Integration Project

Gantry Crane

by

Máté Bakos
Thomas Booij

Student numbers:	4714962 & 4316932	
Project duration:	April 23, 2018 – June 22, 2018	
Supervisors:	Prof. dr. Robert Babuska	TU Delft, supervisor
	Ir. Sander Bregman	TU Delft

TU Delft

Contents

1 Introduction **1**

2 System description **3**

3 System Identification **5**

3.1 Method selection 5

3.2 Obtaining data from the set-up 6

3.2.1 Choosing input signal 6

3.2.2 Validation data 6

3.3 Performing the identification 7

3.3.1 6th order model 7

3.3.2 4th order model 9

4 Controller Design **13**

5 Results and Comparison **15**

6 Conclusion **17**

Bibliography **19**

1
Introduction

In this report, the analysis and control design of a gantry crane setup is discussed. First, a system identification is performed to obtain a model of the system, next two controllers are designed and implemented: PID and LQR. The performance is compared between these controllers and results are discussed.

Write introduction at the end.

Remove blank pages between chapters.

2

System description

The system considered is a gantry crane. This can be modeled as a pendulum hanging from a cart. Figure 2.1 shows the geometrical parameters of the system. The control objective of this project is anti-sway of the container, while tracking the position of the cart. Modeling of such a system has been widely performed in the literature.[1] [3]. In chapter 3, a system identification is performed without the use of equation of motions based on first principles. The resulting model has been used for simulation and control design. Nevertheless,a non-linear model is derived from first principle to give insight in the system dynamics.

The introduction m
be a better place fo
ences.

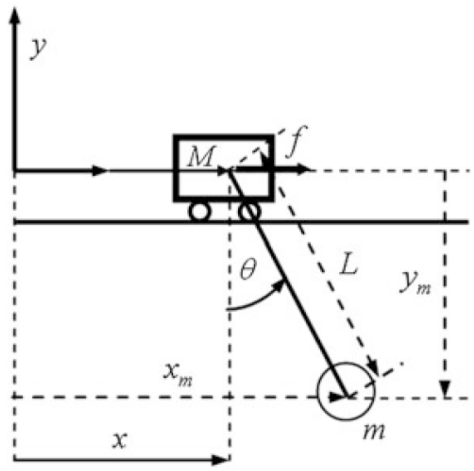


Figure 2.1: Schematic representation of the inverted pendulum.[2]

3

System Identification

In order to achieve accurate simulations, conduct control design and model-based control methods off-line, a well fitted model is required. This can be accomplished several ways. In this section, the process for finding a model is described including the design decisions made.

3.1. Method selection

The domain of system identification provides the following general approaches:

1. **White-box Model**
   The equations of motions derived in chapter 2 are known completely, with accurate parameters. In this case, the system is already identified, the model can be used for control design.

   either derive those tions or change thi

2. **Grey-box Model**
   The equations of motions derived in chapter 2 are used as a framework. These equations depend on a series of unknown or uncertain parameters. An experiment is performed on the setup and input and output data is recorded. The obtained data and the model framework are passed to an optimizer, one possible implementation is `greyest()` from the System identification toolbox in Matlab. Next, the estimated parameters can be retrieved in addition to the corresponding state-space.

   either derive those tions or change thi

3. **Black-box Model**
   Contrary to the Grey-Box variant, Black-Box identification doesn't require said model-framework, it operates without the knowledge of any of the equations of motion or system parameters. The routine `ssest()` estimates a state-space model, using similar input- output data of the form:

   N4SID

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t) \tag{3.1}$$

$$y(t) = Cx(t) + Du(t) + e(t) \tag{3.2}$$

   In addition, the order of the system has to be supplied to the routine. Other options will be discussed in section 3.3.

   refer to singular va next subsection

In this report, the third option, black-box system identification has been chosen and implemented successfully. Arguments for this choice contain the hard-to-model motor jitter which can be relatively easy estimated using this method and the simplicity of the method. Pitfalls of black-box identification is naturally that the estimated state-space is harder to interpret, in case of an unsuccessful identification, it is harder to pinpoint the cause, furthermore a decision has to be made concerning the order, and the measurement has to be conducted more carefully, as false physical characteristics can be easily introduced by faulty measurements.

For these reasons, careful analysis has been performed on the obtained models to create correct interpretation and the order of the system is carefully selected. This process is elaborated on in section 3.3. The remaining part of this section will therefore focus on this method.

3.2. Obtaining data from the set-up

3.2.1. Choosing input signal

In identification, one of the most important parameters to obtain a correct model is a well chosen input signal. The signal has to be chosen such that all modes of the system are excited. Examples of popular signals are

1. **Bernoulli noise**, also known as binary noise,
2. **Chirp Signal**, a sine wave with increasing frequency,
3. **White noise**,
4. **Multisine**, a linear combination of sine waves at different frequencies.

White noise is discarded in a large number of real settings. The reason this is, that high frequency excitation can potentially have a destructive effect on the setup. Furthermore, the multisine is not considered, as it only excites a fixed amount of frequencies. Naturally, the first signal considered is the Bernoulli noise.

To avoid the problem of having a too high frequencies in the Bernoulli noise, a new input signal has been constructed. This signal and the result of the response of the real system can be inspected in figure 3.1.

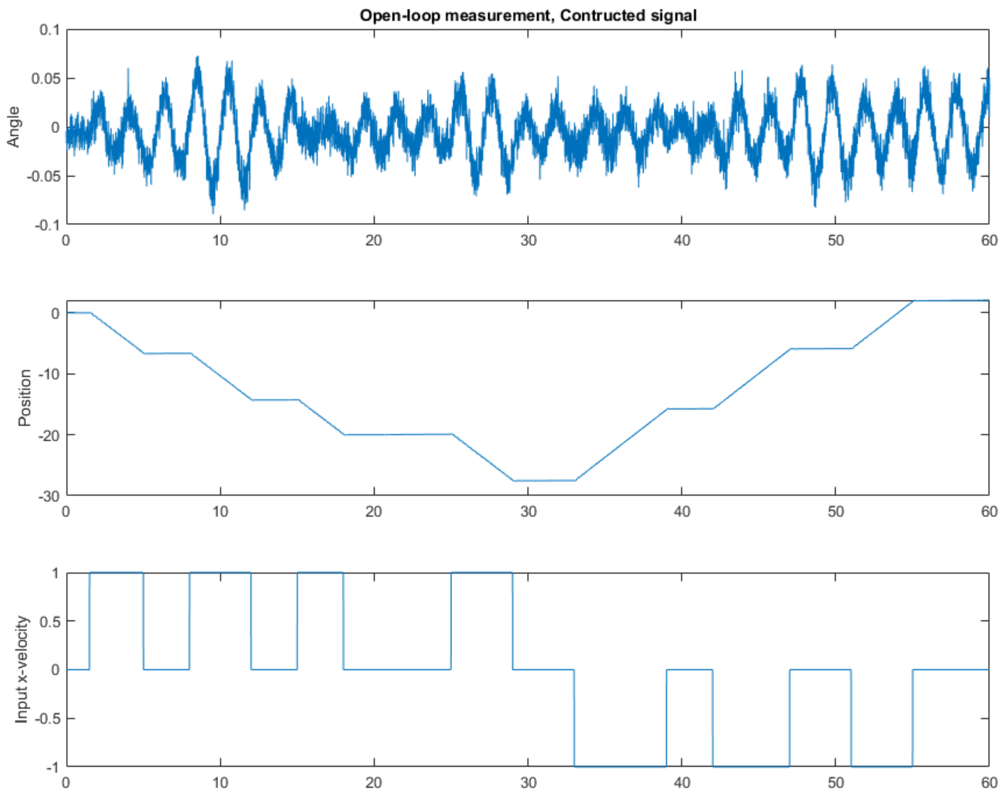


Figure 3.1: Open loop measurement for system identification. A hand constructed input-signal is used.

Post-processing of the data

To obtain better identification, the data is post-processed. The offset of the angle is removed by detrending the angle measurement.

3.2.2. Validation data

To test how well the identified model generalizes, a validation dataset is obtained using the joystick of the setup. This signal and the corresponding response can be observed in figure 3.2

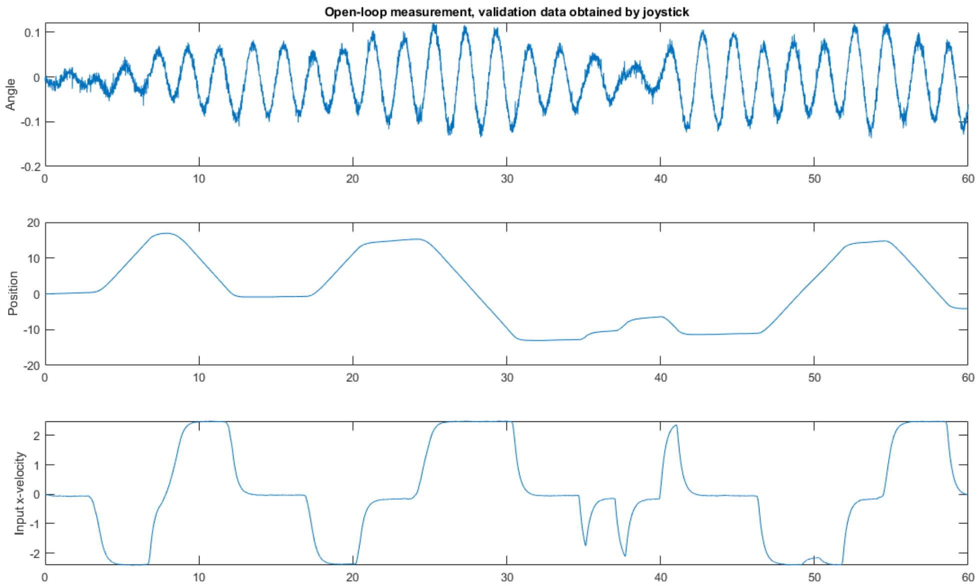


Figure 3.2: Open loop measurement for system validation. A joystick constructed input-signal is used.

create higher resol
plot

3.3. Performing the identification

By analyzing the system, during the measurements, intuition is created for some physical characteristics of the system. The modes that need to be recognized in the identified model are:

1. A pure integrator
2. a negative real pole corresponding to the damping ratio of the system.
3. A complex pole pair to model the natural frequency of the pendulum. (appr 0.5 Hz)
4. (Optional) A complex pole pair to model the jitter in the motor. (appr. 3 Hz)

factcheck this, I'm
if it works like this

Based on these remarks, two separate models are identified, which will be tested later, to see which one describes the real system more accurately; a 4th order model excluding motor jitter, and a 6th order model including the 3 Hz jitter. The identification has been implemented in matlab with the routine called `ssest` as described earlier.

3.3.1. 6th order model

A 6th order black-box identification is performed using the input signal described in the previous subsection. The input is fed into the obtained model to generate a simulated response. Then, this response is compared to the measurements, and a fit-score is generated. The result is shown in figure 3.3. It is worth mentioning that the model is retrieved with the original measurement data, but in the plot is compared against a low-pass filtered version. This is done to get a more trustworthy performance metric i.e. that the score is not artificially high or low due to the noise.

optional:Explain th
mula for this fit-m

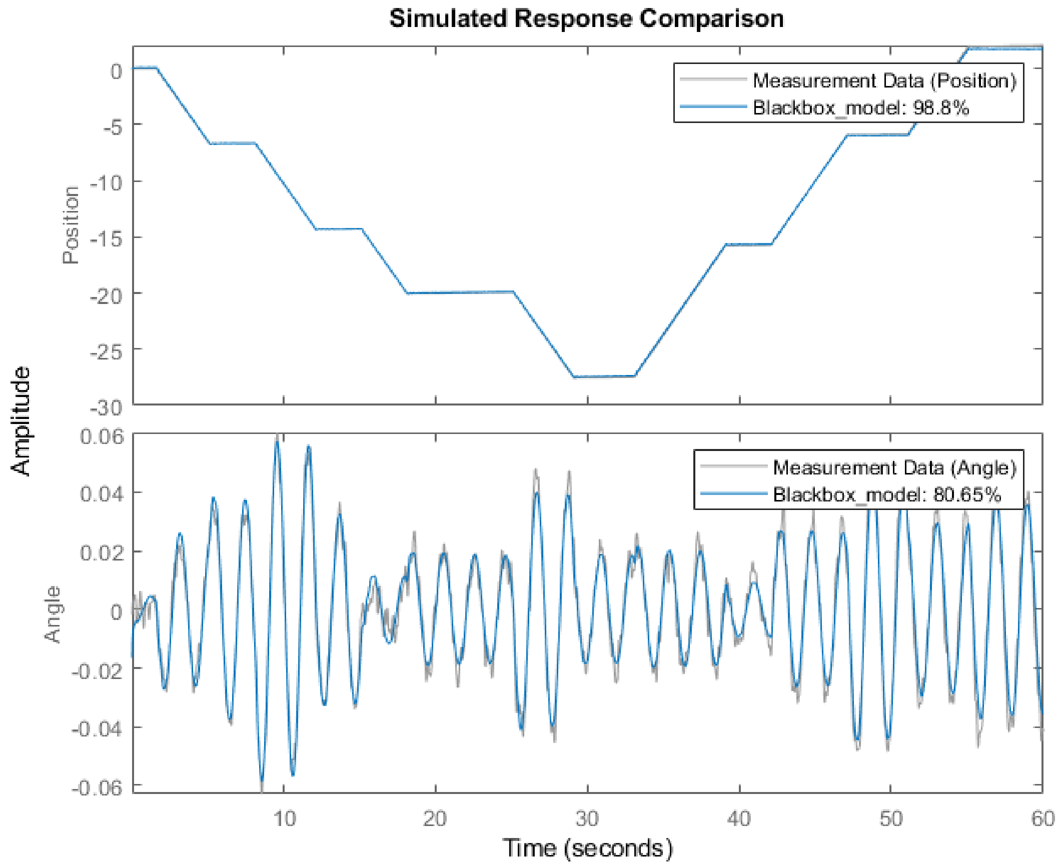


Figure 3.3: insertCaption

gure to new best

This model achieves a fit of 98.8% and 80.65% for the position and angle respectively. However, since this data was what the algorithm optimized for, such a good fit might be due to overfitting, leading to a model, which may not generalize well. It is therefore necessary to check the model by using the before mentioned validation dataset. This input signal is generated by hand using the joystick, the fit can be seen in figure 3.4

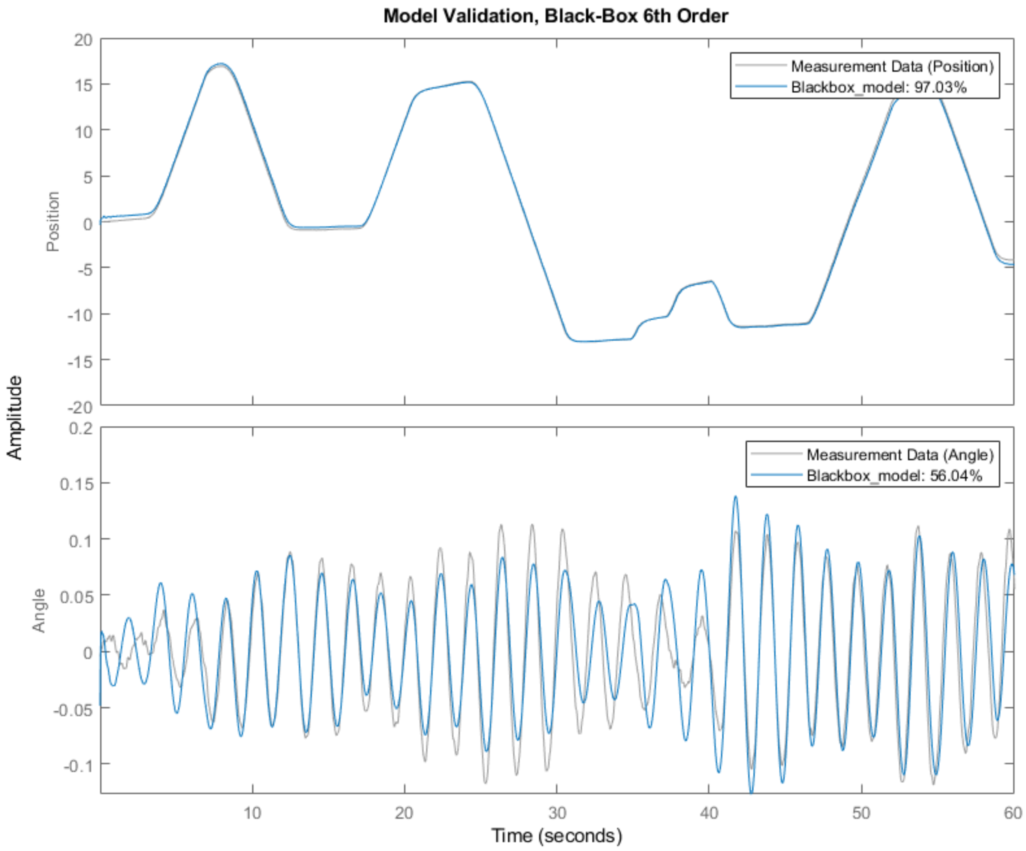


Figure 3.4: The 6th order model performing on a validation input signal.

As anticipated, the fit on the angle has decreased. The fits are now 97,03% and 56,04% for the position and angle respectively.

change to next bes

Add ZPK analysis

3.3.2. 4th order model

When given the choice, a lower order model is preferred for both overfitting and computational resource reasons. It is therefore worthwhile to investigate whether a good fit can be achieved with a 4th order model. When the same estimation input sequence is applied for the black-box estimation however, a bad fit is obtained as seen in figure 3.5.

it doesn't really bec
more comp heavy,

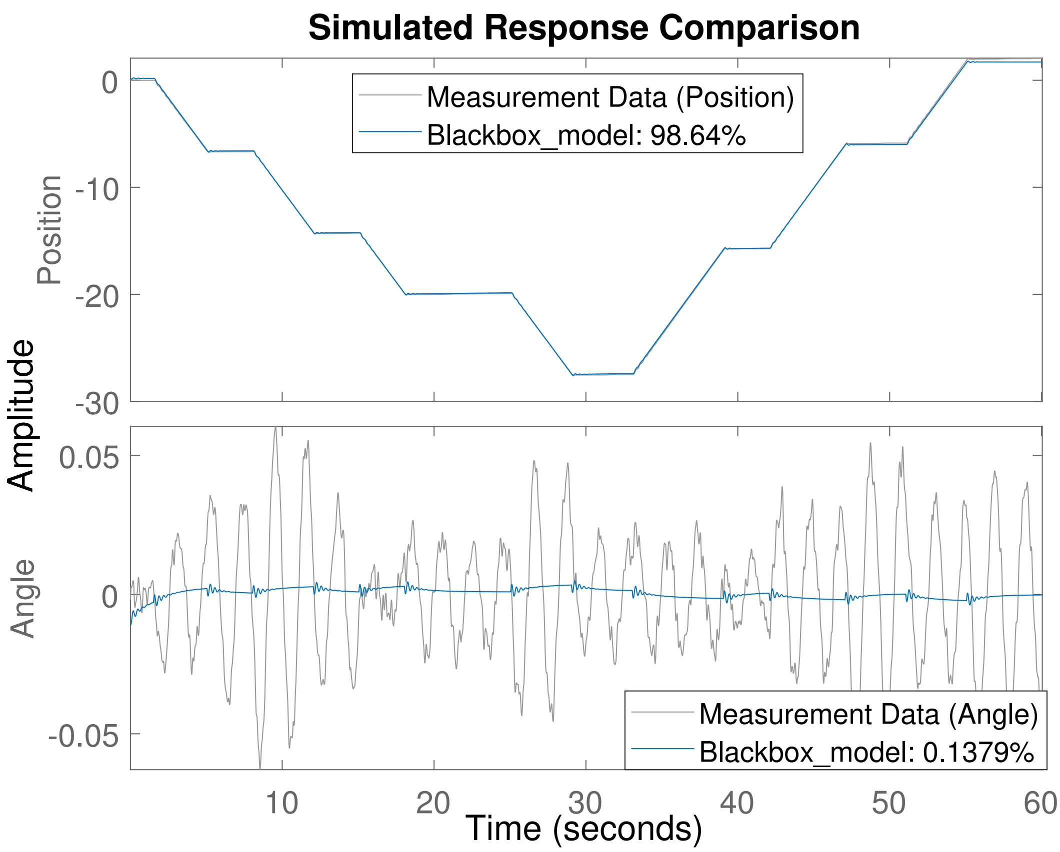


Figure 3.5: insertCaption

The problem can be found when looking at the zero-pole-gain (ZPK) factorization of the transfer function from the input to the angle:

$$\frac{\theta}{u} = \frac{0.067644(s+0.04041)(s^2+0.2315s+11.04)}{(s+0.7225)(s+0.0007846)(s^2+5.078s+439.4)} \tag{3.3}$$

The poles of the system can be interpreted as:

1. A negative real pole corresponding to the damping.
2. A pole very close to zero acting as an approximation of a pure integrator.
3. A complex pole pair corresponding to $\frac{\sqrt{439.4}}{2\pi} = 3.34$ Hz.

This last pole-pair is the problem. The identification procedure tried to fit the motor-jitter and viewed it as a dominant frequency, based on the fitting function. Since the error caused by the absence of this frequency would have resulted in a higher fitting error overall, the natural frequency of 0.5 Hz is therefore not modeled and causes the poor fit regarding the angle. This is absolutely unacceptable for an anti-sway controller.

Through output weighting the optimization function, we are able to force the identification to include the natural frequency and neglect the motor-jitter in the 4th order model. With this modification, the following ZPK was obtained:

$$\frac{\theta}{u} = \frac{0.025849(s-41.35)(s-0.3298)(s+0.03457)}{(s+17.2)(s+0.0008959)(s^2-0.01066s+9.596)} \tag{3.4}$$

In this ZPK-factorization the natural frequency is recognized. This will result in a better model which can be observed in figure 3.7

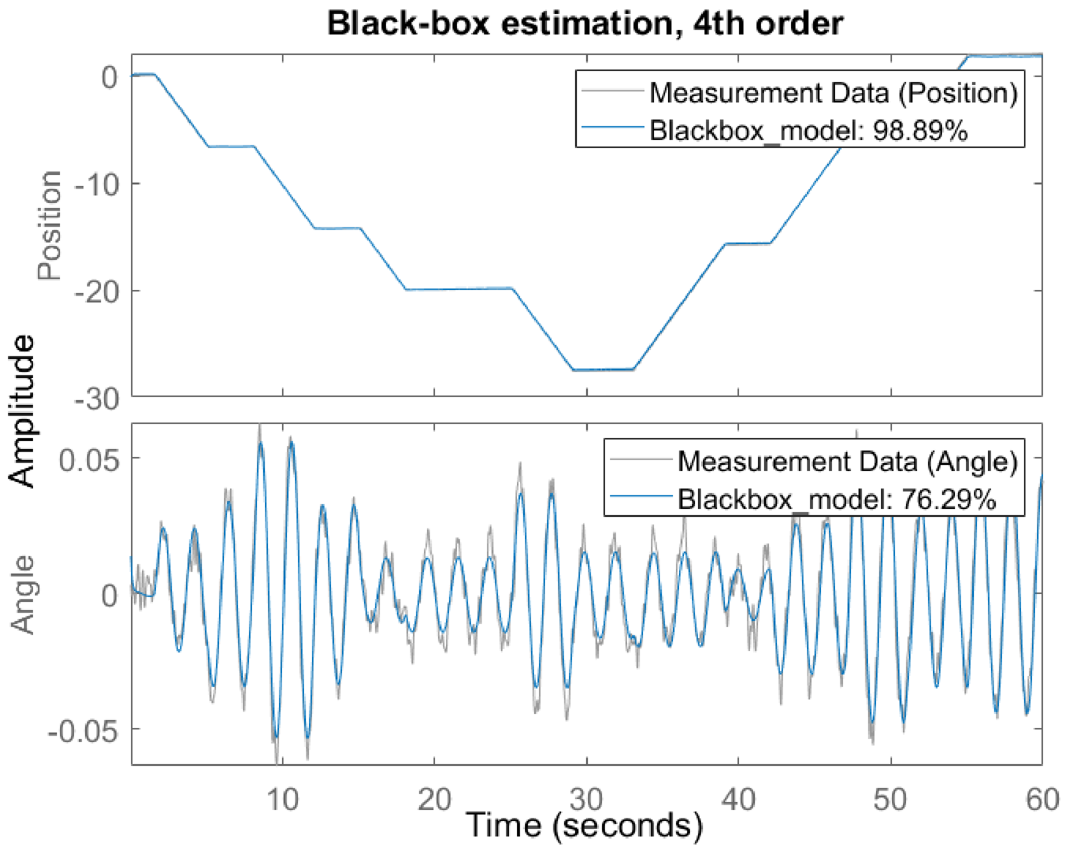


Figure 3.6: Improved 4th order black-box identification. Compared to the filtered estimation data.

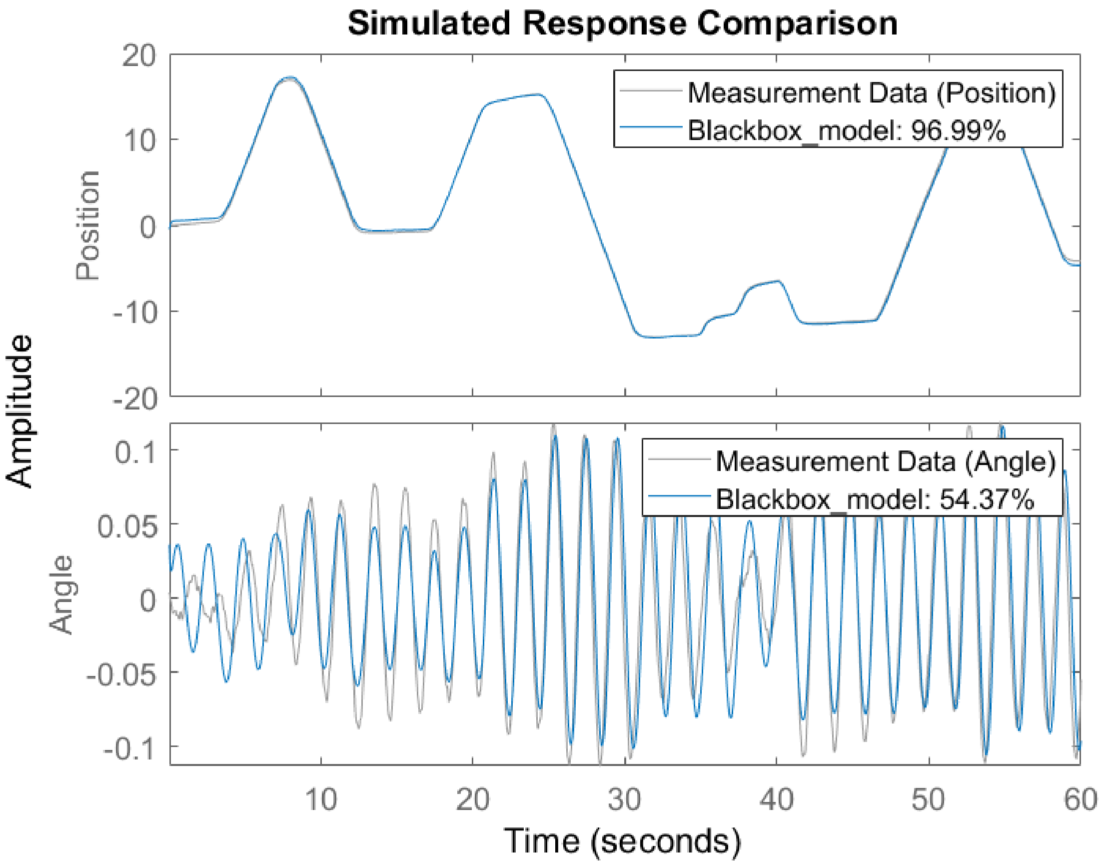
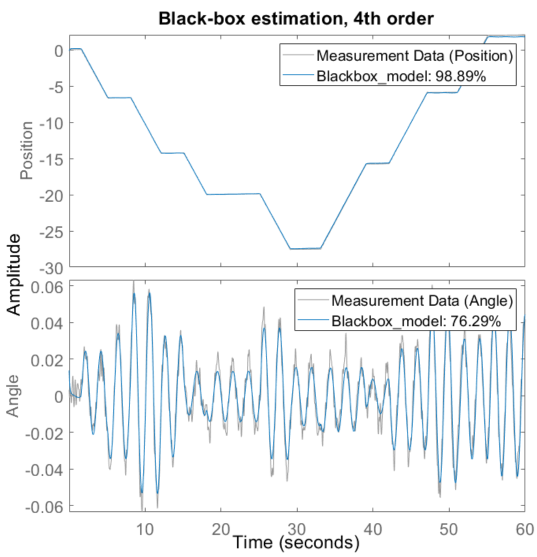
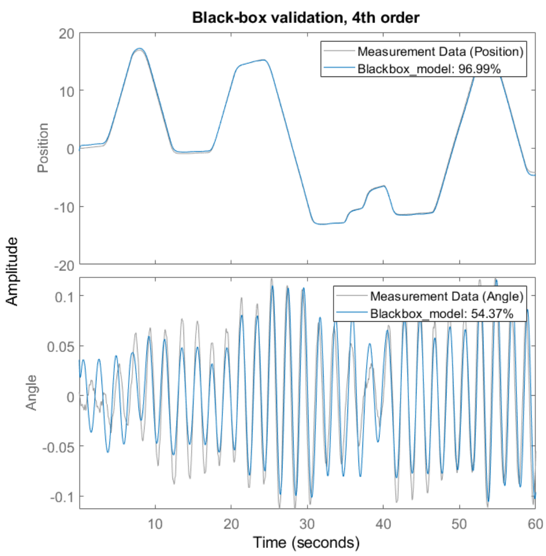


Figure 3.7: 4th order black-box validation. Compared to the filtered validation data.



(a) insertCaption



(b) insertCaption

Figure 3.8: insertCaption

4
Controller Design

5

Results and Comparison

6 Conclusion

Bibliography

[1] S. Bruins. Comparison of different control algorithms for a gantry crane system. *Intelligent Control and Automation, Vol. 1 No. 2, 2010, pp. 68-81*, Vol. 1 No. 2, 2010, pp. 68-81, 2010. URL `https://www.scirp.org/journal/PaperInformation.aspx?PaperID=3142`.

[2] Jianqiang Yi Dianwei Qian. *Hierarchical Sliding Mode Control for Under-actuated Cranes*. Springer-Verlag GmbH, 2015. ISBN 3662484153. URL `https://www.ebook.de/de/product/24789258/dianwei_qian_jianqiang_yi_hierarchical_sliding_mode_control_for_under_actuated_cranes.html`.

[3] Ehsan Omidi. Modeling and nonlinear control of gantry crane using feedback linearization method. *CoRR*, abs/1405.5926, 2014. URL `http://arxiv.org/abs/1405.5926`.