LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
LUT School of Energy Systems
LUT Mechanical Engineering

Mika Sokka

DYNAMIC SIMULATION MODEL AND VIRTUAL SENSOR IMPLEMENTATION FOR ELEVATOR HOISTING MACHINERY

Examiners: Prof. Jussi Sopanen
D.Sc. (Tech.) Janne Heikkinen

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
LUT School of Energy Systems
LUT Mechanical Engineering

Mika Sokka

Hissikoneiston dynaaminen simulaatiomalli ja virtuaalianturi-sovellus

Diplomityö

2018

61 sivua, 25 kuvaa ja 2 taulukkoa

Tarkastajat: Prof. Jussi Sopanen
D.Sc. (Tech.) Janne Heikkinen

Hakusanat: hissikoneisto, digitaalinen kaksonen, dynaaminen simulaatiomalli, virtuaalianturi

Tämän diplomityön tavoitteena oli luoda dynaaminen simulaatiomalli KONE MX-sarjan hissikoneistosta Simulink-ympäristöön. Parametrinen simulaatiomalli voi toimia osana hissin digitaalista kaksosta, johon pyritään peilaamaan kaikki hissin oleelliset ominaisuudet, käyttäytyminen ja data. Mallin tavoitteena oli myös mahdollistaa sellaisten suureiden saaminen ulos Simulink-ympäristöstä, joita ei fyysisillä antureilla voida mitata. Lisäksi mallia tulisi mahdollisuuksien mukaan kyetä simuloimaan reaaliajassa, jotta virtuaaliantureiden ja digitaalisen kaksosen kaikki hyödyt olisivat myöhemmin saavutettavissa.

Malli perustuu yksinkertaistettuun viiden vapausasteen jousi-massa-vaimennin -systeemiin. Vapausasteet ovat staattorin pystysuuntainen liike, staattorin kulma-asennon muutos, roottorin pyörimisliike, hissikorin pystysuuntainen liike sekä vastapainon pystysuuntainen liike. Liikeyhtälöt johdettiin Lagrangen menetelmällä ja mallinnettiin Simulink-ympäristöön, jossa liikeyhtälöiden herätteenä käytettiin roottoriin vaikuttavaa vääntöä. Lisäksi Simulink-malliin tehtiin ohjaus nopeusreferenssin perusteella.

Mallin verifiointi suoritettiin vertaamalla sen vastetta mittauksista saatuun dataan. Mitatut suureet olivat roottorin kulma-asento ja pyörimisnopeus sekä hissikorin paikka ja nopeus. Tulokset olivat mittauksista saadut fyysisen hissin vasteet kolmella eri kuormituksella sekä mallin simuloidut vasteet saman herätteen vaikutuksesta. Lisäksi jokaiselle vasteelle piirrettiin käyrät todellisen virheen suuruudelle referenssistä.

Kokonaisuudessaan Simulink-malli toimi hyvin, liikereferenssin seuraaminen onnistui ja absoluuttinen virhe oli suuruusluokaltaan todella pientä koko liikerataan verrattuna. Lisäksi mallista saatiin ulos myös dataa jota ei fyysisestä hissistä päästä mittaamaan. Simulaatiomallille löytyi myös useita parannus- ja jatkokehitysehdotuksia joiden avulla sitä voidaan kehittää tarkemmaksi ja tiettyihin tarkoituksiin sopivaksi.

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Mika Sokka

Dynamic simulation model and virtual sensor implementation for elevator hoisting machinery

Master's thesis

2018

61 pages, 25 figures and 2 tables

Examiners: Prof. Jussi Sopanen
D.Sc. (Tech.) Janne Heikkinen

Keywords: elevator, hoisting machinery, digital twin, dynamic simulation model, virtual sensor

The goal of this thesis was to create a dynamic simulation model of a KONE MX-series hoisting machinery in Simulink. The parametric simulation model can act as a part of the elevators digital twin, which is aimed to mirror all relevant properties, behavior and data of the elevator. A goal of the model was also to enable the extraction of such quantities from Simulink that cannot be measured with physical sensors. Lastly the model should also, if possible, be able to simulate in real time in order to eventually obtain all benefits that virtual sensors and digital twins have to offer.

Model is based on a simplified five degree-of-freedom mass-spring-damper -system. The degrees of freedom are stator vertical movement, stator rotational movement, rotor rotational movement, elevator car vertical movement and counterweight vertical movement. The equations of motion were derived using Lagrange's method and modelled in Simulink, where the input used for the equations of motion was torque applied to the rotor. Also the input to the Simulink model is in a form of speed reference, in order to enable speed control.

Model verification was performed by comparing its response to measured data. The measured quantities were rotor angle and rotation speed as well as elevator car position and speed. The results were responses obtained from measuring the physical elevator with three different loading cases, as well as simulating the model responses with similar conditions. In addition to that all plots also include actual error from the reference signal.

Overall the Simulink model behaved as expected, it was able to follow the speed reference and the absolute error magnitude was minor compared to the movement trajectory. Also the model was able to output data that could not be measured from the physical elevator. In addition to that several improvement and additional development ideas were found for the model that can be used to improve its accuracy and suitability for various tasks.

ACKNOWLEDGEMENTS

I would like to thank Jussi Sopanen and Janne Heikkinen from Machine Dynamics group in LUT Mechanical Engineering for guidance during this thesis. This work would not have been possible without their expertise in the modelling and simulation process.

I would also like to thank Antti Kärkkäinen and Henri Wenlin from KONE for providing this interesting topic for the thesis, an opportunity to gain insight on the elevator system and conduct the measurements in Hyvinkää, as well as their valuable knowledge on the subject.

Mika Sokka
Lappeenranta 27.11.2018

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

a	Acceleration [m/s^2]
A	Cross-section area [m^2]
c	Damping coefficient [Ns/m]
E	Young’s modulus [GPa]
j	Jerk [m/s^3]
J	Mass moment of inertia [kgm^2]
k	Stiffness [N/m]
k_θ	Torsional stiffness [N/rad]
L	Length [m]
m	mass [kg]
M	Torque [Nm]
q	Generalized coordinate
$\dot{q}$	Generalized velocity
Q	Non-conservative force
T	Kinetic energy
U	Potential energy
x	Translational displacement [m]
v	Velocity [m/s]
θ	Angular displacement [rad]
ρ	Density [kg/m^3]
τ	Time constant [s]
ω	Angular velocity [rad/s]

CAD	Computer-Aided Design
CNC	Computer Numerical Control
CPS	Cyber-Physical System
IoT	Internet of Things
VS	Virtual Sensor

1 INTRODUCTION

Digital twin is a digital model or simulation that replicates a physical counterpart, such as a part, a product, a manufacturing process or a plant. It uses all available relevant data, previously obtained or gathered real-time, to mirror the current state of the physical twin. Digital twins have emerged in the 21st century. The term itself was introduced in 2010 by NASA and further defined in relation to Industry 4.0 and manufacturing in 2013. (Negri, Fumagalli & Macchi 2017)

Digital twins are parts of cyber-physical systems (CPS). CPS consists of the virtual part that is the digital twin, physical part which is the other twin, and layers of sensors and networking to support the operation. A concept image is shown in Figure 1. The benefits of these systems originate from ideally having all parts connected in real-time, allowing them to operate based on each other's data. (Al-Ali, Gupta & Nabulsi 2018)

Since the digital twin is essentially mirroring the state if the current twin, it can be used to introduce different operating conditions that might not be feasible with the physical counterpart, and the responses can be simulated without the need for physical changes to the system. Such analysis can be applied for example in fault injection, lifecycle analysis and predictive maintenance. (Negri, Fumagalli & Macchi 2017)

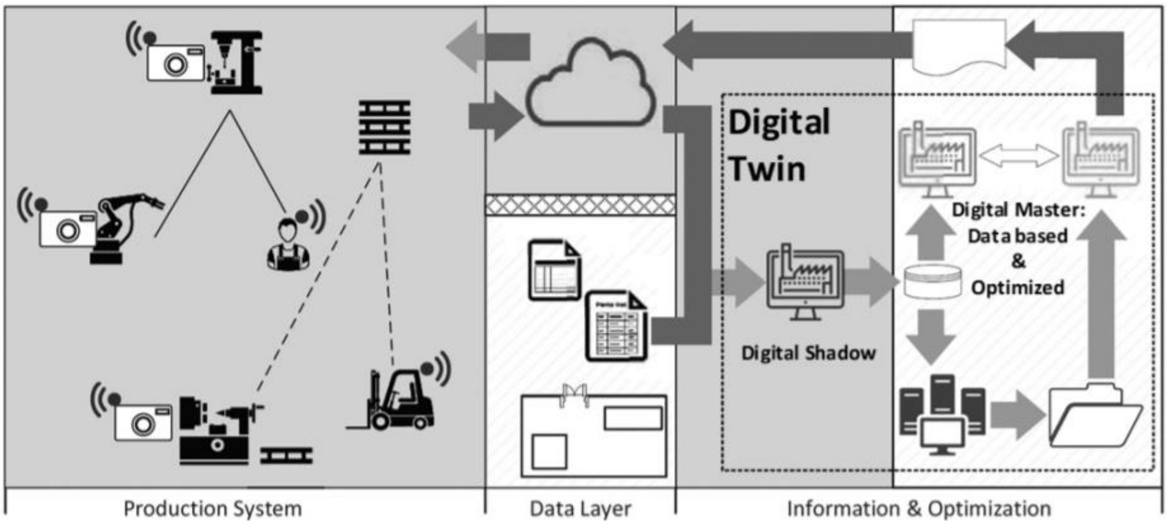
Digital Twin
Digital Master: Data based & Optimized
Digital Shadow
Production System
Data Layer
Information & Optimization

Figure 1. Digital twin in a cyber-physical system concept. (Rodic 2017)

1.1 Background

The motivation for this works comes from KONE, as having a comprehensive digital tool, a sort of a replicate of their product, would be a significant aid in R&D fields such as sensor development, condition monitoring and fault simulation. This would allow them to avoid creating individual hardware-in-the-loop simulations for each component, and instead plug the desired components into a complete twin.

Since the digital twin project at KONE is only in the starting phase, they are experimenting with various different angles that could be used to approach the concept. One of the options is a parametric, dynamic model of an elevator system that can be used to simulate the behavior of various machine configurations, and also include virtual sensors that provide information obtained from the model that is not physically measurable.

For this project the model is constructed in MATLAB/Simulink. This brings several benefits in addition to the software being already in use within KONE. Block diagrams in Simulink allow viewing and plotting of each signal that is present in the model, as well as having real-time capabilities that are desirable when mirroring a physical system. MATLAB and Simulink also have wide interfacing capabilities which allow expanding and connecting the various digital platforms.

1.1.1 Digital twin

Digital twin as a concept has several different descriptions in literature, instead of a single specific definition. Mostly there is an agreement in the literature that the digital twin is a digital object mirrored from an identical physical object. These objects should also have a method of communication that links the data from both objects to each other.

According to Tao et. al. (2017) a generally accepted definition "was given by Glaessegen and Stargel in 2012: digital twin is an integrated multi-physics, multi-scale, probabilistic simulation of a complex product and uses the best available physical models, sensor updates, etc., to mirror the life of its corresponding twin. Meanwhile, digital twin consists of three parts: physical product, virtual product, and connected data that tie the physical and virtual product."

According to Hehenberg & Bradley (2016) “The vision of the Digital Twin itself refers to a comprehensive physical and functional description of a component, product or system, which includes more or less all information *which could be useful* in all lifecycle phases.”

It should be noted that the cyber-physical system and the digital twin within it should focus on only the relevant information. Not all available data, as according to Hehenberg & Bradley (2016) the volume is “huge, diverse and totally unstructed”. The original data can be stored in the existing IT systems and only relevant information should be extracted to the twin. This in turn requires well defined architecture to support data flow. Also lifecycle research often focuses on physical product instead of virtual model. Lack of convergence between the two data sources and isolated as well as fragmented data causes issues with connecting the two. (Hehenberg & Bradley 2016)

Benefits of the digital twin can also be found in literature. Using digital twins in product design can reduce time-to-market as well as provide simulation models, optimization tools and development aid to where they may be needed. For example modifications can be made virtually to check the behavior of new system. (Hehenberg & Bradley 2016)

Interest in digital twins is shown in the increasing number of recent publications studying practical applications and proofs of concept for it in connection with such concepts as Industry 4.0 and Internet of Things (IoT) as well as Computer-Aided Design (CAD). For example Tao et. al. (2018) proposed digital twin -driven prognostics and health management for complex equipment such as wind turbines. Miller, Alvarez & Hartman (2018) suggested extending 3D-CAD models with behavioral data to create a digital twin to achieve properties that are currently hardly available in CAD models. Haag & Anderl (2017) developed a proof of concept for a digital twin by mirroring a beam bending test machine and running tests simultaneously on both the virtual mirror and the physical testing unit. Their digital twin was first created as a CAD representation, then turned functional with multibody and finite element methods to run bending tests virtually. Moreno et. al. (2016) created a virtualization process, functioning like a basic digital twin, of a commercial sheet metal punching machine. They conclude that the twin could be utilized as a design tool of optimal Computer Numerical Control (CNC) programs.

1.1.2 Virtual/soft sensors

According to Shenghui et. al. (2011) “Virtual Sensors (VS) are software algorithms which exploit a set of available measurements to compute an estimate of a physical quantity of interest”. Virtual sensors are often called in literature also as soft sensors, referencing to using software- instead of hardware-based sensors (Fortuna et. al. 2007).

Virtual sensor can be implemented as either an additional real sensor, or only as a validation method for the real sensors. Difference being that as an additional sensor, the virtual sensor is included in the vote when determining which sensor value is closest to the actual, and can be latched in case of a failure in it. Problem with this is that as the virtual sensor might not be as accurate as the real ones, some false alarms can be generated from it in case of sudden changes in operation. (Oosterom & Babuska 2000)

Virtual sensors can be useful in two ways. Firstly they are used to replace hardware sensors to decrease both weight and hardware costs of the application. Secondly they can provide additional data compared to a physical sensor which often measures a single parameter. Virtual sensor can use the data it monitors primarily to observe other parameters derived from the original data as well.

Virtual sensors have been researched and applied in various fields. Inamdar, Allemang & Phillips (2016) suggested that they can be used for estimating frequency and damping in a rotating system. Gutierrez et. al. (2016) studied virtual sensors for determining frequency response, deflection and stress in a flexible structure. Shenghui et. al. (2011) developed a sensor for evaluating vehicle sideslip angle without direct measurement. Oosterom & Babuska (2000) used a fuzzy model approach for fault detection and isolation in aircraft sensors and Fortuna et. al. (2007) describes several industrial virtual sensor applications, with tasks such as observation of emission levels, tank contents, product quality, chemical concentrations, thermal and mechanical stresses as well as line densities.

1.1.3 Real-time simulation

Running simulations in real time means that each time step of the simulation is executed at the same time as it happens in reality, in equal duration. In such case the simulation progresses at a constant, predictable rate, meaning that results of a real-time simulation are evaluated based on not only what results were obtained and how, but also when (TimeSys 2002). This also means that the simulated system can be run in parallel with for example a physical system, and both can be observed simultaneously.

The challenges of real-time simulation arise from the purpose that the simulation speed needs to be optimized in a way that it would match the application in real time. This can be achieved by for example choosing appropriate integration settings to speed up simulating of the model, or running the simulation with powerful enough hardware if available. (Bishop 2008, p. 2-13)

1.1.4 Elevator hoisting machinery dynamics and specifications

Previous research found in literature on the topic of elevators has focused primarily on either the drive modelling aspects such as electric drive torque, current, voltage and vector control, or the behavior and even teaching of dynamics of the elevators cart and the hoisting ropes (Esteban et. al. 2016; Al-Sharif et. al. 2014; Arrasate et. al. 2013, Sul 2011), with less emphasis given to modelling of the dynamics in the middle of these, the hoisting machinery.

This thesis focuses on KONE MX-series hoisting machines. The motors in these hoisting machines are axial flux machines. An axial flux machine is a variation of a rotating electric motor. It has a stator and a rotor centered around the same axis. The stator is fixed to the structure. Rotor is supported by a single bearing, and it also has the traction sheave attached to it, as well as an encoder for measuring rotation angle and rotation speed of the motor. In this research relevant sensors that are found in the machinery are digital encoders that provide position and velocity data from both motor axis as well as elevator car.

1.2 Goals

The goals of this thesis were drafted in cooperation with KONE in order to ensure that the final result matches their needs. The purpose of this thesis is to create a dynamic model of an elevator hoisting machinery that could be used for simulating (possibly in real-time depending on sensor data acquisition) the dynamic response of the system. It would also serve as a starting point for a digital twin that can be built on and improved over time. The model is created in MATLAB/Simulink to take advantage of the various computational benefits and ease of integration of other systems, as well as allow running the simulation on any computer that has MATLAB. Simulink also offers real-time functionality if at some point that is desired from the model.

1.2.1 Research problem

The current problem is that KONE is lacking a dynamic model of their elevator system. While they have various models of several parts of their system, a Simulink model would act as a building part for comprehensive digital twin that can include all relevant aspects and data required in the R&D process. Also literature research suggests that there are very few dynamic models of the hoisting machinery or articles researching them, while research has been conducted with electric drives, hoisting ropes and elevator carts.

The main problem of this research is dynamic modelling of the elevator system, with focus on the hoisting machinery. Another problem is to include requirements from the digital twin concept so that the model is compatible with them. Next problem is the implementation of virtual sensors into the model, and lastly it is important to note the most significant phenomena that must be accounted in such system, also when simulating in real time.

1.2.2 Research questions

This thesis aims to answer the following questions. How does a dynamic Simulink model of an elevator hoisting machinery behave when compared to a physical system? Can it be made accurate and comprehensive enough to aid research and development, and is it possible to implement virtual sensors into the model to obtain relevant data? And finally can the model be simulated in real time?

1.2.3 Hypothesis

Hypothesis of this research is that as a result a functioning dynamic simulation model is obtained in Simulink and it can be verified with reasonable accuracy. Also this research should result in suggestions on how to improve and expand on the model to make it more comprehensive.

1.3 Scope

The scope of this thesis is limited to creating a fairly simple dynamic model of the elevator hoisting machinery that can later be expanded into a digital twin. The modelling is done in

Simulink. The model is verified using experimental data that is collected from measurements during this research. Specific product details such as hoisting rope properties and braking dynamics are excluded from the research.

2 METHODS FOR MODELLING AND MEASUREMENTS

This chapter focuses on first deriving the equations of motion for the hoisting machinery, implementing them in MATLAB/Simulink and finally the methods used for verification of the simulation model. The physical system must be analyzed in order to form the equations of motion which can be used for simulating its behavior. The equations of motion can be utilized in figuring out the systems response, for example with modal-, time domain- or frequency domain analysis. An overview picture of an example KONE axial flux machine is shown in Figure 2 and an exploded view with example construction is shown in Figure 3.

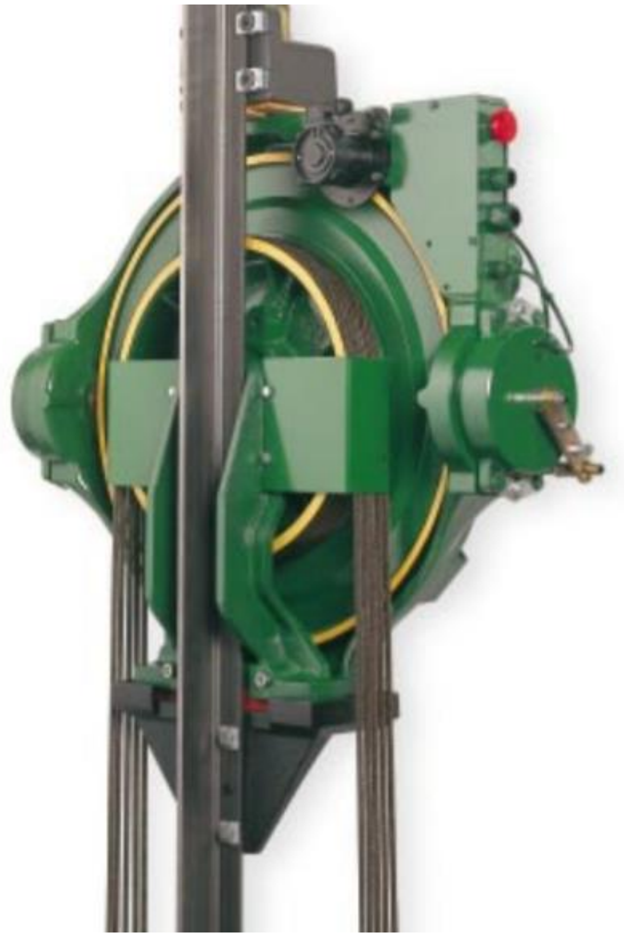

Figure 2. Overview of the system. (KONE 2001)

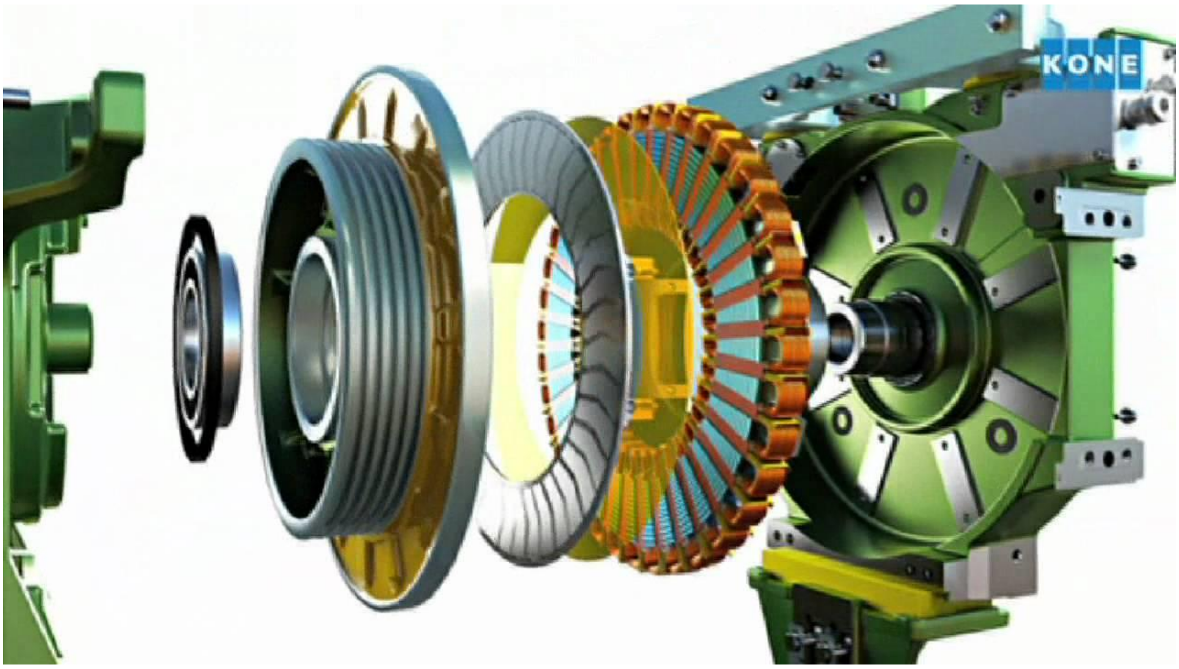

Figure 3. Exploded view of the system. (KONE 2001)

2.1 Physical system

The system itself consists of four parts that are relevant in observing the dynamic behavior. Firstly there is a frame that has the stator attached to it, this joint is assumed to be fixed and the whole frame-stator pair is assumed to be a single body that is supported by rubber pads providing stiffness and damping into the part. The rotor is supported on a shaft by a bearing, and an electromagnetic torque is applied to it by the stator in order to rotate it. Hoisting ropes are wound around the traction sheave that is fixed on the rotor, and one end of the rope moves the cart and the other end moves the counterweight. In a 1:1 roping scheme the rope is directly attached to the cart on one side of the sheave, and the same setup is done to the counterweight on the other side. Figure 4 shows a schematic diagram of the system.

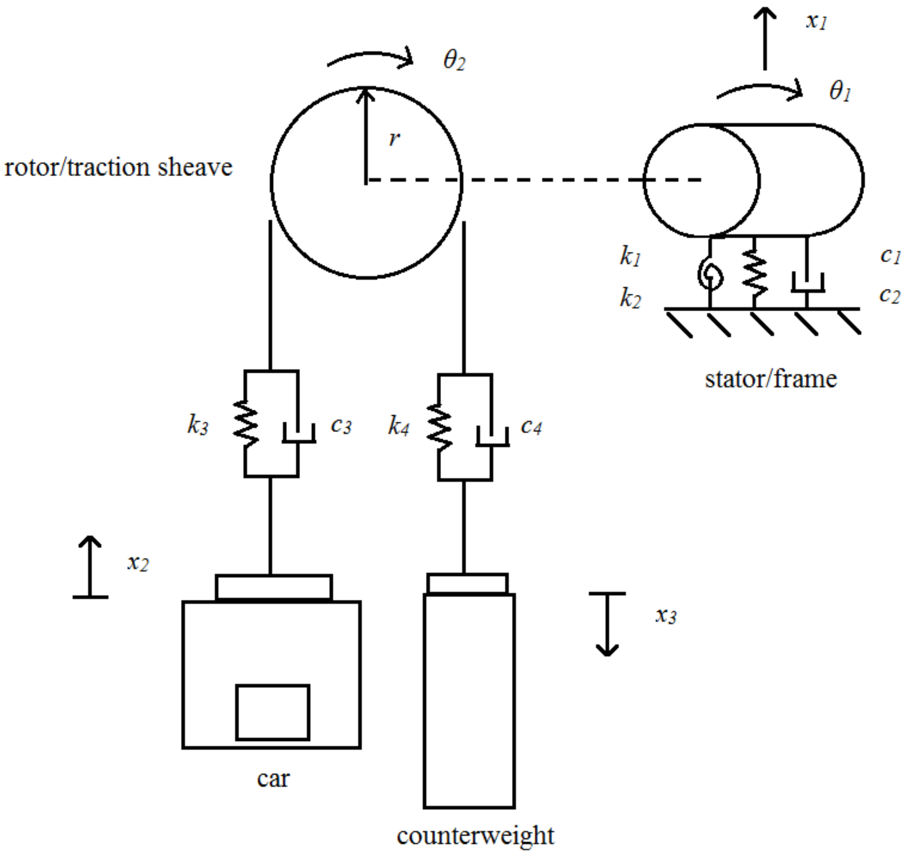

Figure 4. Schematic diagram of the modelled system.

2.2 Modelling

The hoisting machinery in this case can be simplified into a five degree-of-freedom spring-mass-damper system. Inman (2001) describes two ways of deriving the equations of motion for multiple degree-of-freedom systems, Newton's method which is a force-based method (p. 24) and Lagrange's method which is an energy-based method (p. 300). Lagrange's method is suitable for systems consisting of both translational and rotational generalized coordinates, and as such is also applied here in the case of hoisting machinery.

The five chosen DOFs are vertical movement of the stator/frame combination (x_1), rotational movement of the stator/frame (θ_1), rotation of the rotor/sheave combination (θ_2), vertical movement of the elevator cart (x_2) and vertical movement of the counterweight (x_3). These will be represented by the five generalized coordinates q_i.

Simplifications are made into the system to reduce unnecessary information. Elevator ropes are modelled as having stiffness, as well as minor damping in order to ease simulating the system, and the rope masses are lumped into single points on the cart and the counterweight. The bearing connecting the rotor to main shaft is assumed rigid. Friction between rope and traction sheave is assumed to be large enough that no relative movement between them occurs.

Cart and counterweight location should be taken into account when modelling rope properties, since the stiffness and mass of the rope on each side depends on the rope length. It can be assumed that there is a linear relationship between rope length and mass. In order to simplify the model the stiffness of the rope is calculated with the equation

$$k=\frac{EA}{L} \tag{1}$$

where

k	stiffness of the rope
E	Young’s modulus
A	cross-section area of the rope
L	length of the rope.

2.2.1 Equations of motion

Lagrange’s method is an energy-based method, and as such the first step is to form equations for systems kinetic and potential energy based on the schematic diagram. The equations are written based on generalized coordinates chosen from the system coordinates representing the individual degrees of freedom. It should be noted that as the mechanism is using a 1:1 roping scheme, tangential velocity of the traction sheave, and thus velocity of the rope, is approximately equal to that of the vertical velocity of the cart and the counterweight, with small fluctuations occurring due to flexibility of the rope.

Lagrange’s formulation can be written as (Inman 2001, p. 301)

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right)-\frac{\partial T}{\partial q_i}+\frac{\partial U}{\partial q_i}=Q_i,\ i=1,\ 2,\ \ldots,\ n \qquad (2)$$

where

q_i	generalized coordinate
$\dot{q}_i = \partial q_i / \partial t$	generalized velocity
T	kinetic energy of the system
U	potential energy of the system
Q_i	non-conservative forces corresponding to q_i
n	number of generalized coordinates.

Kinetic energy of a translational system can be calculated with the equation (Inman 2001, p. 26)

$$T=\frac{1}{2}m\dot{x}^2 \qquad (3)$$

where

m	mass of the system
$\dot{x} = dx/dt$	velocity of the system

and kinetic energy of a rotating system can be calculated with the equation (Inman 2001, p. 27)

$$T=\frac{1}{2}J\dot{\theta}^2 \qquad (4)$$

where

J	mass moment of inertia of the system
$\dot{\theta} = d\theta/dt$	rotational velocity of the system.

Potential energy of a translational spring can be obtained from the equation (Inman 2001, p. 25)

$$U=\frac{1}{2}kx^2 \tag{5}$$

where

k stiffness of the spring

x displacement from equilibrium

and potential energy of a rotational spring can be obtained from the equation (Inman 2001, p. 28)

$$U=\frac{1}{2}k_\theta\theta^2 \tag{6}$$

where

k_θ torsional stiffness of the spring

θ angular displacement.

Damping force proportional to velocity (Rayleigh's) (Inman 2001, p. 306)

$$F=\frac{1}{2}\sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}\dot{q}_r\dot{q}_s \tag{7}$$

where

c_{rs} damping coefficient.

This can be implemented to Lagrange's formulation as a non-conservative force

$$Q_i=-\frac{\partial F}{\partial \dot{q}_i} \tag{8}$$

In case of the elevator hoisting machinery, five degrees of freedom can be obtained from the schematic diagram. Thus five generalized coordinates are chosen, each representing a single degree of freedom. Following generalized coordinates will be applied in the system under observation:

$q_1 = x_1$	vertical translational displacement of the stator/frame
$q_2 = \theta_1$	rotational displacement of the stator/frame
$q_3 = \theta_2$	rotational displacement of the rotor/traction sheave
$q_4 = x_2$	vertical translational displacement of the elevator cart
$q_5 = x_3$	vertical translational displacement of the counterweight.

Using these generalized coordinates the kinetic energy of the system becomes

$$T=\frac{1}{2}m_1\dot{q}_1^2+\frac{1}{2}J_1\dot{q}_2^2+\frac{1}{2}J_2\dot{q}_3^2+\frac{1}{2}m_2\dot{q}_4^2+\frac{1}{2}m_3\dot{q}_5^2 \qquad (9)$$

where

m_1	mass of the stator/frame
J_1	mass moment of inertia of the stator/frame
J_2	mass moment of inertia of the rotor/sheave
m_2	mass of the cart side
m_3	mass of the counterweight side.

Potential energy of the system becomes

$$U=\frac{1}{2}k_1q_1^2+\frac{1}{2}k_2q_2^2+\frac{1}{2}k_3(q_1+rq_3-q_4)^2+\frac{1}{2}k_4(q_1-rq_3+q_5)^2 \qquad (10)$$

where

k_1	stiffness of the stator/frame support
k_2	rotational stiffness of the stator/frame support
k_3	stiffness of the ropes, cart side
k_4	stiffness of the ropes, counterweight side
r	radius of the traction sheave.

Assuming that the damping forces are proportional to velocities, Rayleigh dissipation function can be used to model them. Based on Inman’s (2001, pp. 306-307) derivation of the damping coefficients and looking at the schematic diagram in Figure 4 to conclude that each of the spring and damper forces affect on the same coordinates, it can be assumed that the stiffness and damping matrix are of the same form and that damping is proportional.

Next step is to calculate the partial derivatives in the Lagrange’s formulation for each generalized coordinate and combine them into matrix form, yielding the following result:

$$\begin{bmatrix} m_1 & 0 & 0 & 0 & 0 \\ 0 & J_1 & 0 & 0 & 0 \\ 0 & 0 & J_2 & 0 & 0 \\ 0 & 0 & 0 & m_{\mathrm{c}}+m_{\mathrm{rc}}+m_{\mathrm{L}} & 0 \\ 0 & 0 & 0 & 0 & m_{\mathrm{cw}}+m_{\mathrm{rcw}} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \\ \ddot{q}_5 \end{bmatrix}$$

$$+\begin{bmatrix} c_1+c_3+c_4 & 0 & r(c_3-c_4) & -c_3 & c_4 \\ 0 & c_2 & 0 & 0 & 0 \\ r(c_3-c_4) & 0 & r^2(c_3+c_4) & -rc_3 & -rc_4 \\ -c_3 & 0 & -rc_3 & c_3 & 0 \\ c_4 & 0 & -rc_4 & 0 & c_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix}$$

$$+\begin{bmatrix} k_1+k_3+k_4 & 0 & r(k_3-k_4) & -k_3 & k_4 \\ 0 & k_2 & 0 & 0 & 0 \\ r(k_3-k_4) & 0 & r^2(k_3+k_4) & -rk_3 & -rk_4 \\ -k_3 & 0 & -rk_3 & k_3 & 0 \\ k_4 & 0 & -rk_4 & 0 & k_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} = \begin{bmatrix} 0 \\ -M-M_{\mathrm{bal}} \\ M+M_{\mathrm{bal}} \\ -m_2 g \\ m_3 g \end{bmatrix} \qquad (11)$$

where the right side of the equation in this case represents the input torque. The parameters of equation (11) are listed in Table 1. The equations of motion can be represented in general matrix form

$$\mathbf{M}\ddot{\boldsymbol{q}}+\mathbf{C}\dot{\boldsymbol{q}}+\mathbf{K}\boldsymbol{x}=\boldsymbol{Q(t)} \qquad (12)$$

2.2.2 MATLAB/Simulink

The Simulink model is based on the schematic in Figure 5. It consists of a block diagram with several specific parts. On the left side is the reference signal. The reference signal is fed into the torque block after it has been connected to the negative feedback loop, i.e. current velocity. The torque block combines static torque M_{bal} calculated from the mass imbalance on both sides of the traction sheave and dynamic torque M calculated from the current velocity error using the equation

$$\dot{M} = \frac{K_m(\omega_{ref} - \omega_{act}) - M}{\tau} \tag{13}$$

where

M	dynamic torque reference
K_m	motor-specific constant
ω_{ref}	speed reference
ω_{act}	actual speed value
τ	time constant of the motor.

The torque reference is fed into the “Equations of motion” -block. This simulates the total torque acting on the rotor. Possible outputs include position, velocity and acceleration of each of the five DOFs, as well as additional information such as motor torque, forces, rope masses and stiffnesses that are solved during the simulation.

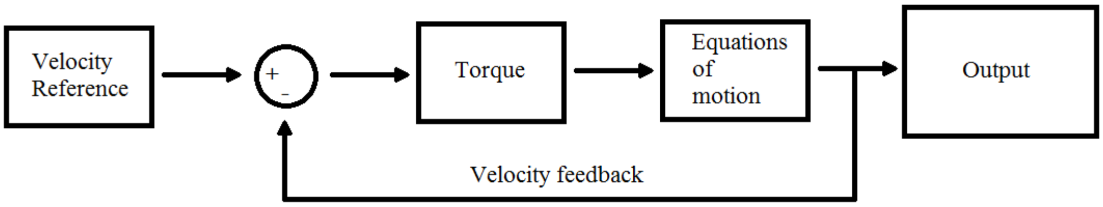

Figure 5. Block diagram schematic of the Simulink model.

In order to run the Simulink model, all parameters must first be specified using either MATLAB workspace or alternatively the file settings in Simulink itself. In this project the constants are loaded into the workspace using a .m-script that contains all relevant data. All variables that are not constants, so changing during the simulation, are wired inside the Simulink block diagram to all necessary locations. Lastly integrator settings are defined in the Simulink file settings. Table 1 lists all parameters used in this model. Simulations are carried out using variable-step ode45 -solver with a simulation time of 50 s.

Table 1. Simulation model parameters.

Parameter	**Value**
Gravity, g [m/s^2]	9.81
Rope density, ρ [kg/m^3]	7800
Young's modulus, E [Pa]	$210 \cdot 10^9$
Rope cross-section area, A [m^2]	$5 \cdot \pi \cdot 0.01^2 = 1.6 \cdot 10^{-3}$
Rope length beyond movement range, L_0 [m]	5
Total movement length, L_{tot} [m]	49.7
Initial car side rope length, L_1 [m]	$L_0 = 5$
Initial counterweight side rope length, L_2 [m]	$L_0 + L_{tot} = 54.7$
Stator+rotor mass, m_1 [kg]	1750+604 = 2354
Stator moment of inertia, J_1 [kgm^2]	175
Rotor moment of inertia, J_2 [kgm^2]	65
Car mass, m_c [kg]	1350
Initial car side rope mass, m_{rc} [kg]	$\rho \cdot A \cdot L_1 = 61.3$
Load mass in three cases, m_L [kg]	0; 320; 665
Counterweight mass, m_{cw} [kg]	2070
Initial counterweight side rope mass, m_{rcw} [kg]	$\rho \cdot A \cdot L_2 = 670.2$
Initial total car side mass, m_2 [kg]	$m_c + m_{rc} + m_L = 1411.3$
Initial total counterweight side mass, m_3 [kg]	$m_{cw} + m_{rcw} = 2740.2$
Traction sheave radius, r [m]	0.2
Spring constant corresponding to q_1, k_1 [N/m]	$1 \cdot 10^{11}$
Spring constant corresponding to q_2, k_2 [N/m]	$1 \cdot 10^8$
Initial car side rope spring constant, k_3 [N/m]	$2 \cdot E \cdot A / L_1 = 6.6 \cdot 10^7$
Initial counterweight side rope spring constant, k_4 [N/m]	$2 \cdot E \cdot A / L_2 = 6.0 \cdot 10^6$
Damping coefficient corresponding to q_1, c_1 [Ns/m]	$0.0001 \cdot k_1 = 1 \cdot 10^7$
Damping coefficient corresponding to q_2, c_2 [Ns/m]	$0.01 \cdot k_2 = 1 \cdot 10^6$
Car side rope damping coefficient, c_3 [Ns/m]	$0.0001 \cdot k_3 = 6.6 \cdot 10^3$
Counterweight side rope damping coefficient, c_4 [Ns/m]	$0.0001 \cdot k_4 = 603.0$
Initial static torque around traction sheave, T_{bal} [Nm]	$r \cdot g\ (m_2 - m_3) = -2.6 \cdot 10^3$
Motor-specific constant, K_m [Nms]	$1 \cdot 10^7$
Motor time constant, τ [s]	$1 \cdot 10^{-3}$

2.3 Model verification

The specific machine that is used in verification measurements is a MX-32. The difference between a simple axial flux machine and this machine is that the MX-32 has two axial flux motors facing each other, connected with the traction sheave. This configuration allows the two rotors to act as one to improve the behavior of this larger machine. The stators connected to the same base can also be assumed to be a single body. A picture of the actual system is shown in figure 6.

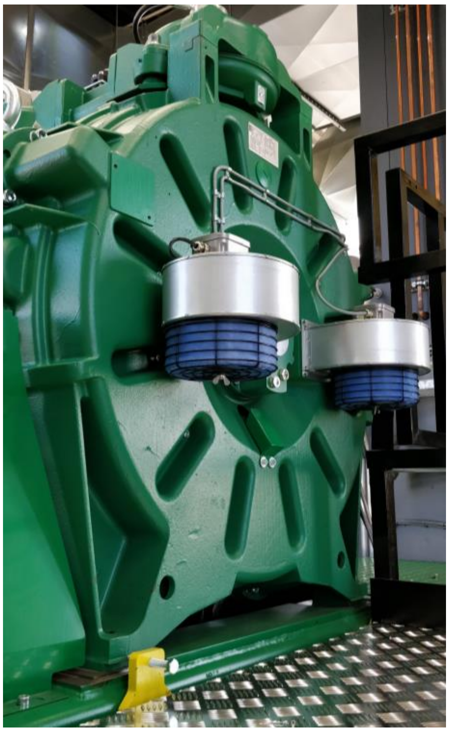

Figure 6. Picture of the actual machine.

Model verification data (position and velocity) is collected from two encoders, one attached to the rotor axle and the other on the top of the elevator car. A total of three different load configurations are tested. Load case 1 corresponds to 0 kg (0 %) load, case 2 to 320 kg (25 %) load and case 3 to 665 kg (48 %) load. Each load case is tested using a specific reference signal five times. Originally measurements are conducted with a sampling rate of 200000 and the data is imported to MATLAB with a sampling time of 0.005 s.

These five data sets are averaged for each load case. This is enough to obtain usable position data, however the velocity signal has so much noise that finally the average velocity data sets for each of the three load cases are filtered in MATLAB using one dimensional 100thorder median filter to remove largest disturbance spikes. This corresponds to median over a time frame of 0.5 s.

These data sets are plotted against response plots obtained from Simulink using the same reference signal as an input for the model. This allows comparison of the model and the physical system response.

The used reference signals are shown in Figure 7. They are obtained by first designing a jerk reference curve that matches the desired travel path of the elevator. This jerk reference is then integrated once for acceleration reference, twice for velocity reference and third time for position reference. Table 2 shows design parameters of the reference signal. These include maximum values for jerk, acceleration, velocity and displacement.

Table 2. Reference signal parameters.

Parameter	**Value**
Maximum jerk, $j_{\max}$ [m/s^3]	1.84
Maximum acceleration, $a_{\max}$ [m/s^2]	1
Maximum velocity, $v_{\max}$ [m/s]	6
Maximum displacement, $L_{\max}$ [m]	49.7

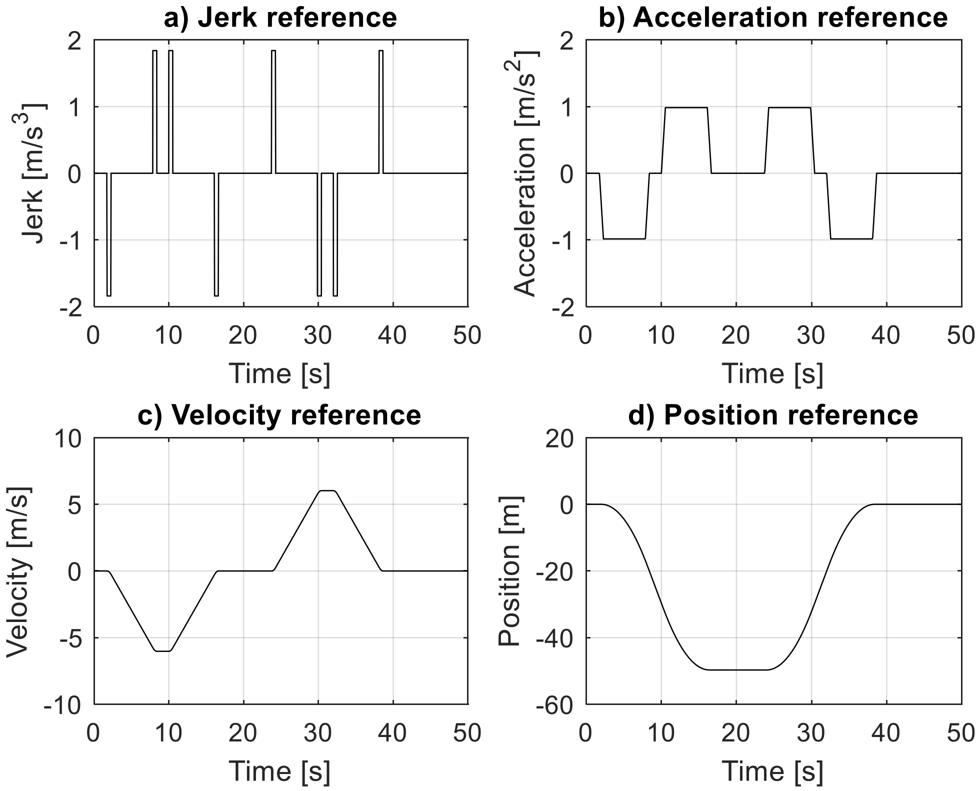

Figure 7. Reference signals.

Figure 7 shows that in order to achieve one displacement cycle of moving from zero to a reference value and then back to zero, it requires two velocity cycles in opposite directions. These two velocity cycles then require four acceleration cycles, one for each moment in time where velocity changes, meaning the derivative of velocity i.e. acceleration, is not zero, and specifically in the same direction as the change. Based on this the derivative of acceleration signal, jerk, will require eight cycles in order to achieve the desired acceleration, velocity and therefore position signal.

3 RESULTS AND SYSTEM RESPONSES

This chapter introduces results obtained from both the simulation model as well as the experimental measurements. The corresponding results are plotted in the same graphs so that they can be compared. Also, for dynamic responses, in each figure the error from reference signal is plotted for both measured and simulated signals.

Figure 8 shows the static position response of the rotor in the simulation model, obtained by keeping the input signal at zero and plotting the response of the model. Rotor angle changes linearly from zero to -6·10^{-3} radians. Average velocity steady state error is 1.2·10^{-4} rad/s. Velocity response oscillates with an amplitude of 2.7·10^{-3} rad/s at the beginning of the simulation and decays almost completely during the first 15 seconds.

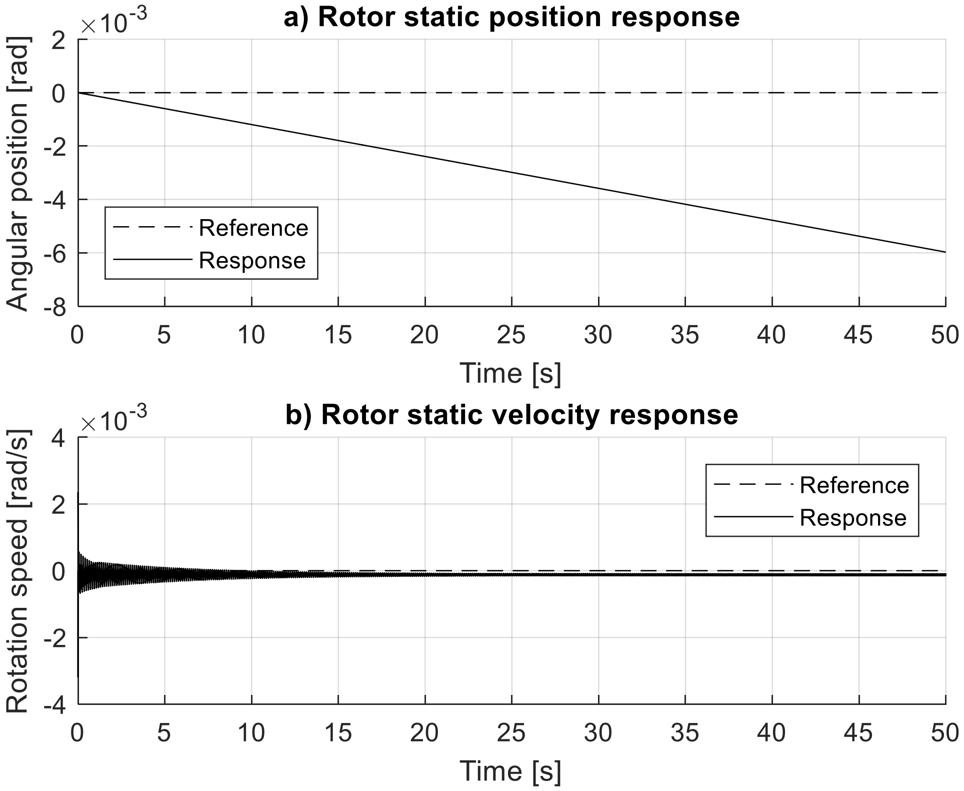

Figure 8. Static response of the model. a) Position response. b) Velocity response.

3.1 Loading case 1, no additional mass

Figures 9 to 12 present the measured and simulated responses with 0 % loading (0 kg) as well as corresponding error signals. Figure 9 shows the measured and simulated rotor rotations when the elevator car was first driven from 0 to -49.7 meters and then driven back to the starting point. This corresponds to rotating the rotor from 0 to -248.5 radians and back as shown in Figure 9a.

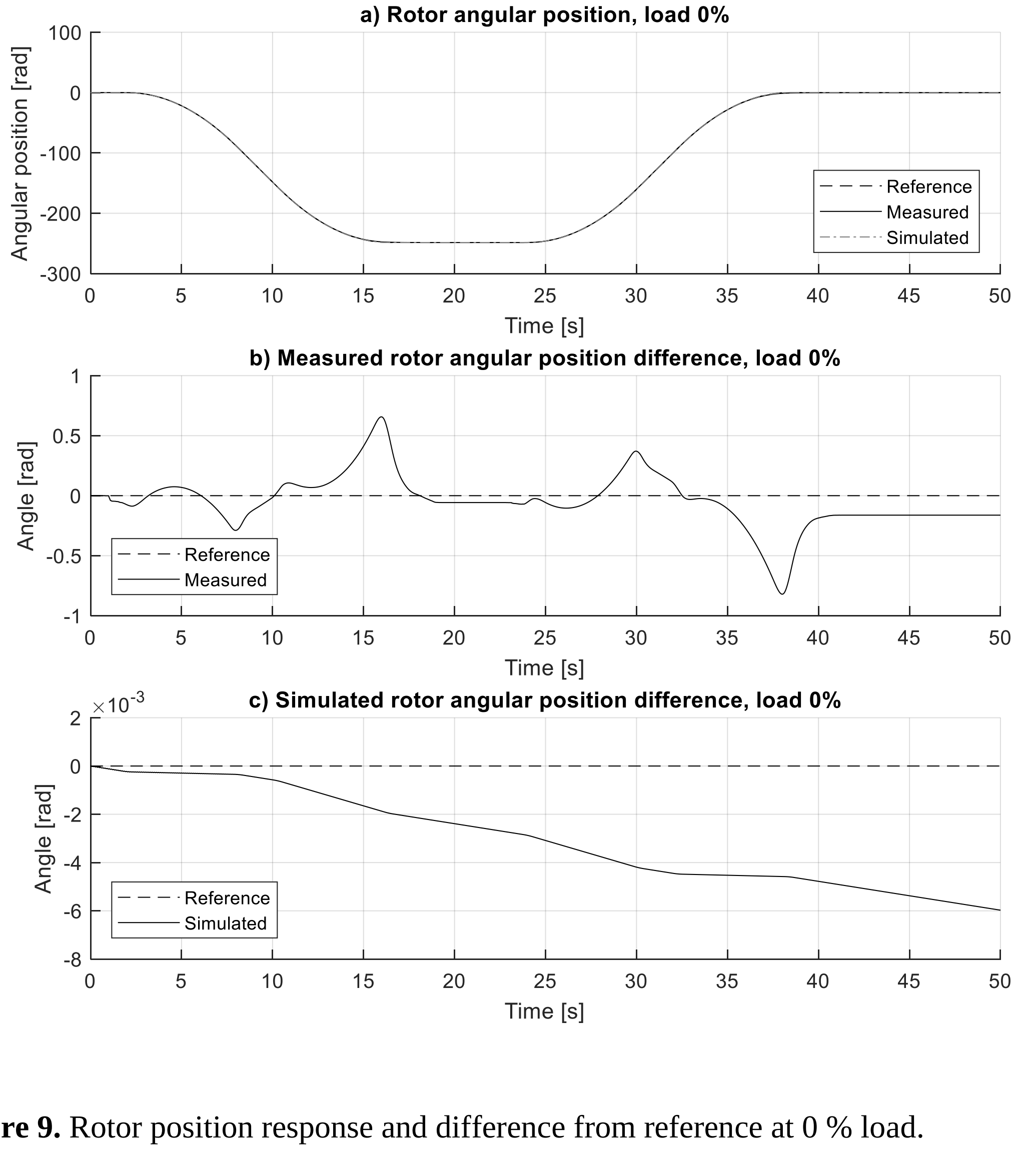

Figure 9. Rotor position response and difference from reference at 0 % load.

During the run largest measured errors occurred during rotor acceleration and deceleration as shown in Figure 9b. The measured signal is leading the reference during acceleration and lagging during deceleration. Starting from zero the rotor angle difference first fluctuates slightly before reaching a local minimum of -0.29 radians at the end of the first acceleration phase. Then the difference progresses to opposite direction and reaches a global maximum of 0.66 radians at the end of decelerating to the desired reference point, after which it returns to a steady state. While at reference the steady state error is at -0.06 radians. While the elevator is moved back to the starting point the measured rotor angle difference reaches a local maximum of 0.37 radians after the accelerating phase and gradually progresses to global minimum of -0.82 radians at the end of the deceleration phase before returning to a steady state. After the entire movement cycle the rotor steady state error is -0.16 radians.

During the simulation the error accumulates over the run, with the error reaching a maximum of $6 \cdot 10^{-3}$ radians after the simulation is complete as shown in Figure 9c. While the overall shape of the simulated error graph is relatively straight, some dynamic behavior can be seen in the plot. At the first phase of the movement cycle, during acceleration the simulated angle is lagging behind the reference and the difference progresses slower, whereas during deceleration the difference progresses faster. While system returns to the starting point at the second phase, these effects behave in the opposite way.

Figure 10a shows the measured and simulated car position when the elevator car was first driven from 0 to -49.7 meters and then driven back to the starting point. During the run largest measured errors occurred during car acceleration and deceleration as shown in Figure 10b. The measured signal is leading the reference during acceleration and lagging during deceleration Starting from zero the car position difference first fluctuates slightly before reaching a local minimum of -0.04 meters at the end of the first acceleration phase. Then the difference progresses to opposite direction and reaches a global maximum of 0.13 meters at the end of decelerating to the desired reference point, after which it returns to a steady state. While at reference the steady state error is at -1.66·millimeters. While the elevator is moved back to the starting point the measured car position difference reaches a local maximum of 0.09 meters after the accelerating phase and gradually progresses to global minimum of -0.11 meters at the end of the deceleration phase, and after that the system returns to a steady state again. After the entire movement cycle the car steady state error is -0.02 meters.

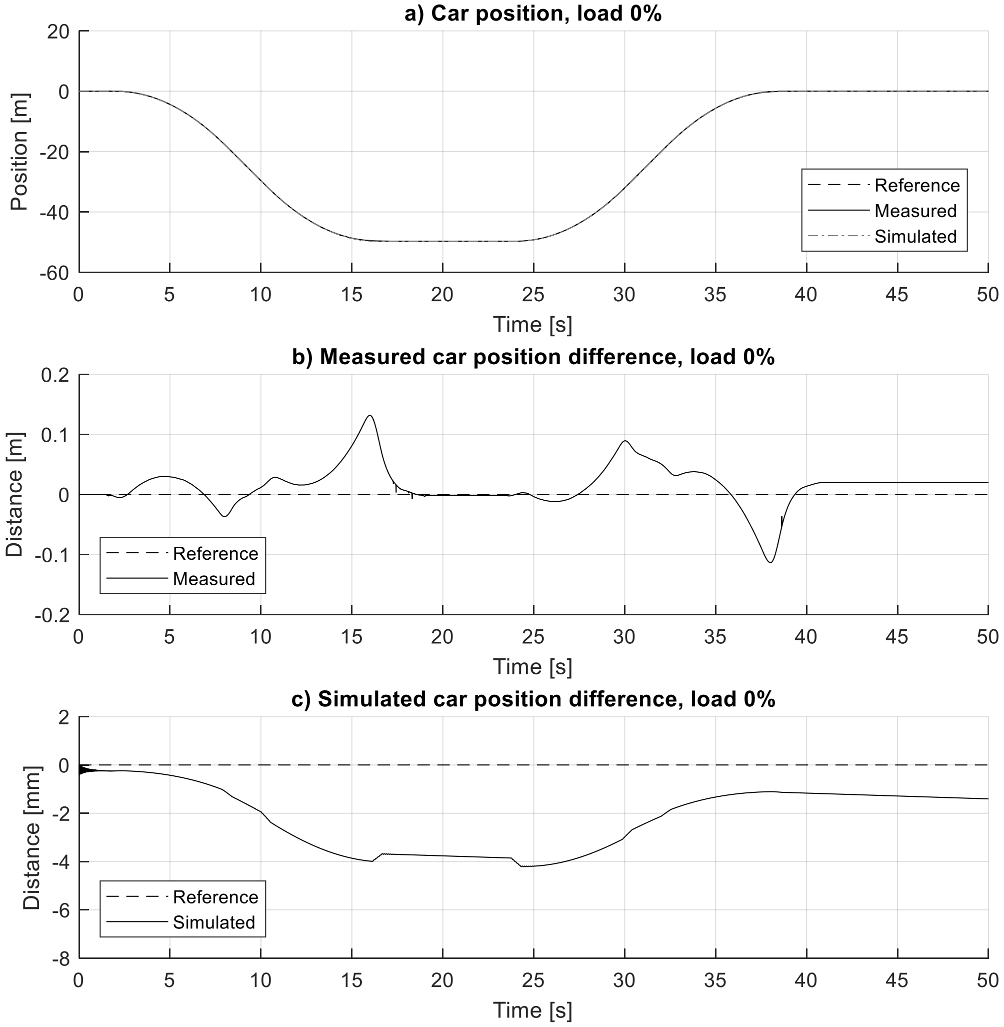

Figure 10. Elevator car position response and error at 0 % load.

The simulated car position error is shown in Figure 10c. At the start of the run the difference oscillates around -2.4·10^{-4} meters with an amplitude of 1.7·10^{-4} meters, however this oscillation decays completely withing the first two seconds of the simulation. After the car begins to move the difference progresses in a similar fashion to the car movement graph, reaching a local minimum of -4.0·millimeters before bouncing back slightly and staying nearly constant for the duration of the reference steady state. When the car begins to move again the simulated difference behaves similarly as while the car was going down, first descending to a global minimum of -4.2·millimeters before progressing to the final position at -1.1·millimeters, exept that the final value is not a constant and instead the steady state error keeps slowly increasing after the full movement cycle.

Rotor rotation speed response is shown in Figure 11, when the elevator car was first driven from 0 to -6 m/s, then back to 0 m/s, then from 0 to 6 m/s and finally back to 0 m/s. This corresponds to rotor rotation speed of -30 rad/s while descending and 30 rad/s while ascending as shown in Figure 11a.

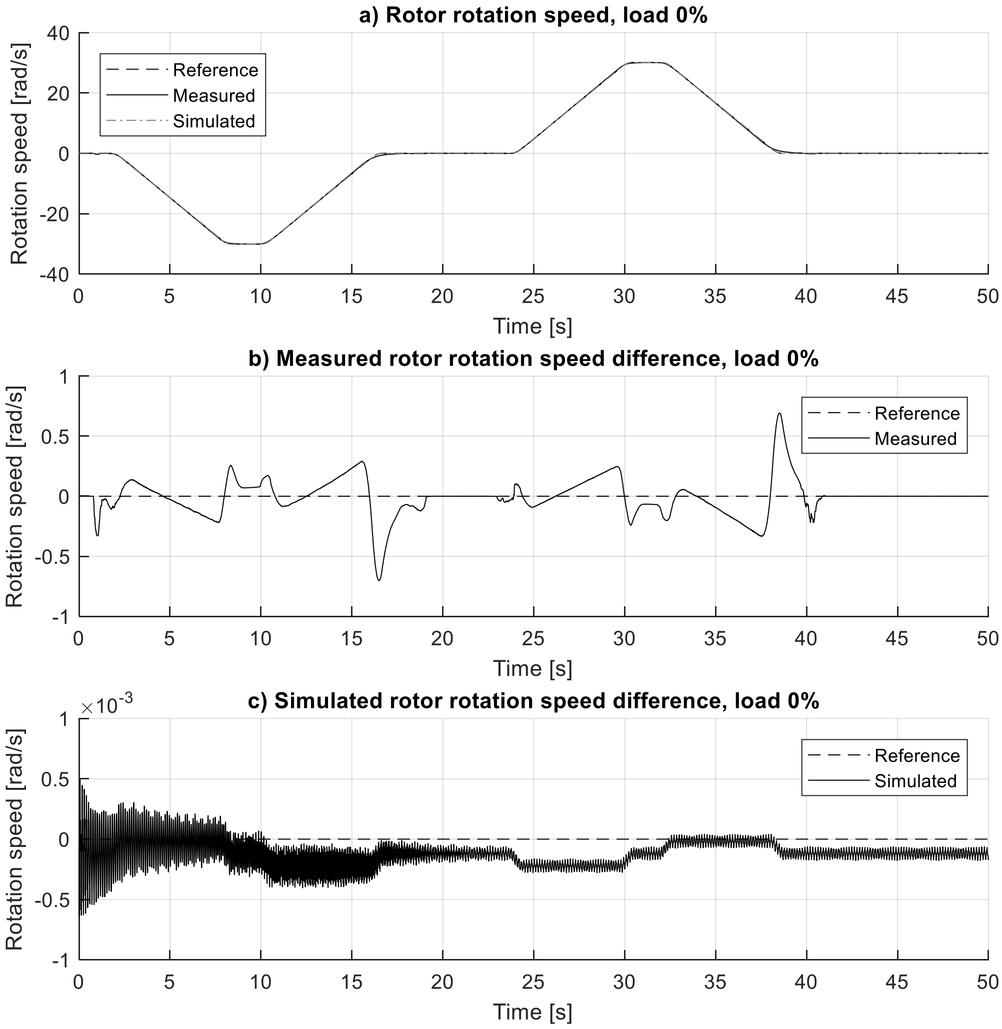

Figure 11. Rotor velocity response and error at 0 % load.

Measured rotor speed difference in Figure 11b shows some minor fluctuation in addition to larger peaks that are associated with beginning and ending of acceleration and deceleration phases, these are also the moments where jerk reference is not equal to zero. The global minimum and maximum values of -0.70 rad/s and 0.69 rad/s are located at times of 16.5

seconds and 38.5 seconds respectively, this is when the measured signal is lagging behind the reference as the rotor position approaches the desired level.

Simulated rotation speed difference shown in Figure 11c has some oscillation during the entire simulation, although the effect is significantly larger at the start and mostly decays before 20 seconds of simulation time. The simulated difference is closer to zero during positive acceleration phases at times of 1.75 seconds and 32 seconds, averaging $1.9 \cdot 10^{-5}$ rad/s, and furthest from zero during negative acceleration phases starting at times of 10 seconds and 23.75 seconds, averaging $2.2 \cdot 10^{-4}$ rad/s. Simulated rotation speed steady state error is $1.2 \cdot 10^{-4}$ rad/s on average, this is during the simulation times that the reference is zero and the rotor is supposed to be stationary.

Car velocity response is shown in Figure 12, where the elevator car was first driven from 0 to -6 m/s, then back to 0 m/s, then from 0 to 6 m/s and finally back to 0 m/s. This is shown in Figure 12a.

The measured car velocity difference in Figure 12b behaves similarly to the measured rotor speed, with the difference that it has both minor noise during the whole operation and especially during acceleration and deceleration phases, but also that it has a large noise spike at the end of the run just as the car is stopping at the final position.

The simulated car velocity difference in Figure 12c behaves like the simulated rotor speed, although it also has significantly larger oscillations both at the beginning of the simulation as well as at the beginning of each jerk phase. These oscillations decay completely within approximately 2 seconds of simulation time.

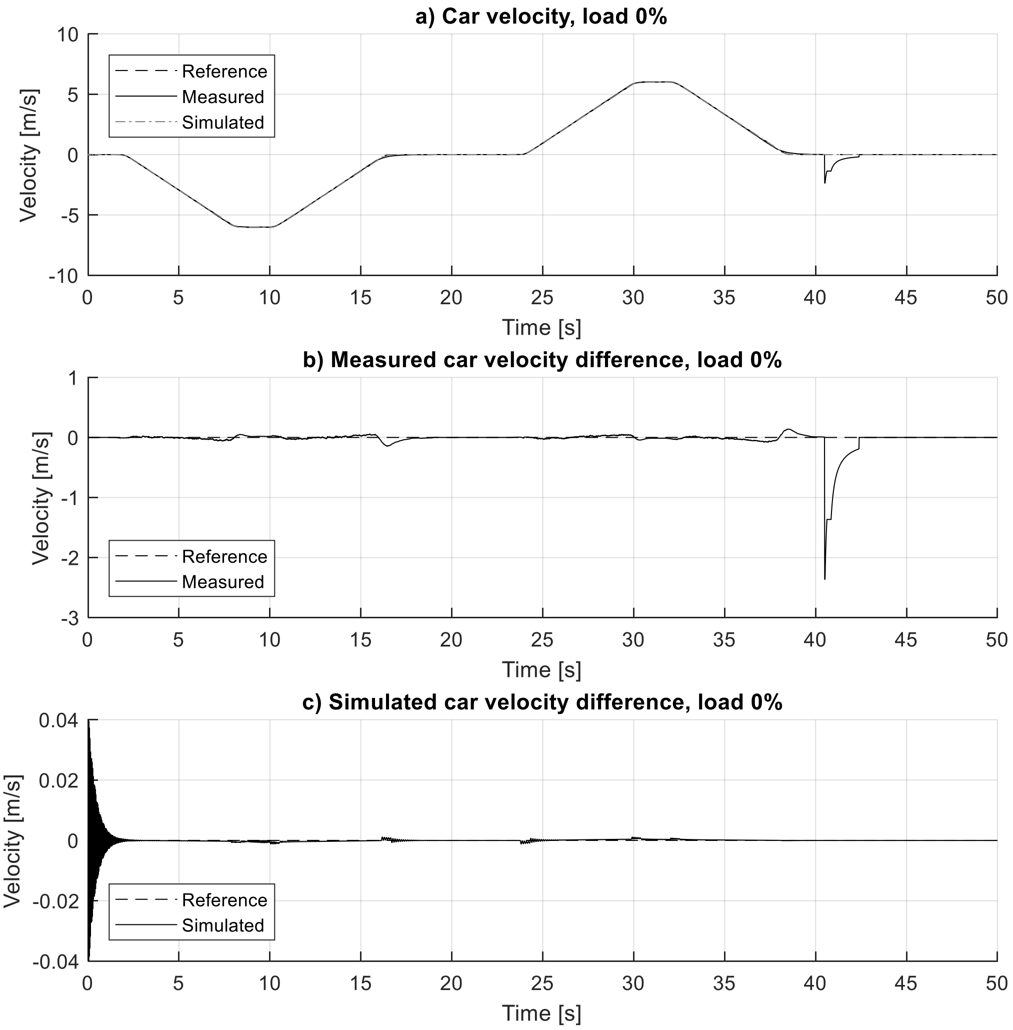

Figure 12. Elevator car velocity response and error at 0 % load.

3.2 Loading case 2, 320 kg additional mass

Figures 13 to 16 show measured and simulated responses with 25 % loading (320 kg). Corresponding error signals are also shown in each figure. The responses of this load case are very much similar to those with no additional load and the plots are largely in same shape, as they mostly only have slightly different sized error or slightly delayed response. These effects are compared later in this chapter.

Figure 13 shows the measured and simulated rotor rotations when the elevator car was first driven from 0 to -49.7 meters and then driven back to the starting point. This corresponds to rotating the rotor from 0 to -248.5 radians and back as shown in Figure 13a.

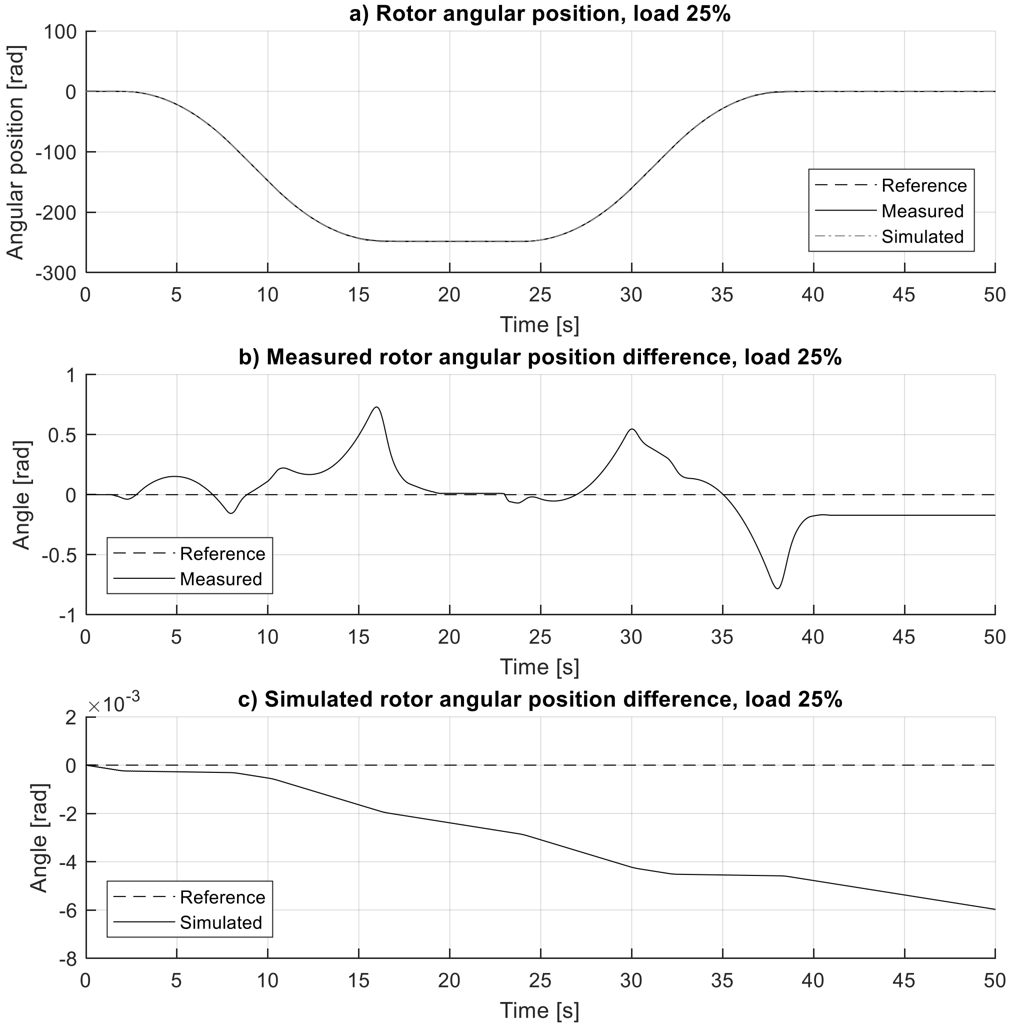

Figure 13. Rotor position response and error at 25 % load.

During this run largest measured errors were observed during rotor acceleration and deceleration as shown in Figure 13b. The measured signal leads the reference during acceleration and lags during deceleration. Starting from zero the rotor angle difference first fluctuates slightly before reaching a local minimum of -0.16 radians at the end of the first acceleration phase, after which it progresses to opposite direction and reaches a global maximum of 0.73 radians as it approaches to the desired reference point, after which it returns to a steady state. While at reference the steady state error is at 0.01 radians. As the elevator is moved back to the starting point the angle difference reaches a local maximum of 0.55 radians after accelerating and then progresses to global minimum of -0.78 radians at the end of deceleration before returning to a steady state. After the movement cycle the steady state error stays at -0.17 radians.

During the simulation the error accumulates over the run, with the error reaching a maximum of $6\cdot10^{-3}$ radians after the simulation is complete as shown in Figure 13c. While the overall shape of the simulated error graph is relatively straight and similar to no load case, some dynamic behavior can be seen in the response that differs from the first case. At the first phase of the movement cycle, during acceleration the simulated angle is lagging behind the reference more than in the previous case and the difference progresses even slower, whereas during deceleration the difference progresses even faster. These effects behave in the opposite way when the system returns to the starting point at the second phase, also at a slightly larger magnitude than in the first case.

Figure 14a shows the measured and simulated car position with the car driven from 0 to -49.7 meters and then back to the starting point. During the run largest measured errors occurred during car acceleration and deceleration as shown in Figure 14b. The measured signal is leading the reference during acceleration and lagging during deceleration. Starting from zero the car position difference first fluctuates slightly, reaches a local minimum of -0.02 meters at acceleration, then progresses to opposite direction and reaches a global maximum of 0.13 meters at deceleration before returning to a steady state. While at reference the steady state error is at $-0.78\cdot10^{-4}$ meters. When moving back to the starting point the position difference reaches a local maximum of 0.13 meters after accelerating, progresses to global minimum of -0.10 meters at deceleration, and after that the returns to a steady state again. After the cycle the car steady state error is -0.03 meters.

Figure 14c shows the simulated car position error. The difference oscillates in a similar manner as in the first load case and also decays in a same time frame of 2 seconds. While the car is moving the difference progresses in similar shape as the position graph before reaching a local minimum of -4.6·millimeters, bouncing back and returning to steady state that has some drifting away from reference. Going back up the difference first descends to a global minimum of -4.8·millimeters before progressing to the final desired position of -1.2·millimeters, where it also slightly drifts away from the reference.

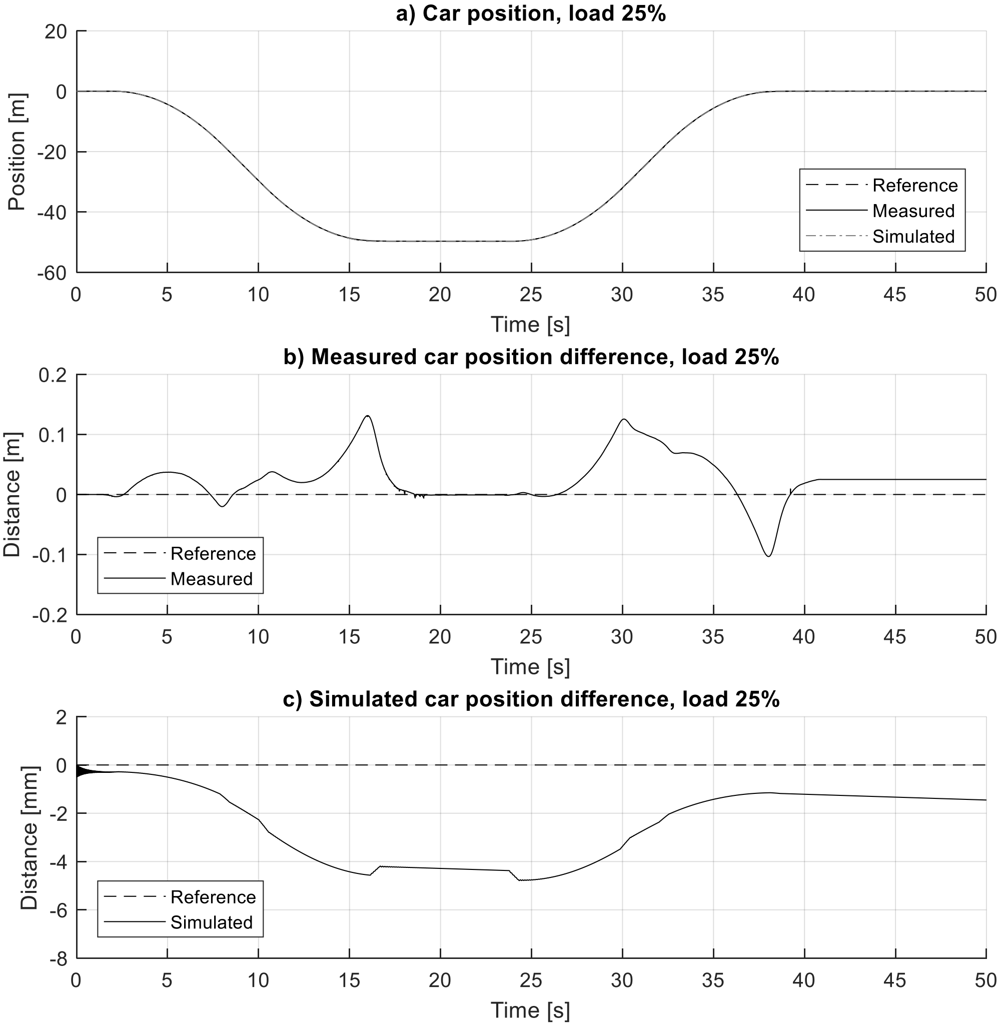

Figure 14. Elevator car position response and error at 25 % load.

Rotor rotation speed response is presented in Figure 15. There the elevator car was first driven from 0 to -6 m/s, then back to 0 m/s, then from 0 to 6 m/s and finally back to 0 m/s, corresponding to rotor rotation speed of -30 rad/s while descending and 30 rad/s while ascending. This is shown in Figure 15a.

Measured difference in Figure 15b shows some minor fluctuation. Larger peaks, that are associated with start and end of acceleration and deceleration, are also the moments where jerk reference is not equal to zero. The global minimum and maximum values of -0.70 rad/s and 0.68 rad/s are located at 16.5 seconds and 38.5 seconds respectively. This is when the measured signal lags the reference as the rotor position approaches the desired level.

Simulated difference shown in Figure 15c has some oscillation during the whole simulation. The oscillation is significantly larger at the beginning and mostly decays before 20 seconds of simulation time in this case too. The simulated difference is closer to zero while speed changes after times of 1.75 seconds and 32 seconds, averaging -1.2·10^{-5} rad/s, and further while speed changes after times of 10 seconds and 23.75 seconds, averaging -2.3·10^{-4} rad/s. Simulated rotation speed steady state error is -1.3·10^{-4} rad/s on average, this happens while the reference is zero and the rotor is desired to be stationary.

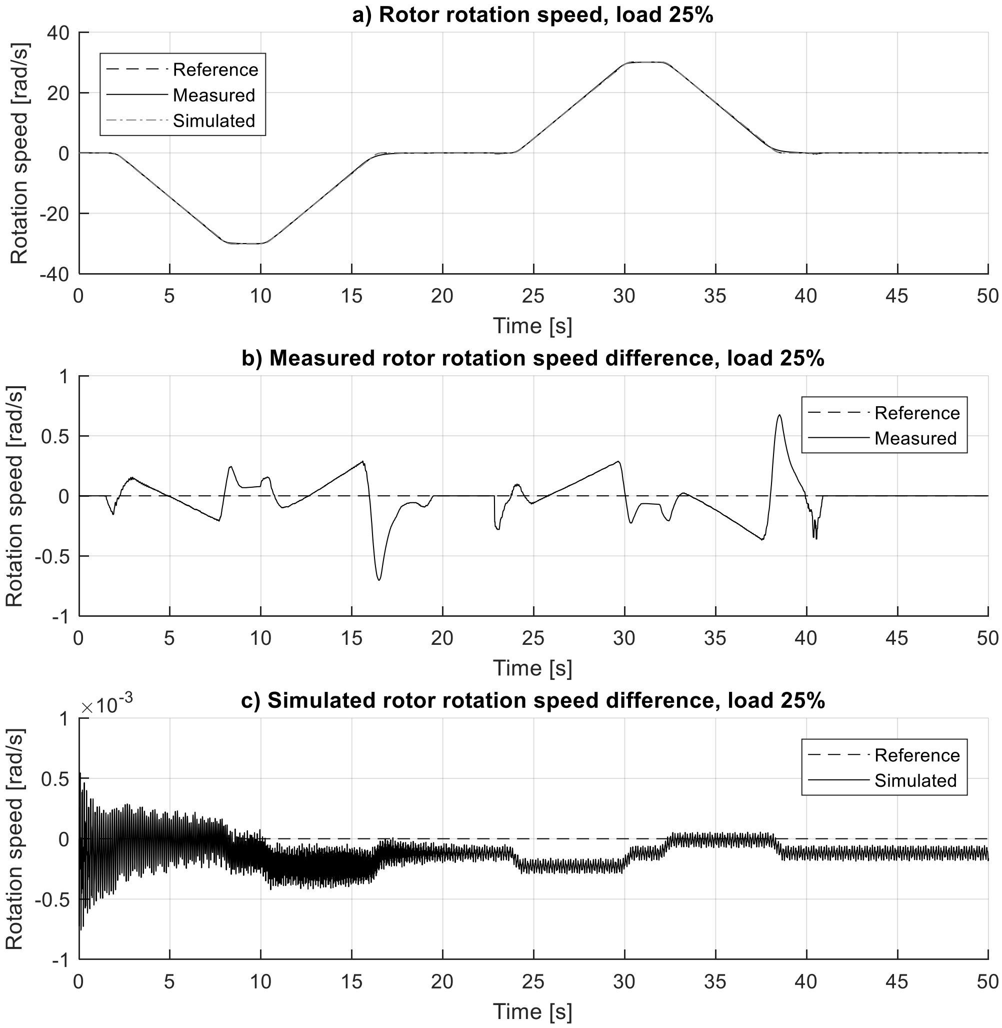

Figure 15. Rotor velocity response and error at 25 % load.

Car velocity response is shown in Figure 16. The elevator car was first driven from 0 to -6 m/s, then back to 0 m/s, then from 0 to 6 m/s and finally back to 0 m/s as shown in Figure

16a. The measured car velocity in Figure 16b behaves similarly to the measured rotor speed, it has minor noise during the operation and especially during acceleration and deceleration. It also has a large noise spike at the end of the run just as the car is stopping at the final position, similarly to the peak seen in the first load case. The simulated car velocity in Figure 16c behaves like the simulated rotor speed, although it also has significantly larger oscillations both at the beginning of the simulation as well as at the beginning of each jerk phase. These oscillations decay completely within approximately 2 seconds of simulation time.

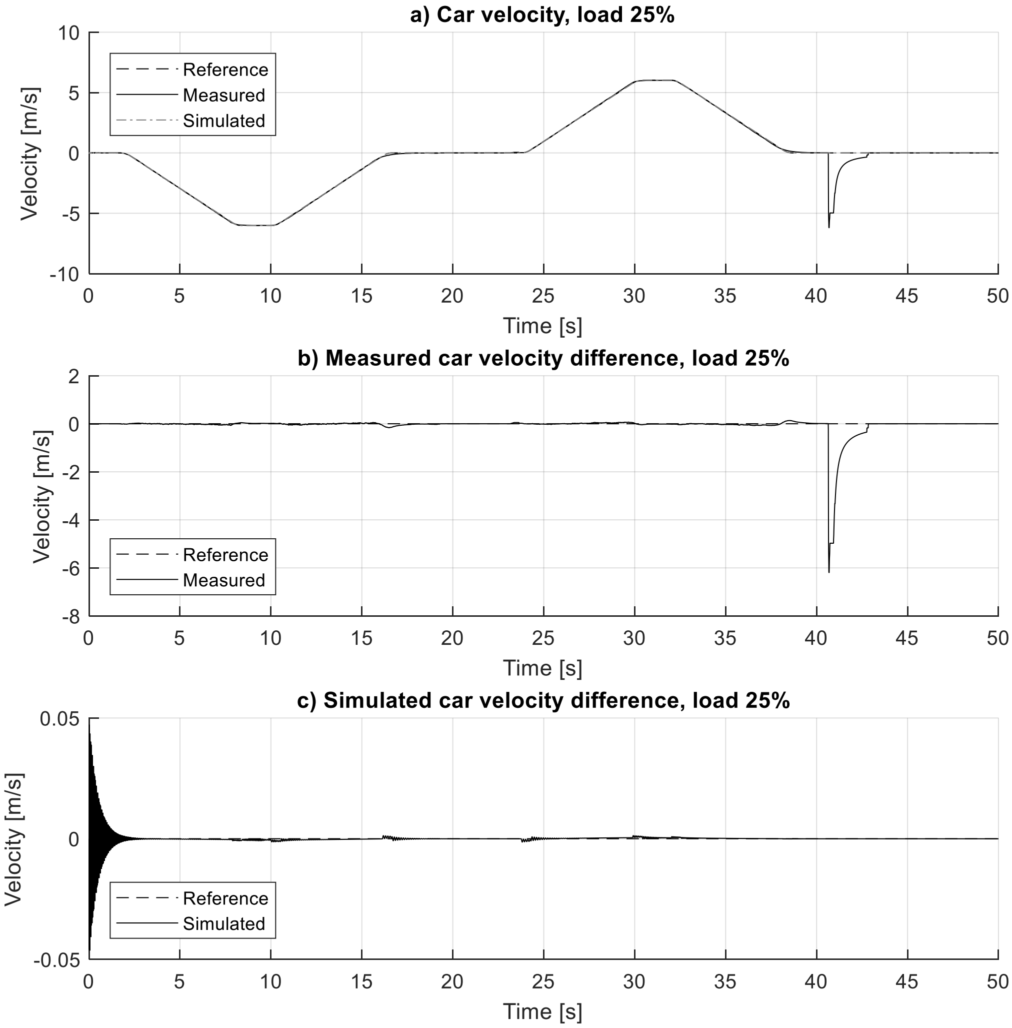

Figure 16. Elevator car velocity response and error at 25 % load.

3.3 Loading case 3, 665 kg additional mass

Figures 17 to 20 show measured and simulated responses with 48 % loading (665 kg). Corresponding error signals are also plotted in each figure. The responses of the last load case are very much similar to the other two and the plots here are also largely in same shape, as they also mostly only have slightly more difference in error size or slightly more delayed response. These effects are included in the comparison later in this chapter.

Figure 17 shows the measured and simulated rotor rotations as the elevator car was first driven from 0 to -49.7 meters and then driven back to the starting point. This corresponds to rotation of 0 to -248.5 radians and back as shown in Figure 17a.

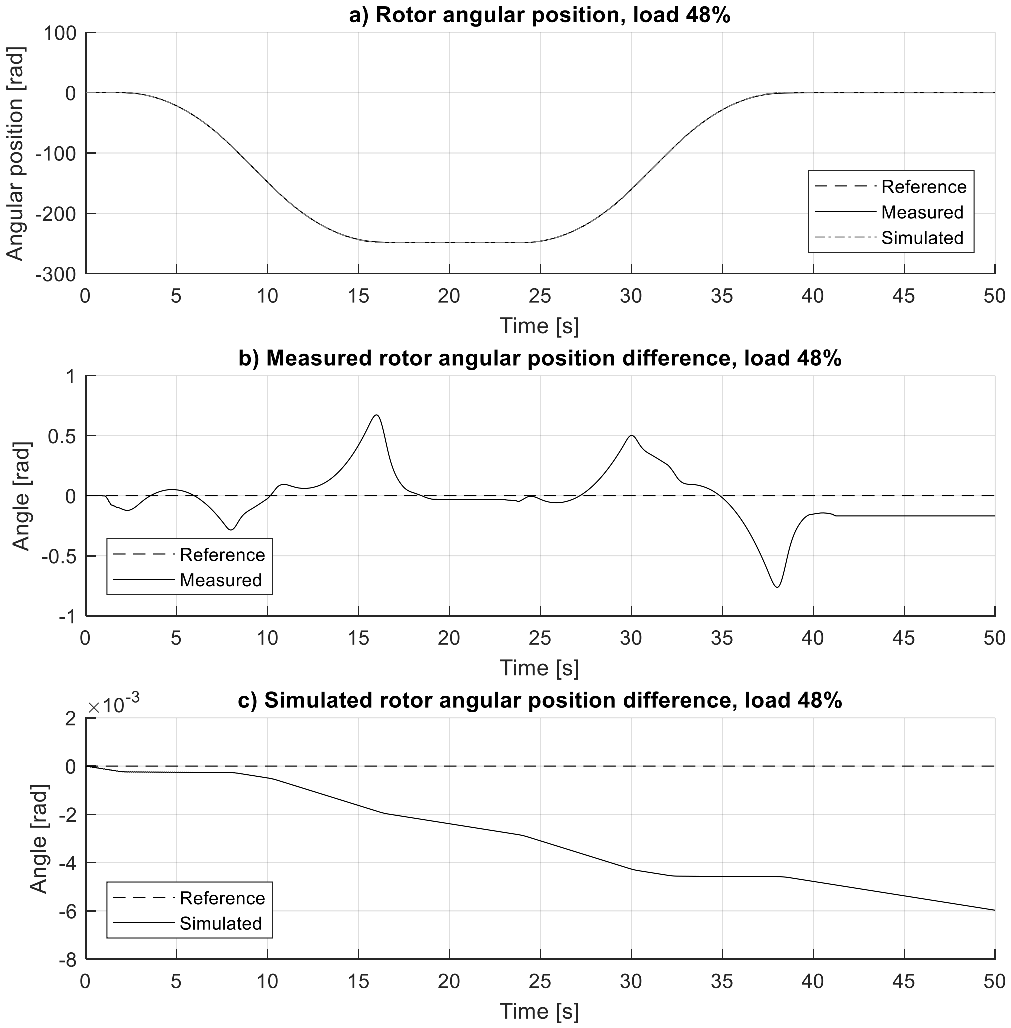

Figure 17. Rotor position response and error at 48 % load.

During this run largest measured errors were observed during rotor acceleration and deceleration as shown in Figure 17b. Similarly to other cases the measured signal leads during acceleration and lags during deceleration. Rotor angle difference starts from zero, fluctuates slightly before reaching a local minimum of -0.29 radians at the end of the first acceleration phase, progresses to opposite direction and reaches a global maximum of 0.67 radians as it approaches to the desired reference point, and lastly returns to a steady state. The steady state error is 0.03 radians. As the elevator is returned to the starting point the angle difference reaches a local maximum of 0.50 radians after accelerating, progresses to global minimum of -0.76 radians at the end of deceleration, and returns to a steady state. After the complete cycle the steady state error is -0.17 radians.

During the simulation the error accumulates over the run similarly to the other two cases, with the error reaching a maximum of $6 \cdot 10^{-3}$ radians after the simulation is complete as shown in Figure 17c. Overall shape of the simulated error graph is again relatively straight and similar to other cases, some dynamic behavior is visible in the response that differs from the other two cases. At the first phase of the movement cycle, during acceleration the simulated angle is lagging behind the reference more than in the previous cases and the difference progresses slowest, whereas during deceleration the difference progresses fastest. These effects again behave in the opposite way with the system returning to the starting point, at a slightly larger magnitude than in the first two cases.

Figure 18a shows the measured and simulated car position with the car driven from 0 to -49.7 meters and back to the start. During the run largest measured errors occurred during car acceleration and deceleration as shown in Figure 18b. The measured signal is leading during acceleration and lagging during deceleration. Car position difference first fluctuates slightly from zero, reaches a local minimum of -0.03 meters while accelerating, then progresses to opposite direction and reaches a global maximum of 0.13 meters while decelerating before returning to a steady state. At reference the steady state error is $-0.74 \cdot 10^{-4}$ meters. After that, while moving back to the start the position difference reaches a local maximum of 0.11 meters while accelerating, progresses to global minimum of -0.11 meters while decelerating, and after that the returns to a steady state again. After the cycle the car steady state error is -0.02 meters.

Simulated car position error is shown in Figure 18c. The difference oscillates in a similar manner as in the first load cases and also again decays in a same time frame of 2 seconds. While moving the car the difference progresses in similar shape as the position graph, reaches a local minimum of -5.2·millimeters, bounces back and returns to steady state that also has some drifting away from reference like the previous cases. When going up the difference first descends to a global minimum of -5.4·millimeters and progresses to the final desired position of -1.2·millimeters, where it also slowly drifts away from the reference.

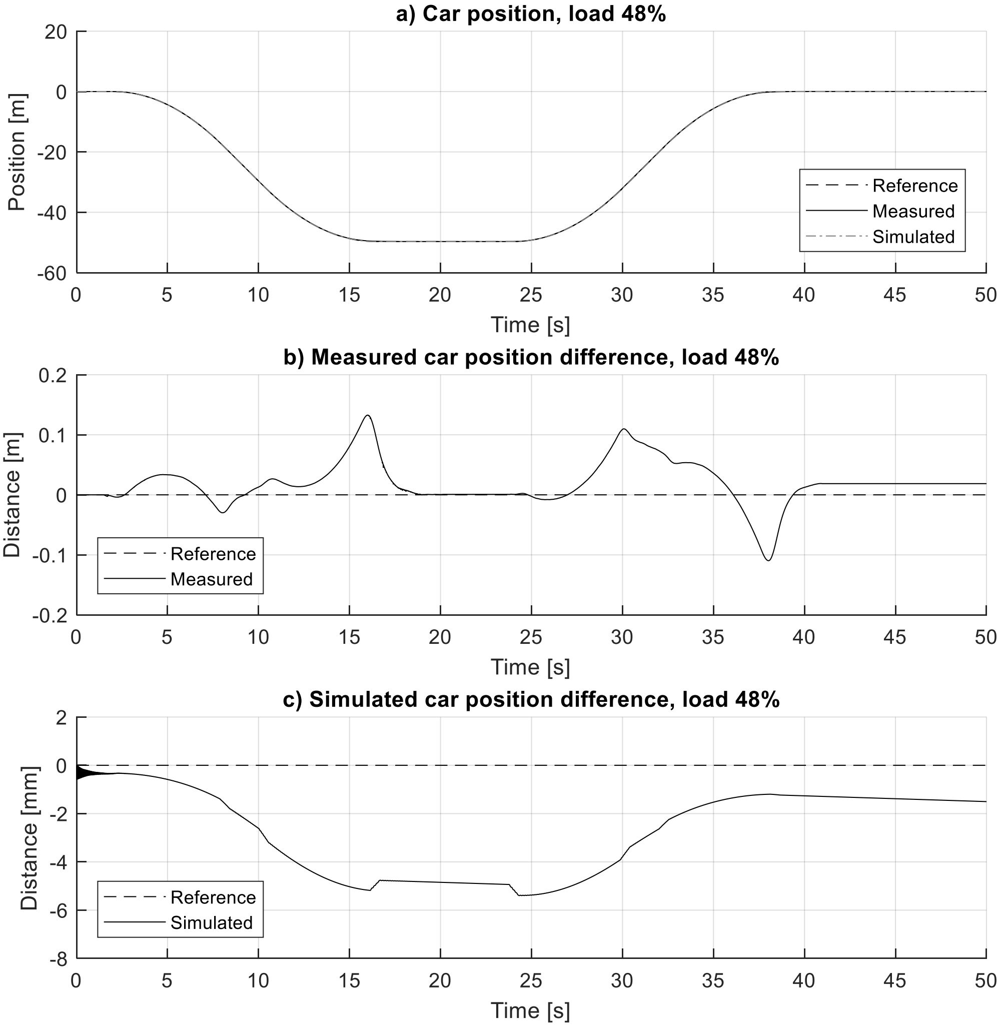

Figure 18. Elevator car position response and error at 48 % load.

Rotor rotation speed response is presented in Figure 19. The elevator car was driven from 0 to -6 m/s, then -6 to 0 m/s, then from 0 to 6 m/s and finally from 6 to 0 m/s, corresponding to speed of -30 rad/s while descending and 30 rad/s while ascending as shown in Figure 19a.

Measured difference in Figure 19b shows minor fluctuation. Larger peaks associated with start and end of acceleration and deceleration, that are also the moments where jerk reference is not equal to zero, can be found by looking at the global minimum and maximum values of -0.70 rad/s and 0.67 rad/s that are located at 16.5 seconds and 38.5 seconds respectively. At these locations the measured signal lags the reference when the rotor position approaches the target level.

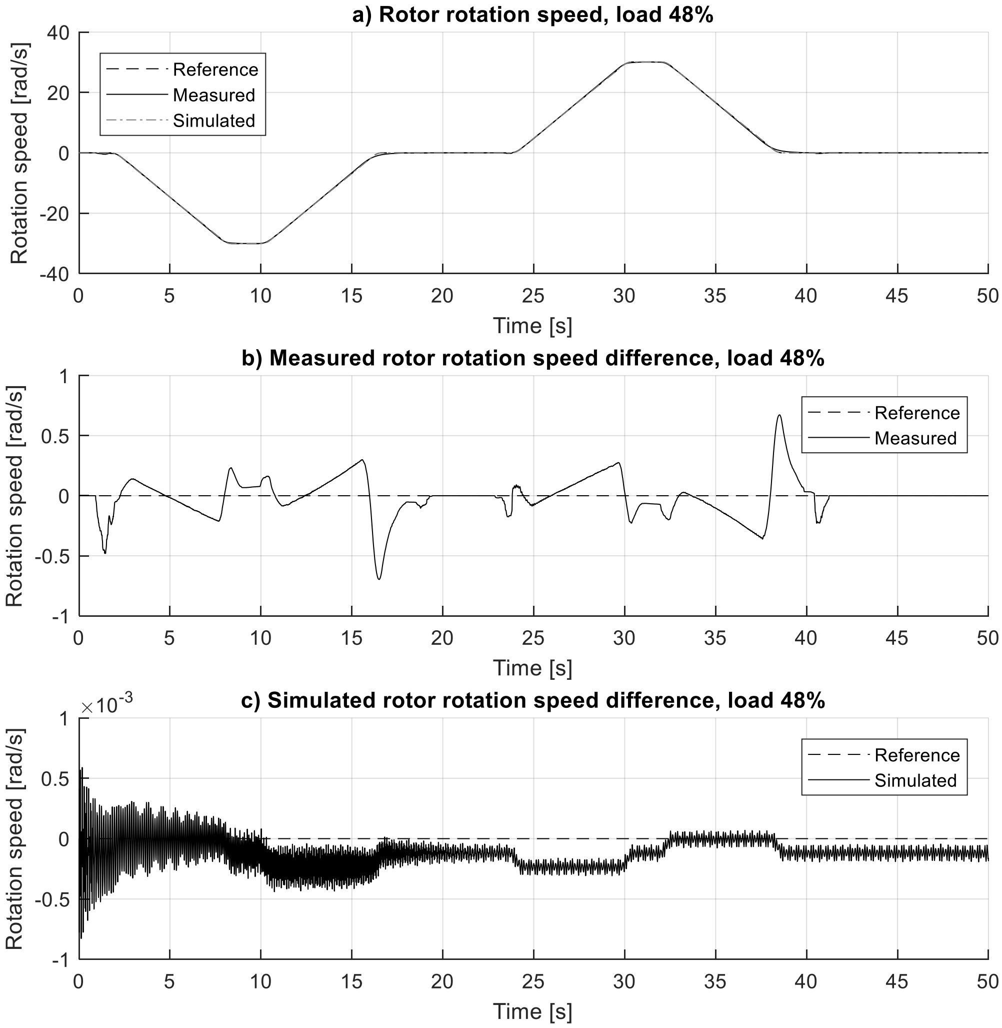

Figure 19. Rotor velocity response and error at 48 % load.

Simulated difference shown in Figure 19c has some oscillation during the whole simulation. The oscillation is significantly larger at the beginning and mostly decays before 20 seconds of simulation time like in the other two cases. The oscillation here also has a greater amplitude than the other two cases. The simulated response is closer to reference while speed changes after times of 1.75 seconds and 32 seconds, averaging -5.1·10^{-6} rad/s, and further while speed changes after times of 10 seconds and 23.75 seconds, averaging -2.4·10^{-4} rad/s. Simulated steady state error is -1.2·10^{-4} rad/s on average, this happens while the speed reference is kept at zero and the rotor is aimed to stay stationary. Figure 20 shows car velocity response along with differences compared to reference.

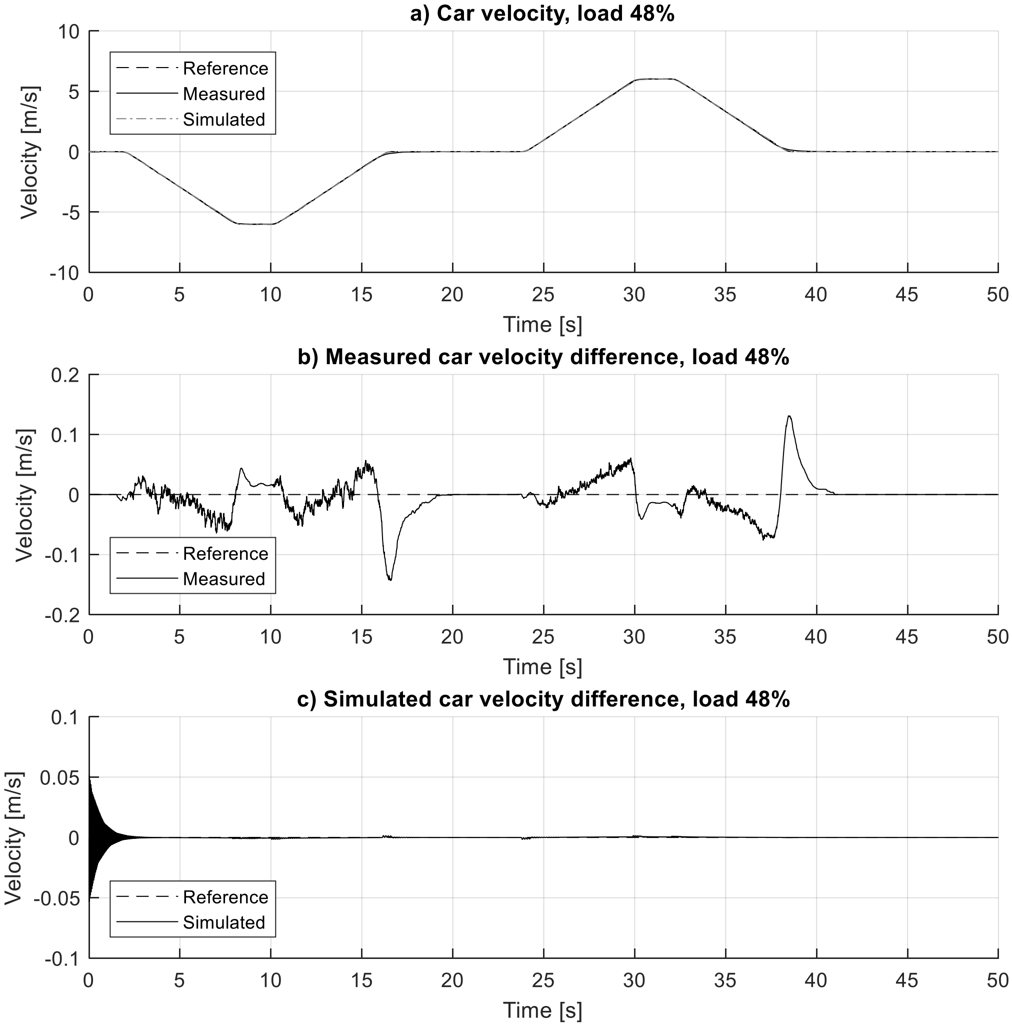

Figure 20. Elevator car velocity response and error at 48 % load.

Shown in Figure 20a is the elevator car that was first driven from 0 to -6 m/s, after that it went back to 0 m/s, then from 0 to 6 m/s and finally returned back to 0 m/s. The measured car velocity in Figure 20b behaves similarly to the measured rotor speed, it has some noise during the operation and especially during acceleration and deceleration. However it is missing the large noise spike at the end of the run just as the car is stopping at the final position, which was present in the first two load cases. The simulated car velocity in Figure 20c behaves like the simulated rotor speed, although it also has significantly larger oscillations both at the beginning of the simulation as well as at the beginning of each jerk phase. These oscillations decay almost completely within approximately 2 seconds of simulation time.

3.4 Comparison and virtual data

Figure 21 shows car position deviation from reference signal for each measured and simulated loading case. These are plotted in the same figure to aid comparison of the graphs.

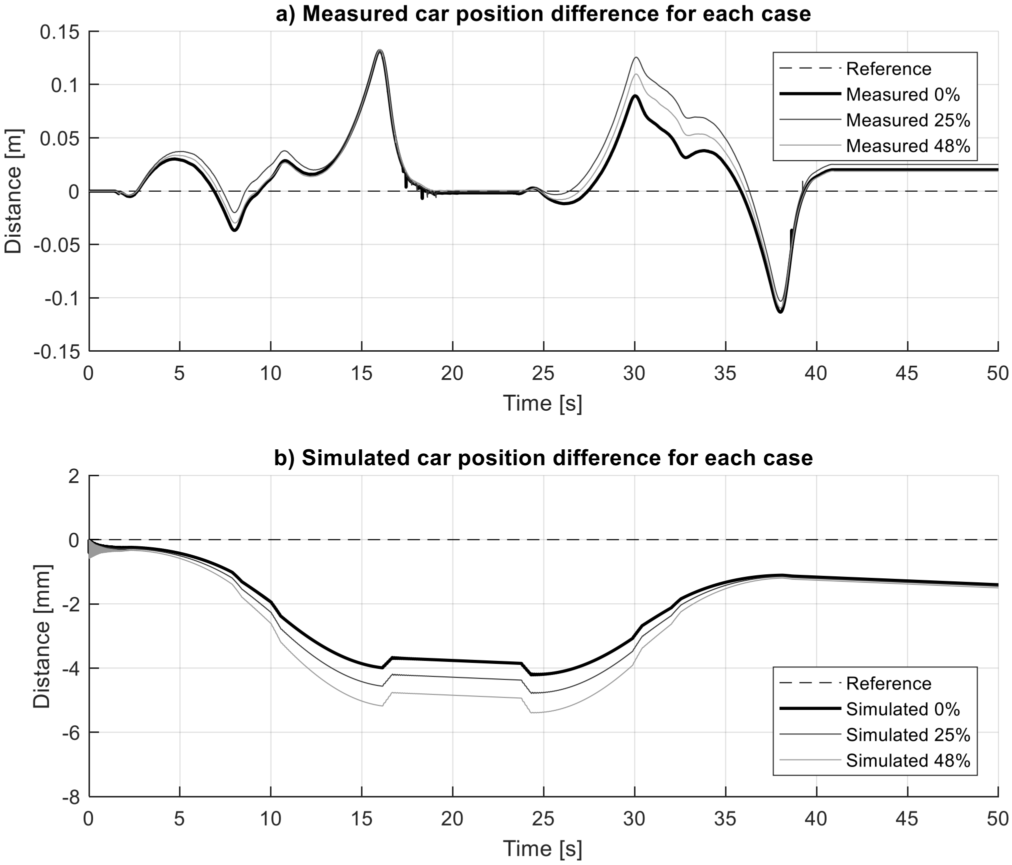

Figure 21. Car position error for each case in a single figure.

Measured differences are shown in Figure 21a. While the reference is at steady state or just beginning to accelerate, the lowest differences are seen in 0 % load case, followed by 25 % load case and highest differences are seen in 48 % load case. However during the phases that have a larger velocity reference value, while the 0 % load case is still producing lowest differences, the two other load cases switch positions and during that time the 25 % load case has the highest difference values. With simulated differences the results in Figure 21b follow the dynamics of the model. During the entire movement cycle the 0 % load case response is closest to the reference value, followed by the 25 % load case, and the 48 % load case response is furthest from the reference signal.

Figures 22 and 23 examples of the virtual sensor concept that could be implemented in a Simulink model. They are extracted from the simulation run at 0 % load. These graphs include no reference value or measured data as such information does currently not exist.

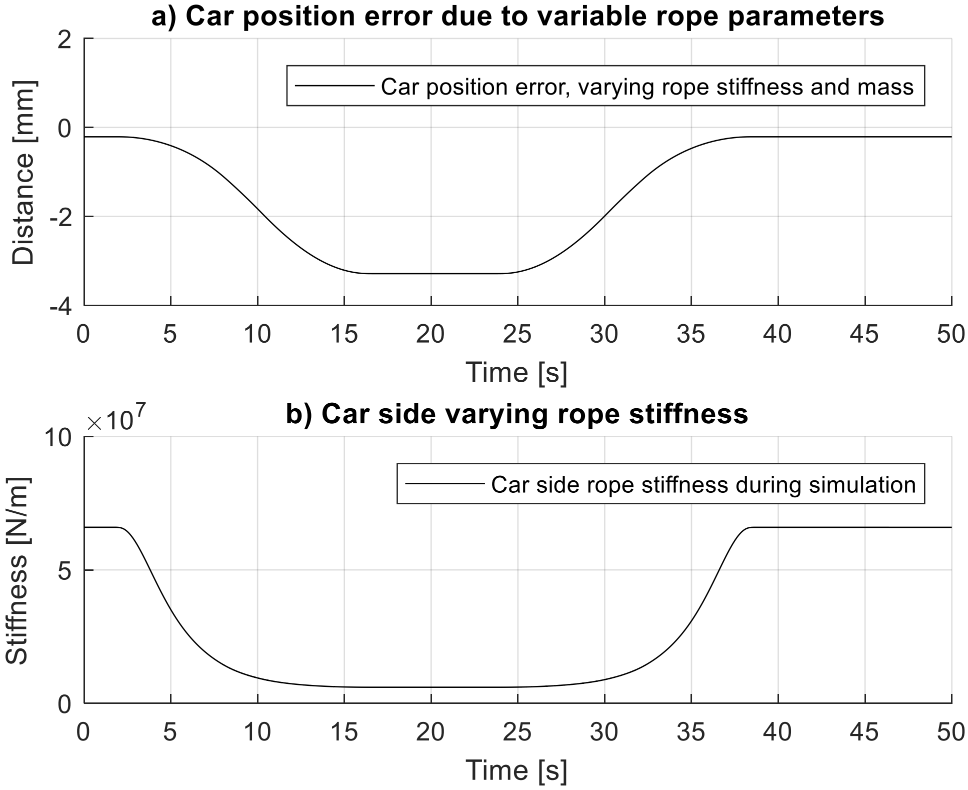

Figure 22. Example of a virtual sensor; simulated car position error due to rope parameters.

Figure 22 shows how part of the car position error is forming, which is based on the simulated rope stiffnesses and masses. They are functions of the rope length and therefore also functions of the car position. Figure 22a shows that the error ranges from 0.21·millimeters when the car is at highest position and the rope is at its shortest at only 5 meters, up to 3.3·millimeters when the car is at lowest position and the rope is at its longest at 54.7 meters.

Figure 22b shows how the rope stiffness changes based on car position. This is calculated using the equation (1). When the car is at the highest position the rope stiffness on that side is $6.6 \cdot 10^{7}$ N/m, and when the car is at the lowest point the rope stiffness on that side goes down to $6.0 \cdot 10^{6}$ N/m. The forces used in calculating the position error of Figure 22a are obtained from varying the rope mass which is proportional to the rope length.

Figure 23a shows the simulated torque that is acting on the rotor during the simulation run. The torque has a large amount oscillation at the start of the run but this decays over time. The average torque ranges from -2500 Nm to 2100 Nm based on the velocity reference. Since same torque is affecting the stator but in the opposite direction, the stator angular displacement can be obtained by accounting the torsional stiffness and damping of the stator mounting. This can be seen in Figure 23b. Using the parameters of this model the average simulated rotation ranges from $-2.0 \cdot 10^{-5}$ radians to $2.4 \cdot 10^{-5}$ radians.

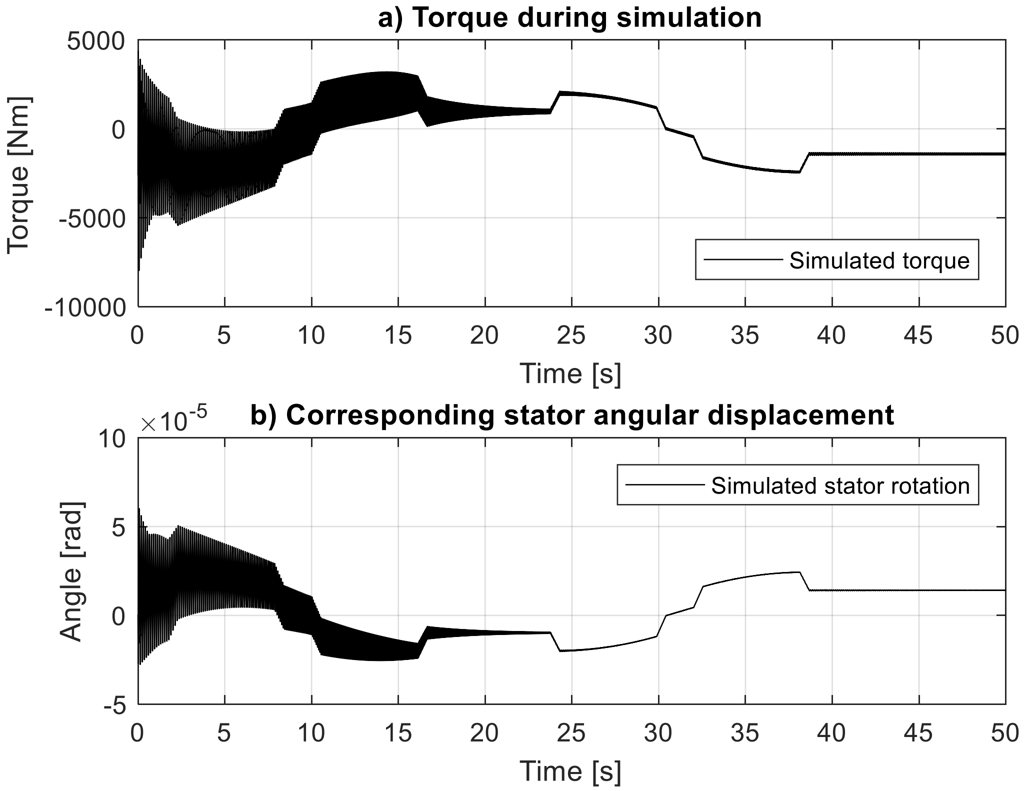

Figure 23. Example of a virtual sensor; simulated torque affecting rotor and corresponding angular displacement of stator.

4 DISCUSSION

Overall the results of the simulations follow the reference signals nicely, even though compared to the averaged measurement data there are some phenomena that are not included. Ultimately the simulation could be made even more accurate even in its current form, however after experimenting with various variable-step solvers (fixed-step was not an option due to nonexistent stability) it was clear that more accurate solvers would also multiply the execution time.

The raw measurement data had a lot of noise, especially the velocity obtained from the car. This was likely caused by both the distance the information had to travel along the wire, as well as the somewhat unreliable nature of the encoder used. These outliers were mostly not too difficult to remove, and the ones that remained were noticeable by visually comparing the plots in the same graph.

Based on Figures 9-20 it can be seen that in all cases the simulated response error is at most below the magnitude of $1\cdot10^{-2}$ which makes it somewhat irrelevant as the reference goes up to 50, meaning that the relative error is fractions of a percent. Of course when discussing the position error of an elevator car even a difference of 1 cm can cause issues during operation, which means that in order to use this model for purposes that require higher accuracy, one would need to adjust the model to operate in such way.

The shape of the simulated car position error can possibly be explained by looking at Figures 21 and 22. Since the car position error depends on both the gravity force affecting on the car as well as the spring force that is resisting this error, it makes sense that all of the simulated plots in Figure 21 follow the shape of the parameter effect plot shown in Figure 22a. This is caused by simultaneously increasing the car side mass and reducing the car side rope stiffness by moving the car downwards and therefore increasing the car side rope length. Also since this error is directly related to the gravity force and therefore the mass, it makes sense that the additional mass applied to the car increases the absolute error in car position. This effect in these results is slightly too large because of the entire rope mass being modelled as a lumped part of the mass at the end of the rope. In reality the rope mass would be evenly distributed over the length of the rope.

The large transient that occurs in the beginning of the simulation, especially noticeable in Figure 23, is likely caused by the simulated system not being in equilibrium when simulation starts. This can happen due to one or more initial conditions being defined incorrectly. A solution for this could be to first simulate the model with a zero reference until the oscillation finishes before applying the input signal. Another way to avoid this could be to first calculate initial values for the generalized coordinates using static analysis, and applying these results to the Simulink model integrators to force the simulation to start from equilibrium.

The measured responses had significantly larger differences from the reference value that the simulations were based on, some in the range of 0.10-0.15. These can however be mostly explained by dynamics of the physical elevator and the probable additions that are not present in the simulation. Some of the sources that may have caused this are the unreliability of the encoders used in measurements, an additional position control loop in the physical system that does not exist in the model, slipping phenomena between hoisting rope and the traction sheave during measurements which was also not modelled into the virtual system, and other control algorithms that may contribute to rider comfort.

4.1 About results

Although the results of the simulation plotted in the same graph as measurements are largely quite similar, the graph also shows several characteristics of the elevator system that is not accounted for in the model. When approaching the target position, the actual system is in fact not following the strict jerk reference that is provided, but rather slows down in a calmer manner. This behavior is difficult to include in the model without knowing the actual control scheme that is used in the physical elevator. This can be seen close up in Figures 24 (filtered) and 25 (not filtered).

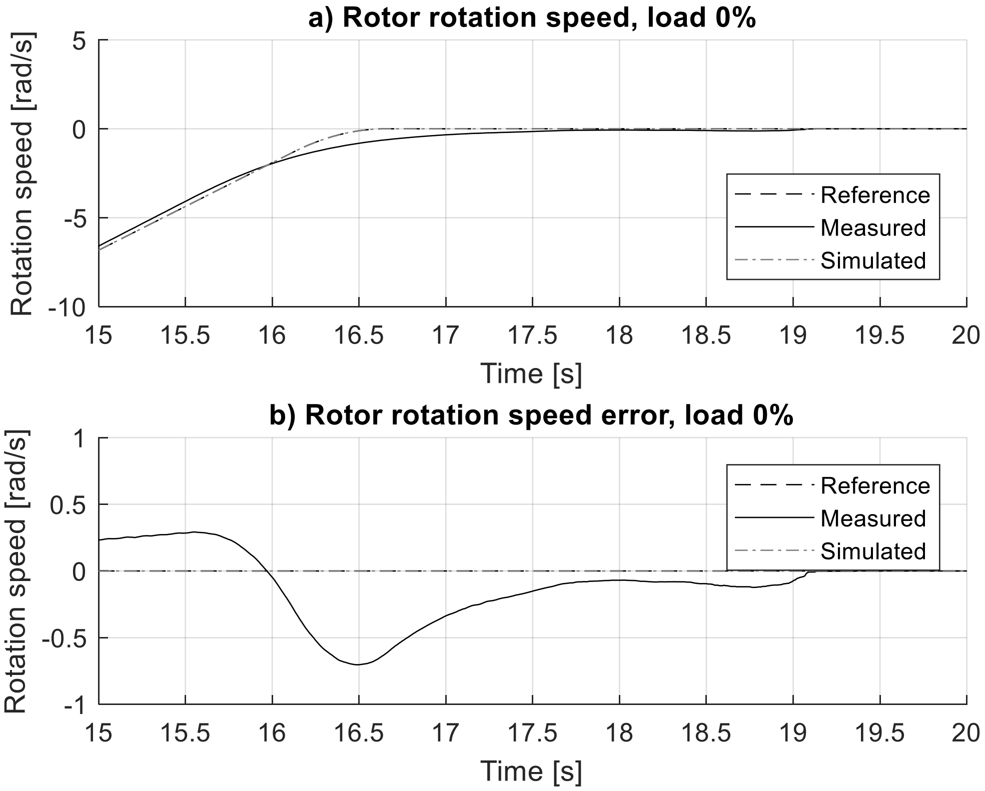

Figure 24. Rotor velocity error when approaching target position.

As can be seen in Figure 24 the rotation speed in measured data slows down quite early before reaching the target velocity, and in addition to that right before reaching the set target it slightly accelerates before going to a full stop. The first slowdown is probably implemented to improve rider comfort, as this ensures that the elevator basically glides to the target level without a sudden stop, unlike the current simulation model. The second part, additional adjustment, is done by a position controller to ensure that the elevator car is at level with the floor that its supposed to reach. This can be better seen in Figure 24, as after a specific point the velocity signal begins vibrating as if it is attempting to carefully adjust the position correctly before coming to a full stop.

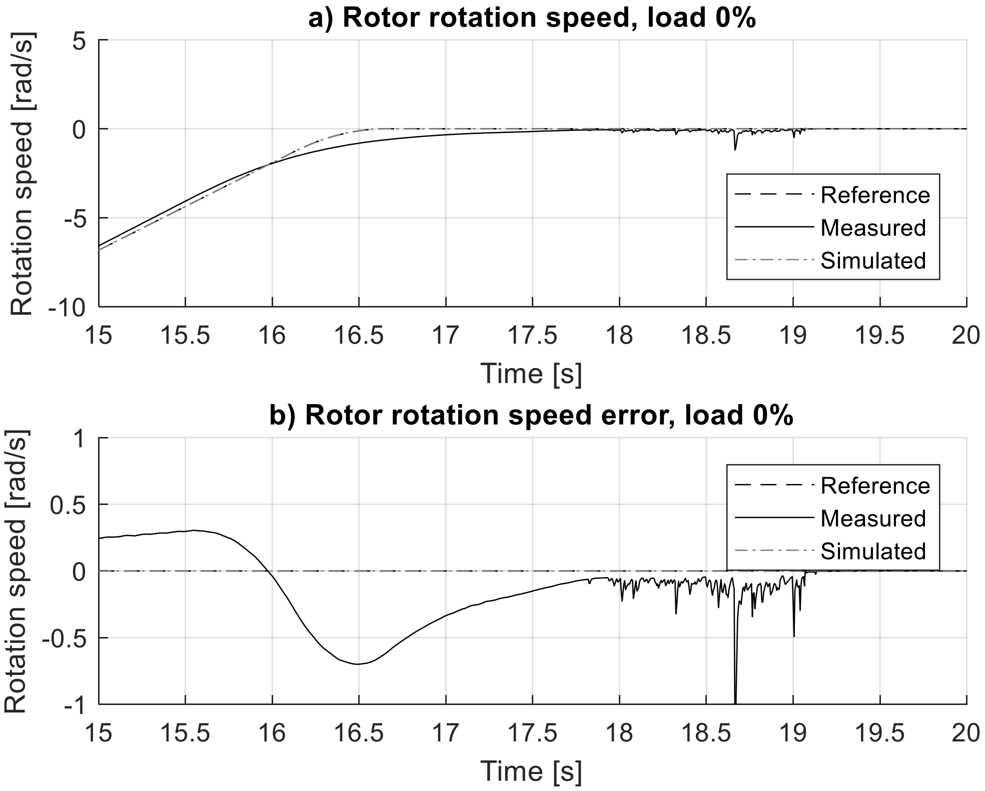

Figure 25. Unfiltered rotor velocity error when approaching target position.

Another example of where the model could use an improved position control is that since currently the feedback loop is only handling velocity control, it means that if at any point of operation, a position error occurs, the model is unable to correct this. In reality the actual system has some kind of position control as well, but without knowing its exact operating principles it is not possible to mirror it in the simulation model. This was also not feasible to complete in the scope of this thesis. It can also be seen in the static response simulation in Figure 8, where the steady state error of the velocity signal causes the position response simulation to creep away from zero.

In the literature this is solved by adding either a position control loop or a current control loop inside or outside the velocity control loop. This ensures that the possible steady state error from velocity control is not reflected into the actual rotor position. One other possibility could be to add an integral part into the velocity controller, which could help mitigate the steady state error. However this is still not guaranteed to give flawless position responses.

The errors between measured data and the reference signal used here can also be caused by the two other factors. The large amount of noise and fluctuation in the signals suggests that the encoder used for data acquisition could have been functioning better. This could have removed some of the error that was present.

The steady state error that was present in all load cases was possibly caused by minor slipping between hoisting rope and traction sheave, which meant that if the car slipped while slowing down for the lower floor level, in order for the car to be in the correct position it would have to travel a longer distance to reach the upper floor level. This also explains why the car steady state error in our cases was positive and rotor steady state error was negative, since in all cases the counterweight side was heavier than the car side, causing more slipping to happen into the counterweight side.

Several integrator settings provided in Simulink were tested during the tuning of the model. Best results were obtained with either variable step ode45 or ode23. Others had either significantly larger vibration or longer execution times, and fixed-step solvers gave unstable results. Out of the two acceptable solvers ode45 had significantly shorter simulation time while ode23 might have been slightly more accurate. Therefore ode45 was chosen for this model since it gave reasonably accurate results and because the real-time simulation viewpoints require faster execution.

The simulation model theoretically has the possibility of outputting more data than position and velocity information of the motor axis and elevator car. The 5-DOF Simulink block diagram also has wiring for example for torque, stator vibrations (both vertical and rotational), counterweight movement and rope parameters. This data can be extracted from the model and plotted as shown in Figures 22 and 23. The issue with this type of data is that without physical sensors the readings cannot be verified. They could however still be considered virtual sensors.

4.2 Reliability analysis

In order to minimize issues caused by sensor errors the measurement results were averaged from five separate runs. This was unfortunately not enough as especially the elevator car velocity data had large spikes that were likely caused by the data collection methods.

Inaccuracy of the encoder and the fact that the data had to travel over 50 m over a cable in the elevator shaft provided numerous faulty readings, and in order to obtain meaningful plots for comparison, the data had to be filtered quite a bit. Running the data through a 100th order median filter means picking the median value over 100 data points, which in this experiment corresponds to a 0.5 s time step. This will inevitably remove some properties that are present in the raw data, for example any oscillation when stopping would be difficult if not impossible to determine.

In the case of this study the loss of some minor data properties was not a main concern since not only would the oscillation be difficult to separate from the large amount of noise that was present in the raw data, but also the simulation model is unable to reproduce this type of fine details without specifically modelling the control system to focus on it.

Another point is that the great stiffness of the system caused by several integrators, feedback signals and large coefficients will limit the usability of the model in capturing small oscillations. This means that the model should not be used for that type of detailed analysis without fine-tuning it for that specific purpose.

Even though the encoders used in the measurements were capable of much greater resolutions, the data was imported to MATLAB using a time step of 0.005 s. This is sufficient when comparing the model as running the simulation with significantly smaller time steps than this would increase run times unnecessarily. Especially when the simulation time was 50 seconds, the value of smaller integration steps would only be achievable after further optimizing the model parameters.

Reliability and relevance of the sources used in the literature research should also be discussed in this chapter. Out of the total of 22 references cited in this thesis there were 11 scientific articles, 4 conference publications and 5 books. The rest of the references include for example web documents and commercial handbooks included for some specific figures or data. From the scientific articles and conference papers 11 of them were published in 2016 or later, so the references are fairly recent. In addition to that the books cited were either also quite recent or otherwise widely used if they were older.

4.3 Future research

There are some ways to improve the model studied in this research. Probably one of the most important changes that could be made would be to implement a similar control scheme that is used in the physical system so that the parallel operation of physical and simulated system would not only produce similar output, but would also have similar internal functions. This would require researching the current control logic that is present in actual systems.

Some smaller additions to the current model include a physical rotor that is likely to have some imbalance that could excite for example the stator frame, this could be introduced in the model as a sinusoidal excitation directly at the stator position vertical coordinate. This would allow researching the effects of external excitation using the simulation model.

Another smaller addition to the model would be to include the friction phenomena between the traction sheave and the ropes. This would allow the model to simulate slipping of the cart and counterweight, since it can cause position error to occur in single instances and also to accumulate. This would not only work during both normal elevator movement but could also enable the possibility of using this model to simulate irregular events such as emergency braking.

Lastly more work would probably need to be conducted in order to optimize the model, so that the real-time simulation would be possible with as little resources as possible. Currently the simulation speed is quite close to actual time passing, however speeding up the model significantly would make the implementation not only much easier but possibly also perhaps at all possible. This is because of the delays that are introduced in parts of the real-time system other than the model itself, which would need to be accounted for in the model execution speed.

5 CONCLUSIONS

The goal of this thesis was to create a dynamic simulation model of a KONE MX-series hoisting machinery in Simulink. The purpose for this task was that as KONE is interested in working towards having a comprehensive digital twin of their elevator system, and in order to eventually achieve it they require various digital tools for development, one of which could be a dynamic simulation model of an elevator. The parametric simulation model can be used to act as a part of the elevators digital twin, since a digital twins goal is to mirror all relevant properties, behavior and data of the elevator, and a parametric model can easily be adjusted to use for example a range of different properties and initial conditions. A goal of the model was also to study the use of virtual sensors in order to enable the extraction of such quantities from Simulink that cannot be measured with physical sensors. Finally the model should also, if it is possible, be able to simulate various cases in real time, since the real-time support and possibilities of Simulink are some of the benefits of the program. This is done in order to eventually obtain all benefits that virtual sensors and digital twins have to offer, since these are especially valuable in real-time when considering fault injection and detection, condition monitoring and sensor development.

Model is based on a simplified five degree-of-freedom mass-spring-damper -system. The five degrees of freedom are stator vertical movement, stator rotational movement, rotor rotational movement, elevator car vertical movement and counterweight vertical movement. The equations of motion were derived using Lagrange’s method and modelled in Simulink as a block diagram. The input used for the equations of motion was torque applied to the rotor in order to simulate the actual motor functioning.

In order to have the model function similarly to the physical system, the input to the entire model was made by integrating a jerk reference curve twice to obtain a speed reference curve, which was then converted to torque and used as an input to the system. A speed control feedback loop was implemented in order to get the model to follow the speed reference.

Model verification was completed by comparing the models response to measured data that was obtained during verification measurements conducted with a physical elevator system. The measured quantities were rotor position and speed as well as elevator car position and speed. The results that were plotted were responses obtained from measuring the physical elevator with three different loading cases, 0 % (0 kg), 25 % (320 kg) and 48 % (665 kg), as well as simulating the model responses with similar loads and same input references. In addition to that the actual response plots the figures also include actual error of each response when compared to the reference signal. This allows comparing the responses on a significantly more accurate scale, rather than just comparing the overlapping response plots.

Overall the Simulink model behaved quite well and pretty much as expected. It was able to follow the speed reference with no significant issues, and the absolute error was several orders of magnitude below the measured differences. This was also true for position reference even though the model has no position or current feedback loop, meaning that the error would keep increasing over time since the velocity response had steady state error present. When ignoring the two large noise spikes in elevator car velocity measurements that were likely caused by the encoder, the maximum measured error was in the magnitude of 0.10 to 0.15 and the simulated error in the magnitude of 0.005 to 0.010.

Also the model was able to output simulated data that could not be measured from the physical elevator, such as stator rotational vibrations, actual torque affecting the rotor, and hoisting rope stiffness. Even though this data cannot be used as-is since it cannot be verified without additional physical sensors and measurements, it serves as a proof that such aspects of the model could very well be suitable for applying the virtual sensor concept.

Several improvement and additional development ideas were found for the model that, while not currently present in the simulation model, can eventually be used to improve its accuracy and suitability for various tasks. Implementing a position controller would improve the position responses of the model. This however would require studying the control logic that is currently in place in the elevators, and therefore was not feasible to be considered in terms of this project. Another smaller improvement to the torque block would be to apply some kind of limiting elements to it based on actual motor properties, so that the simulated maximum torque would not exceed the capabilities of the actual axial flux motor.

Smaller additions that could also be made to improve the current model include aspects such as not ignoring all friction behavior between the traction sheave and the rope, such as the rope slipping when the elevator accelerates or brakes, however this was also outside of the scope of the thesis.

Another smaller addition would be to include any imbalances of the rotor in the model by simulating a sinusoidal harmonic force affecting to the stator frame. This would allow the model to simulate the effects of such imbalance on for example the elevator car vibrations. However mirroring such imbalance in the model would require measurements of the physical rotor in order to determine the amplitude of the imbalance, and as such is not really feasible for studying specific equipment properties, but rather just for general effect.

REFERENCES

Al-Ali, A. R., Gupta, R. & Al Nabulsi, A. 2018. Cyber Physical Systems Role in Manufacturing Technologies. AIP Conference Proceedings, Vol. 1957, Iss. 1. pp. 1-7.

Al-Sharif, L., Tutunji, T. A., Ragab, D. & Kayfi, R. 2014. Using Elevator System Modelling and Simulation for Integrated Learning in Mechatronics Engineering. 15th International Workshop on Research and Education in Mechatronics, pp. 1-8.

Arrasate, X., Kaczmarczyk, S., Almandoz, G., Abete, J. M. & Isasa, I. 2013. The modelling, simulation and experimental testing of the dynamic responses of an elevator system. Mechanical Systems and Signal Processing, Vol. 42, pp. 258-282.

Bishop, R.H. 2008. The Mechatronics Handbook, Second Edition, Mechatronic systems, sensors, and actuators: fundamentals and modeling. CRC Press, 692 p.

Esteban, E., Salgado, O., Iturrospe, A. & Isasa, I. 2016. Model-based approach for elevator performance estimation. Mechanical Systems and Signal Processing, Vol. 68-69, pp. 125-137.

Fortuna, L., Graziani, S., Rizzo, A. & Xibilia, M. G. 2007. Soft Sensors for Monitoring and Control of Industrial Processes. Springer-Verlag, 284 p.

Gutierrez, H., Javani, B. S., Kirk, D., Su, W., Wolf, M. & Griffin, E. 2016. Fiber Optic Sensor Arrays for Real-Time Virtual Instrumentation and Control of Flexible Structures. Structural Health Monitoring, Damage Detection & Mechatronics, Vol. 7, pp. 9-22.

Haag, S. & Anderl, R. 2017. Digital twin – Proof of concept. Manufacturing Letters, Vol. 15, pp. 64-66.

Hehenberger, P. & Bradley, D. 2016. Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers. Springer International Publishing, 259 p.

Inamdar, S., Allemang, R. J. & Phillips, A. W. 2016. Estimation of Frequency and Damping of a Rotating System Using MEOT and Virtual Sensor Concept. Special Topics in Structural Dynamics, Vol. 6, pp. 23-35.

Inman, D. J. 2001. Engineering Vibration, second edition, Prentice-Hall, Inc., 621 p.

KONE 2001. Monospace brochure. [Web document] Published 2001 [Referenced 5.11.2018] Available at: https://toolbox.kone.com/media/mpb/tools/quicktraffic2/UI_prototype/Concept%20Proto/Concept%20Proto%20Demo%20HTML/pdf/monospace_brochure.pdf

Miller, A. M., Alvarez, R. & Hartman, N. 2018. Towards and extended model-based definition for the digital twin. Computer-Aided Design & Applications, Vol. 15, Iss. 6, pp. 880-891.

Moreno, A., Velez, G., Ardanza, A., Barandiaran, I., Infante, A. R. & Chopitea, R. 2016. Virtualisation process of a sheet metal punching machine within the Industry 4.0 vision. International Journal on Interactive Design and Manufacturing, Vol. 11, Iss. 1, pp. 365-373.

Negri, E., Fumagalli, L. & Macchi, M. 2017. A review of the roles of Digital Twin in CPS-based production systems. Procedia Manufacturing, Vol 11. pp. 939-948.

Oosterom, M. & Babuska, R. 2000. Virtual sensor for fault detection and isolation in flight control systems - fuzzy modeling approach. Proceedings of the 39th IEEE Conference on Decision and Control, pp. 2645-2650

Rodic, B. 2017. Industry 4.0 and the New Simulation Modelling Paradigm. Organizacija, Vol. 50, Iss. 3, pp. 193-207.

Shenghui, P., Chuan, L., Menghe, L. & Lezhu, C. 2011. Virtual Sensor for Vehicle Sideslip Angle Based on Extended Kalman Filter. 2011 Third International Conference on Measuring Technology and Mechatronics Automation, pp. 1131-1134.

Sul, S. K. 2011. Control of Electric Machine Drive Systems. IEEE Press series on power engineering, 401 p.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H. & Sui, F. 2017. Digital twin-driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology, Vol. 94, Iss. 9-12, pp. 3563-3576.

Tao, F., Zhang, M., Liu, Y. & Nee, A. Y. C. 2018. Digital twin driven prognostics and health management for complex equipment. CIRP Annals – Manufacturing Technology, Vol. 67, pp. 169-172.

TimeSys 2002. The Concise Handbook Of Real-Time Systems. TimeSys Corporation, 65 p.