NTNU
Norwegian University of
Science and Technology

Investigating an approach for Estimation of Forces using Inverse FEA Utilizing Digital Twin

-

Lars Markus Eie

Mechanical Engineering
Submission date: June 2018
Supervisor: Terje Rølvåg, MTP

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

MASTER 2018
FOR
STUD.TECHN. LARS MARKUS EIE

DIGITAL TWIN TOWER

Digitalt Tvilling Tårn

Fedem Technology AS is developing digital twin solutions for predictive maintenance and monitoring of structural integrity. The company has therefore developed hardware and software solutions for instrumentation of physical structures and mechanisms. These solutions are currently in a prototype phase and Fedem Technology AS wants to benchmark these systems on physical twins.

Tasks include:

1. Modify the tower (inspect and remove local damages/imperfections)
2. Update the tower FEM according to the modifications in task 1
3. Identify sensors and locations to detect applied tower loads on the physical tower twin
4. Connect the virtual and physical tower twin
5. Develop methods to synthesize applied loads based on physical sensor outputs
6. Apply loads on the physical twin and evaluate how well the sensors are capturing the load based on the deflections of the digital Fedem twin

If time permits:

7. Introduce imperfections to benchmark the digital twins ability to monitor structural integrity
8. Write a scientific digital twin paper with the supervisors

Contact:
At the department (supervisor, co-supervisor): Terje Rølvåg and Bjørn Haugen
From Fedem Technology AS: Stein Ove Erikstad and Runar H. Refsnæs

Abstract

The industry is moving quickly into a new era, the era of *Internet of Things* and *Industry 4.0*. This means that everything should have some sort of virtual information database where information such as environmal factors should be stored. For embedded systems the connection to internet is not a very far-fetched idea, but for structural object, such as a bridge or a windmill, the method for connecting it to internet is not that straight forward. The solution to this is the so called *Digital Twin*. A digital twin is an as-built Finite Element Model (FEM) which runes a continous real-time Finite Element Analysis (FEA) by using sensor inputs from the real life structure. The information will then be stored, and it is possible to view the load history on the structure or each substructure within a superstructure. Using a Digital Twin the predictive maintenance can hopefully be estimated with great confidence. In this thesis it has been investigated and approach for how to estimate a force (its direction and magnitude), and how to further implement this in Fedem. A comparison of real strain gauge values versus the ones calculated from the input force in Fedem has been done.

Sammendrag

Industrien er på vei inn i en ny era, eraen av *Internet of Things* og *Industry 4.0.* Det betyr at alt skal en form for virtuel informasjonsdatabase, hvor informasjon som miljøet skal være lagret. For innebyde systemer er det ikke så vanskelig å tenke seg til hvordan den kan kobles opp mot internett, men for konstruksjoner, en bru eller vindmølle, er ikke fremgangsmåten så rett frem. Løsningen på nevnte problmen er en så kalt *Digital Tvilling.* En digital tvilling er en as-built Finite Element modell som kjører en kontinuerlig Finite Element analyse ved å brukte sensorverdier fra den ekte konstruksjonen. Informasjonen vil så bli lagret og det vil være mulig å se lasthistorikken av strukturen eller små komponenter i en stor konstruksjon. Ved å bruke en Digital Tvilling kan man med stor sikkerhet anslå når det vil være tid for å gjøre vedlike på ulike komponenter. I denne oppgaven har det blitt sett på en mulig løsning for å anslå laster en struktur blir utsatt for, det være seg retning og størrelse. Tøyningsverdier fra den ekte strukturen har blitt sammenlignet med tøyningsverdier beregnet i Fedem ved bruk av estimert kraft.

Preface

This master thesis is the final dissertation for the M.Sc. degree at the Department of Design and Materials at Norwegian University of Science and Technology.

I would like to thank my supervisor Terje Rølvåg and Fedem Technology for guiding me through this project and giving me freedom to choose approach as I saw fitted, and allowing me to write a thesis on such an interesting topic. I would also like to thank co-supervisor Bjørn Haugen for discussion of the mechanics of the tower and guiding me trough the selected approach.

Lastly I would like to thank Halvard Støwer for both helping and teaching me how to mount and connect the sensors. Several hours were used in both preparing and installing sensors, and troubleshooting the system would have been impossible without him.

I am grateful for every hour these people has dedicated to this thesis.

Contents

Nomenclature

γ	Forgetting factor
ω	Frequency
$\mathbf{A}$	Transition matrix
$\mathbf{A}_c$	Continuous-time system matrix
$\mathbf{B}$	Input-to-space matrix
$\mathbf{B}_c$	Continuous-time state to input matrix
$\mathbf{B}_s$	Sensitivity matrix
$\mathbf{C}$	State to output matrix
$\mathbf{C}_\xi$	Damping matrix
$\mathbf{F}$	Input force vector
$\mathbf{G}$	Process noise to state matrix
$\mathbf{K}$	Stiffness matrix
$\mathbf{K}_a$	Kalman gain
$\mathbf{K}_s$	Correction gain
$\mathbf{M}$	Mass matrix
$\mathbf{M}_s$	Sensitivity matrix
$\mathbf{P}$	Filters error covariance matrix
$\mathbf{P}_s$	Error covariance matrix for input estimator
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Measurement noise covariance matrix
$\mathbf{S}$	Innovation covariance
$\mathbf{u}$	Displacement vector
$\mathbf{x}$	State vector

$\mathbf{y}$	Measurement matrix
$\tilde{\mathbf{y}}$	Innovation
ξ	Damping ratio
C_ξ	Damping coefficient
K	Stiffness coefficient
M	Mass

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

The knowledge of dynamic forces acting on a mechanical structure is a vital concept for design. Structural design, codes, dynamic analysis, risk and reliability all rely on knowledge and characterization of external loads. Structural health monitoring, fatigue analysis and life estimation is all regarded as the premise of structural design and optimization. Forces occurring in service conditions can induce non-visible damage in structures. In engineering applications, input forces are quite hard or even impossible to measure directly, that includes loads from wind, waves, earthquakes, airplane wings deflections, windmill blades and so on. Measurement of system response is far more convenient in such cases. Estimating of loads based on system response is known as an inverse problem. This thesis is written in cooperation with Fedem Technology which is developing a digital twin solution for predictive maintenance and structural health monitoring.

“A digital twin is a software construct that bridges physical systems and the digital world” [1]. One could say it is the missing link between physical systems and the computer model. As of 2018 there are no consensus on the definition of the word. Siemens uses the word as a virtual model in virtual world used to validate and optimize industrial processes (i.g. welding, robot motion etc.) [2]. In this project the Fedem’s definition of digital twin will be investigated, in this concept the digital twin uses input in from a real structure and by the use of actuator deforms a FE model, thereby a full state of the structure is calculated including strain/stresses, load history etc.

Some of the potential uses for a digital twin includes, but are not limited to [4]:

- Validating design parameters of the model with the real-world.
- Warn anomalies in the operating conditions.
- Identify root causes for failures.
- Design feedback.
- Discover new ways to operate the system.

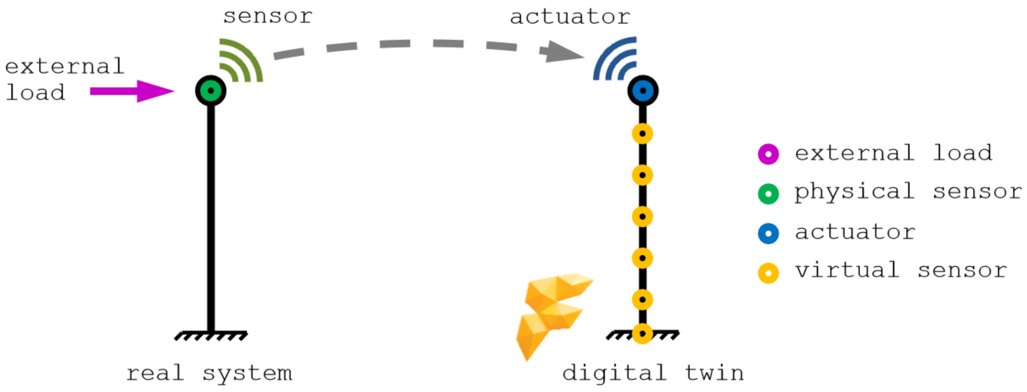

Figure 1.1: Digital twin concept of Fedem. Picture retrived from [3].

1.2 Scope of work

Digital twin is still in a developing phase, it was therefore desirable to look at it in a different approach; an "inverse FEA". That means investigating how it would be possible to identify what kind of load (magnitude, direction etc.) the structure was exposed to based on the response of the system. The tasks for this project was as followed:

- Modify the tower (inspect and remove local damages/imperfections)
- Update the tower FEM according to the modifications in task 1
- Identify sensors and locations to detect applied tower loads on the physical tower twin
- Connect the virtual and physical tower twin
- Develop methods to synthesize applied loads based on physical sensor outputs
- Apply loads on the physical twin and evaluate how well the sensors are capturing the load based on the deflections of the digital Fedem twin

If time permits:

- Introduce imperfections to benchmark the digital twins ability to monitor structural integrity
- Write a scientific digital twin paper with the supervisors

This thesis will mainly be the work of others, with the exception of the implementation on a digital twin. It has been tried to keep the approach as general as possible such that it can be of value to other structures far more complex than what is presented here. The filter itself has not been investigated that means advanced load cases such as rapid changing force, forces applied to several nodes and reduction of sensors has not been preformed.

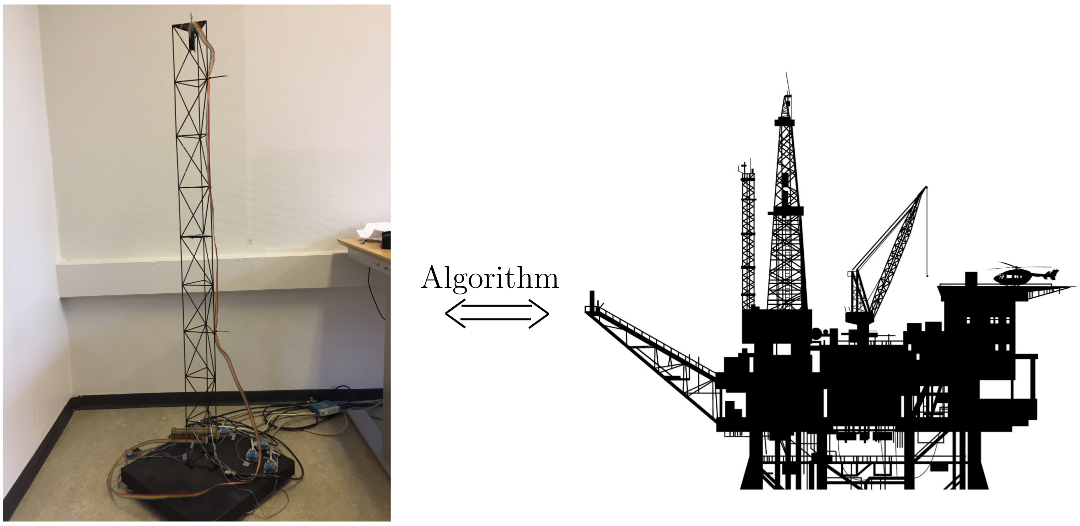

Figure 1.2: The same algorithm applied in this thesis is applicable to any structure as long as state can be measured.

Chapter 2

Theoretical bacground

2.1 Strain gauges

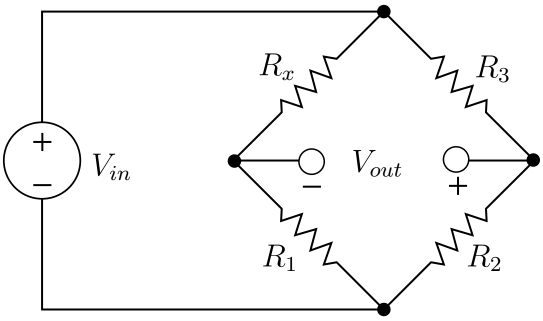

Figure 2.1: A Wheatstone bridge

A strain gauge is a device with a thin conductive strips in a zig-zag pattern. If the strain gauge is subjected to a tensile strain the cross section of the strip is reduced, hence increasing the resistance. For a gauge subjected to compression the opposite is true. To measure the voltage drop over the strain gauge a *Wheatstone bridge* is used. This allows for very accurate measurement of the change in voltage. A Wheatstone bridge consist of four resistors oriented as shown in figure 2.1. The voltage V_{in} is applied across the bridge and then the voltage is measured at V_{out}. Since this is a uni-axially loaded member only a quarter bridge is needed R_x, that means a dummy resistors are needed in parallel to the strain gauge. In this case a $120\,\Omega$ resistor were put in parallel to the strain gauge while the two remaining resistors to complete the Wheatstone bridge are located inside *Spider8*. Strain gauges has a fundamental parameter called *gauge factor* (GF) and is defined as followed

$$\mathrm{GF} = \frac{\Delta R/R}{\Delta L/L} = \frac{\Delta R/R}{\varepsilon}$$
$$V = IR$$
$$\therefore \varepsilon \propto \Delta V$$

2.2 Accelerometer

An accelerometer is a device which measures the acceleration experienced by the device. It has three axis which are oriented orthogonal. An accelerometer resting on a surface at earth will measure one g, it should therefore be oriented in such a way that the gravitational component only acts on one of the axis, that way it does not influence accelerations measured by the two remaining axes. By integrating the acceleration twice it is mathematically possible to find the distance traveled. This have however proved to be a challenging task due to drifting and despite correction for drifting and both low and high pass filters it has yet to deem satisfactorily results [5]. Accelerometer should therefore only be used for the purpose of measuring acceleration, and for the reasons just mention it will not be used for state estimations in this thesis since the load rates will be of such low magnitudes that inertia of the structure is negligible. An accelerometer will however be mounted to the structure and be used to obtain the natural frequency and damping of the structure.

2.3 State space

Any linear structural system consisting of η degrees of freedom subjected to a time varying force $F(t)$ can be express by the following ordinary linear differential equation for describing the behavior of a structure

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}_\xi\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{F}(t), \tag{2.1}$$

where $\mathbf{M}$ denotes the $\eta \times \eta$ mass matrix, $\mathbf{C}_\xi$ the $\eta \times \eta$ damping matrix, $\mathbf{K}$ the $\eta \times \eta$ stiffness matrix, $\mathbf{F}$ the $\eta \times 1$ input force vector. $\mathbf{u}$ is the $\eta \times 1$ displacement vector, and the dots represents differentiation with respect to time.

By using the redundant equation $\dot{\mathbf{u}} - \dot{\mathbf{u}} = 0$ the equation 2.1 can be written as

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \ddot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{C}_\xi \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} [\mathbf{F}] \tag{2.2}$$

thus

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \ddot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C}_\xi \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} [\mathbf{F}] \tag{2.3}$$

By defining

$$\mathbf{x}(t) \overset{\text{def}}{=} \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \tag{2.4}$$

where $\mathbf{x}(t)$ is the state vector of size $n = 2\eta$. The equation can now be converted to a first order differential equation state-space model

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{F}(t) \tag{2.5}$$

where

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_{\eta\times\eta} & \mathbf{I}_{\eta\times\eta} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C}_\xi \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \mathbf{0}_{\eta\times\eta} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{2.6}$$

$\mathbf{A}_c$ is known as the continuous-time system matrix and $\mathbf{B}_c$ as the continuous-time state-to-input matrix. The solution to equation 2.5 at any give time t is the following convolution integral [6]

$$\mathbf{x}(t) = e^{\mathbf{A}_c t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}_c t}\mathbf{B}\mathbf{F}(\tau)\mathrm{d}\tau \tag{2.7}$$

where $\mathbf{x}(0)$ is the initial state.

2.3.1 Discrete Time State-Space Representation

Measurements are taken at distinct points in time, therefore a discrete formulation of equation 2.7 is required. Defining a time step Δt and assuming that that the state response at time $t = k\Delta t$ is known then the step $k+1$ is given as

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} e^{\mathbf{A}_c((k+1)\Delta t-\tau)}\mathbf{B}_c\mathbf{F}(\tau)\mathrm{d}\tau \tag{2.8}$$

where $\mathbf{A} = e^{\mathbf{A}_c\Delta t}$. Since the force is unknown, assumptions about the force needs to be made in order to solve the integral. Two common assumptions that is made is either *Zero Order Hold* or *First Order Hold*, meaning that the force is kept constant or is linearly increasing from one time step to the next step respectively. Zero Order Hold is only suited for moderately changing forces or if the sampling rate is high compared to the force impact rate. Zero Order Hold is defined as followed [6]

$$\mathbf{B} = \int_{k\Delta t}^{(k+1)\Delta t} e^{\mathbf{A}_c((k+1)\Delta t-\tau)}\mathbf{B}_c\mathrm{d}\tau = (\mathbf{A}-\mathbf{I})\mathbf{A}_c^{-1}\mathbf{B}_c \tag{2.9}$$

2.4 Kalman filter

The Kalman filter has been used in different applications for more than 50 years and its most famous and early uses was in the navigation system for Apollo 11. The filter combines a theoretical mathematical model with real life measurements, it smooths the noisy measurements data and provides estimations for the parameters of interest. Some of the key benefits of the filter is that it does not need to keep any history other than the previous state and the calculations are very fast, this makes the filter suited for real time calculations.

2.4.1 Derivation of the filter

True state in discrete time

In the following subsection the discrete Kalman filter for a linear system is derived. It is assumed that there is a finite number of measurable quantities which gives the state $\mathbf{x}$. Considering a system subjected to deterministic input $\mathbf{u}(t)$, unmeasured disturbances

$\mathbf{w}(t)$ and the system has measurements $\mathbf{y}(t)$ that is linearly related to the state vector $\mathbf{x}(t)$. The state is considered to be without uncertainties i.e., the state is the "true" state. In its most general form the state and measurement equations are as followed

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k \tag{2.10}$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{2.11}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times r}$ and $\mathbf{G} \in \mathbb{R}^{n\times r}$ are transition, input to space and process noise to state matrices respectively. $\mathbf{C} \in \mathbb{R}^{m\times n}$ is the state to output matrix and is the relationship between state matrix and the measurement vector $\mathbf{y}_k \in \mathbb{R}^{m\times 1}$. $\mathbf{w}_k \in \mathbb{R}^{r\times 1}$ is the process noise and $\mathbf{v}_k \in \mathbb{R}^{m\times 1}$ is the measurement noise. It is assumed that both noise vectors are Gaussian white noise with zero mean and known covariance, namely

$$\mathbb{E}[\mathbf{w}_k] = 0$$
$$\mathbb{E}[\mathbf{v}_k] = 0$$

and

$$\mathbb{E}[\mathbf{w}_k\mathbf{w}_j^{\mathrm{T}}] = \mathbf{Q}\delta_{kj}$$
$$\mathbb{E}[\mathbf{v}_k\mathbf{v}_j^{\mathrm{T}}] = \mathbf{R}\delta_{kj}$$
$$\mathbb{E}[\mathbf{w}_k\mathbf{v}_j^{\mathrm{T}}] = 0$$

where $\mathbb{E}[\cdot]$ is the expected value and δ_{kj} is the Kronecker delta function. $\mathbf{Q}$ and $\mathbf{R}$ represents the process and measurements noise covariance matrices respectively. It is also assumed that the state vector is uncorrelated with the process and measurement noise, which means

$$\mathbb{E}[\mathbf{x}_k\mathbf{v}_j^{\mathrm{T}}] = 0$$
$$\mathbb{E}[\mathbf{x}_k\mathbf{w}_j^{\mathrm{T}}] = 0$$

Given the same system as before, but this time the state is an estimate. The state estimator has the form

$$\hat{\mathbf{x}}_{k+1} = \mathbf{H}_k\hat{\mathbf{x}}_k + \mathbf{Z}_k\mathbf{u}_k + \mathbf{L}_k\mathbf{y}_k \tag{2.12}$$

where $\hat{\mathbf{x}}_k$ is the estimator of $\mathbf{x}$. The matrix $\mathbf{L}_k$ is the observer gain, that is how much influence the measurements should have on the estimate. The matrices $\mathbf{H}_k$ and $\mathbf{Z}_k$ has no physical meaning, they are chosen to maximize the estimated state. The difference between the estimate and the true value is the error, therefore the error at step k can be expressed as

$$\boldsymbol{\varepsilon}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$$

where ε denotes the error. The same expression is true for step $k+1$, and if the expressions for $\mathbf{x}_{k+1}$ and $\hat{\mathbf{x}}_{k+1}$ is inserted we get

$$\boldsymbol{\varepsilon}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k - \mathbf{H}_k\hat{\mathbf{x}}_k - \mathbf{Z}_k\mathbf{u}_k - \mathbf{L}_k\mathbf{y}_k$$

Substituting in equation 2.11 and collecting terms

$$\boldsymbol{\varepsilon}_{k+1} = (\mathbf{A} - \mathbf{L}_k\mathbf{C})\mathbf{x}_k + (\mathbf{B} - \mathbf{Z}_k)\mathbf{u}_k + \mathbf{G}\mathbf{w}_k - \mathbf{H}_k\hat{\mathbf{x}}_k - \mathbf{L}_k\mathbf{v}_k$$

Instead of using the true state, the previous error and state estimate can be used

$$\boldsymbol{\varepsilon}_{k+1} = (\mathbf{A} - \mathbf{L}_k\mathbf{C})(\boldsymbol{\varepsilon}_k + \hat{\mathbf{x}}_k) + (\mathbf{B} - \mathbf{Z}_k)\mathbf{u}_k + \mathbf{G}\mathbf{w}_k - \mathbf{H}_k\hat{\mathbf{x}}_k - \mathbf{L}_k\mathbf{v}_k$$

this can be rewritten as

$$\boldsymbol{\varepsilon}_{k+1} = (\mathbf{A} - \mathbf{L}_k\mathbf{C})\boldsymbol{\varepsilon}_k + (\mathbf{B} - \mathbf{Z}_k)\mathbf{u}_k + (\mathbf{A} - \mathbf{L}_k\mathbf{C} - \mathbf{H}_k)\hat{\mathbf{x}}_k + \mathbf{G}\mathbf{w}_k - \mathbf{L}_k\mathbf{v}_k$$

If we now take the expected value of the equation both the noise vectors will be zero because of the assumption of zero mean. Therefore the expected value of the error is

$$\mathbb{E}[\boldsymbol{\varepsilon}_{k+1}] = (\mathbf{A} - \mathbf{L}_k\mathbf{C})\mathbb{E}[\boldsymbol{\varepsilon}_k] + (\mathbf{B} - \mathbf{Z}_k)\mathbb{E}[\mathbf{u}_k] + (\mathbf{A} - \mathbf{L}_k\mathbf{C} - \mathbf{H}_k)\mathbb{E}[\hat{\mathbf{x}}_k]$$

The expected value of the estimation state vector and input vector is most likely none-zero, therefore in order to achieve zero mean of the estimation error these terms needs to vanish. To achieve the latter the following is required

$$\mathbf{Z}_k = \mathbf{B} \quad \text{and} \quad \mathbf{H}_k = \mathbf{A} - \mathbf{L}_k\mathbf{C}$$

By substituing the values for $\mathbf{Z}_k$ and $\mathbf{H}_k$ in the original state estimation equation (2.12), the equation becomes

$$\hat{\mathbf{x}}_{k+1} = (\mathbf{A} - \mathbf{L}_k\mathbf{C})\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{L}_k\mathbf{y}_k$$

Rewritten to a more known form

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{L}_k\tilde{\mathbf{y}}_k \tag{2.13}$$

where $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k$. We have now arrived at an optimal expression with physical quantities for the state estimation. The term $\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k$ is called *innovation* and is the difference between the real observer and the estimated observer.

Predictor and Corrector Estimation

The state estimator derived in section 2.4.1 is a one step prediction form. The prediction and measurement happens at the same time, that means that the measurements lags one time step. In order to improve the estimate the equation is split into two, one *predictor* happening at time $k-1$ and then an *corrector* happening at time k. This is often referred to as *prior* and *posterior* estimate.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}\mathbf{u}_{k-1} \tag{2.14}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{L}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{2.15}$$

The state is estimated from the previous step, then updated by the difference between the estimated state and the values measured in step k. In order to keep the estimation as accurate as possible we seek to minimize the error both before and after the update

$$\begin{aligned}\boldsymbol{\varepsilon}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\ \boldsymbol{\varepsilon}_k^+ &= \mathbf{x}_k - \hat{\mathbf{x}}_k^+\end{aligned}$$

We define the covariance of the estimate as

$$\begin{aligned}\mathbf{P}_k^- &= \mathbb{E}[\boldsymbol{\varepsilon}^-\boldsymbol{\varepsilon}^{-\mathrm{T}}] = \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^{\mathrm{T}}] \\ \mathbf{P}_k^+ &= \mathbb{E}[\boldsymbol{\varepsilon}^+\boldsymbol{\varepsilon}^{+\mathrm{T}}] = \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^{\mathrm{T}}]\end{aligned}$$

The approach for finding the error is analogous to the approach in section 2.4.1, thus the expressions for the states are inserted

$$\boldsymbol{\varepsilon}_k^- = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{G}\mathbf{w}_{k-1} - \mathbf{A}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}\mathbf{u}_{k-1}$$

This reduces to

$$\boldsymbol{\varepsilon}_k^- = \mathbf{A}\boldsymbol{\varepsilon}_{k-1}^+ + \mathbf{G}\mathbf{w}_{k-1}$$

Post-multiplying this equation with itself gives

$$\boldsymbol{\varepsilon}_k^-\boldsymbol{\varepsilon}_k^{-\mathrm{T}} = \mathbf{A}\boldsymbol{\varepsilon}_{k-1}^+\boldsymbol{\varepsilon}_{k-1}^{+\mathrm{T}}\mathbf{A}^{\mathrm{T}} + \mathbf{G}\mathbf{w}_{k-1}\boldsymbol{\varepsilon}_{k-1}^{+\mathrm{T}}\mathbf{A}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\varepsilon}_{k-1}^+\mathbf{w}_{k-1}^{\mathrm{T}}\mathbf{G}^{\mathrm{T}} + \mathbf{G}\mathbf{w}_{k-1}\mathbf{w}_{k-1}^{\mathrm{T}}\mathbf{G}^{\mathrm{T}}$$

The reader is reminded the noise is Gaussian and the system is linear, therefore one can state that the error is also a Gaussian random signal and since the noise is white there should be no correlation between the error and the noise. By taking the expected values of the expression just derived and applying the logic just mentioned we get the following

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} \tag{2.16}$$

In order to find an expression for the update covariance we need to go back to equation 2.15 and insert the expression for true state and measured value

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{L}_k(\mathbf{C}\mathbf{x}_k + \mathbf{v}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)$$

Substituting the expressions for error into this equation gives

$$\mathbf{x}_k - \boldsymbol{\varepsilon}_k^+ = \mathbf{x}_k - \boldsymbol{\varepsilon}_k^- + \mathbf{L}_k(\mathbf{C}\mathbf{x}_k + \mathbf{v}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)$$

thus

$$\boldsymbol{\varepsilon}_k^+ = \boldsymbol{\varepsilon}_k^- - \mathbf{L}_k(\mathbf{C}\boldsymbol{\varepsilon}_k^- + \mathbf{v}_k)$$

Now to obtain the covariance for the update step we post-multiply this equation with its transpose as was done to obtain the estimation covariance matrix.

$$\boldsymbol{\varepsilon}_k^+\boldsymbol{\varepsilon}_k^{+\mathrm{T}} = [\boldsymbol{\varepsilon}_k^- - \mathbf{L}_k\mathbf{C}\boldsymbol{\varepsilon}_k^- - \mathbf{L}_k\mathbf{v}_k][\boldsymbol{\varepsilon}_k^{-\mathrm{T}} - \boldsymbol{\varepsilon}_k^{-\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} - \mathbf{v}_k^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}}]$$

$$\begin{aligned}\boldsymbol{\varepsilon}_k^+\boldsymbol{\varepsilon}_k^{+\mathrm{T}} = \boldsymbol{\varepsilon}_k^-\boldsymbol{\varepsilon}_k^{-\mathrm{T}} - \boldsymbol{\varepsilon}_k^-\boldsymbol{\varepsilon}_k^{-\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} - \boldsymbol{\varepsilon}_k^-\mathbf{v}_k^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} - \mathbf{L}_k\mathbf{C}\boldsymbol{\varepsilon}_k^-\boldsymbol{\varepsilon}_k^{-\mathrm{T}} + \mathbf{L}_k\mathbf{C}\boldsymbol{\varepsilon}_k^-\boldsymbol{\varepsilon}_k^{-\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} + \\ \mathbf{L}_k\mathbf{C}\boldsymbol{\varepsilon}_k^-\mathbf{v}_k^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} - \mathbf{L}_k\mathbf{v}_k\boldsymbol{\varepsilon}_k^{-\mathrm{T}} + \mathbf{L}_k\mathbf{v}_k\boldsymbol{\varepsilon}_k^{-\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} + \mathbf{L}_k\mathbf{v}_k\mathbf{v}_k^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}}\end{aligned}$$

Then the expected value

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} - \mathbf{L}_k\mathbf{C}\mathbf{P}_k^- + \mathbf{L}_k\mathbf{C}\mathbf{P}_k^-\mathbf{C}^{\mathrm{T}}\mathbf{L}_k^{\mathrm{T}} + \mathbf{L}_k\mathbf{R}\mathbf{L}_k^{\mathrm{T}}$$

We seek to find an estimate which minimizes the error covariance for the update prediction, this is equivalent to minimizing the diagonal sum of covariance matrix. This is also referred to as least square solution or minimum-variance unbiased estimator. With the covariance minimized it is obvious that $\hat{\mathbf{x}}_k$ is the best possible estimation of $\mathbf{x}_k$. Therefore we seek the minimum point with respect to $\mathbf{L}_k$ as followed

$$\frac{\partial \mathrm{trace}(\mathbf{P}_k^+)}{\partial \mathbf{L}_k} = -2(\mathbf{L}_k\mathbf{P}_k^-)^{\mathrm{T}} + 2\mathbf{L}_k\mathbf{R} + 2\mathbf{L}_k\mathbf{C}\mathbf{P}_k^-\mathbf{C}^{\mathrm{T}} = 0$$

$$\mathbf{L}_k = \mathbf{P}_k^- \mathbf{C}^\mathrm{T}[\mathbf{C}\mathbf{P}_k^- \mathbf{C}^\mathrm{T} + \mathbf{R}]^{-1}$$

This expression for $\mathbf{L}_k$ is the optimal weighing for the residual, and is called *Kalman gain* and from now on redefined as $\mathbf{K}_a$.

Rewriting $\mathbf{P}_k^+$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{C}^\mathrm{T}\mathbf{K}_k^\mathrm{T} - \mathbf{K}_k\mathbf{C}\mathbf{P}_k^- + \mathbf{K}_k(\mathbf{C}\mathbf{P}_k^-\mathbf{C}^\mathrm{T} + \mathbf{R})\mathbf{K}_k^\mathrm{T}$$

Now inserting the expression for $\mathbf{K}_{a,k}$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{C}^\mathrm{T}\mathbf{K}_k^\mathrm{T} - \mathbf{K}_k\mathbf{C}\mathbf{P}_k^- + \mathbf{P}_k^-\mathbf{C}^\mathrm{T}\mathbf{K}_k^\mathrm{T}$$

Simplifying

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^- \tag{2.17}$$

All equations that complete the entire Kalman filter algorithm has now been derived and it can now be summarized

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}\mathbf{u}_{k-1} \tag{2.18a}$$
$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^\mathrm{T} + \mathbf{G}\mathbf{Q}\mathbf{G}^\mathrm{T} \tag{2.18b}$$

$$\mathbf{S}_k = \mathbf{C}\mathbf{P}_k^-\mathbf{C}^\mathrm{T} + \mathbf{R} \tag{2.19a}$$
$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^\mathrm{T}\mathbf{S}^{-1} \tag{2.19b}$$
$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^- \tag{2.19c}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{2.19d}$$

The equations 2.18 are the prediction equations, and equations 2.19 are the update equations. A visualization is shown in figure 2.2. The Kalman filter is sometimes written without the superscript $+/-$ and instead the subscript $k|k-1$, which is Bayesian notation; meaning k given the evidence of $k-1$. The convenience of Bayesian notation is that it is more evident that the filter is based around statistics, and that the state is a result of uncertainties in measurement and predictions.

2.4.2 Observability

Observability is a vital concept in control theory, and it is a measure of how well internal states can be uniquely defined based on the amount, position and type of observation the system has. Following are two definitions of observability, one for continuous-time and the second one for discrete time, directly cited from [7]:

- *A continuous-time system is observable if for any initial state $\boldsymbol{x}(0)$ and any final time $t > 0$ the initial state $\boldsymbol{x}(0)$ can be uniquely determined by knowledge of the input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$ for all $t \in [0, t]$.*

- *A discrete-time system is observable if for any initial state $\boldsymbol{x}_o$ and some final time k the initial state $\boldsymbol{x}_0$ can be uniquely determined by knowledge of the input $\boldsymbol{u}_i$ and output $\boldsymbol{y}_i$ for all $i \in [0, k]$.*

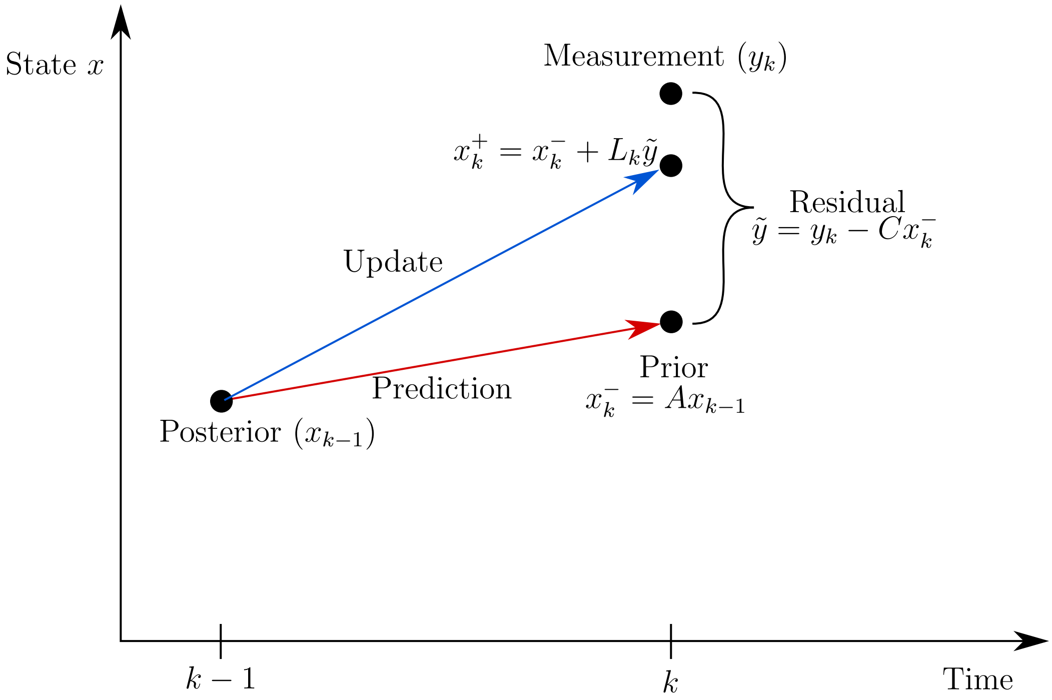

Figure 2.2: A visualization of how the filter predicts and updates the state from step $k-1$ to k

For both continuous-time and discrete time linear system the observability matrix is

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{2.20}$$

The system is observable if and only if the *rank* of $\mathcal{O}$ is equal to n.

2.5 Deriving the input estimator

Going back to equation for estimated state and letting the deterministic input be the force, that is redefining $\mathbf{u} = \mathbf{F}$. It is not possible to advance the state since one does not know what the input actually is. Tuan et al. [8] has derived an recursive least square estimation algorithm which will be derived in the following seciton. The input estimator consists of two parts: The first one is kalman filter, but without the input term and the second part is the recursive least square estimator.

$\mathbf{F}_k$ is considered to be a constant input from time n_0 to n_f

$$\mathbf{F}_k = F \tag{2.21}$$

n_o is start time of force and n_f is the final time. Introducing yet another state estimator $\bar{x}$. This is the estimator $\hat{x}$, but without the $\mathbf{BF}_{k-1}$ term. Going back to the difference between true state and $\Delta\mathbf{x} = \hat{\mathbf{x}} - \bar{\mathbf{x}}$. Letting $\Delta x = \mathbf{M}_{s,k}\mathbf{GF}$ [9] yields

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{M}_{s,k}\mathbf{GF} \tag{2.22}$$

where

$$\mathbf{M}_{s,k} = [\mathbf{I} - \mathbf{K}_{a,k}\mathbf{C}][\mathbf{AM}_{s,k-1} + \mathbf{I}] \tag{2.23}$$

The observed value of the residual $\bar{\mathbf{y}}$ with unknown force, can be relaterd to the innovation's residual sequence, $\hat{\mathbf{y}}$ i.e. if the force were known as followed

$$\bar{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{C}\bar{\mathbf{x}}_k^- = \mathbf{y}_k - \mathbf{CA}\bar{\mathbf{x}}_{k-1}^+ \tag{2.24}$$

and

$$\hat{\mathbf{y}}_k = \mathbf{y}_k - C\hat{\mathbf{x}}_k^- = \mathbf{y}_k - \mathbf{CA}\hat{\mathbf{x}}_{k-1}^+ - \mathbf{CGF}_{k-1} \tag{2.25}$$

By subtracting 2.25 from 2.24 and using the assumption of constant force we get the following

$$\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_k = \mathbf{CA}\hat{\mathbf{x}}_{k-1}^+ - \mathbf{CA}\bar{\mathbf{x}}_{k-1}^+ + \mathbf{CGF}_{k-1} \tag{2.26}$$

by factoring out and using the defined value for Δx we get the following

$$\bar{\mathbf{y}}_k = \hat{\mathbf{y}}_k + \mathbf{C}[\mathbf{AM}_{s,k-1} + \mathbf{I}]\mathbf{GF} \tag{2.27}$$

Defining $\mathbf{B}_{s,k} = \mathbf{C}[\mathbf{AM}_{s,k-1} + \mathbf{I}]\mathbf{G}$ the expression can be made more pleasing

$$\bar{\mathbf{y}}_k = \hat{\mathbf{y}}_k + \mathbf{B}_{s,k}\mathbf{F} \tag{2.28}$$

Letting $k = n+1, n+2, \ldots, n+l$, equation 2.28 can be written as

$$\mathbf{Y}_N = \mathbf{B}_{s,N}\mathbf{F} + \boldsymbol{\epsilon}_N \tag{2.29}$$

where

$$\mathbf{Y}_N = \begin{bmatrix} \bar{\mathbf{y}}_{n+1} & \bar{\mathbf{y}}_{n+2} & \cdots & \bar{\mathbf{y}}_{n+l} \end{bmatrix}^{\mathrm{T}}$$
$$\boldsymbol{\epsilon}_N = \begin{bmatrix} \hat{\mathbf{y}}_{n+1} & \hat{\mathbf{y}}_{n+2} & \cdots & \hat{\mathbf{y}}_{n+l} \end{bmatrix}^{\mathrm{T}}$$

and

$$\mathbf{B}_{s,N} = \begin{bmatrix} \mathbf{B}_{s,n+1} \\ \mathbf{B}_{s,n+2} \\ \vdots \\ \mathbf{B}_{s,n+l} \end{bmatrix} = \begin{bmatrix} \mathbf{CG} \\ \mathbf{C}[\mathbf{AM}_{s,n+1} + \mathbf{I}]\mathbf{G} \\ \vdots \\ \mathbf{C}[\mathbf{AM}_{s,n+l-1} + \mathbf{I}]\mathbf{G} \end{bmatrix} \tag{2.30}$$

The covariance of the innovation $\hat{\mathbf{y}}_k$ is $\mathbf{S}_k$, more precisely

$$\mathbb{E}[\hat{\mathbf{y}}_k\hat{\mathbf{y}}_k^{\mathrm{T}}] = \mathbf{S}_k$$

By the same logic

$$\mathbb{E}[\boldsymbol{\epsilon}_N\boldsymbol{\epsilon}_N^{\mathrm{T}}] = \boldsymbol{\Sigma}_N$$

where

$$\boldsymbol{\Sigma}_N = \begin{bmatrix} \mathbf{S}_{n+1} & 0 & \ldots & 0 \\ 0 & \mathbf{S}_{n+2} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \ldots & \mathbf{S}_{n+l} \end{bmatrix} \tag{2.31}$$

The problem is to estimate the unknown input $\mathbf{F}$ in equation 2.29. Using the least-squares estimation from [10], the input estimation is given as

$$\hat{\mathbf{F}}_N = [\mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{B}_{s,N}]^{-1} \, \mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{Y}_N \tag{2.32}$$

The covariance of the estimation error is gives as

$$\mathbf{P}_{s,N} = \mathbb{E}[(\mathbf{F} - \hat{\mathbf{F}}_N)(\mathbf{F} - \hat{\mathbf{F}}_N)^{\mathrm{T}}] = [\mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{B}_{s,N}]^{-1} \tag{2.33}$$

The matrix in equation 2.31 can be rewritten in terms of a fading memory factor $\gamma, 0 < \gamma < 1$. This method is well suited for real-time algorithms because the distant history will eventually fade away and the most recent estimation will weigh more than the older ones. The weighing matrix is as followed

$$\mathbf{\Sigma}_N^{-1} = \begin{bmatrix} \mathbf{S}_{n+1}^{-1} \gamma^{\,l-1} & 0 & \dots & 0 \\ 0 & \mathbf{S}_{n+2}^{-1} \gamma^{\,l-2} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \mathbf{S}_{n+l}^{-1} \end{bmatrix} \tag{2.34}$$

When new measurements are made at time $k = n + l + 1$ equation 2.28 becomes

$$\bar{\mathbf{y}}_{N+1} = \hat{\mathbf{y}}_{N+1} + \mathbf{B}_{s,N+1} \mathbf{F} \tag{2.35}$$

thus

$$\mathbf{Y}_{N+1} = \mathbf{B}_{s,N+1} \mathbf{F} + \boldsymbol{\epsilon}_{N+1} \tag{2.36}$$

where

$$\mathbf{Y}_{N+1} = \begin{bmatrix} \mathbf{Y}_N \\ \bar{\mathbf{y}}_{N+1} \end{bmatrix} \tag{2.37}$$

$$\mathbf{B}_{s,N+1} = \begin{bmatrix} \mathbf{B}_{s,N} \\ \mathbf{B}_{s,N+1} \end{bmatrix} \tag{2.38}$$

$$\boldsymbol{\epsilon}_{N+1} = \begin{bmatrix} \boldsymbol{\epsilon}_N \\ \hat{\mathbf{y}}_{N+1} \end{bmatrix} \tag{2.39}$$

and the weighing matrix

$$\mathbf{\Sigma}_{N+1}^{-1} = \begin{bmatrix} \mathbf{\Sigma}_{N+1}^{-1} & 0 \\ 0 & \mathbf{S}_{N+1}^{-1} \end{bmatrix} \tag{2.40}$$

Now the estimation for the next measurement

$$\hat{\mathbf{F}}_{N+1} = [\mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{\Sigma}_{N+1}^{-1} \mathbf{B}_{s,N+1}]^{-1} \, \mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{\Sigma}_{N+1}^{-1} \mathbf{Y}_{N+1} \tag{2.41}$$

By inserting equations 2.37,2.38 and 2.39 in equation 2.41 yields

$$\begin{aligned} \hat{\mathbf{F}}_{N+1} = [\gamma \mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{B}_{s,N} + \mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{S}_{N+1}^{-1} \mathbf{B}_{s,N+1}]^{-1} \\ \times [\gamma \mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{Y}_N + \mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{S}_{N+1}^{-1} \bar{\mathbf{y}}_{n+1}] \end{aligned} \tag{2.42}$$

Inserting expression for covariance

$$\hat{\mathbf{F}}_{N+1} = [\gamma \mathbf{P}_{s,N}^{-1} + \mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{S}_{N+1}^{-1} \mathbf{B}_{s,N+1}]^{-1} [\gamma \mathbf{B}_{s,N}^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{Y}_N + \mathbf{B}_{s,N+1}^{\mathrm{T}} \mathbf{S}_{N+1}^{-1} \bar{\mathbf{y}}_{n+1}] \tag{2.43}$$

now the covariance for $N+1$

$$\begin{aligned}\mathbf{P}_{s,N+1} &= [\mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{\Sigma}^{-1}_{N+1}\mathbf{B}_{s,N+1}]^{-1}\\ &= [\gamma\mathbf{B}^{\mathrm{T}}_{s,N}\mathbf{\Sigma}^{-1}_{N}\mathbf{B}_{s,N} + \mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\mathbf{B}_{s,N+1}]^{-1}\\ &= [\gamma\mathbf{P}^{-1}_{s,N} + \mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\mathbf{B}_{s,N+1}]^{-1}\end{aligned} \tag{2.44}$$

Noticing that this expression is also found in equation 2.43. By applying the inversion lemma, which can be found in[10] to equation 2.44 yields

$$\begin{aligned}\mathbf{P}_{s,N+1} = \gamma^{-1}\mathbf{P}_{s,N} - \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1}&\\ \times [\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]^{-1}\mathbf{B}_{N+1}\gamma^{-1}\mathbf{P}_{s,N}&\end{aligned} \tag{2.45}$$

substituting this back into equation 2.43 and using $\hat{F}_N = \mathbf{P}_s\mathbf{B}^{\mathrm{T}}_{s,N}\mathbf{\Sigma}^{-1}_N\bar{\mathbf{y}}_N$

$$\begin{aligned}\hat{\mathbf{F}}_{N+1} = \big[\gamma^{-1}\mathbf{P}_{s,N} - \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} \times [\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]^{-1}\mathbf{B}_{N+1}\gamma^{-1}\mathbf{P}_{s,N}\big]&\\ \times [\gamma\mathbf{B}^{\mathrm{T}}_{s,N}\mathbf{\Sigma}^{-1}_{N}\mathbf{Y}_N + \mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}]&\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{F}}_{N+1} = \hat{\mathbf{F}}_N + \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}&\\ - \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1}[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]^{-1}\mathbf{B}_{N+1}&\\ \times [\hat{\mathbf{F}}_N + \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}]&\end{aligned}$$

Now the we insert the identity

$$[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]^{-1}[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]$$

Between $\mathbf{B}_{s,N+1}$ and $\mathbf{S}^{-1}_{N+1}$ in the second term, and at the same time defining

$$\mathbf{K}_{s,N+1} = \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1}[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]^{-1} \tag{2.46}$$

we get

$$\begin{aligned}\hat{\mathbf{F}}_{N+1} = \hat{\mathbf{F}}_N + \mathbf{K}_{s,N+1}[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1} + \mathbf{S}_{N+1}]\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}&\\ - \mathbf{K}_{s,N+1}\mathbf{B}_{N+1}[\hat{\mathbf{F}}_N + \gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}]&\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{F}}_{N+1} = \hat{\mathbf{F}}_N + \mathbf{K}_{s,N+1}[\mathbf{B}_{s,N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,n+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1} + \bar{\mathbf{y}}_{n+1}&\\ - \mathbf{B}_{N+1}\hat{\mathbf{F}}_N - \mathbf{B}_{N+1}\gamma^{-1}\mathbf{P}_{s,N}\mathbf{B}^{\mathrm{T}}_{s,N+1}\mathbf{S}^{-1}_{N+1}\bar{\mathbf{y}}_{n+1}]]&\end{aligned}$$

Simplifying

$$\hat{\mathbf{F}}_{N+1} = \hat{\mathbf{F}}_N + \mathbf{K}_{s,N+1}[\bar{\mathbf{y}}_{n+1} - \mathbf{B}_{N+1}\hat{\mathbf{F}}_N] \tag{2.47}$$

By replacing $N+1$ by k, we have now arrived at a recursive least square estimation of the unknown input force by the use of the residual from Kalman filter. Now we can summarize all the steps in the algorithm for the modified state and force estimation

$$\bar{\mathbf{x}}^{-}_{k} = \mathbf{A}\bar{\mathbf{x}}^{+}_{k-1} \tag{2.48a}$$

$$\mathbf{P}^{-}_{k} = \mathbf{A}\mathbf{P}^{+}_{k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} \tag{2.48b}$$

$$\mathbf{S}_k = \mathbf{C}\mathbf{P}_k^- \mathbf{C}^\mathrm{T} + \mathbf{R} \tag{2.49a}$$
$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^\mathrm{T} \mathbf{S}^{-1} \tag{2.49b}$$
$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{C})\mathbf{P}_k^- \tag{2.49c}$$
$$\bar{\mathbf{x}}_k^+ = \bar{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\bar{\mathbf{x}}_k^-) \tag{2.49d}$$

$$\mathbf{B}_{s,k} = \mathbf{C}[\mathbf{A}\mathbf{M}_{s,k-1} + \mathbf{I}]\mathbf{G} \tag{2.50a}$$
$$\mathbf{M}_{s,k} = [\mathbf{I} - \mathbf{K}_{a,k}\mathbf{C}][\mathbf{A}\mathbf{M}_{s,k-1} + \mathbf{I}] \tag{2.50b}$$
$$\mathbf{K}_{b,k} = \gamma^{-1}\mathbf{P}_{b,k-1}\mathbf{B}_{s,k}^\mathrm{T}[\mathbf{B}_{s,k}\gamma^{-1}\mathbf{P}_{b,k-1}\mathbf{B}_{s,k}^\mathrm{T} + \mathbf{S}_k]^{-1} \tag{2.50c}$$
$$\mathbf{P}_{b,k} = [\mathbf{I} - \mathbf{K}_{b,k}\mathbf{B}_{s,k}]\gamma^{-1}\mathbf{P}_{b,k-1} \tag{2.50d}$$
$$\hat{\mathbf{F}}_k = \hat{\mathbf{F}}_{k-1} + \mathbf{K}_{b,k}[\bar{\mathbf{y}}_k - \mathbf{B}_{s,k}\hat{\mathbf{F}}_{k-1}] \tag{2.50e}$$

$\mathbf{B}_s$ and $\mathbf{M}_s$ sensitivity matrices. $\mathbf{K}_b$ correction gain, $\mathbf{P}_b$ error covariance. $\hat{\mathbf{F}}$ estimated input force vector.

2.6 Filter design considerations

The algorithm needs some initial values, for $\mathbf{P}_0$ and $\mathbf{P}_{b,0}$, C.-k. et al. [11] suggested initializing them with large numbers such as 10^6 and 10^2 respectively. By letting the error covariance have such high magnitude the first few estimates are ignored, then after a few steps the estimation results converges to their actual value rapidly, and the initial estimate becomes less important.

To better understand the effect of observation noise one can erroneously write the equation for the kalman gain as

$$\mathbf{K}_{a,k} = \lim_{R\to\infty} \frac{\mathbf{P}_k^- \mathbf{C}^\mathrm{T}}{\mathbf{C}\mathbf{P}_k^- \mathbf{C}^\mathrm{T} + \mathbf{R}} = 0 \tag{2.51}$$

meaning that for the case of $\mathbf{R} \to \infty$ the state will only be estimated from the mathematical model, so observations will have zero influence. For $R \to 0$ the opposite is true.

One must do some educated guess for $\mathbf{R}$,$\mathbf{Q}$ and γ. A lot of study has been done on it, and some of the suggestions are not easily implemented [12]. One easy way to estimate the measurement noise is to do a static measurement, and then calculate the standard deviation of the noise from the sensors, and assuming no correlation between them. The solution is not as simple for the case of $\mathbf{Q}$ and γ, often times these values are chosen by educated guesses and/or trail and error[8, 11]. Both these values affect how fast the adaptive the algorithm adapts to changes, one can for the sake of simplicity state that if the value of $\mathbf{Q}$ is huge the value of γ could be chosen as unity, this is however only suited for constant-parameter systems.

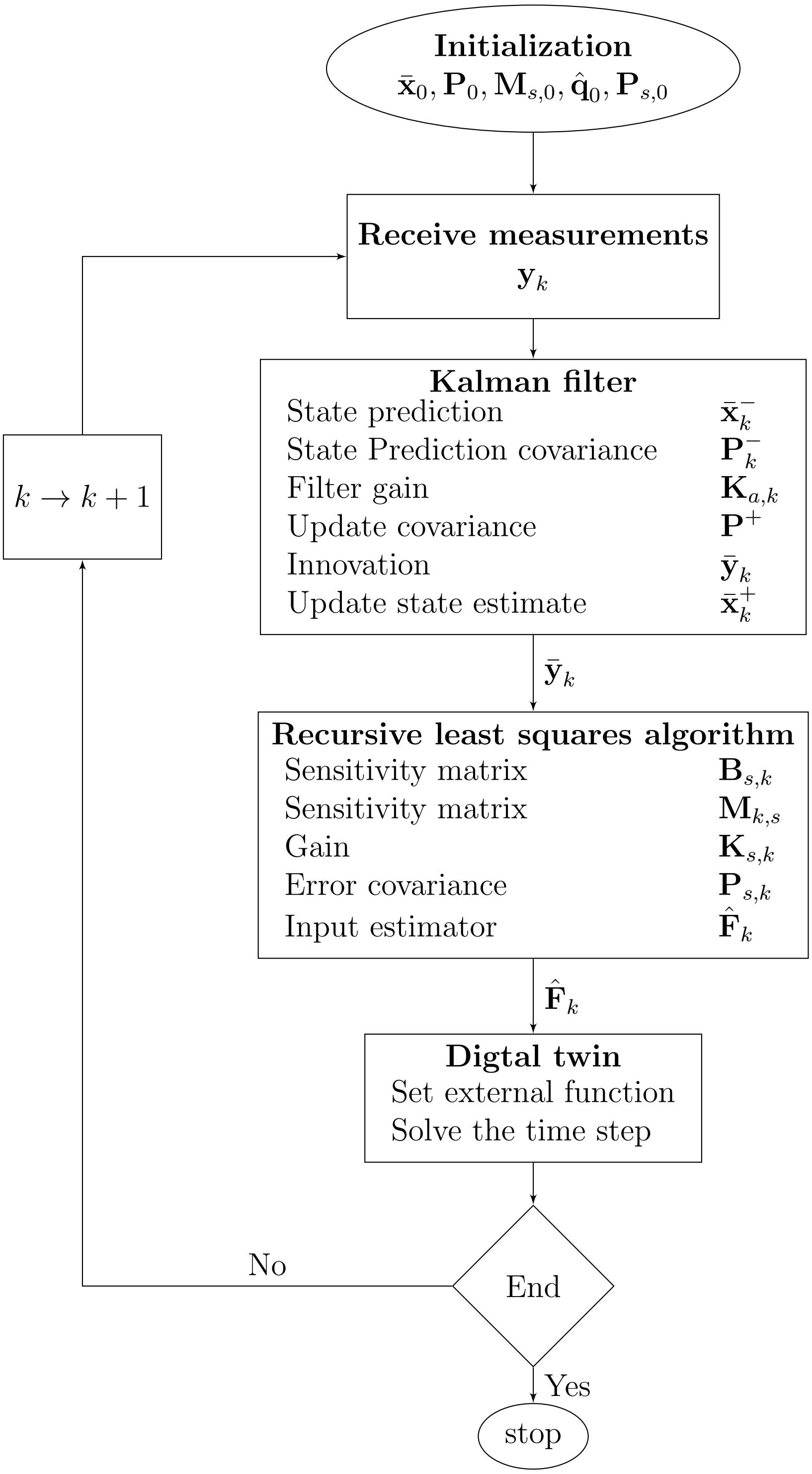
Initialization
$\bar{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{M}_{s,0}, \hat{\mathbf{q}}_0, \mathbf{P}_{s,0}$
Receive measurements
$\mathbf{y}_k$
Kalman filter
State prediction $\bar{\mathbf{x}}_k^-$
State Prediction covariance $\mathbf{P}_k^-$
Filter gain $\mathbf{K}_{a,k}$
Update covariance $\mathbf{P}^+$
Innovation $\bar{\mathbf{y}}_k$
Update state estimate $\bar{\mathbf{x}}_k^+$
$\bar{\mathbf{y}}_k$
Recursive least squares algorithm
Sensitivity matrix $\mathbf{B}_{s,k}$
Sensitivity matrix $\mathbf{M}_{k,s}$
Gain $\mathbf{K}_{s,k}$
Error covariance $\mathbf{P}_{s,k}$
Input estimator $\hat{\mathbf{F}}_k$
$\hat{\mathbf{F}}_k$
Digtal twin
Set external function
Solve the time step
End
No
Yes
stop
$k \to k+1$

Figure 2.3: Flowchart for the algorithm

Chapter 3

Experimental setup

3.1 Tower details

For this project a truss tower made of steel was used. The tower was built around 1994 and was used in a PhD investigating Active Vibration Damping of Large Flexible Space Structures. It had been moved around a few times since then, and some of the trusses was largely deformed as a result of this. At the bottom of the tower there were mounted two actuators, there were no use in these anymore so they were removed resulting in a shorter tower. The tower now had a total height of 1.65 meter. The tower is clamped to a heavy plate using screws. All trusses measures 3 mm in diameter and are welded together. The rods used to make the horizontal planes are one single rod bent to 60°.

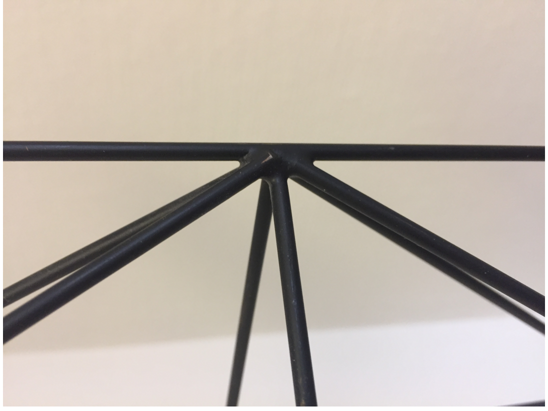

Figure 3.1: Shows the welded joint

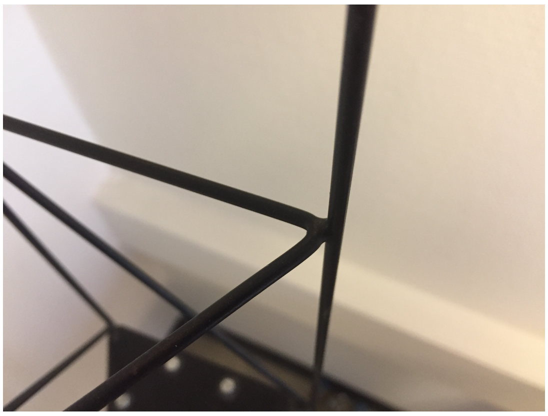

Figure 3.2: Shows the rod which is bent to 60°

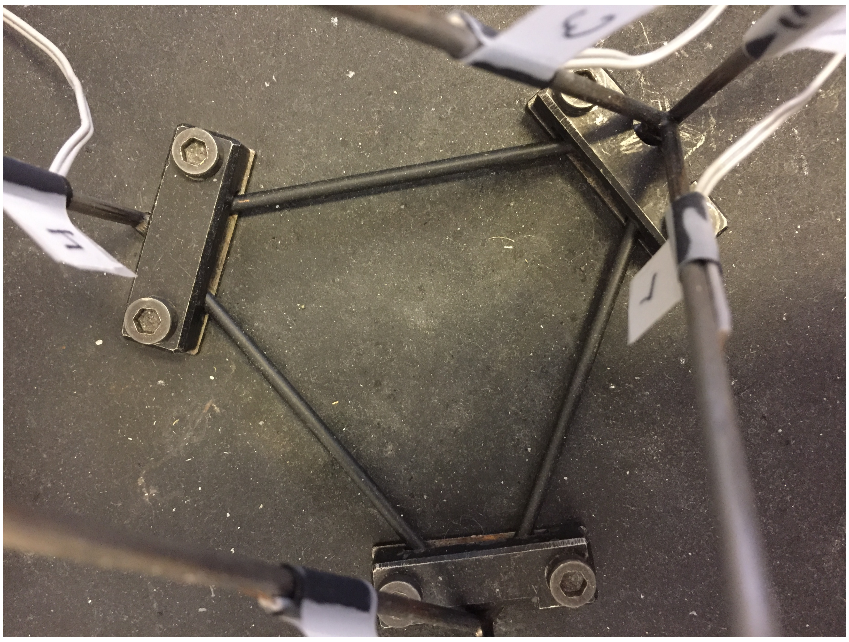

Figure 3.3: The structure is clamped to the plate using screws. The vertical rods continues about 30 mm downward into holes in the plate.

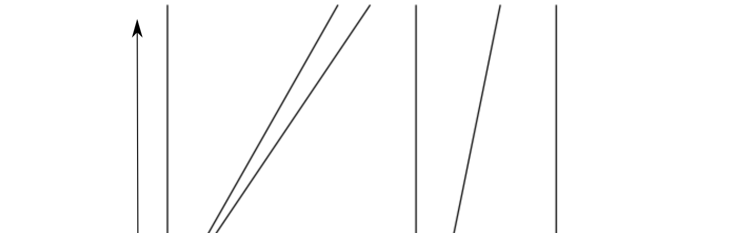

Figure 3.4: Dimensions for tower

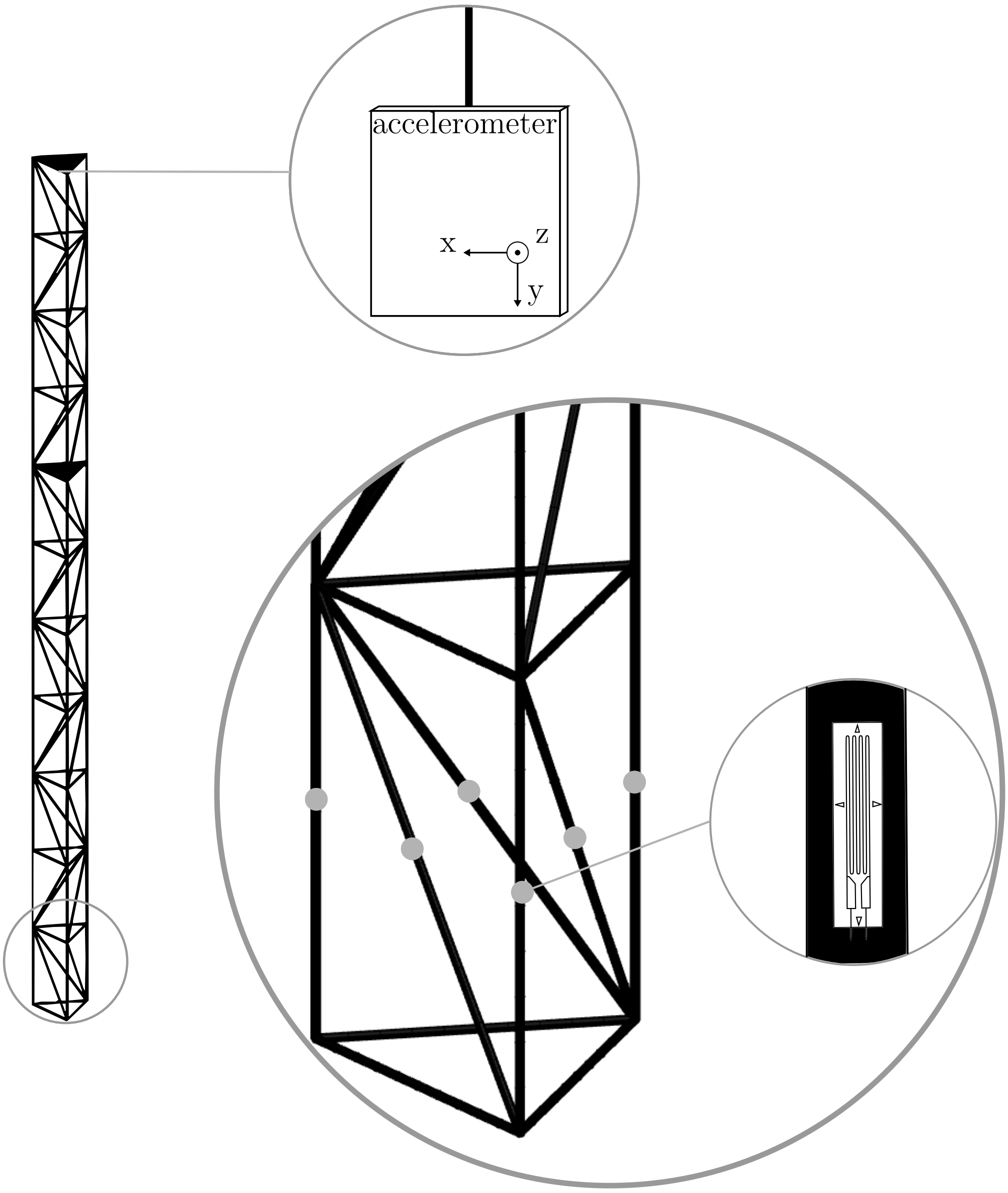

Figure 3.5: Locations for strain gauges and accelerometer

3.2 Strain gauges

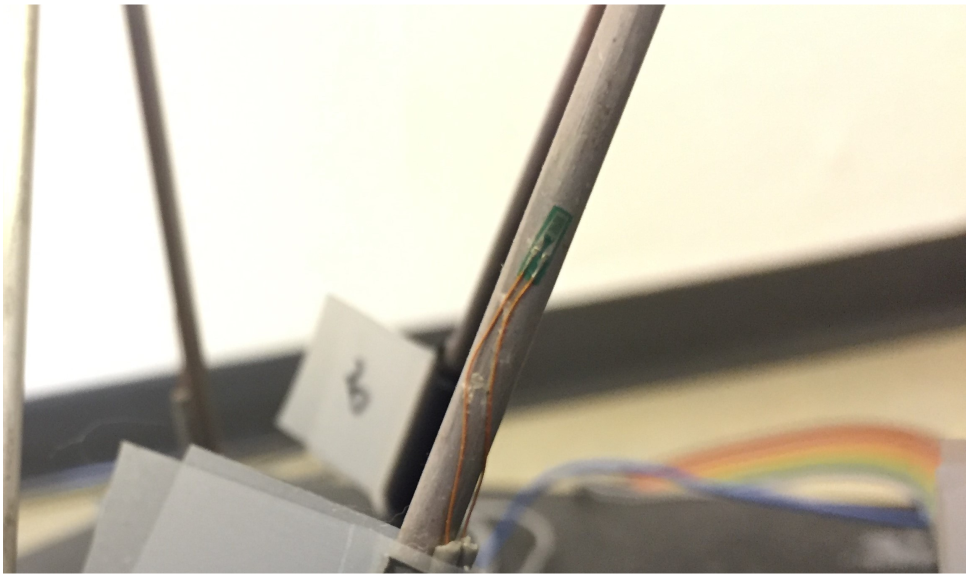

Figure 3.6: Shows the polished surface with strain gauge mounted

Before the strain gauges can be mounted on the structure the surfaces of the bonding areas needs to be prepared thoroughly. All of surface finishes (i.e. paint), dirt and other pollutants has to be removed, this was done using sandpaper. To increase the bonding forces it is recommended that the surface should then be roughened to enlarge the surface contact area, this was not done and it is believed that the sandpaper was of such high gradient that the surface was already rough. The area was then cleaned with Acetone in order to remove grease and dust from the sanding. A glue suited for gluing strain gauges to steel was used and then, using a sheet of Teflon, pressure was added for approximately one minute. Figure 3.6 shows the finished product.

3.3 Accelerometer

At the top of the tower there is an accelerometer installed. This will be used to observe the frequency and the damping of the structure. The accelerometer was glued to a piece of plastic housing and a screw was inserted at the top. This assembly was rigid with very little backlash and could be rotated if there was need for. The mount also made the accelerometer always stay in the center of the structure. Unfortunately the Spider8 only had eight input channels, six of which were used for strain gauges so there were only two slots available for the accelerometer. It was not a huge problem that only two axes could be monitored because the accelerometer was oriented in such a way that the gravitational acceleration was parallel to the unmonitored axis.

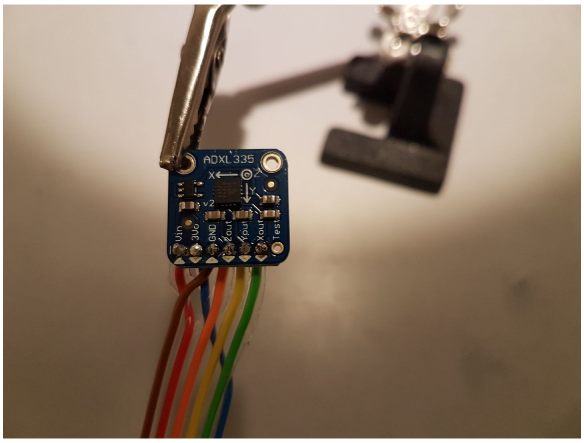

Figure 3.7: Shows the accelerometer soldered and ready for mounting.

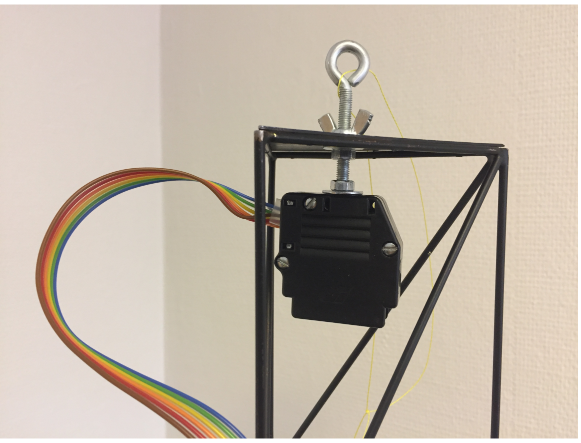

Figure 3.8: Shows the accelerometer housing and the bolt used to load the structure.

3.4 Data transfer

Since this thesis is a proposal for implementation on a digital twin, it was desirable to be able to measure the sensor values in real time. The software used to obtain the measurements is called *Catman*. Catman is a Data Acquisition Software (DAQ) that is able to measure, visualize and calculate the data in real time, however there is no really good way to export the data quickly. There are a few options which involves saving the files on the run, but this was both too slow or too demanding on the computer power once the files got up to a few megabytes. The solution to this problem was to obtain a second computer, one computer running Catman and a second one receiving the data transferred form the first one through an ethernet cable. This was achieved using *User Datagram Protocol* (UDP), notoriously referred to as *Unreliable Datagram Protocol*. The reason for the latter is that UDP provides no guarantee that the message sent is received [13].

In the Catman help documentation the following is stated: *The UDP output is not intended for real-time control, because the measurements are only sent approx. every 50 to 100 ms (the typical UDP output rate is 10 Hz). New measurements may therefore only be available after 100 ms*, however this turned out to not be the case and every measurment was sendt at 300 Hz. By the use of python function *time.clock()* and taking the difference from after sampling to before the sampling reoccurs it was found that the code takes roughly 0.6 ms to run, so the code keeps up with the broadcaster. Catman can send either 32-bit (4-byte single precision) or 64-bit (8-byte double precision) packets. For this project 32-bit was used, that means 4 characters contains the information for one channel. The python function *struct.unpack(fmt, string)* were used to unpack the sting with float as format.

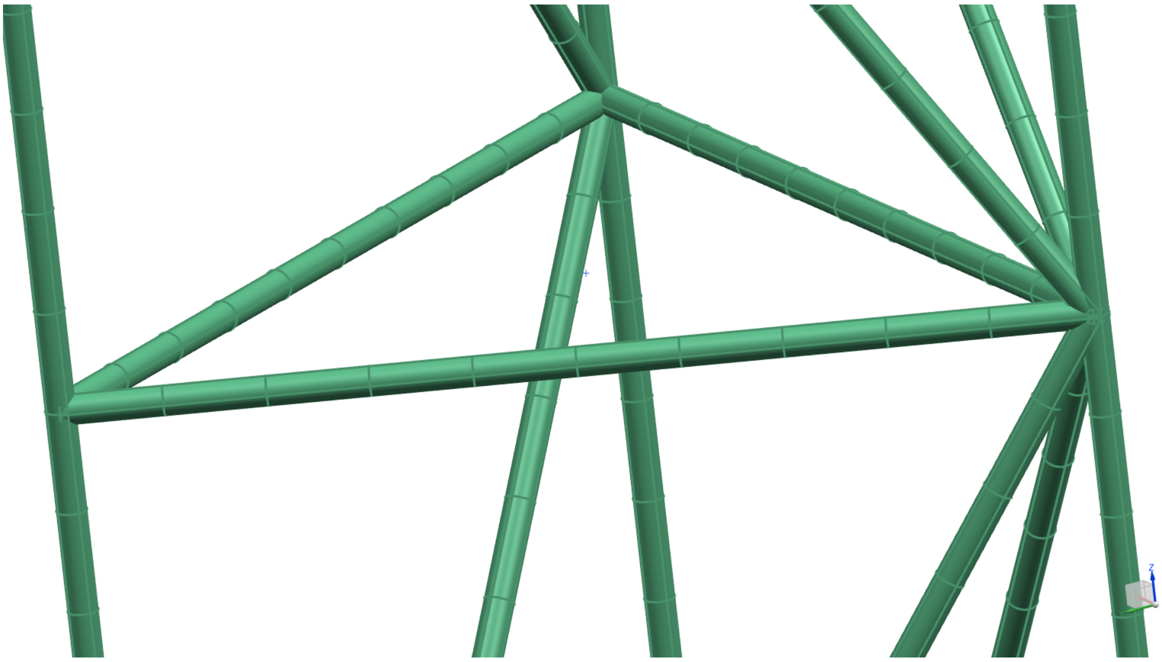

Figure 3.9: The mesh used in NX, each element is 10 mm.

3.5 Finite element model

The tower was modeled in Siemens NX. The FEM model is to be imported to Fedem as well as used for comparison of real strain values. 1D *PBEAM* elements were used as mesh with elements of 10 mm. Giving a total of 204 elements and 470 nodes for the whole structure. Figure 3.3 shows how the structure was fastened to the plate. This fastening was assumed to be equivalent to a fully clamped boundary condition, thus the nodes at each corner of the lower triangle was fixed in all degrees of freedom.

Table 3.1: Material properties for steel from NX

Property	Value
Young's modulus	$206.94{\times}10^3$ [MPa]
Possion's ratio	0.29 [–]
Density	7.829×10^{-6} [kg/mm^3]

3.6 Fedem

The FEM model was exported as a .nas file from NX and then imported to Fedem. At the corner of every other level a triad was added. At the top only one triad was added at the center of the triangle, this is where the force is applied. In-depth details on how the Fedem solver works will not be discussed in this thesis. In order to transfer the estimated force from the Kalman filter three *external functions* are defined in Fedem.

Three orthogonal forces are created and each force is defined by its respective function. The forces are applied to the triad at the top of the tower. The Fedem model .fmm file is loaded into the same code as the Kalman filter and then for each loop the estimated forces are exported to Fedem. The observation vector $\mathbf{y}_k$ consists of all measured strains for each time step, thus it is easy to export these values as well to Fedem as an external function analogous to the force. Fedem needs to be running a continuous solver and therefore a special version of Fedem is needed. The time step should be equal to the sampling time.

The model is imported as beam elements and Fedem does not calculate strain in these elements. To calculate the strain a relative sensor is created between two nodes. This relative sensor will measure the distance between this two nodes throughout the simulation. To calculate the strain one has to create a control system with the relative sensor as input and subtracting the inital length from this value and divide by the initial length. This shown in figure 3.12. The location of this relative sensor should be as close as possible to the actual strain gauge position. The beam elements are 10 mm in length, thus the strain is calculated over a larger area than the real strain gauges, this is a source of potential error, but in order to keep the computational time as low as possible the mesh was chosen to this level of refinement.

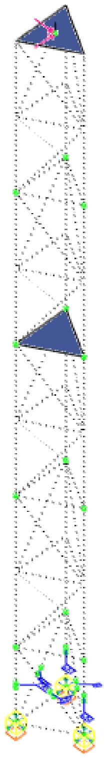

Figure 3.10: The model in Fedem with the forces applied at the top.

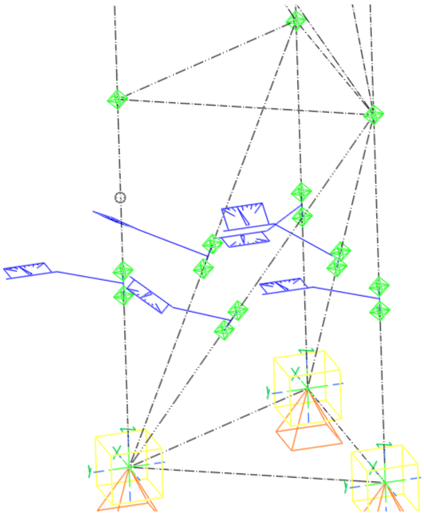

Figure 3.11: Shows the triads made to create strain gauges.

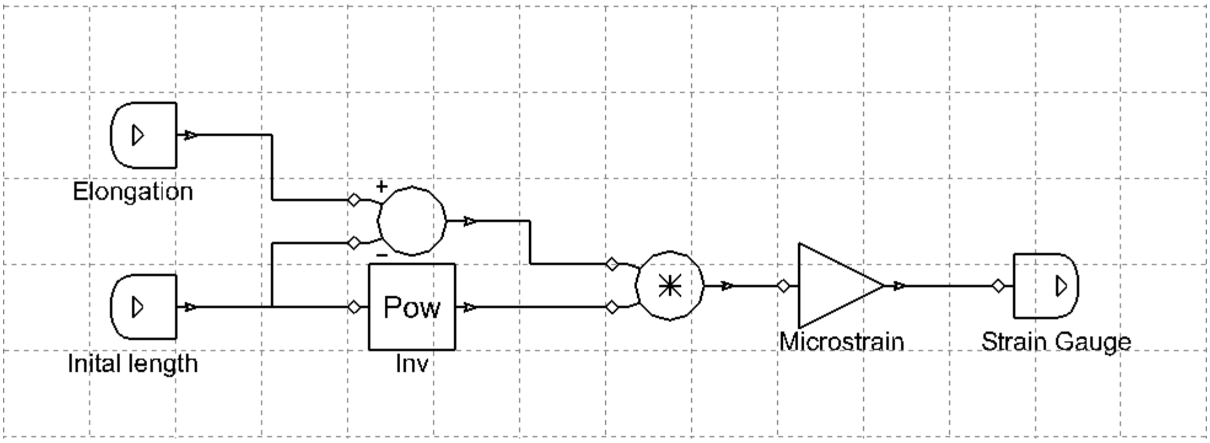

Figure 3.12: The control system that makes the relative sensor between two triads behave like a strain gauge.

Chapter 4

Analysis and results

4.1 Strain gauges

One of the first test that was done to the structure was a simple force equilibrium calculation. Since all the forces going through the tower is observed the sum of forces through each member should by equilibrium equal the applied load if it was a real truss structure. By assuming that the strain is purely axial the force each member experience can be calculated from the strain the following way

$$\sigma = E\varepsilon, \quad \sigma = \frac{F}{A} \Rightarrow F = AE\varepsilon \tag{4.1}$$

By assuming tensile forces inn all members in the sliced cross section area the forces can be decomposed as shown in table 4.1. Summarization of forces can now be done in x,y and z direction. This is shown in figure 4.1.

Table 4.2 shows that the sum of forces is not equal to the applied load on the tower, this suggests that there are some rotational stiffness in the boundaries of the members which induces bending strain; and the structure is not a truss structure by definition. Since bending is present beam elements with stiffness against rotation was used as mesh.

Table 4.1: Decomposition of forces

Force	F_x	F_y	F_z
F_1	$-F_1 \cos\phi \sin\alpha/2$	$F_1 \cos\phi \cos\alpha/2$	$-F_1 \sin\phi$
F_2	0	0	$-F_2$
F_3	$F_3 \cos\phi \sin\alpha/2$	$F_3 \cos\phi \cos\alpha/2$	$-F_3 \sin\phi$
F_4	0	0	$-F_4$
F_5	$F_5 \cos\phi$	0	$-F_5 \sin\phi$
F_6	0	0	$-F_6$

$\phi = 56.5752°$, $\alpha = 60°$

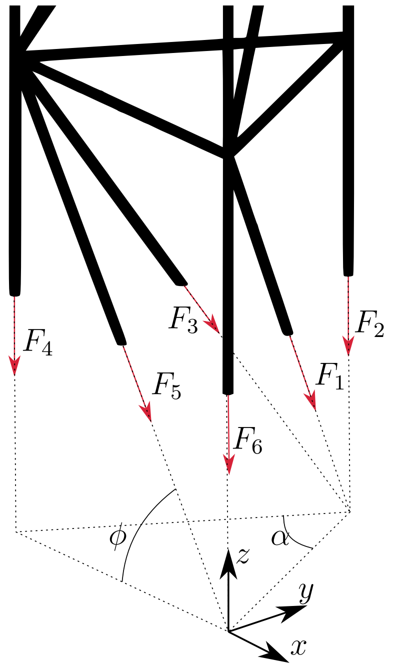

Figure 4.1: By slicing the tower through each strain gauge force equilibrium can be calucluated.

Table 4.2: Comparison of real values versus values extracted from FEA using the same load load case of 2 kg (19.62N)

Strain gauge	Measured strain $[\mu]$	FEM strain $[\mu]$
1	-20.75	-9.22
2	-210.59	-192.91
3	-8.57	-18.79
4	156.70	101.99
5	-4.10	-0.47
6	121.93	98.39
Sum forces:	61.94 N	30.98 N

The bending of the trusses induces yet another problem, the strain is matter of direction, meaning the strain value is depending on which way the member bends. It does matter if the strain gauge is on the compression, tension or neutral side of the bent member since it will measure different values. Rods does not have a strong and weak axis therefore there is nothing that obviously suggests which direction it will bend. The strain gauges was however mounted on the side that the author believed was most likely to be the tension side, however as it will turn out later section it does not really matter which side it is on as long as it is within reasonably linear region.

It is worth mentioning the issue of misorientation angle of strain gauges; The angle between a properly align strain gauge and the mounted strain gauge. The affect of misorientation can be analyzed via 2D coordinate transform equation for normal strain given as

$$\varepsilon'_{xx} = \varepsilon_{xx}\cos^2\theta + \varepsilon_{yy}\sin^2\theta + 2\left(\frac{\gamma_{xy}}{2}\right)\sin\theta\cos\theta \tag{4.2}$$

where ε'_{xx} is the normal strain measured by the strain gauge at the misorientated angle θ. ε_{xx} is the value the strain gauge would measure if it were properly mounted, i.e. at $\theta = 0°$. ε_{yy} is the strain normal to the local x direction and γ_{xy} is the shear strain. For a misalignment of 6° it would give

$$\begin{aligned}\varepsilon'_{xx} &= \varepsilon_{xx}\cos^2 6° + \varepsilon_{yy}\sin^2 6° + 2\left(\frac{\gamma_{xy}}{2}\right)\sin 6°\cos 6° \qquad (4.3)\\ &= 0.99\varepsilon_{xx} + 0.01\varepsilon_{yy} + 0.1\gamma_{xy} \qquad (4.4)\end{aligned}$$

Since the rods are long and slender the transverse and shear strain will be close to zero, but still 99% of the strain are observed, thus one can assume that all the strain gauges are mounted within reasonable accuracy with respect to misorientation.

4.2 Stiffness matrix

The system was simplified as a beam clamped in the bottom and three degrees of freedom at the top. In order to create the stiffness matrix enforced unit displacement in each direction was preformed in NX. The resultant force was then extracted and thereby giving the stiffness in each direction. The stiffness is presented in table 4.3

Table 4.3: Stiffness matrix

Direction	Value [N/mm]
K_{xx}	8.55
K_{yy}	8.63
K_{zz}	316.67

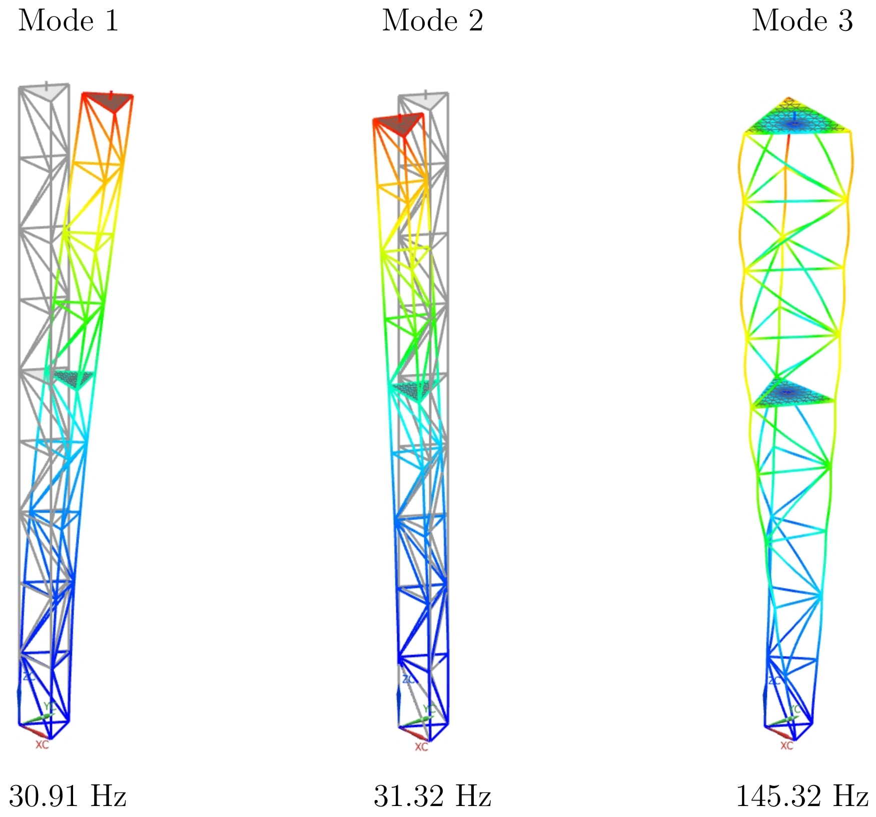

Figure 4.2: Plot of three lowest natural frequency modes

4.3 Frequency analysis

The stiffness matrix extracted from NX will most likely be different from the real stiffness of the structure due to imperfections in the real structure that are not in the FE model. One commonly used trick to scale the stiffness matrix is to compare the natural frequency of the real structure to the natural frequency calculated in the FE model. The solver *SOL103 Real Eigenvalues* was used to obtain the natural frequencies of the structure, as can be seen in figure 4.2. Mode 1 and mode 2 are very close to each other, whereas mode 3 are almost 100 Hz higher. The only modes of interest are the two lowest, since the third one is twisting. Both mode 1 and 2 is a combination of movement in x and y direction, however mode 1 is mostly dominated by x displacement whereas mode 2 in y direction, therefore a simplification is done and mode 1 is defined to be determined by stiffness in x direction and mode 2 by y direction.

To obtain the natural frequency the structure was struck by a hammer and left to vibrate until it came to a complete stop. The accelerometer data was measured the whole time and this is shown in figure 4.3. It is apparent that both the lowest modes are activated, and the structure drifts between the two. The author was unable to to fit the exponential expression for damping using a logarithmic decrement as is normally done. The reason of this was believe to be caused by the two modes being activated at once, thus making the decrements not occurring in a logarithmic manner. Knowing the expression for damping

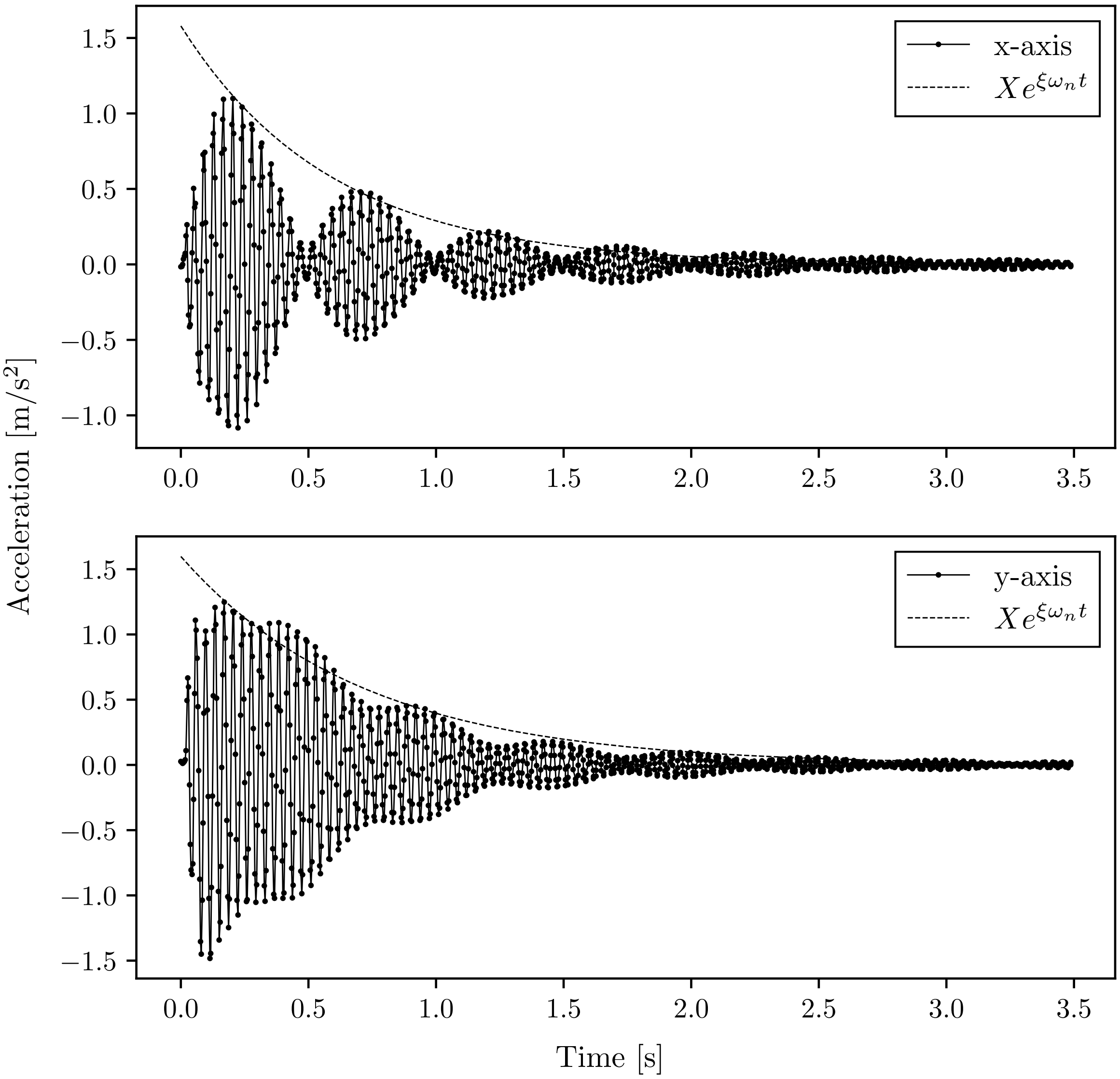

Figure 4.3: Accelerometer measurements of the tower after being struck by impulse load.

is expressed as Ae^{Bt}, the solution to this problem was solved as followed

$$a(t) = Ae^{Bt}$$
$$\ln a(t) = \ln A + Bt$$

where $B = \xi\omega_n$.

In order to find the constants A and B two points from the graph where chosen, giving the following expression for x-axis

$$a_x(t) = 1.579e^{-1.702t} \tag{4.5}$$

and for y-axis

$$a_y(t) = 1.598e^{-1.397t} \tag{4.6}$$

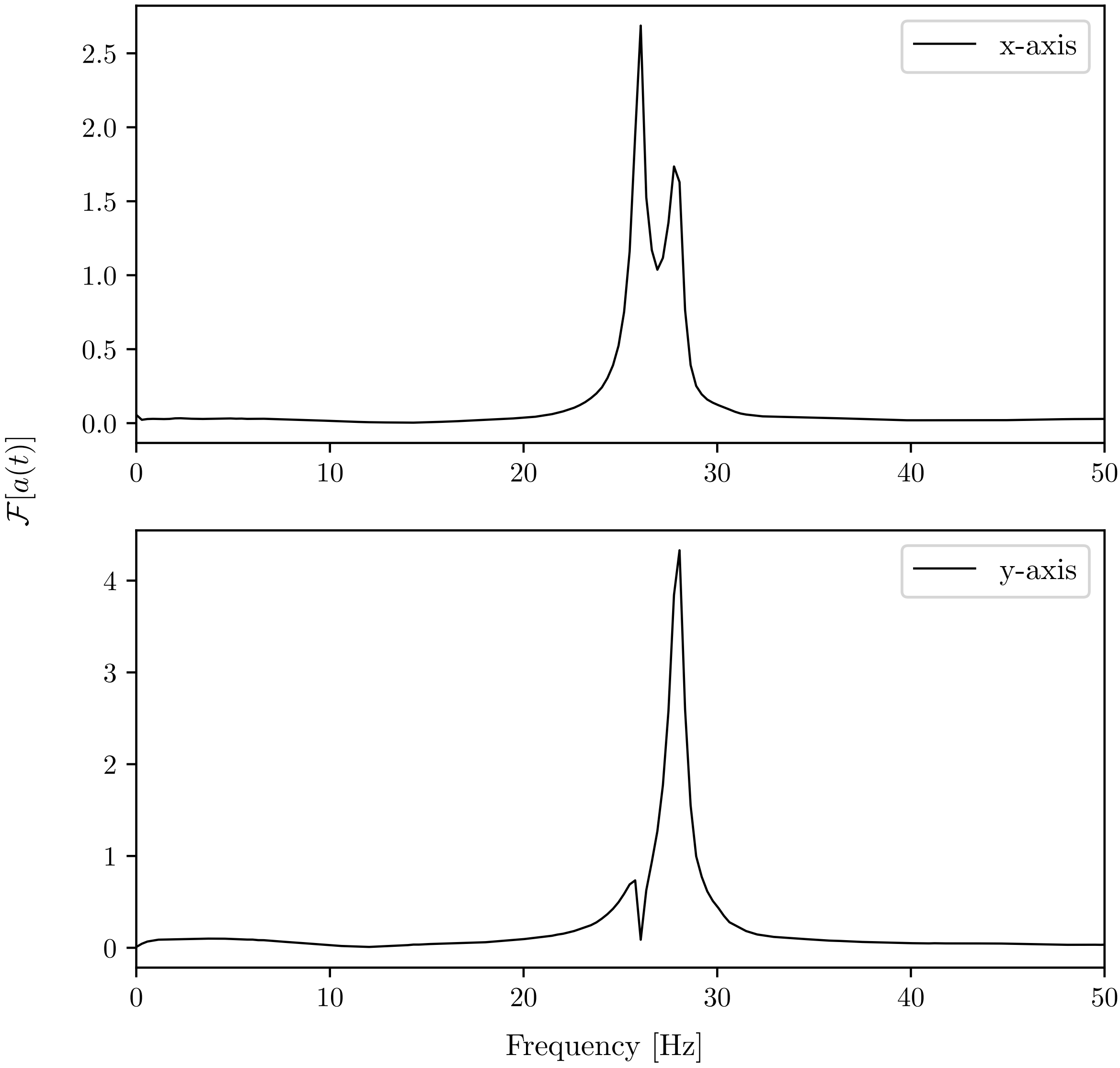

Figure 4.4: FFT of the accelerometer measurements after the impulse load.

Now the natural frequency is needed to extract from the constant. The natural frequency was obtained by doing a *Fast Fourier transform* of the accelerometer signal. The FFT of the acceleration is shown in figure 4.4, it is apparent that both the x and y direction captures both modes. This confirms that both modes are activated at once and the structure alternates between the two modes. From x and y direction the frequencies is found to be 26 Hz and 28 Hz respectively. Going back to equations 4.5 and 4.6 it is now possible to find the damping ratio

$$\xi_x = \frac{1.702}{26} = 0.065 \tag{4.7}$$

$$\xi_y = \frac{1.397}{28} = 0.050 \tag{4.8}$$

This is however only valid under the assumption that the damped natural frequency is equal to the undamped natural frequency. The relationship between the them is $\omega_d =$

$\sqrt{1-\xi^2}\omega_n$. If the damping ratio is inserted in the equation we get the following ratio $\omega_d = 0.99\omega_n$ for both direction, thus confirming the assumption that the damped and free natural frequency can be assumed equal. Now that the natural frequency is obtain it is possible to scale the stiffness matrix by the ratio of real frequency to the one calculated in NX as explained earlier, namely

$$\frac{\omega_{real}}{\omega_{FE}} = \sqrt{\frac{k_{real}/m}{k_{FE}/m}}$$

Assuming the mass is the same we get the following relationship between the real stiffness matrix and the measured

$$k_{real} = \left(\frac{\omega_{real}}{\omega_{FE}}\right)^2 k_{FE} \tag{4.9}$$

by once again assuming that mode one is determined by x direction and mode 2 by y direction the stiffness can be scaled accordingly

$$k_{xx} = 8.55\left(\frac{26}{30.91}\right)^2 = 6.05 \quad \text{[N/mm]} \tag{4.10}$$

$$k_{yy} = 8.63\left(\frac{28}{31.32}\right)^2 = 6.90 \quad \text{[N/mm]} \tag{4.11}$$

4.4 Measurement Matrix

The measurement matrix is the transition from state space to measurement space. This matrix has to be defined for unit displacement. If the system is linear then this matrix is constant.

Table 4.4: Linearity test in x direction

Strain gauge	$\mu\varepsilon$/7.85 N	$\mu\varepsilon$/19.62 N	$\mu\varepsilon$/29.43 N	Average
SG1	-0.150	-0.164	-0.151	-0.155
SG2	-0.107	-0.169	-0.056	-0.110
SG3	0.210	0.258	0.271	0.246
SG4	-16.639	-16.525	-17.309	-16.824
SG5	1.027	0.912	0.918	0.952
SG6	11.202	10.297	10.104	10.534

The tables shown in tables 4.4–4.6 is the strain values for the given load case divided be the load applied. Every load case has been applied 5 times and then averaged, giving a total of 15 measurements for each direction, with the exception of z axis, which has only two load cases. Any system that is linear should have a constant input to output value. By division of the strain (output) by force (input) a constant value should be found for

Table 4.5: Linearity test in y direction

Strain gauge	$\mu\varepsilon/7.85$ N	$\mu\varepsilon/19.62$ N	$\mu\varepsilon/29.43$ N	Average
SG1	-1.148	-1.058	-1.065	-1.090
SG2	-11.469	-10.733	-10.610	-10.937
SG3	-0.500	-0.437	-0.428	-0.455
SG4	8.980	7.987	8.028	8.332
SG5	-0.218	-0.209	-0.206	-0.211
SG6	7.042	6.215	6.334	6.530

Table 4.6: Linearity test in z direction

Strain gauge	$\mu\varepsilon/7.85$ N	$\mu\varepsilon/16.68$ N	Average
SG1	-0.012	0.010	-0.001
SG2	-0.003	0.246	0.121
SG3	-0.010	-0.000	-0.005
SG4	0.868	0.627	0.748
SG5	-0.024	-0.020	-0.022
SG6	0.189	0.084	0.137

all values if and only if the structure and strain gauges truly were linear, namely

$$Const = \frac{\varepsilon_{measured}}{F_{applied}}$$

By averaging this constant value by all the load cases a constant that should be valid for the whole load spectrum is obtained. One can now use the modified stiffness matrix and the constant value calculated in the tables to to create the measurement matrix, i.e. the strain value for unit deflection

$$\mathbf{C} = \begin{bmatrix} -0.937 & -7.523 & -0.347 & 0 & 0 & 0 \\ -0.668 & -75.468 & 38.441 & 0 & 0 & 0 \\ 1.491 & -3.139 & -1.564 & 0 & 0 & 0 \\ -101.787 & 57.488 & 236.784 & 0 & 0 & 0 \\ 5.760 & -1.454 & -6.958 & 0 & 0 & 0 \\ 63.732 & 45.058 & 43.237 & 0 & 0 & 0 \end{bmatrix} \tag{4.12}$$

The rank of this matrix is 6, thus we have full observability for the simplified system. As it was mention in section 4.1 that it does not matter if the strain gauges was mounted on tensile or compression side of the rod, with respect to force estimation, since the strain is defined by the state, thus the state will always create the same strain values they will only be scaled differently. The first column in this matrix is equivalent to unit deflection in the x direction and second and third for y and z respectively. Inspecting this matrix on can get a feel for how much the structure differ from an idealized structure. The second row in the first column should be 0 because this member is in the neutral axis of cross section of the structure. Looking at the third column every even numbered row should have more or less the same value if the structure was ideal because all the members should carry equal amount of force.

4.5 Assembling the State Transition Matrix

Since this is a simplified system there is no obvious or straight forward way to include the mass, therefore the mass will be used to tune the system further. The mass will be defined between the total mass of structure and zero. The damping values might not be 100% correct values, but since the loading will happen in such a slow manner and stiffness of the structure will be the dominating force its is more a matter of making the matrix complete and making the speed state converge. The reader is once again reminded that the system matrix is given as

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -k_{xx}m^{-1} & 0 & 0 & -2\xi_x\omega_{n1} & 0 & 0 \\ 0 & -k_{yy}m^{-1} & 0 & 0 & -2\xi_y\omega_{n2} & 0 \\ 0 & 0 & -k_{zz}m^{-1} & 0 & 0 & -Cm^{-1} \end{bmatrix} \tag{4.13}$$

The damping coefficient has been substituted with $2\xi_x\omega_{n1}$ for the x and y direction, but since no frequency that gave a pure vibration in the z axis was found it will just be assumed to of the same magnitude as x and y direction.

Using the definition of matrix exponential

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k \tag{4.14}$$

where X^0 is the identity matrix, and by taking the two first terms the state transition matrix is obtained

$$\mathbf{A} = \mathbf{I} + \mathbf{A}_c \Delta t \tag{4.15}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ \dfrac{-6.05\Delta t}{m} & 0 & 0 & 1-3.4\Delta t & 0 & 0 \\ 0 & \dfrac{-6.90\Delta t}{m} & 0 & 0 & 1-2.8\Delta t & 0 \\ 0 & 0 & \dfrac{-316.70\Delta t}{m} & 0 & 0 & 1-3\Delta t \end{bmatrix} \tag{4.16}$$

The filter was first tuned to give good results in the y direction by scale the mass matrix, then in order to obtain correct values in the x direction the stiffness matrix was modified by trail end error. The final values was a followed: $k_x = 6.55$ N/mm, $k_y = 6.9$ N/mm, $k_z = 316.7$ N/mm and mass 0.45 Kg.

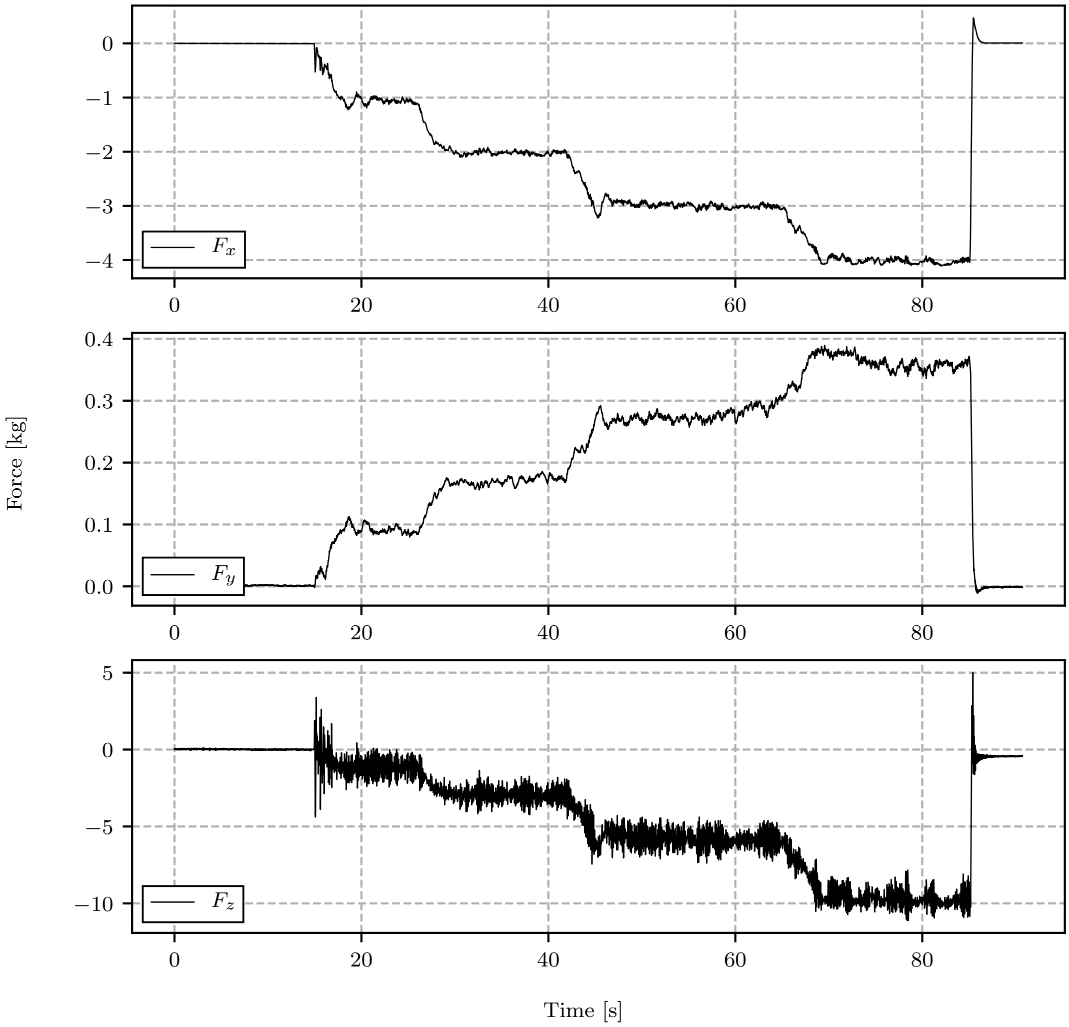

Figure 4.5: Force estimated in each direction. The load has been applied in the x-direction and kept constant for a few seconds at each whole number. Q_z is set to 1

4.6 Force Estimation

Every time the structure is displaced in a direction all the strain gauges will give out a value, and even though the deflection is in a pure direction, the readings will make the filter calculate values to the other two directions. The reason behind this is because all three states are defined by the same strain gauges.

The structure is very stiff in the z direction compared to the x and y stiffness. Even though the uncertainty (fluctuations in the measurements) in the state is the same for all direction it will affect force estimation in the z direction greater because of the stiffness being greater. The values for Q_x and Q_y was by a lot of trial and error eventually chosen as 10^{-4} and the forgetting factor γ as 0.8. This was found to be a suitable value for fast adaptive capability yet good estimation accuracy. The $\mathbf{R}$ matrix was set to $\mathbf{I} \times 10^{-9}$. This is a fairly high value, but it was chosen in order to "cover" all the standard deviations which might occur during the whole loading spectrum and in every direction. In the figures 4.5- 4.7 the filter has been tuned to give accurate results for loading in x and y

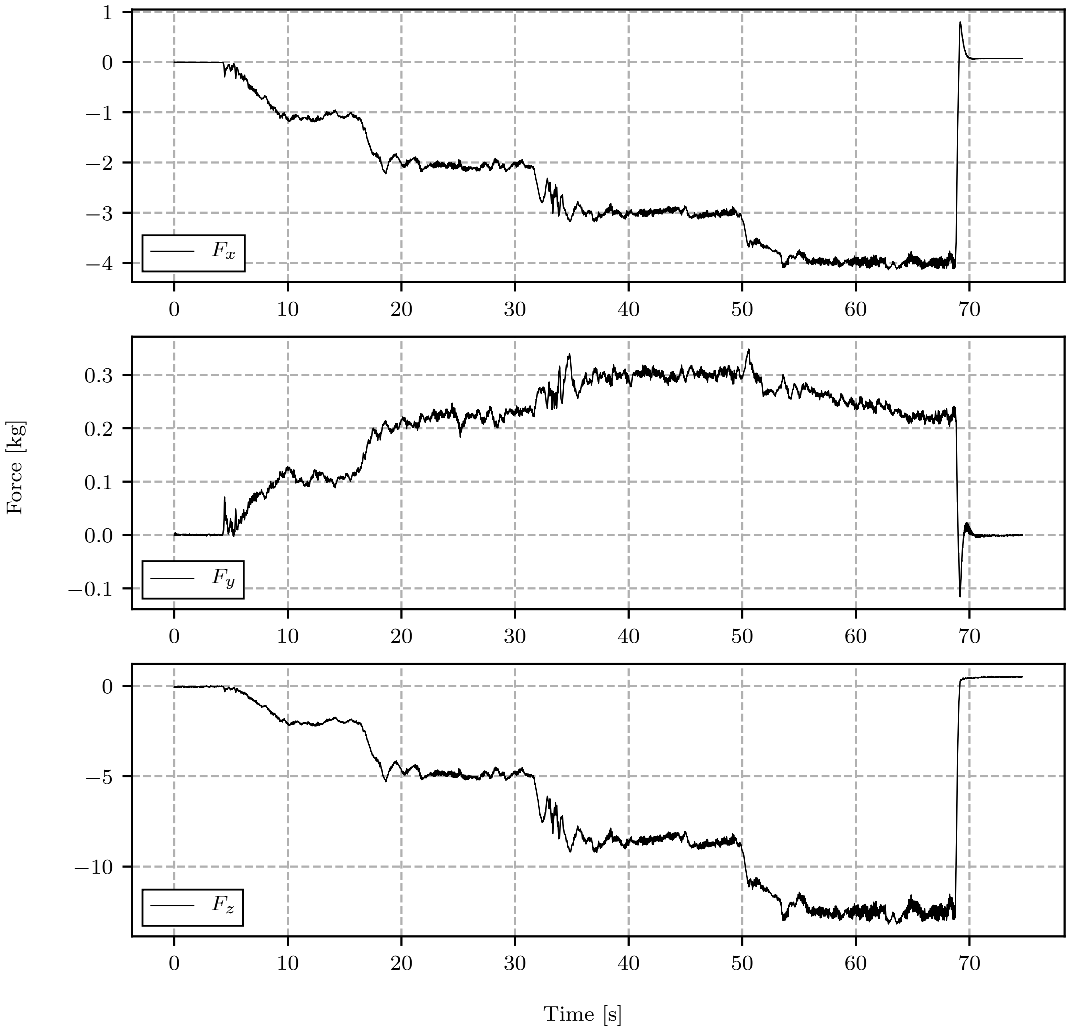

Figure 4.6: Force estimated in each direction. The load has been applied in the x-direction and kept constant for a few seconds at each whole number. Q_z is set to 10^{-4}

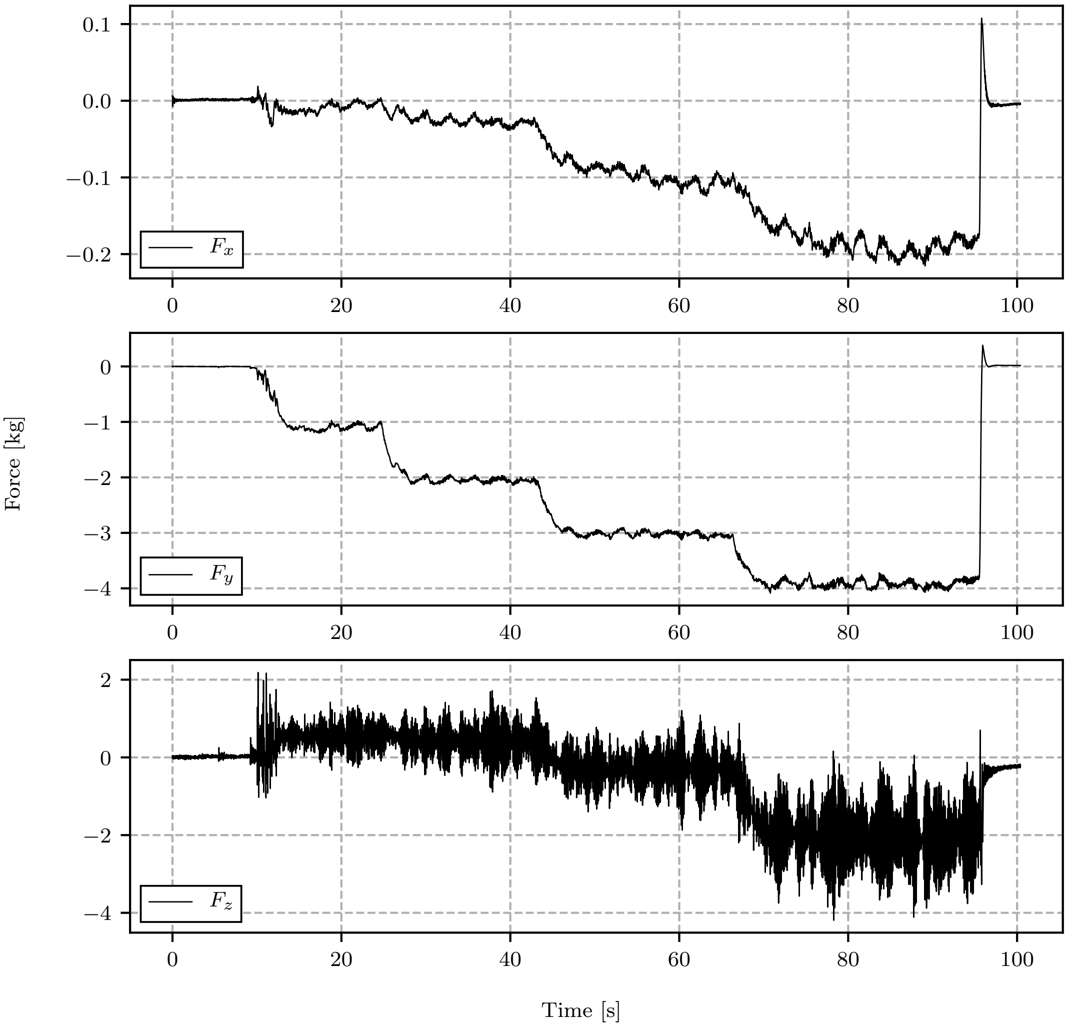

Figure 4.7: Force estimated in each direction. The load has been applied in the y-direction and kept constant for a few seconds at each whole number.

direction. For the load at 3 kg the fluctuation is less than 0.1 kg, that gives an accuracy of more than 96% for both loading in x and y direction. It is apparent how the value of Q affects the estimated force in Z direction. When its close to unity it is closer to the real value which is close to 0, but it fluctuates very much. When it is chosen as a lower value it is much more stable so it appears as a more confident estimation, but the opposite is true. The author was unable to achieve good values when load was applied purely in z and still have good results when loaded in x and y direction, perhaps a full measurement noise matrix with correlation noise could give better results. The recursive least square method has been proven to give very good results for both force and heat estimation in 2D plane [8, 11, 14, 15], but no study was found where the algorithm was utilized in three dimensions as in this thesis. It is believe that if the states was not all defined by the same sensors, i.e. if unique sensor could be used for each state it would give better results.

4.7 Import to Fedem

When the force estimation algorithm was tuned to give estimates that was within acceptable tolerance (with accuracy of >95% the applied load) the next step was to implement Fedem, thus closing the algorithm loop. The stiffness of the structures was scaled by the same factor as $\sqrt{\omega_{real}/\omega_{FEM}} = 0.707$. The two lowest natural frequencies of the structure was then found to be 27.19 Hz and 27.55 Hz respectively, which is in the middle of the ones found in the real structure. The estimated forces as well as the observation vector $\mathbf{y}$ was imported to Fedem as described in section 3.6. The F_z force was however not imported, this was because it was estimated too high, and therefore would induce artificially large deformations in the Fedem model.

Force in x direction

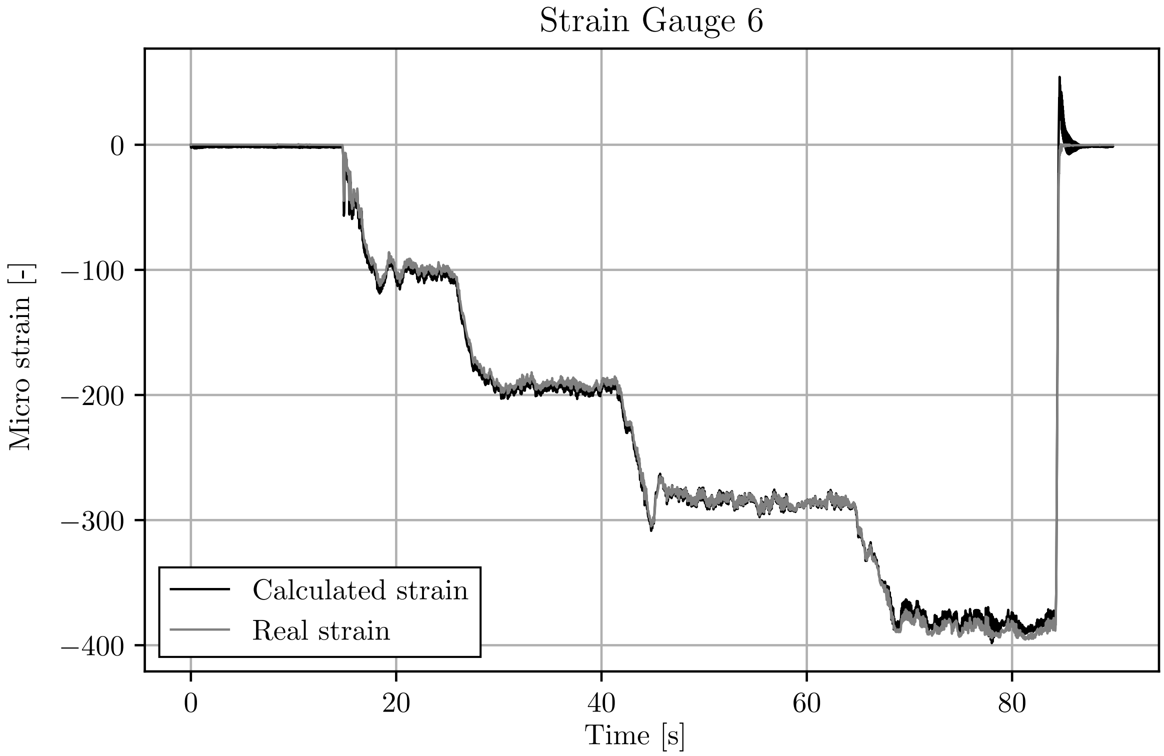

Figure 4.8: The real strain and strain calculated in Fedem. The calculated strain has been scaled 0.85

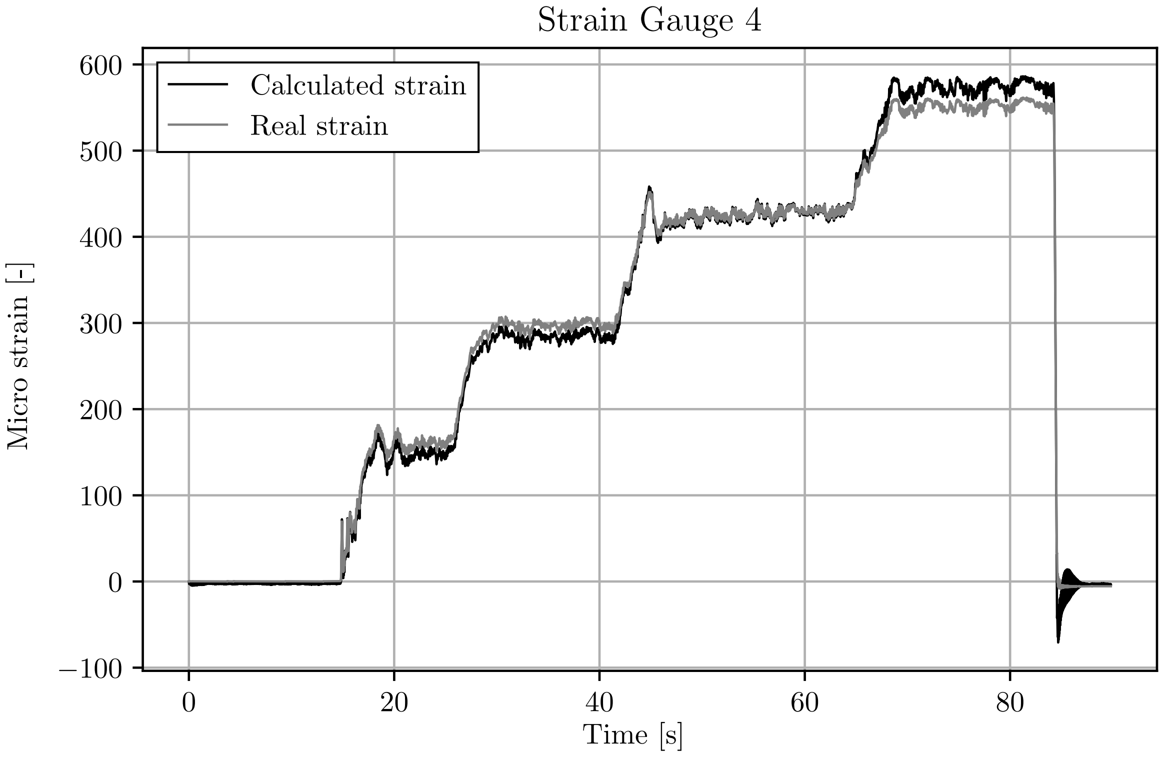

Figure 4.9: The real strain and strain calculated in Fedem. The calculated strain has been scaled 1.04

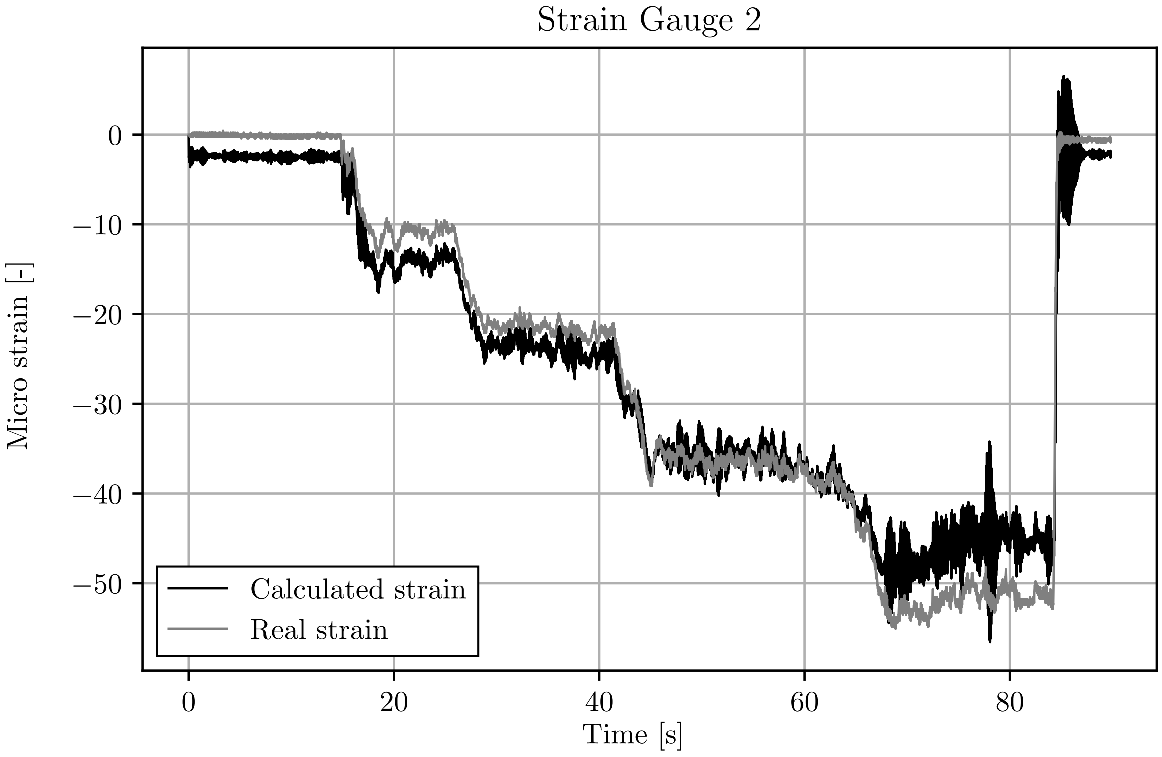

Figure 4.10: The real strain and strain calculated in Fedem. The calculated strain has not been scaled.

Force in y direction

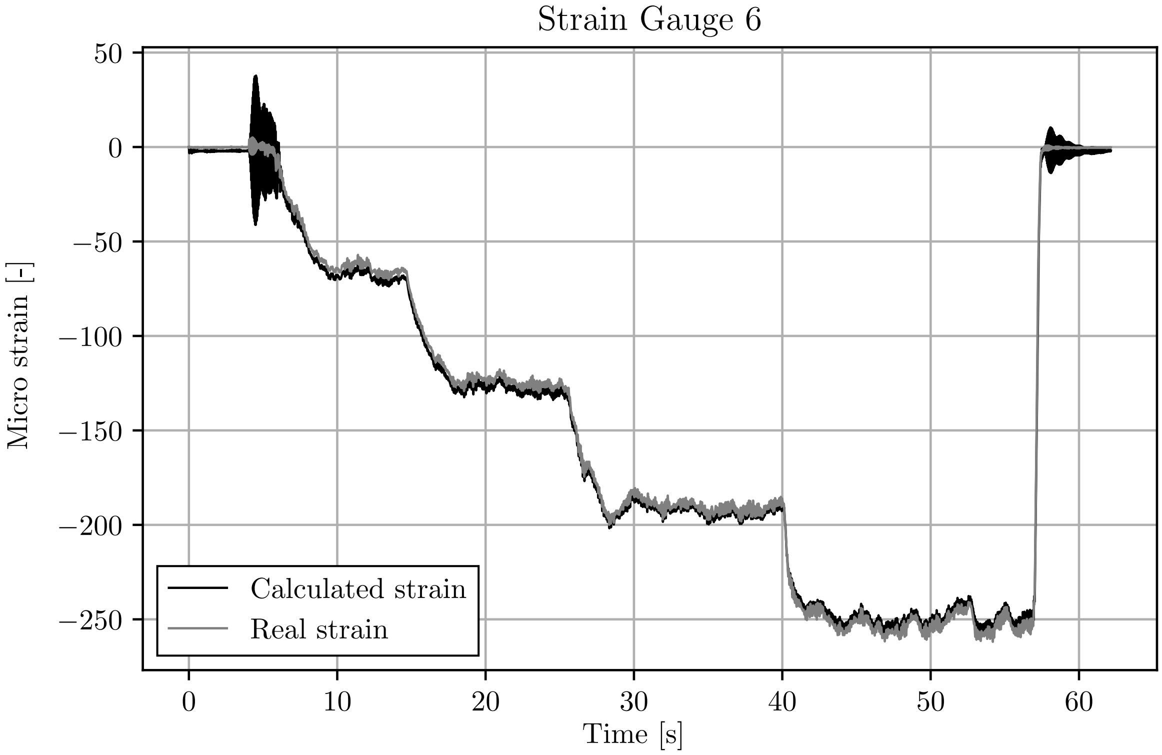

Figure 4.11: The real strain and strain calculated in Fedem. The calculated strain has been scaled 0.9.

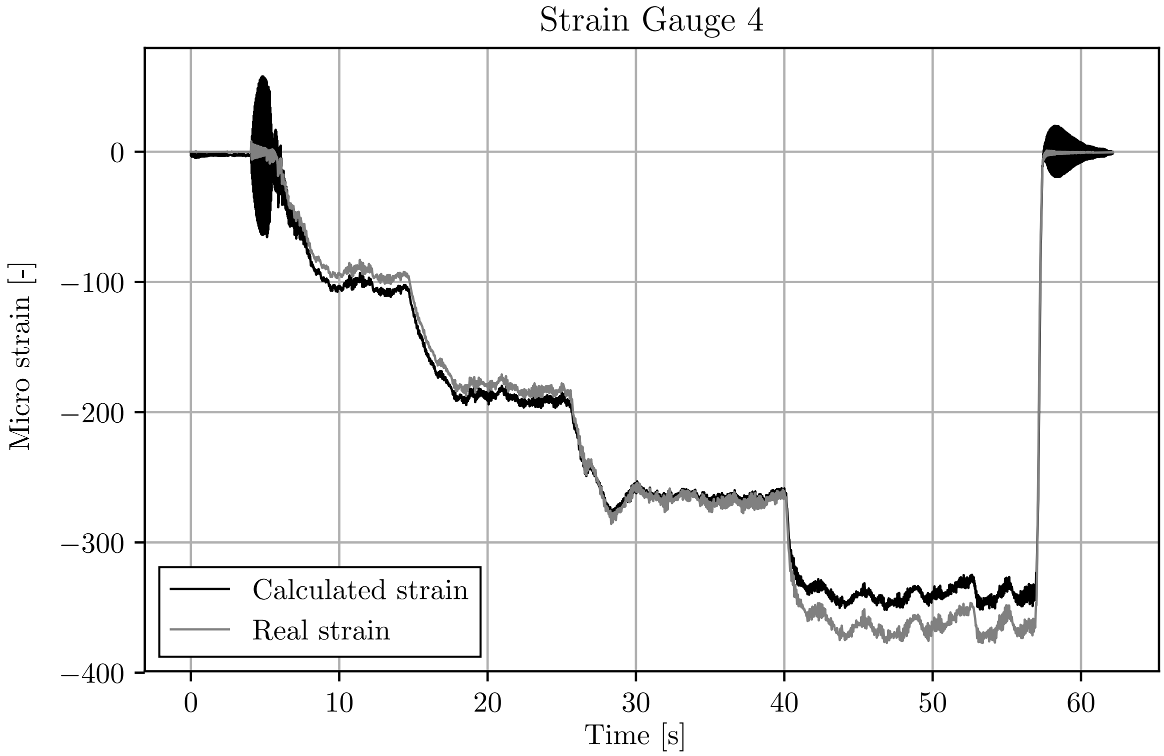

Figure 4.12: The real strain and strain calculated in Fedem. The calculated strain has been scaled 1.15.

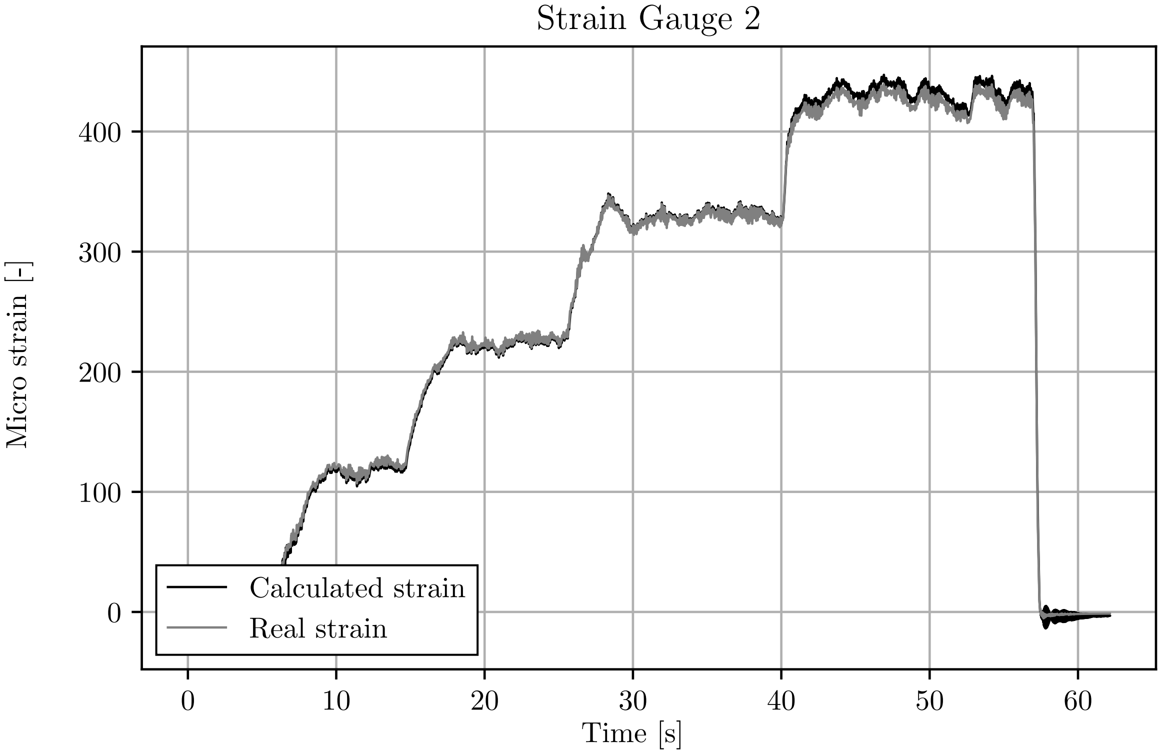

Figure 4.13: The real strain and strain calculated in Fedem. The calculated strain has been scaled 0.82.

All the plots have been scaled in such a way that the strains coincide at the force of 29.43 N (3 kg), meaning the third flat level on the plots. The calculated and real strain fluctuate in the same manner, and they are very similar but with different amplitudes. Similar looking plots means that the forgetting factor γ is of such low value that fluctuations in the measurements are countered. If however the forgetting factor was higher or the system noise vector were lower there would be a noticeable delay between the real value and the calculated because the force would lag. The best value of gamma is a value that gives good estimates and yet adapts quickly to changes.

There is no uniform difference between the measured strain and the calculated strain when comparing the different strain gauges, and it is not easy to pinpoint the exact reason behind this. There is the possibility that the strain gauges might give wrong value because of faulty wiring and/or uneven gluing etc., thus the real values are the incorrect ones. However a more likely reason is that the model does not represent the real structure closely enough with respect to imperfections, plus the strain is calculated in Fedem by just a simple axial deformation, thus no bending strain is calculated. It was shown in table 4.2 that it is highly likely that bending strain is present, and if this is the case it is pretty obvious that the Fedem model does not match the measured strain.

None of the diagonal members have been presented here, the reason for this was because they were either dominated by a lot of fluctuation, very large error or it was not possible to make the calculated and real strain plots coincide. It was believed the reason for all of the above was because the deformation in the diagonal members are so small that computer precision will affect the computation.

4.8 Model Imperfections and Uncertainties

The welds in the structure makes it neither a pinned connected structure nor are the welds stiff enough to make it behave like beams. For this reason the FE model (which had beam elements) was stiffer than the real structure. Creating a FE model that closely resembles the real structure as possible is a difficult and time demanding task, for some structures it might be impossible due to the complexity. If the structure can be modelled with great precision assembling of the matrices will be a lot quicker since the state transition matrix and the measurement matrix can all be created from the FE model. Once it is operational further tuning can be done through the **Q** and **R** matrices. For this structure it was easy to trigger and measure the natural frequency, and by the assumption of equal mass in the real structure and the FE model the stiffness was scaled accordingly. This is a fairly accurate and quick way to find the stiffness of the structure. Some structures might have lots of frequencies in a narrow specter, making the scaling by frequency ratio a more time consuming task.

A good estimation with respect to the F_z force remains unsolved, but perhaps implementation of more sophisticated version of the filter could solve the problem. Given the limited knowledge the author had prior to this project it was too time consuming to implement such a filter, but Kalman filtering extends way beyond what is presented in this thesis and there are several ways to handle non-linearities. Two common ones for the latter are *Extended Kalman filter* (EKF) and *Unscented Kalman filter* (UKF). Lee et al [14] implemented EKF for a nonlinear tower where the recursive least square algorithm was implemented, thus a very similar setup as presented in this thesis. The author concluded: *The proposed method was proven effective in estimating the unknown input. The proposed algorithm was further compared by alternating between the adaptive weighting and constant factors to demonstrate its excellent performance.* In addition to non-linear filtering two different versions of the kalman filter has been developed to account for model uncertainties: *Robust Kalman Filtering* (RKF) and *Adaptive Kalman Filtering* [6]. This shows how great potential this filter has for future implementations.

4.9 Thoughts on the Algorithm and Digital Twin

The author had very little knowledge on observation and state space models prior to this project, and a lot of time was put into gaining knowledge of the problem. A few different methods was attempted before finally arriving at the use of Kalman filter. One of the main problems was how to utilize more sensors than states. Kalman filter deemed very perfect for this. The great strength of the Kalman filter is the fact the state can be estimated trough a variety of sensors and it scales well with increased size and complexity, all one need is the relationship between system output and state. For this reason it is well suited for implementation on systems where it is either impossible or unfeasible to install a sensor at the desired location.

The algorithm used in this thesis did prove to capture loads applied to one node in three dimensions fairly well in the xy direction but the same precision in z direction remains unsolved. The algorithm is only as good as the mathematical model, thus a good model is required. No impulse loads or rapidly changing forces was considered in this thesis, nor loads applied to several nodes at the same time, it is worth mentioning that Ma et al [11] applied forces to two nodes at the same time with one node in between, and the algorithm did estimate the force on the applied nodes and no force on the unloaded one. Even though this was only done on a FE model it is fair to assume it should work on a real life application as well.

The algorithm presented in the thesis is a quick and computationally light algorithm for force estimation, no other than the previous step is needed to estimate both the state and forces. For this reason the author believes it is well suited as force input for implementation to a digital twin.

4.9.1 Solving time

If this algorithm was to be implemented on another structure in the future and the model is suppose to run real-time, the time it takes to solve one time step is something one should be aware of. If the solver is slower than the sampling rate and the model is suppose to run 24 hours a day 7 days a week the model would lag further and further behind the real time and it would never catch up. One possible solution to this would be to have an increased increased forgetting factor, thus averaging the force over a longer

	Known forces	Unknown forces
Known response	OK	Possible risk
Unknown response	Possible risk	Technologic blind spot

Figure 4.14: Possible scenarios the Digital Twin could be working under. Inspidred by Johari Window.

time period (ideally the same time it takes Fedem to solve one time step) and then export this force into fedem. This way the time in Fedem would be synchronized with the real time, however this comes at the cost of precision and short quick loads,such as an impulse load, would be neglected.

4.9.2 Structural Monitoring

One of the great potentials of a Digital Twins is using it for structural health monitoring. If all the forces a structure is subjected to are found and imported to the model it is be possible to monitor local stress hot-spots in the structure. By reviewing the stress history one can have a very confident estimate for when the part should be serviced with respect to fatigue, wear etc. Further inspection in the Digital Twin might also reveal stress hot-spots on locations different from what was first believed in the design process, this might be because the forces behaves differently than what was assumed. Using a Digital Twin for structural monitoring gives rise to some interesting scenarios, the so called *known knowns and unknown unknowns*, shown in figure 4.14. The top left one, the known force and known response is best possible scenario, because then all forces the structure is exposed to are known and therefore one can be certain than the stresses are correct and that nothing unforeseen will occur, this is however very unlikely as systems tends be pretty complex. The top right and bottom left squares are the Known response but unknown force and unknown force but known response is both scenarios with possible risk. The numbers extracted from the Digital Twin operating in these conditions needs to be handled with care since there is a lot of uncertainties. The Digital Twin will work as a supplement to regular visual inspections to the areas that are believed to be at risk of failure. The last scenario, the bottom right, is unknown forces and unknown response and this is potentially the most dangerous one. One can never know if this actually exist hence the “blind spot”. No one knows where to look for wear and no ones know what could cause a wear. A Digital Twin operating in this condition could potentially decrease the safety because of the belief that everything is counted for and nothing in the model indicates failer so no furher inspection is done.

Chapter 5

Conclusion

5.1 Main results and findings

This thesis has proposed a possible approach for the inverse problem to be implemented for a Digital Twin. A complete loop from retrieving measurements, estimating force and then calculating the strain in Fedem to see how close they are to the measurements has been performed. Even though it has been implemented on a fairly simple structure, the thesis aims to be as general an possible such that it is valid for structures of more complexity. The algorithm which has been utilized to estimate unknown input forces by the use of recursive least square estimator has been proven prior to this thesis to give good results in a Finite Element model. The big advantage of the Kalman filter is the fact that state is very easy to measure on a structure, and many different sensors can be used in the measurement matrix to relate state and and output. It does however require a fairly accurate Finite Element model in order to keep the estimated input forces precise.

The estimator algorithm utilized in this thesis is a very quick and computational light algorithm. No other info than the previous step needs to be stored to estimate both the state and forces in the next time step, thus it deems perfect for a real-time computation. After a lot of trial and error it eventually gave very good results to loads in the x and y direction, however the estimated force in z direction would in these cases tend to overestimate. This problem remained unsolved.

It has been investigated how this force can be exported into Fedem, and the real strain gauge values was compared to the ones calculated in Fedem. The latter did not give satisfactorily results, this was believed to be due to absence of bending strain in Fedem. Some recommendations on how to run the solver with respect to the filter has also been proposed.

5.2 Future work

Further investigation should be to benchmark the the algorithm with respect to sensor reductions. One should always aim to use as few as possible to sensors to capture the most precise estimations, thus reducing the costs of big structures. Implementation of a more sophisticated version of the filter and/or a full noise matrix, one that includes correlations should be tried in order to archive good estimates in the z direction when loading in x or y direction. Furthermore more load cases should be explored and the adaptive capability and response time of the algorithm should be investigated, that includes impulse load, loading at several different positions at the same time and quickly changing loads.

Bibliography

[1] GE digital. Digital twin technology and the outcomes you should expect. https://www.ge.com/digital/blog/digital-twin-technology-and-outcomes-you-should-expect, January 2018.

[2] Siemens. Digital enterprise – on the way to industrie 4.0. https://www.siemens.com/content/dam/internet/siemens-com/customer-magazine/old-mam-assets/print-archiv/advance/adv152-en-screen.pdf, November 2015.

[3] Fedem Technology. How does a digital twin work? http://www.fedem.com/digital-twin/how-does-a-digital-twin-work/, January 2018.

[4] Engineering.com. What is the digital twin and why should simulation and iot experts care? https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/16070/What-is-the-Digital-Twin-and-Why-Should-Simulation-and-IoT-Experts-Care.aspx, January 2018.

[5] NXP Semiconductors. Implementing positioning algorithms using accelerometers. https://www.nxp.com/docs/en/application-note/AN3397.pdf, January 2018.

[6] Yalcin Bulut. Phd thesis on applied kalman filter theory. http://people.duke.edu/~hpgavin/SystemID/References/Balut-KalmanFilter-PhD-NEU-2011.pdf, August 2011.

[7] Dan Simon. *Optimal State Estimation: Kalman, H_∞ and Nonlinear Approaches.* John Wiley & Sons, 2006. Ch. 1.7.2, pg. 40.

[8] Pan-Chio Tuan, Ching-China Ji, Li-Wei Fong, and Wen-Tang Huang. An input estimation approach to on-line two-dimensional inverse heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals*, 29(3):345–363, 1996. doi: 10.1080/10407799608914986. URL https://doi.org/10.1080/10407799608914986.

[9] Y. T. Chan, A. G. C. Hu, and J. B. Plant. A Kalman Filter Based Tracking Scheme with Input Estimation. *IEEE Transactions on Aerospace Electronic Systems*, 15:237–244, March 1979. doi: 10.1109/TAES.1979.308710.

[10] Jerry M. Mendel. *Lessons in Digital Estimation Theory.* Prentice-Hall, Inc., 1987.

[11] C.-K. MA, J.-M. CHANG, and D.-C. LIN. Input forces estimation of beam structures by an inverse method. *Journal of Sound and Vibration*, 259(2):387 – 407, 2003. ISSN 0022-460X. doi: https://doi.org/10.1006/jsvi.2002.5334. URL `http://www.sciencedirect.com/science/article/pii/S0022460X02953344`.

[12] Ka-Veng Yuen, Ka-In Hoi, and Kai-Meng Mok. Selection of noise parameters for kalman filter. *Earthquake Engineering and Engineering Vibration*, 6(1):49, Mar 2007. ISSN 1993-503X. doi: 10.1007/s11803-007-0659-9. URL `https://doi.org/10.1007/s11803-007-0659-9`.

[13] IPv6.com. Udp – user datagram protocol. `https://www.ipv6.com/general/udp-user-datagram-protocol/`, August 15 2006.

[14] Ming-Hui Lee and Ying-Wei Liu. Input load identification of nonlinear tower structural system using intelligent inverse estimation algorithm. *Procedia Engineering*, 79:540 – 549, 2014. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2014.06.377. URL `http://www.sciencedirect.com/science/article/pii/S1877705814009564`. 37th National Conference on Theoretical and Applied Mechanics (37th NCTAM 2013) & The 1st International Conference on Mechanics (1st ICM).

[15] Yang-Hsiung Ko, Jiu-Zhang Lu, Kai-Tai Lu, and Tsao-Fa Yeh. Inverse estimation problem of determining the unknown timewise-varying strength of a primer rapid heat source. *Procedia Engineering*, 79:295 – 304, 2014. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2014.06.346. URL `http://www.sciencedirect.com/science/article/pii/S1877705814009254`. 37th National Conference on Theoretical and Applied Mechanics (37th NCTAM 2013) & The 1st International Conference on Mechanics (1st ICM).