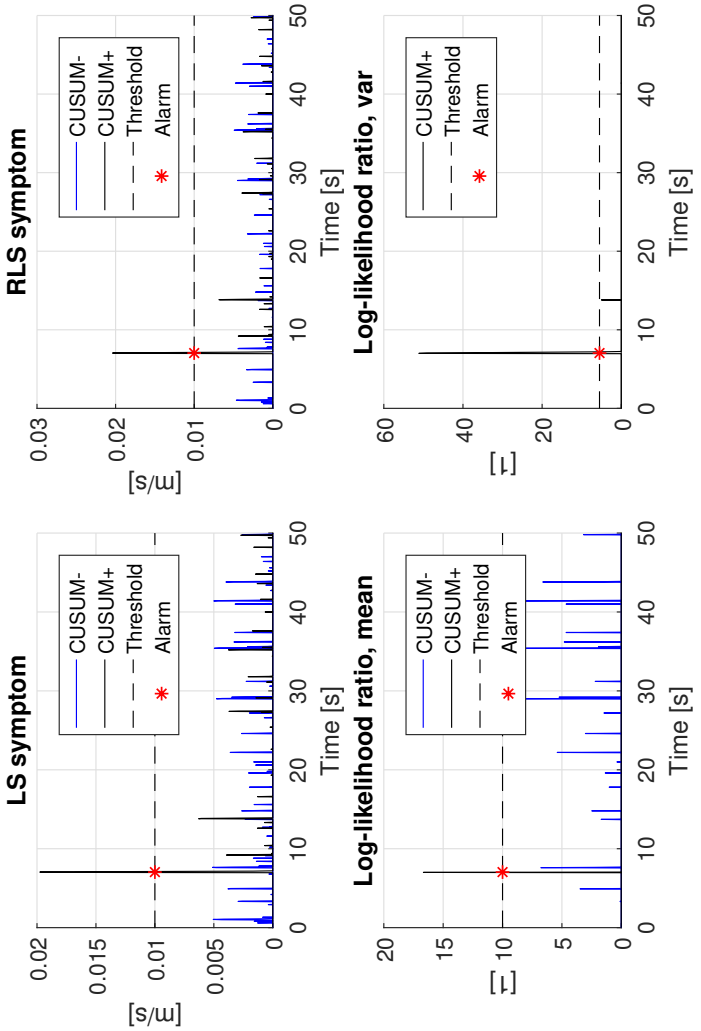

Figure 5.9: *CUSUM decision rule applied to residuals from a fault injection where an abrupt bias change of +0.02 is introduced at $t = 7$s. CUSUM- and CUSUM + indicate the decision rules for detection of a negative and a positive change, respectively, see (4.34). The fault is detected for all different distance measures.*

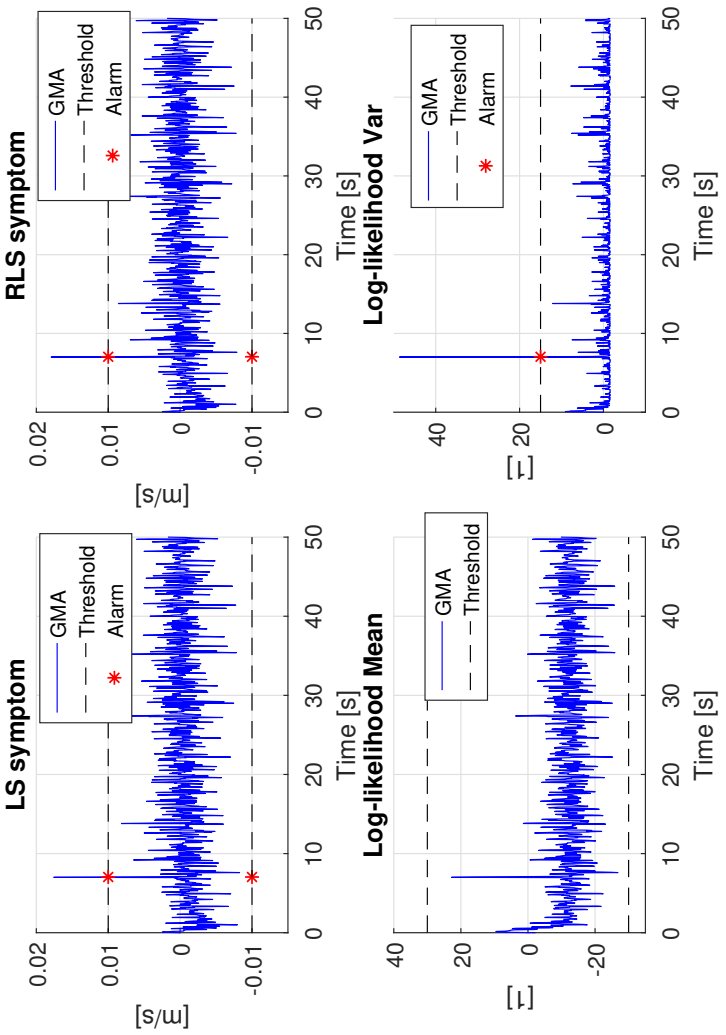

Figure 5.10: *GMA decision rule applied to residuals from a fault injection where an abrupt bias change of +0.02 is introduced at $t = 7$s. The fault is detected for all distance measures, except the sufficient log-likelihood statistic for a change in mean.*

Allowance of a drift gain change in the hoist speed sensor

A drift change of the gain, with a positive incline, leads to a deceleration of the speed. For a slow deceleration this fault is not a dangerous, so an alarm should not be raised. A fast deceleration could lead to swaying of the load which is potentially dangerous considering the sizes of the loads carried by overhead cranes.

Figure 5.11 displays a comparison between the expected behaviour of the crane, and the reaction to a drift gain change of incline 3% introduced at $t = 7$s.

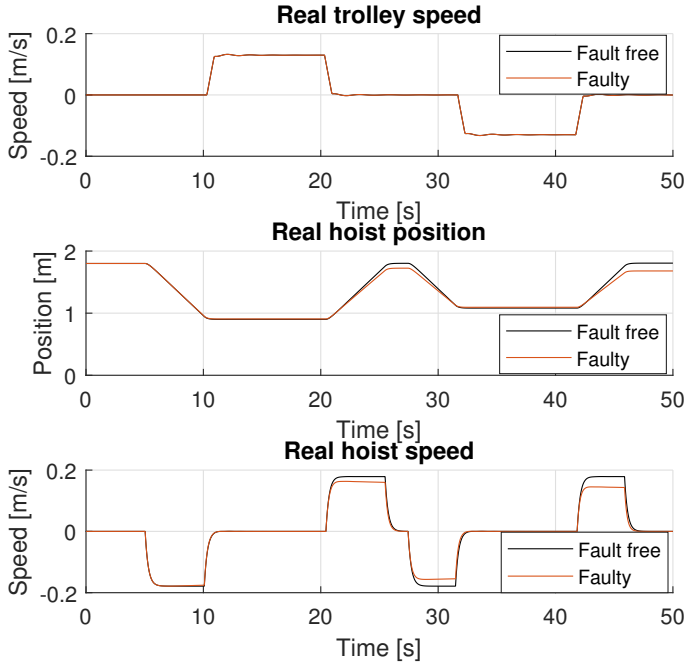

Figure 5.11: *Comparison of the nominal behaviour of the crane and the reaction to a fault injection consisting of a drifting gain change with an incline of 3% in the hoisting speed sensor at $t = 7$ s*

The hoisting position and hoist speed residuals from this test are found in Figure 5.12. The change in gain is smooth enough not to cause any sharp peaks in the residuals.

The distance measures when direct thresholding is applied are displayed in Figure 5.13, the results from CUSUM are shown in Figure 5.14, and Figure 5.15 shows the results using GMA. The fault is not detected using any of the methods.

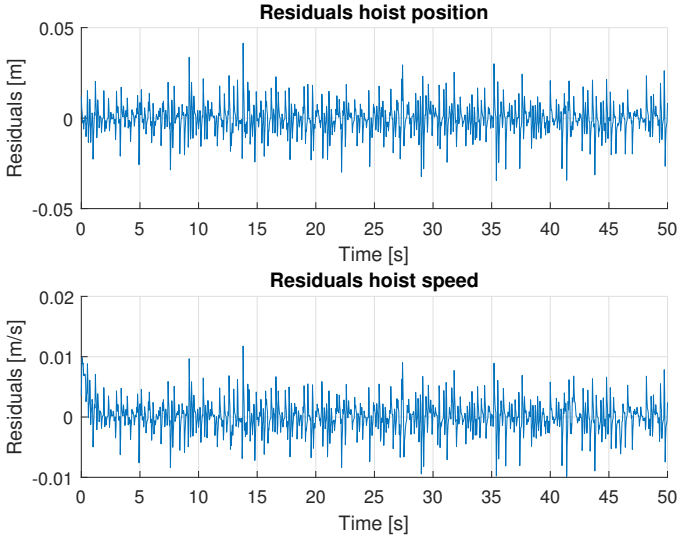

Figure 5.12: *Hoist speed and position residuals from a fault injection where a drift gain change of +3% is introduced at $t = 7s$.*

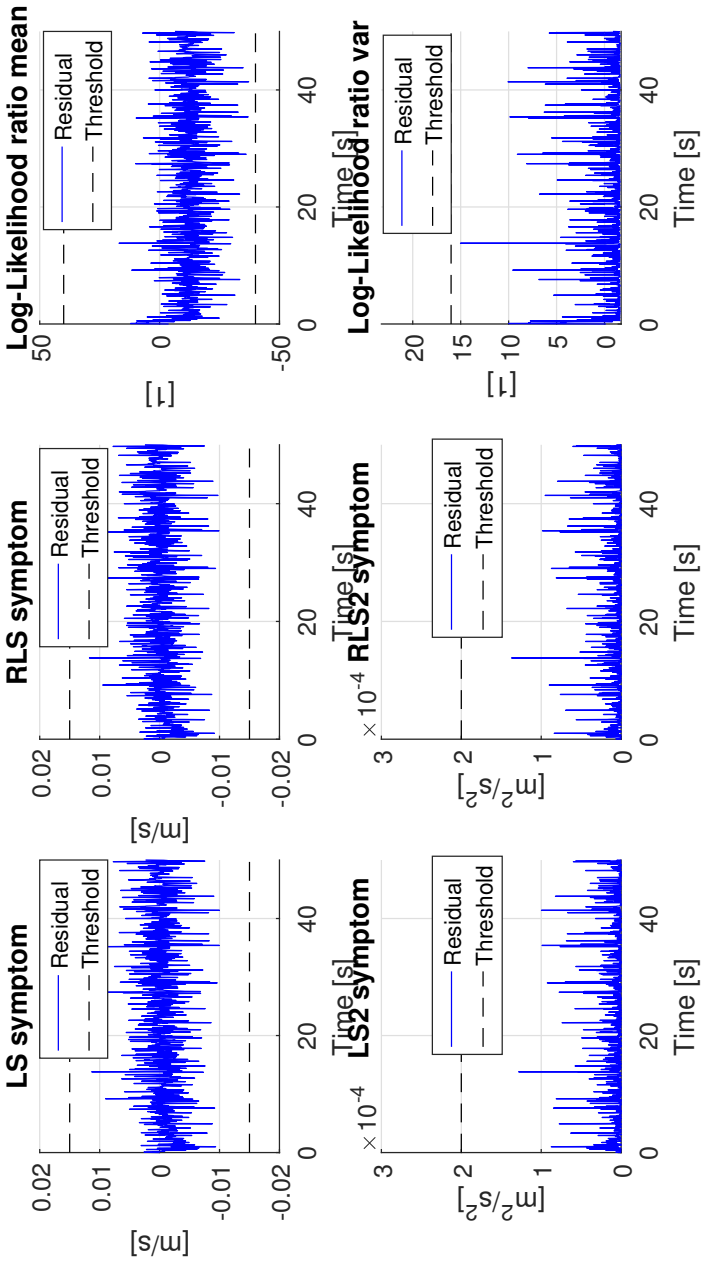

Figure 5.13: *Direct thresholding performed on residuals from a fault injection where a drift gain change of +3% is introduced at $t = 7s$. The fault is not detected.*

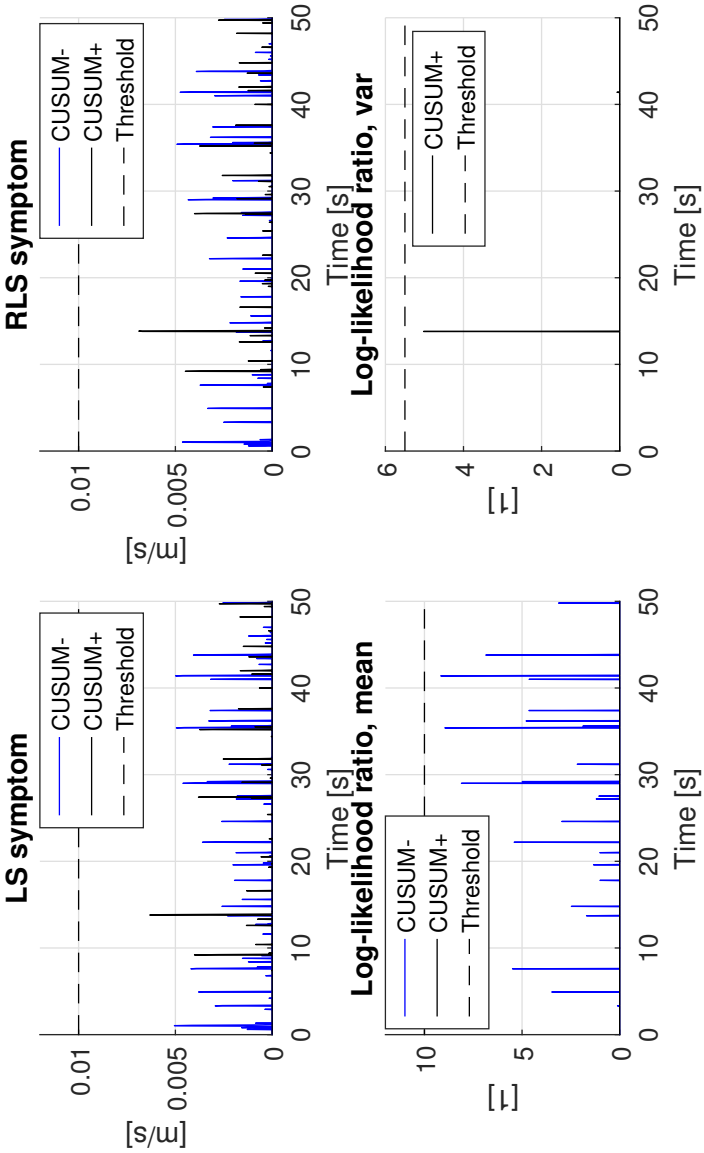

Figure 5.14: *CUSUM decision rule applied to residuals from a fault injection where a drift gain change of +3% is introduced at $t = 7$s. The fault is not detected.*

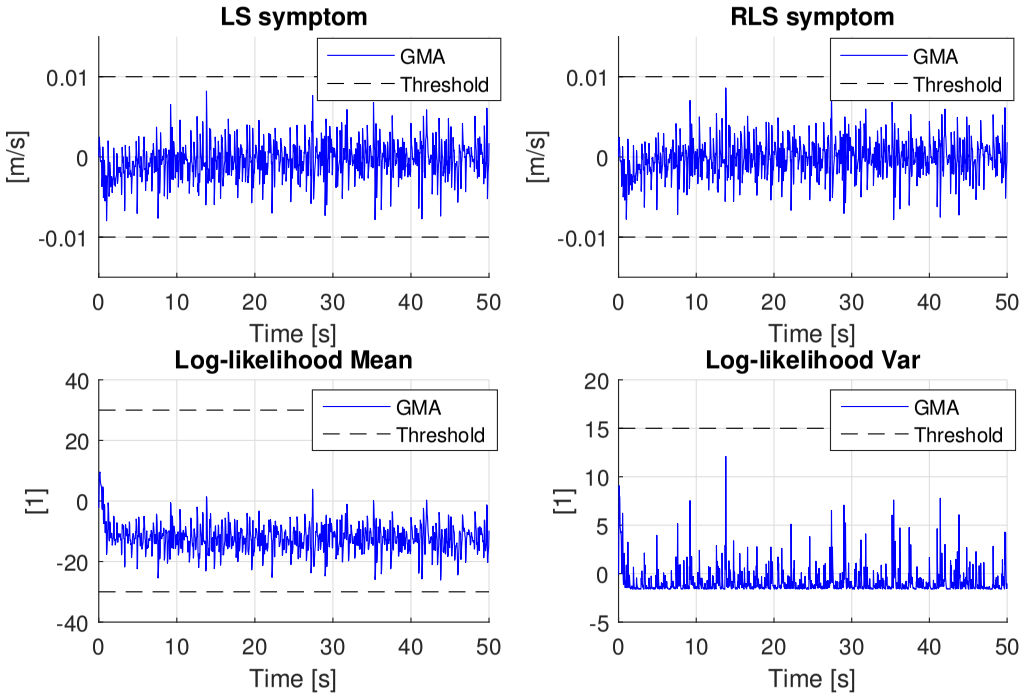

Figure 5.15: *GMA decision rule applied to residuals from a fault injection where a drift gain change of +3% is introduced at $t = 7s$. CUSUM- and CUSUM + indicate the decision rules for detection of a negative and a positive change, respectively, see* (4.34)*. The fault is not detected.*

Detection of load sensor faults

The load sensor fault injection is different in the sense that the fault does not change the system during the test, it is present from the start of the test. During this simulation, the true load is not outside the accepted range of the test crane, but it does represent a significant load sensor error.

All the thresholds and other parameters were chosen using the load mass m = 23 kg, which was the load used during the data collection. The load fault injection was performed by using $m_{measured}$ = 23 kg as the mass in the observer, but the model of the crane used another value, m = 40 kg. The expected behaviour would be that the crane lifts the load slower than expected, and lowers it faster.

Figure 5.16 shows the true speed of the system with a load of 23 kg, compared to the system with a load of 40 kg. The most clear difference is in the acceleration of the hoisting. The question is whether the fault detection system can detect a difference when the observer is based on the system behaving like the load is 23 kg, but the true load is 40 kg.

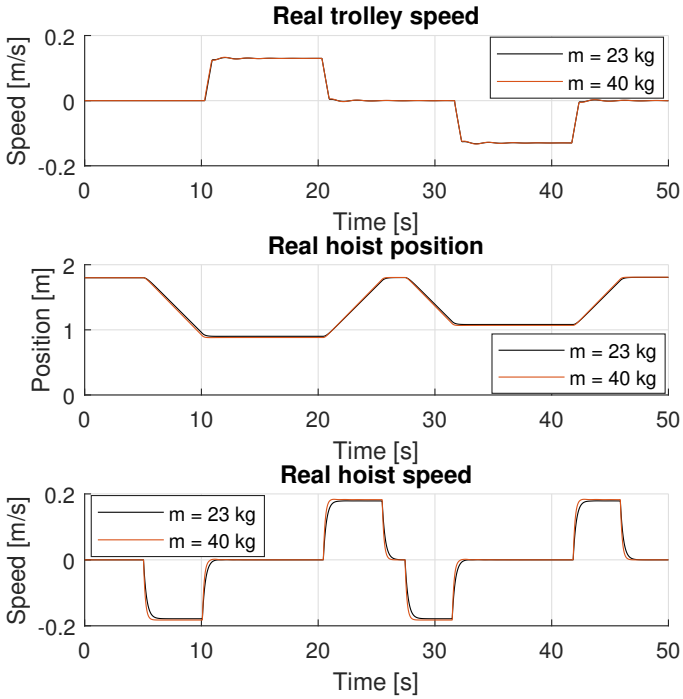

Figure 5.16: *Comparison between the behaviour of the system with a load of m = 23 kg, and a load of m = 40 kg.*

The residuals of this simulation are found in Figure 5.17. There are some peaks, but nothing that stands out specifically for any of the residuals.

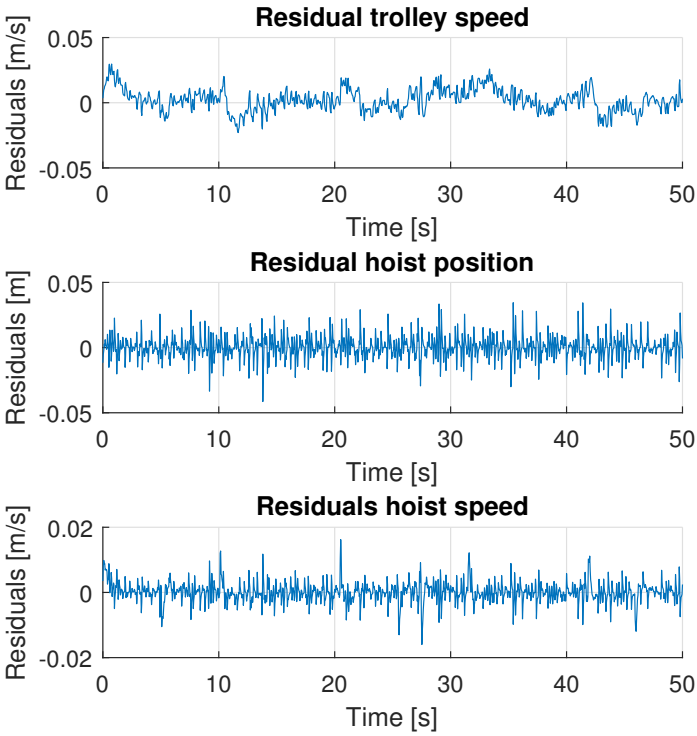

Figure 5.17: *Residuals of a simulation where the measured load used in the observer is 23 kg, but the true load in the system is 40 kg.*

For the hoisting speed residuals, the CUSUM method provides the best results. The results of this method are presented in Figure 5.18. As can be seen in the figure, the fault is detected very quickly after the hoisting has been initiated at t = 5 s for the mean change detection symptoms, and after the hoisting has stopped at t = 10 s, for the variance change detection symptom. The detection times after these events are 3 to 5 samples. The fault cannot be detected before the hoisting is initiated, since it is only through the hoisting that a change between the behaviour of the crane and the observer can occur. CUSUM is also the best method for the trolley speed residuals, which is shown in Figure 5.19. The fault is detected shortly after the trolley reaches its normal speed for the first time. Fault detection using the position sensor does not detect this fault.

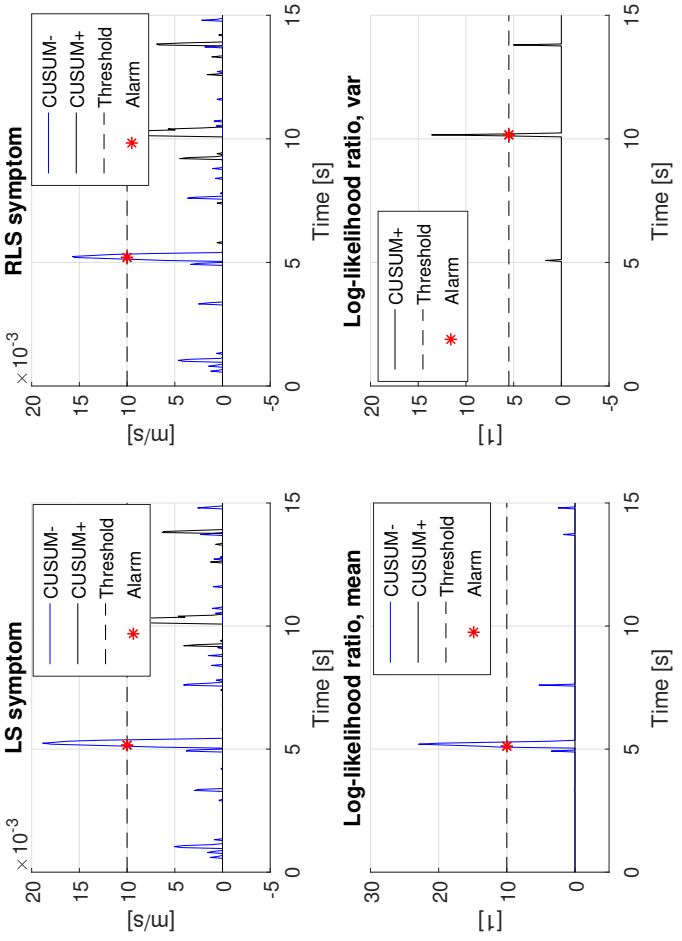

Figure 5.18: *CUSUM algorithm applied to the hoisting speed residuals for a load error simulation where the measured load used in the observer is 23 kg, but the true load in the system is 40 kg. The fault is detected within 5 samples of hoisting start for the mean change detection symptoms, and 4 samples after the first hoisting has stopped for the variance detection symptom. CUSUM- and CUSUM + indicate the decision rules for detection of a negative and a positive change, respectively, see* (4.34).

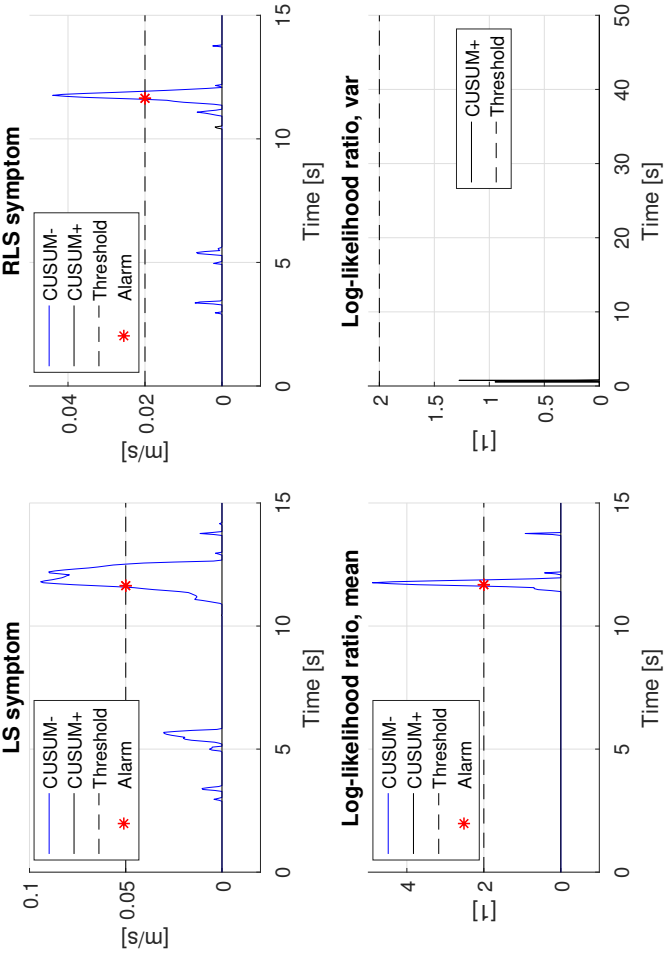

Figure 5.19: *CUSUM algorithm applied to the trolley speed residuals for a load error simulation where the measured load used in the observer is 23 kg, but the true load in the system is 40 kg. The fault is detected when the trolley has reached its normal speed the first time for the LS and RLS symptoms, and the sufficient log-likelihood statistic for detecting mean. The variance detector does not detect the fault. CUSUM- and CUSUM + indicate the decision rules for detection of a negative and a positive change, respectively, see* (4.34).

6

Discussion and conclusion

This chapter contains a discussion of the method and the results. It also brings back objectives, and compares them to the results leading to a conclusion. In the end ideas for future work are presented.

6.1 Method

The model is based on a small crane using a light load. For a larger crane and significantly bigger loads, there are dynamics that are not present in the smaller crane. For example the hoisting wire can stretch due to a large load, leading to vertical oscillations as well as the horizontal sway. Using the model on a larger scale crane can lead to a worse fit.

This far only simulations have been made, no fault injections have been carried out in a real crane. Since the model is validated using fault free data, there is no guarantee that the crane will act as the simulation model when a fault is introduced.

A linearised observer causes residuals in the fault free case, especially for the trolley movement. This means that the observer must be faster, and the thresholds higher, making the overall fault detection less efficient.

In all the method is not perfect, but it is a good method for the initial research. The simplifications made have made it possible to investigate modelling, validation and fault detection within the time frame of a master's thesis. During possible further work with this project all parts should be re-evaluated and adapted to real situations.

6.2 Results

The fault detection system derived in the project is very efficient in detecting abrupt faults, for all the tested methods. These faults cause peaks the residuals, as in Figure 5.7, since the controller, and to a smaller extent the observer, adapt to the large change. These peaks are augmented by the distance measures which square the residuals, such as LS and RLS, making the detection easier. Direct thresholding works well for detecting these peaks, as well as the CUSUM method. Depending on the choice of forgetting factor, GMA can end up smoothing the peaks too much, leading to the fault not being detected.

Incipient faults do not cause this peak in the residuals, meaning that the fault will usually not be detected as fast. Since the changes in the residuals are more gradual, direct thresholding can give a long time to detection. CUSUM can, with its probability basis, give a faster detection for these drift-like changes.

During the project, the idea was formed that the sign of the residuals could be used to make a clear differentiation between safe and unsafe faults. The residuals are created by subtracting the estimated outputs, $\hat{y}$, from the measured ones, y, which gives the residuals signs according to Table 6.1, where higher speeds are considered dangerous, but lower speeds are not.

Table 6.1: *Possible speed faults and the resulting signs of the residuals.*

Measured speed		**Sign**
Higher than expected, positive direction	$y > \hat{y}$	$\varepsilon > 0$
Lower than expected, positive direction	$y < \hat{y}$	$\varepsilon < 0$
Higher than expected, negative direction	$y < \hat{y}$	$\varepsilon < 0$
Lower than expected, negative direction	$y > \hat{y}$	$\varepsilon > 0$

Multiplying the residual with the sign of the estimated speed, would give the dangerous situations positive residuals, and the safe situations negative residuals, as in Table 6.2. The intention was then to use one-sided tests, detecting only the faults causing positive residuals.

Table 6.2: *Possible speed faults and the resulting signs of the residuals multiplied with the sign of the estimated speed.*

Measured speed		**Sign**
Higher than expected, positive direction	$y > \hat{y}$	$\varepsilon \cdot \text{sign}(\hat{y}) > 0$
Lower than expected, positive direction	$y < \hat{y}$	$\varepsilon \cdot \text{sign}(\hat{y}) < 0$
Higher than expected, negative direction	$y < \hat{y}$	$\varepsilon \cdot \text{sign}(\hat{y}) > 0$
Lower than expected, negative direction	$y > \hat{y}$	$\varepsilon \cdot \text{sign}(\hat{y}) < 0$

A problem with this idea is that the method only works under the assumption that the sensors are working. As mentioned in Section 5.4, a negative change in bias or gain of a sensor will lead to a higher speed, in the case of a positive setpoint speed. This will give the residuals signs opposite to Table 6.2.

A second problem that occurs when trying to separate safe and unsafe faults is that it is initially impossible to distinguish a change in bias, which should be detected, from a change in gain, which can be allowed providing that the gain becomes higher, leading to a lower speed. The only difference appears when the setpoint speed becomes low enough that the bias fault leads to a speed in the opposite direction, while the gain fault does not make the speed change sign. By this time, the observer and controller have usually adapted to the change in the sensor signal, which makes the fault undetectable.

6.3 Objectives revisited

Coming back to the objectives of the thesis:

- Is it possible to create a mathematical model of the test crane, based on first principles, to use as a basis for fault detection?
- Is it possible to detect faults within the crane system, measuring only the system inputs and outputs?
- Is it possible to detect only dangerous faults, and allow safe faults to pass?

As described in Section 3.3, the mathematical model derived in Chapter 3 is deemed good enough to use for developing a fault detection system. The project results indicate that it is possible to create a fault detection system for an overhead crane using the input and output signals already present in the system. If the sensors are assumed to be working properly, faults affecting the speed of the hoist or trolley are detected. Faults within the sensors themselves are usually also detected. Dangerous faults are detected as long as they have a large enough effect on the system, and there are safe faults that are allowed, but this is not yet the case for all faults. Whether safe and unsafe faults can be differentiated still needs further work.

In conclusion this project shows that the idea of model based fault detection for overhead travelling cranes has potential. The possible benefits of succeeding with such a system are many. One example is safer workplaces without loss of production due to only dangerous faults being detected. Another is less expensive fault detection since no additional sensors have to be installed.

6.4 Future work

There are many ways to continue with this project, both through further investigations and then possibly development of a final product.

The first step could be to investigate different types of faults, not presented in this thesis. It could be a good idea to perform interviews with users of this type of cranes, or with the producers, to find out which types of faults usually occur, and how they affect the system. When this is done, a more thorough analysis of how these faults present themselves in the systems can be done, potentially leading to a re-evaluation of the choices made in this thesis for the fault detection system.

Another step is to investigate how well the fault detection system works for a real crane, by performing fault injections on the mockup crane and using the observer offline. The code produced during the project can easily be changed to handle measured signals instead of simulated ones. Whether the model performs differently for larger cranes, with much heavier loads, is another intriguing question.

If the linear observer turns out not to be sufficient, another idea is to produce a non-linear observer using for example an extended Kalman filter. Non-linear observers have the downside of often being calculation heavy, but if the performance is good, the increased safety can make it worth it.

It would also be necessary in the long run to extend the model to work in three dimensions, and perform validation. Continuing further it can be interesting to investigate how the model should be adapted to the use of several trolleys and girders performing synchronised motion.

Appendix

A

Derivation of state equations

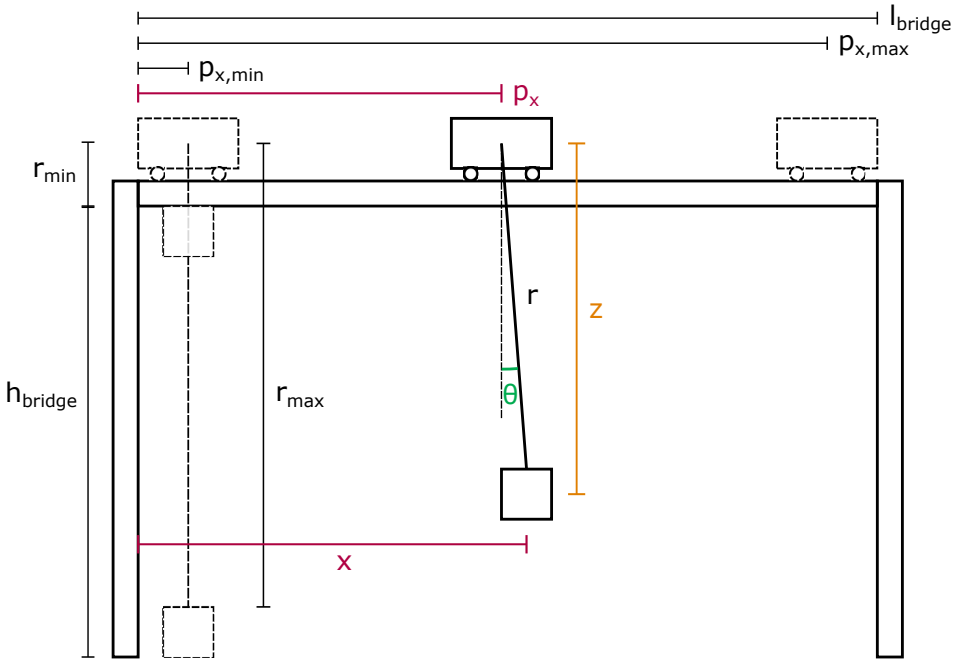

Figure A.1: *Illustration of an overhead crane, used for derivation of state equations.*

The position of the load is described by

$$\begin{aligned} x &= px + r\sin\theta \\ z &= r\cos\theta \end{aligned} \tag{A.1}$$

This gives the following states for the system

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} p_x \\ \dot{p}_x \\ r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{pmatrix} \tag{A.2}$$

A.1 Sway dynamics

The sway dynamics are described by the states θ and $\dot{\theta}$. Newton's second law for the load can be found using the free body diagram in Figure A.2.

$$\begin{cases} m\ddot{x} &= -T\sin\theta - c_{th1}\dot{\theta}\cos\theta \\ m\ddot{z} &= -T\cos\theta + mg + c_{th1}\dot{\theta}\sin\theta \end{cases} \iff \begin{cases} -\frac{T}{m} &= \frac{\ddot{x}}{\sin\theta} + \frac{c_{th1}\dot{\theta}\cos\theta}{m\sin\theta} \\ -\frac{T}{m} &= \frac{\ddot{z}-g}{\cos\theta} - \frac{c_{th1}\dot{\theta}\sin\theta}{m\cos\theta} \end{cases} \tag{A.3}$$

T is the force from the hoist rope on the load, and c_{th} is a friction related constant.

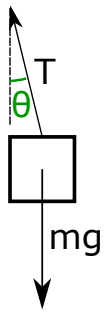

Figure A.2: *Free body diagram of the load*

Differentiating (A.1) twice gives

$$\begin{aligned} \ddot{x} &= \ddot{r}\sin\theta + 2\dot{r}\dot{\theta}\cos\theta + r\ddot{\theta}\cos\theta - r\dot{\theta}^2\sin\theta + \ddot{p}_x \\ \ddot{z} &= \ddot{r}\cos\theta - 2\dot{r}\dot{\theta}\sin\theta - r\ddot{\theta}\sin\theta - r\dot{\theta}^2\cos\theta \end{aligned} \tag{A.4}$$

Combining (A.3) and (A.4), and rescaling the friction related constant to $c_{th} = \frac{c_{th1}}{2m}$ finally gives the sway dynamics

$$r\ddot{\theta} = -2(\dot{r} + c_{th})\dot{\theta} - \ddot{p}_x - g\sin\theta \tag{A.5}$$

A.2 Translational dynamics

The translational dynamics are governed by the forces shown in Figure A.3.

$$\begin{aligned} m_{trolley}\ddot{p}_x &= F + T\sin\theta - F_{fr} \\ &= F + T\sin\theta - c_x\dot{p}_x \end{aligned} \tag{A.6}$$

Where F is the force of the motor moving the trolley forward and c_x is a friction constant. There is no vertical movement.

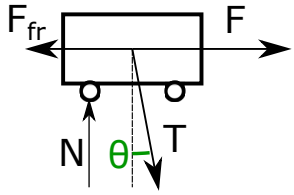

Figure A.3: *Free body diagram of the trolley*

Combining with (A.3), and then inserting (A.4) gives

$$\begin{aligned} m_{trolley}\ddot{p}_x =& F - m\ddot{x} - 2mc_{th}\dot{\theta}\cos\theta - c_x\dot{p}_x \\ =& F - m(\ddot{r}\sin\theta + 2\dot{r}\dot{\theta}\cos\theta + r\ddot{\theta}\cos\theta - r\dot{\theta}^2\sin\theta + \ddot{p}_x) - c_x\dot{p}_x \\ & - 2mc_{th}\dot{\theta}\cos\theta \end{aligned} \tag{A.7}$$

Finally, insertion of (A.5) gives the relation non-dependent on $\ddot{\theta}$, which can be written as

$$\ddot{p}_x = \frac{F - c_x\dot{p}_x + m\sin\theta(-\ddot{r} + g\cos\theta + r\dot{\theta}^2)}{m_{trolley} + m\sin^2\theta} \tag{A.8}$$

A.3 Hoisting dynamics

The hoist contains a wheel on which the rope or chain is rolled. The hoisting motor exerts its force, C, on this wheel, as described in Figure A.4.

Newton's second law for the situation becomes

$$\begin{aligned} J\alpha &= -C_{fr} - C + bT \\ \iff J\ddot{r} &= -c_r\dot{r} - bC + b^2T \end{aligned} \tag{A.9}$$

Here, J is the moment of inertia of the hoist wheel, c_r is a friction constant, and b is the radius of the hoist wheel.

Using (A.3) removes the dependency on T

$$J\ddot{r} = -c_r\dot{r} - bC - \frac{b^2m}{\sin\theta}\ddot{x} - \frac{2b^2mc_{th}\dot{\theta}\cos\theta}{\sin\theta} \tag{A.10}$$

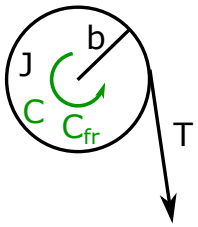

Figure A.4: *Free body diagram of the hoist wheel*

and then inserting (A.4) gives the relation without $\ddot{x}$.

$$\begin{aligned}(J + b^2 m)\ddot{r} = &- c_r\dot{r} - bC - \frac{b^2 m}{\sin\theta}(2\dot{r}\dot{\theta}\cos\theta + r\ddot{\theta}\cos\theta - r\dot{\theta}^2\sin\theta + \ddot{p}_x) \\ &- \frac{2b^2 m c_{th}\dot{\theta}\cos\theta}{\sin\theta}\end{aligned} \tag{A.11}$$

Removing $\ddot{\theta}$ using (A.5) and rearranging gives

$$(J + b^2 m)\ddot{r} = -c_r\dot{r} - bC - b^2 m(-r\dot{\theta}^2 + \ddot{p}_x\sin\theta - g\cos\theta) \tag{A.12}$$

And finally removing the dependency of $\ddot{p}_x$ using (A.8) gives the differential equation for hoisting without dependencies on any other second derivatives

$$\begin{aligned}\ddot{r} = &\frac{-(c_r\dot{r} + bC)(m_{trolley} + m\sin^2\theta)}{J(m_{trolley} + m\sin^2\theta) + b^2 m m_{trolley}} \\ &+ \frac{b^2 m(m_{trolley}(r\dot{\theta}^2 + g\cos\theta) - \sin\theta(F - c_x\dot{p}_x))}{J(m_{trolley} + m\sin^2\theta) + b^2 m m_{trolley}}\end{aligned} \tag{A.13}$$

Having a relation with only $\ddot{r}$ and no other second derivatives, makes it possible to find relations for the other states as well. Using (A.13) in (A.8) gives a relation with only $\ddot{p}_x$

$$\ddot{p}_x = \frac{(F - c_x\dot{p}_x)(J + b^2 m) + m\sin\theta(J(g\cos\theta + r\dot{\theta}^2) + (c_r\dot{r} + bC))}{J(m_{trolley} + m\sin^2\theta) + b^2 m m_{trolley}} \tag{A.14}$$

And finally inserting (A.14) into (A.5) gives the sway dynamics without explicit dependency on the trolley or hoisting accelerations.

$$\begin{aligned}\ddot{\theta} = &- \frac{2(\dot{r} + c_{th})\dot{\theta} + g\sin\theta}{r} \\ &- \frac{\cos\theta}{r}\frac{(F - c_x\dot{p}_x)(J + b^2 m) + m\sin\theta(J(g\cos\theta + r\dot{\theta}^2) + (c_r\dot{r} + bC))}{J(m_{trolley} + m\sin^2\theta) + b^2 m m_{trolley}}\end{aligned} \tag{A.15}$$

Bibliography

[1] Arbetsmiljöverket. Arbetsskador 2016, June 2017. Cited on page 1.

[2] Michèle Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Inc., 1993. Cited on pages 36, 38, and 41.

[3] Mogens Blanke, Michel Kinnaert, Jan Lunze, and Marcel Staroswiecki. *Diagnosis and Fault Tolerant Control*. Springer, 2003. Cited on page 27.

[4] M. Fliess, J. Levine, and P. Rouchon. A simplified approach of crane control via a generalized state-space model. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 736–741 vol.1, 1991. doi: 10.1109/CDC.1991.261409. Cited on page 3.

[5] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6): 3757, 2015. ISSN 1557-9948. Cited on page 3.

[6] A. Garcia, W. Singhose, and A. Ferri. Three-dimensional dynamic modeling and control of off-centered bridge crane lifts. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 139(4), 2017. ISSN 15289028. Cited on page 3.

[7] Humberto Henao, Seyed Mohammad Javad Rastegar Fatemi, Gérard André Capolino, and Sophie Sieg-Zieba. Wire rope fault detection in a hoisting winch system by motor torque and current signature analysis. *IEEE Transactions on Industrial Electronics*, 58(5):1727 – 1736, 2011. ISSN 02780046. Cited on page 3.

[8] The Mathworks Inc. Matlab. URL `https://se.mathworks.com/products/matlab.html`. Cited on page 14.

[9] The Mathworks Inc. Least-squares (model fitting) algorithms. URL `http://se.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html#brnoyes`. Cited on page 15.

[10] The Mathworks Inc. System identification toolbox. URL `https://se.mathworks.com/help/ident/`. Cited on page 14.

[11] Perry Y. Li. Lecture notes: State feedback and observer feedback. Cited on page 30.

[12] Lennart Ljung and Torkel Glad. *Control Theory*. Taylor & Francis, 2000. Cited on page 32.

[13] Lennart Ljung and Torkel Glad. *Modellbygge och simulering*. Studentlitteratur, 2:5 edition, 2004. ISBN 978-91-44-02443-1. Cited on pages 18, 23, and 24.

[14] Samuel Malinen. Afs change detection using signal estimation. master's thesis, 2005. Cited on page 3.

[15] Samuel Malinen, Christian Lundquist, and Wolfgang Reinelt. Fault detection of a steering wheel sensor signal in an active front steering system. *Fault Detection, Supervision and Safety of Technical Processes 2006*, pages 510 – 515, 2007. ISSN 978-0-08-044485-7. Cited on page 3.

[16] Samuel Malinen, Christian Lundquist, and Wolfgang Reinelt. Potentials of change detection algorithms for diagnosis in electronic steering systems, 2007. Cited on page 3.

[17] M. Pohjola, L. Eriksson, and K. Torkkeli. Filtering and fault detection schemes for a trolley crane system. In *2007 9th International Symposium on Signal Processing and Its Applications*, pages 1–4, Feb 2007. doi: 10.1109/ISSPA.2007.4555400. Cited on page 3.

[18] W. Reinelt and C. Lundquist. Observer based sensor monitoring in an active front steering system using explicit sensor failure modelling. Number Proceedings of the 16th IFAC World Congress, IFAC 2005, 2005. Cited on page 3.

[19] Mehrdad Saif and Weitian Chen. Actuator fault diagnosis for a class of nonlinear systems and its application to a laboratory 3d crane. *Automatica*, 47 (7):1435 – 1442, n.d. ISSN 00051098. Cited on page 3.

[20] K. Terashima and A. Kaneshige. Load-position control of overhead travelling crane in terms of fixed-pole approach for 3-d transfer path. In *European Control Conference, ECC 1999 - Conference Proceedings*, pages 1070–1075, 2015. Cited on page 3.