**POLITECNICO**
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Down-scaling of a Wind Turbine Test-Bench and Optimal Sensor Placement

TESI DI LAUREA MAGISTRALE IN
AERONAUTICAL ENGINEERING - INGEGNERIA AERONAUTICA

Author: **Gregory Cottone**

Student ID: 925852
Advisor: Prof. Pierangelo Masarati
Co-advisors: Daniel De Gregoriis, Tommaso Tamarozzi
Academic Year: 2021-22

# Acknowledgements

I would like to express my deepest gratitude to Daniel, my supervisor in Siemens Digital Industries Software, who served as my co-advisor and provided invaluable guidance throughout the completion of this master's thesis. His expertise, patience, and mentorship have been instrumental in shaping my research and ensuring its success.

I would also like to extend my appreciation to Professor Masarati for his support and for overseeing the quality of my thesis.

I am grateful to my family for their support and encouragement throughout my academic journey. Their love, understanding, and belief in my potential have been a constant source of inspiration.

I would like to acknowledge Gabriele for being a good friend and for having shared so many experiences during the last ten years.

I also thank Kevin and Alessandro, my university colleagues and friends with whom I shared most of my university experience.

Lastly, I want to acknowledge the unwavering support of Eliana. I am grateful for her presence in my life and for being my biggest supporter. Her unwavering belief in my abilities and her encouragement during challenging times have been crucial in keeping me motivated and focused.

To Daniel, Professor Masarati, my family, my friends and Eliana, I extend my sincerest appreciation. Your support, guidance, and belief in me have been instrumental in my success.

# Abstract

This work proposes an Augmented Extended Kalman Filter based state-input-parameter estimator for mechanical systems modeled using a flexible multibody formulation which is then applied to estimate the loads, the strain field and parameters on a Wind Turbine Test Bench.

This work demonstrates that a flexible multibody model representation allows to accurately reconstruct the time domain signals of the external loads and of the full strain field, starting from a minimal set of measured strains, hence providing an appealing alternative to direct measurement methods. The numerical validation on the system under analysis shows that all estimated quantities can be accurately reconstructed, given that the system simulation model incorporates an adequate level of accuracy.

The common issue that arises in the estimation of the joint state-input-parameter estimation is the selection of an optimal set of sensors which gives the best estimation. In the Kalman-filtering framework, an Optimal Sensor Placement (OSP) strategy has been proposed and it aims to find the best set of sensors in terms of system observability, which is the minimum requirement of a stable estimation. However, this does not guarantee the most accurate load-parameters estimation. In this contribution, the metric is based on the steady-state error covariance of the estimation which is based on the resolution of the Continuous Algebraic Riccati Equation (CARE).

The Optimal Sensor Placement strategy for state-input-parameter estimation is discussed on an industrial-scale Finite Element Model of the hub of the system under study.

**Keywords:** Multibody dynamics, Augmented Extended Kalman Filter, State-Input-Parameters estimation, Optimal Sensor Placement, Steady-state error covariance, Observability, Virtual sensing

# Contents

# Introduction

In order to assess whether a wind turbine will maintain its structural integrity during its life, i.e. while performing durability analysis, a creation of a test bench that represents and reproduces the same behavior of the system is important, however is often impossible to create test bench of the same dimension of the real wind turbine (they can reach tens of meters), therefore a scaling of the test bench is necessary. Sometimes, the problem could be simply the fact that having different test bench for the different kind of wind turbines could results in prohibitive costs.

Therefore, finding a method that allows to create small-scale test benches that are able to reproduce the same Key Performance Indicators (KPIs) of the real wind turbine, such as deformations, stress, forces exchanged between components, etc., would be of great advantage for the manufacturer.

The goal of this thesis is to find a method that allows to scale the test bench of a wind turbine in such a way as to reproduce the behavior of some states of the real wind turbine.

During the scaling procedure of the test bench, it will be shown that, the magnitude of the loads to apply in the test bench in order to reproduce the same KPI like the strain field of the flexible bodies or the forces exchanged between the bodies is not linear with the scaling of the system it self and therefore, a method that allows to estimate the loads to apply in the scaled test bench is necessary. In this regard, reference will be made to the works of Tamarozzi et al.: [2], [4], [5], [17], [18], [19] and [22]

In the current practice, test engineers perform direct measurements of the loads, in particular the loads acting on the blade center of the wind turbine test bench, during testing campaigns. In such testing campaigns the system is equipped with expensive measurement systems. Although these devices can achieve accurate results, they have the disadvantage of being expensive, intrusive and time-consuming to install and calibrate. For these reasons it could be helpful to develop a so-called virtual sensing strategy where information coming from more affordable and common sensors is combined with system behavioral information coming from available physical models. Depending on the available resources and on the system-under-investigation, several combinations of models and sensors, acting

on several levels of analysis (component, sub-system and system level analysis), can be exploited.

Retrieving back the inputs of a system from the knowledge of its outputs (strain measurements) falls in the category of inverse problems. In most cases inverse problems are ill-posed, meaning that the uniqueness, existence and stability conditions are not concurrently satisfied. Deterministic methods assume that the model is exact, which is never the case in practice. In this sense the use of state-estimation theory for the development of the virtual sensing strategy could be helpful, since it can account for the presence of uncertainties, both in the measurements and in the model.

In this work a Kalman filter will be used. The Kalman filter is a very widely used state estimator and it is a very practical solution to the problem of estimating the states of a system starting from the knowledge of some measurements and of a system model, furthermore can be also used for performing coupled state-input-parameter estimation by using a so-called Augmented Kalman Filter (AKF) which is more correctly defined in our problem as Augmented Extended Kalman Filter (AEKF) since our system is nonlinear.

To work, a Kalman filter needs a physical model of the system and a set of measurements, which can be real, that is, coming from a physically real system or virtual, that is, generated from a virtual model.

The problem of estimating the joint state-input-parameter vector can be obtained by introducing a force and parameter model and rearranging the equations in such a way that it is possible to augment the state vector including the input and the parameters inside it. Hence when estimating the states also the inputs and the parameters are estimated.

Virtual prototypes are used to reduce the number of real prototypes built during the durability development process, hence reducing costs. These virtual prototypes are high fidelity simulation models that accurately reproduce the dynamic behavior of the dynamical system including all the relevant physical phenomena.

This work investigates the use of a flexible multibody model of the wind turbine test bench in combination with strain measurements on that component.

However, sometimes it is not possible to guarantee even the presence of an accurate physical model of the system as some parameters that characterize it are not known, therefore in general the need to find a method that allows to estimate these parameters also arises.

As a second part of the thesis, it will be shown that the position of the sensors for strain measurements influences the accuracy and stability of the estimation of the inputs and

the parameters of the system. Indeed, the selection of the optimal location of sensors is a common issue in experimental design and inverse identification problems. Observability is a minimum requirement to obtain a stable estimation through Kalman filtering techniques. In order to obtain a stable estimation, a number of strain outputs equal or greater to the number of forces to be identified is required. In this work an Optimal Sensor Placement (OSP) strategy is developed for the joint estimation of the states-inputs-parameters estimation as an extension of the work done by Cumbo et al. [6], which enables an accurate a-priori assessment of the impact of sensor selection based on the evaluation of the Steady State Estimation Error Covariance Metric obtained thanks to the solution of the Continuous Algebraic Riccati Equation.

# Part I

# Down-scaling Wind Turbine Test Bench

# 1 | Theoretical background

This chapter summarizes the mathematical background necessary to describe the method of estimating states, inputs and parameters in a multibody model through the use of an Augmented Extended Kalman Filter (AEKF). In particular the first paragraph will show which are the equations describing a generic multibody system consisting of rigid bodies and the numerical methods most used in this context, the second paragraph will show the method used for the implementation of flexible bodies in multibody systems, the third paragraph will show the mathematic method used to model the existing ball bearing in the system under analysis in this thesis and the numerical methods used to solve the resulting implicit nonlinear equations, the fourth paragraph will show the generic formulation of the Kalman filter used in nonlinear systems, going then in the detail of the methods used for the linearization of the implicit non-linear system, finally the fifth paragraph will show the formulation used to integrate the input to be estimated in the generic vector of states in such a way as to be able to estimate the states and the input of the system.

## 1.1.  Multibody Model

Multibody dynamical systems are typically composed of several interconnected bodies that can relatively move between each other, e.g. the several bodies that make up a suspension system, the wind turbine system or the main rotor head system of an helycopter which are hence typically used as high fidelity models for this kind of dynamical systems. Multibody models can then also incorporate all the other additional and necessary non-linear phenomena.

As multibody dynamics deals with the motion of systems, such set of variables must allow one to retrieve the position of all the material points of the system or, equivalently, to determine its configuration. These variables are called generalized coordinates and are collected in the vector of the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_{GC}}$:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{GC} \end{bmatrix} \tag{1.1}$$

The use of redundant generalized coordinates, i.e. in a number that is higher than the actual number of DOFs of the system, is the method used in the development of the multibody model in this thesis. Such redundancy allows a much simpler kinematic description and hence a straightforward and automatable formulation of the equations that govern the motion of the system. The main consequence of using redundant coordinates is that the generalized coordinates are not anymore kinematically independent between each other and algebraic equations that model their interdependence have to be introduced. As a result, the governing equations of the system behavior is a mix of algebraic and differential equations.

Such equations are called constraint equations as they limit the space of the possible configurations and are indicated with:

$$\boldsymbol{\phi}(\boldsymbol{q}) = 0 \tag{1.2}$$

where the constraint vector $\boldsymbol{\phi}$ is a real vector-valued non-linear function of dimension $n_C$, the latter being the number of constraints.

Note that $\boldsymbol{\phi}$ is exclusively a function of the generalized coordinates vector $\boldsymbol{q}$ and not for

instance of its time derivative $\dot{\boldsymbol{q}}$. For this reason Eq. 1.2 is purely an algebraic equation and represents so-called holonomic constraints.

Such type of constraints equations derive from the modeling of all the basic joints typically present in multibody systems and also from the additional redundancy related to a particular generalized coordinates choice.

The solution of a multibody system starts from setting the problem according to Lagrange's formulation of the first type for constrained mechanical systems:

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}(\boldsymbol{q})^T\boldsymbol{\lambda} = \boldsymbol{f}(\dot{\boldsymbol{q}}, \boldsymbol{q}, t) \\ \boldsymbol{\phi}(\boldsymbol{q}) = 0 \end{cases} \tag{1.3}$$

Where $\boldsymbol{M}$ represents the mass matrix, $\boldsymbol{B}(\boldsymbol{q})$ represents the Jacobian matrix for kinematic constraints between bodies, $\boldsymbol{\lambda}$ represents the Lagrange multipliers and $\boldsymbol{f}(\dot{\boldsymbol{q}}, \boldsymbol{q}, t)$ the external generalized forces.

The equation $\boldsymbol{\phi}(\boldsymbol{q}) = 0$ represents the kinematic constraints.

The terms $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ represent the generalized coordinates of position and velocity of the bodies.

For further information on the formulation of the various matrices that make up the system, as well as the various parameterizations of rotations in space and the initialization methods of multibody systems, consult the books of Masarati [12] and Nikravesh [14], [15].

The Jacobi matrix of the kinematic constraints is given by the following relationship with the equations of the constraints:

$$\boldsymbol{B} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{q}} \tag{1.4}$$

This matrix has as many rows as the constraint equations and as many columns as the generalized coordinates.

The method used in this study to solve multibody systems is represented by the GGL formulation (Gear, Gupta and Leimkuhler, [7]) in the discretized version, which represents a set of Differential Algebraic Equations (DAE), for a more detailed study of this formulation consult the book by Simeon [20]:

$$\begin{cases} \dot{\boldsymbol{q}}_{k+1} - \boldsymbol{v}_{k+1} + \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\mu}_{k+1} = 0 \\ \boldsymbol{M}(\boldsymbol{q}_{k+1})\dot{\boldsymbol{v}}_{k+1} + \boldsymbol{f}(\boldsymbol{q}_{k+1}, \boldsymbol{v}_{k+1}) - \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\lambda}_{k+1} = 0 \\ \boldsymbol{B}(\boldsymbol{q}_{k+1})\boldsymbol{v}_{k+1} = 0 \\ \boldsymbol{\phi}(\boldsymbol{q}_{k+1}) = 0 \end{cases} \tag{1.5}$$

The numerical methods that are usually used for solving this type of problems are the BDF (Backward Difference Formulae) of various order, this for two reasons:

- DAE systems tend to be stiff by construction;

- The flexible bodies within multibody systems make such systems stiff.

In particular the BDF schemes have the characteristic of being implicit methods (note that the first order BDF method is the well known Implicit Euler method) and consequently lend themselves well to solving stiff problems.

In our case we will use the BDF method of order 1, or Implicit Euler, not worrying about the numerical dissipation characteristic of some multibody problems when they are integrated with low order BDF numerical methods:

$$\begin{cases} \dot{\boldsymbol{q}}_{k+1} = \frac{1}{h}(\boldsymbol{q}_{k+1} - \boldsymbol{q}_k) \\ \dot{\boldsymbol{v}}_{k+1} = \frac{1}{h}(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k) \end{cases} \tag{1.6}$$

The reason for the choice of a low order BDF scheme will be explained later when we will talk about the implementation of the Kalman filters for the estimation of states, inputs and parameters.

Then substituting the 1.6 to the 1.5 allows to obtain the following system:

$$\begin{cases} \frac{1}{h}(\boldsymbol{q}_{k+1} - \boldsymbol{q}_k) - \boldsymbol{v}_{k+1} + \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\mu}_{k+1} = 0 \\ \boldsymbol{M}(\boldsymbol{q}_{k+1})\frac{1}{h}(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k) + \boldsymbol{f}(\boldsymbol{q}_{k+1}, \boldsymbol{v}_{k+1}) - \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\lambda}_{k+1} = 0 \\ \boldsymbol{B}(\boldsymbol{q}_{k+1})\boldsymbol{v}_{k+1} = 0 \\ \boldsymbol{\phi}(\boldsymbol{q}_{k+1}) = 0 \end{cases} \tag{1.7}$$

So at this point it is sufficient to simply find a numerical method capable of solving this implicit system of nonlinear equations in the unknowns $\boldsymbol{q}_{k+1}$, $\boldsymbol{v}_{k+1}$, $\boldsymbol{\mu}_{k+1}$ and $\boldsymbol{\lambda}_{k+1}$, usually Newton Raphson's method is used.

## 1.2. Flexible Multibody Model

When the flexibility of the bodies plays an important role in the system behavior, flexible bodies have to be considered. In this case additional DOFs related with the flexibilities of the flexible bodies have to be introduced. Here the attention is limited to so-called small-deformation flexible multibody systems, i.e. to systems for which the flexible motion is small in relation to the rigid body motion. For small-deformation flexible multibody systems the most common approach to the generalized coordinates selection is the Floating Frame of Reference Formulation (FFRF). In such an approach the motion of each body is described as the superposition of a large rigid body motion and of a small deformation around a so-called floating frame of reference attached to the body. In terms of generalized coordinates then, the rigid body motion of each body is captured by using Cartesian coordinates collected in a vector of rigid coordinates, while the small deformation around the floating frame of reference for the same body is described by means of flexible coordinates associated with a finite set of deformation shapes of the particular flexible body collected in a vector of flexible coordinates. A typical way to determine such deformation shapes for bodies of general geometrical shape is through finite element analysis. This approach is hence an extension of the Cartesian coordinates approach to include the effects of small deformations associated with small bodies flexibilities.

Since the model under study in this thesis is also made of flexible bodies, we see below what is the procedure for modeling the dynamics of flexible bodies in a multi-body system, taking into account that the following procedure is based on the books of Masarati [11] and Miller et al. [13].

The generalized coordinates used are called flexible natural coordinates and in practice, the generalized coordinates of each body $q$ are composed of a part that allows to describe the global rigid motion of each body, $q_r$, plus a set of coordinates used to describe the part of motion due to the flexibility of bodies, $q_f$.

The reduction method is based on the Craig - Bampton method [3], in which it involves the following steps:

- The nodes of the finite element model are divided into a set of internal nodes and a set of interface nodes with the other bodies of the system.

- The degrees of freedom of the node sets previously described are expressed in terms of modal and boundary coordinates.

- The set of modal coordinates is truncated at a specific cut-off frequency chosen by

us.

- The whole system is reformulated in terms of the reduced degrees of freedom, that is, in terms of the physical coordinates of the interface nodes and of the truncated modal coordinates.

Starting from the undamped nodal motions in finite-element formulation, expressed as:

$$\boldsymbol{M}\ddot{\boldsymbol{z}} + \boldsymbol{K}\boldsymbol{z} = \boldsymbol{f} \tag{1.8}$$

Partitioning the nodes into internal and boundary sets yields:

$$\begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{z}}_b \\ \ddot{\boldsymbol{z}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_b \\ \boldsymbol{z}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{f}_i \end{bmatrix} \tag{1.9}$$

Where the subscript $b$ refers to the boundary nodes and the subscript $i$ refers to the internal nodes.

Expressing the degrees of freedom of the partitioned nodes in terms of boundary and modal coordinates (by the so called Craig Bampton transformation) gives:

$$\begin{bmatrix} \boldsymbol{z}_b \\ \boldsymbol{z}_i \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} \boldsymbol{z}_b \\ \boldsymbol{\eta} \end{bmatrix} \tag{1.10}$$

Where $\boldsymbol{\eta}$ is the set of modal degrees of freedom, the modal amplitudes resulting from the assumption that the boundary nodes are fixed.

The degrees of freedom in $[\boldsymbol{z}_b \ \boldsymbol{\eta}]$ form a hybrid set, with some defined in physical space and some defined in modal space.

$\boldsymbol{H}$ is the Craig Bampton transformation matrix,

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{\Phi} \end{bmatrix} \tag{1.11}$$

where $\boldsymbol{\Xi}$ is a set of interface, or static condensation, modes and $\boldsymbol{\Phi}$ is a set of fixed boundary mode shapes.

The interface modes are obtained simply by imposing $\boldsymbol{z}_b = \boldsymbol{I}\boldsymbol{z}_b$ and considering $\boldsymbol{f}_i = 0$ we can also obtain the static part of the internal modes:

$$K_{ib}z_b + K_{ii}z_i = 0 \tag{1.12}$$

Obtaining:

$$\begin{bmatrix} z_b \\ z_i \end{bmatrix} = \begin{bmatrix} I \\ -K_{ii}^{-1}K_{ib} \end{bmatrix} z_b \tag{1.13}$$

The dynamic part of the internal nodes is simply obtained using the eigenvectors $U_r$, so we have $z_i = U_r\eta$.

Finally the internal nodes are expressed by their static part plus the dynamical part:

$$z_i = -K_{ii}^{-1}K_{ib}z_b + U_r\eta \tag{1.14}$$

Obtaining:

$$\begin{bmatrix} z_b \\ z_i \end{bmatrix} = \begin{bmatrix} I & 0 \\ -K_{ii}^{-1}K_{ib} & U_r \end{bmatrix} \begin{bmatrix} z_b \\ \eta \end{bmatrix} = H \begin{bmatrix} z_b \\ \eta \end{bmatrix} \tag{1.15}$$

Truncating the modal coordinates at a cutoff frequency gives:

$$\begin{bmatrix} z_b \\ z_i \end{bmatrix} \simeq H^* \begin{bmatrix} z_b \\ \eta^* \end{bmatrix} \tag{1.16}$$

where $\eta^*$ is the set of truncated, or reduced, modal degrees of freedom. Rewriting the equation of motion in terms of the reduced degrees of freedom, i.e. substituting this new coordinates inside of the 1.9 gives the final working expression:

$$H^{*T}MH^* \begin{bmatrix} \ddot{z}_b \\ \ddot{\eta}^* \end{bmatrix} + H^{*T}KH^* \begin{bmatrix} z_b \\ \eta^* \end{bmatrix} = H^{*T} \begin{bmatrix} f_b \\ 0 \end{bmatrix} \tag{1.17}$$

External forces are assumed to act exclusively at the boundary nodes (that is $f_i = 0$)

## 1.3.    Flexible Multibody Model of the Test Bench

A flexible model of the Wind Turbine Test Bench embedded in a full non linear framework to capture the non-linear body motion is used in combination with strain measurements to retrieve the loads transmitted from the blade by means of a state-input estimator.

In this section it will be shown an application of the AEKF for input force estimation in a model of the wind turbine test bench.

The model is composed by a flexible hub modeled as a finite element model, a rigid blade connected to the hub through a pitch ball bearing bearing in which more detail will be shown in the next section.

This test bench represents a 3.4 MW wind turbine
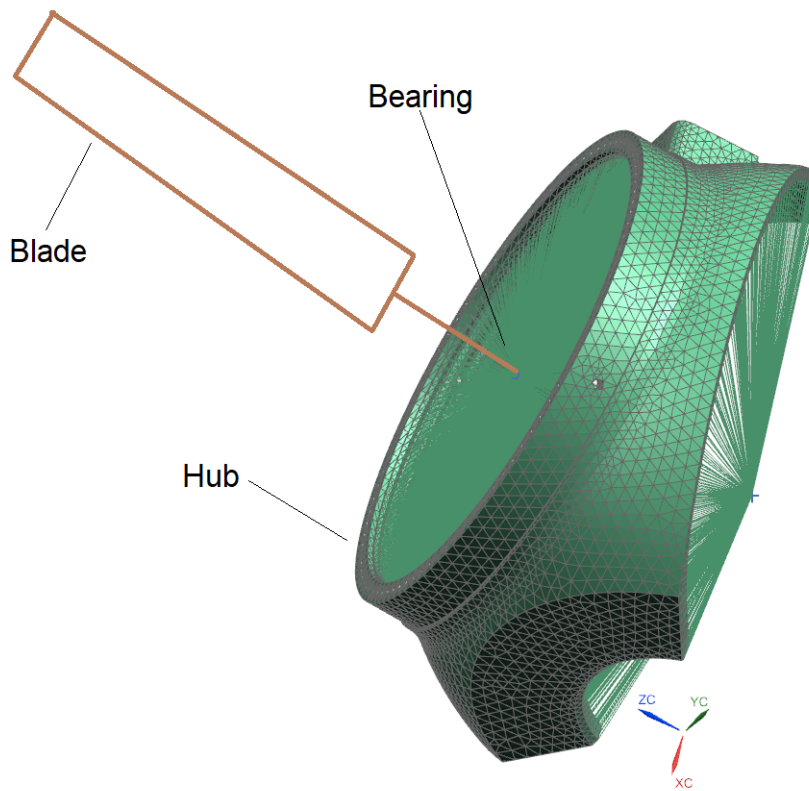


Figure 1.1: Flexible Multibody model of the system

Going more into the detail of the structural model, we observe that it is made up of finite elements of the tetrahedral type with an average size of 77 mm, the connection between the hub and the blade occurs through a connection of the BNDFREE type in all degrees

of nodal freedoms of the connection node, furthermore the node of interest is connected to the structure by means of RBE2 elements, which allows to connect the flexible hub to the bearing force element, the connection with the ground occurs by means of an SPC connection in all degrees of freedom of the connection node.
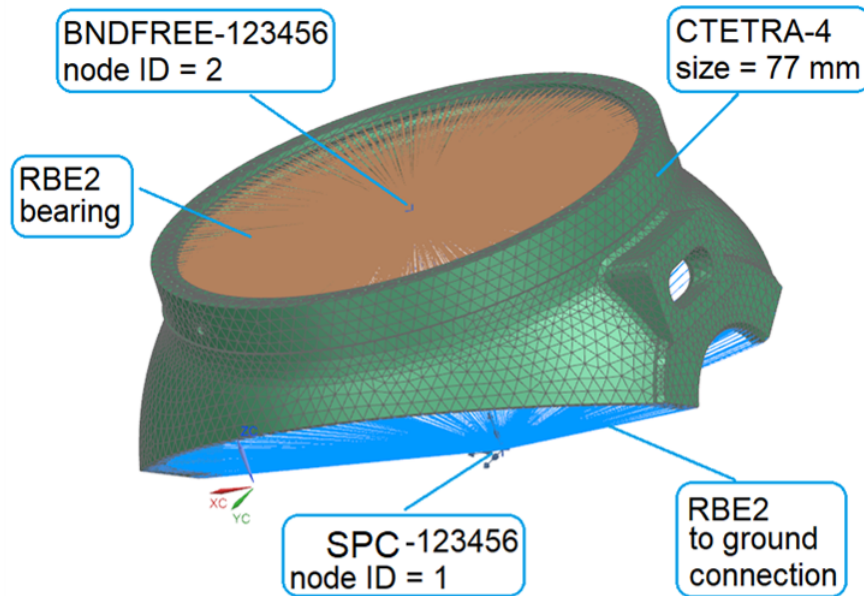


Figure 1.2: Structural Finite Element model of the hub

As already anticipated, the structural model of the hub in the complete multibody system is a reduced model, characterized by its first 10 modes of vibrations shown in the following figure.
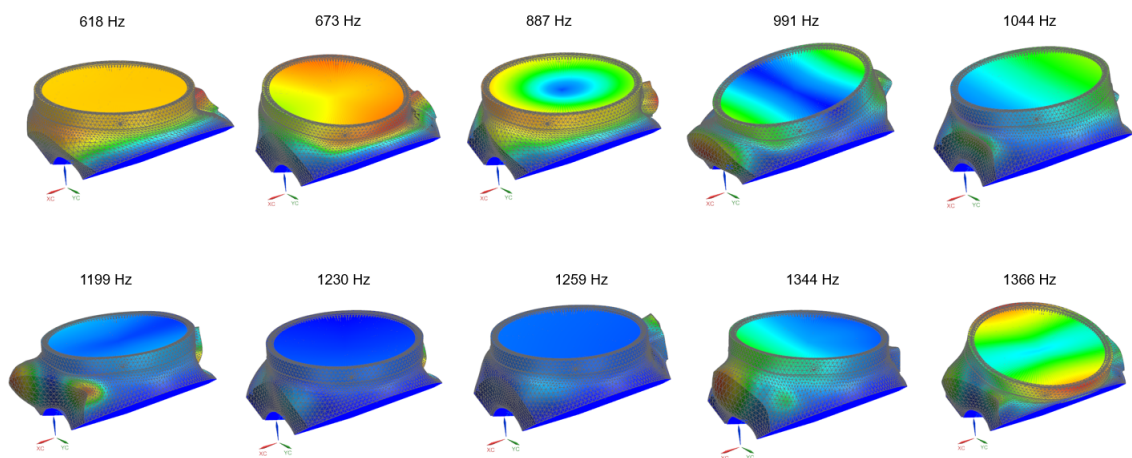


Figure 1.3: Eigenmodes and eigenfrequencies of the hub

## 1.4.   Bearing Model

The component that is located between the blade, modeled as a rigid body and the hub, modeled as a flexible body, is the ball bearing, the mathematical model that describes the behavior of this object is described below, with particular reference to the articles [8], [9] and [10].

The object of study is a four-point contact ball bearing, usually used in applications where the forces exchanged between the inner and outer rings are very high but the rotation speed is relatively low, which reflects the characteristics of our system, since usually the rotation of the blades with respect to the hub occurs at low speed.

The contacts between the spheres and rings are modeled according to Hertz's contact theory.

In this model the effects due to the gyroscopic moment are neglected because the speeds are relatively low.

A fixed reference system with respect to the inner ring and a fixed reference system with respect to the outer ring are defined, the relative displacement between the two rings is defined by the set of coordinates $\boldsymbol{q} = [q_1, q_2, q_3, q_4, q_5]$ where $q_6$ is not defined since the axial rotation of the inner ring with respect to the outer ring is allowed, moreover, in the development of the formulations described by Halpin et al. [8], the outer ring is kept fixed, look at the following figure 1.4:
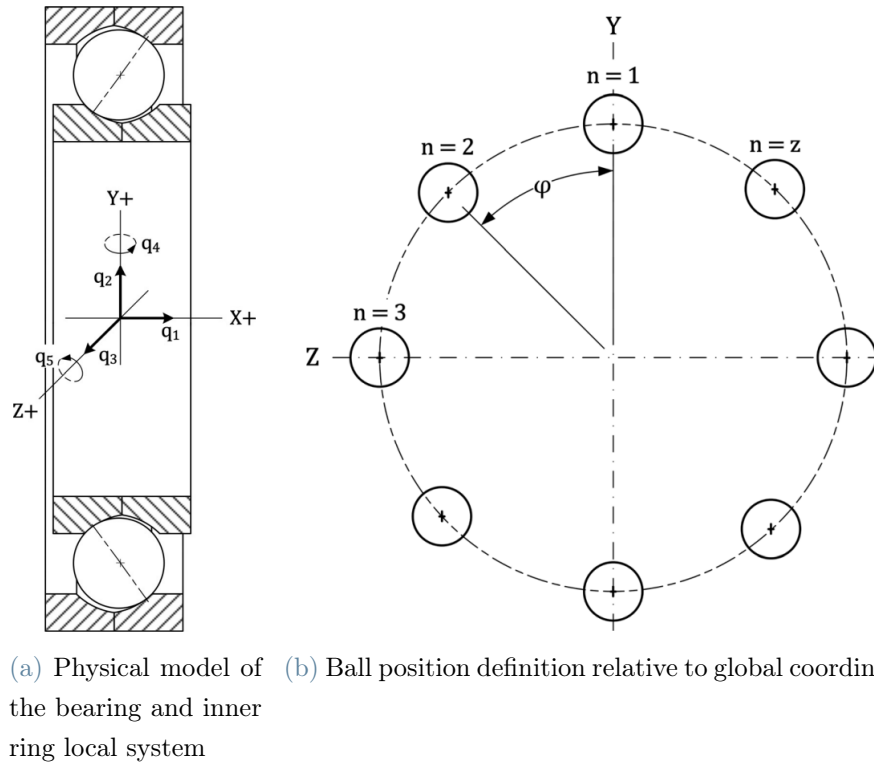
(a) Physical model of the bearing and inner ring local system  (b) Ball position definition relative to global coordinates

Figure 1.4: Physical model of the four point contact bearing, [8]

Then, the normal approach for each contact point, i.e. four contact points for each sphere, is solved under conditions of quasi-static equilibrium, obtaining the position of the spheres.

For each sphere the following system of equations is defined which describes the equilibrium of the forces acting in the sphere in the axial and radial directions, see figure 1.5:

$$\begin{cases} Q_{il}\sin\beta_{il} + Q_{ol}\sin\beta_{ol} - Q_{ir}\sin\beta_{ir} - Q_{or}\sin\beta_{or} = 0 \\ Q_{il}\cos\beta_{il} + Q_{ir}\cos\beta_{ir} - Q_{ol}\cos\beta_{ol} + Q_{or}\cos\beta_{or} + F_c = 0 \end{cases} \tag{1.18}$$
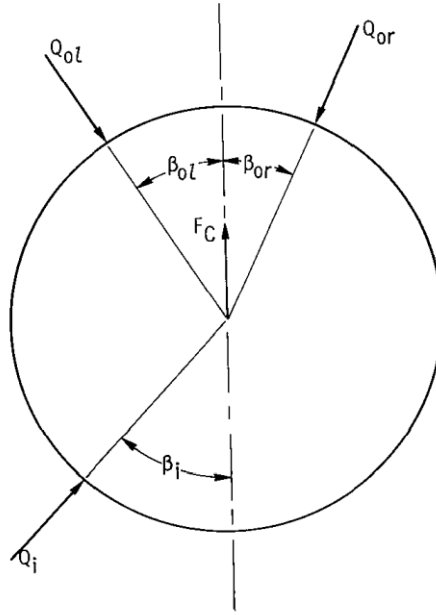
Figure 1.5: Contact forces of the bearing sphere, [9]

Where the normal contact is expressed according to Hertz's contact theory and is given by the following relation:

$$Q_{**} = K_{**}\delta_{**}^{1.5} \tag{1.19}$$

The angles $\beta_{**}$, shown in figure 1.6, represent the contact angles and are a function of the normal approach, the position of the center of the spheres $V$ and $W$ and the vector $\boldsymbol{q}$. Note that the terms $\delta_{**}$ are functions of the position of the spheres $V$ and $W$.
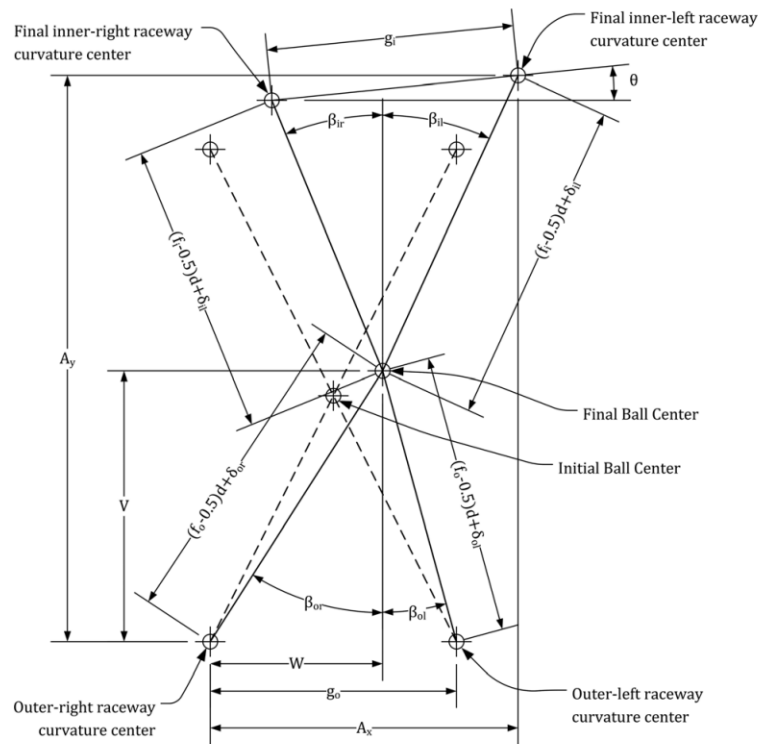
Figure 1.6: Position of ball center and raceway curvature centers at initial and final loaded positions, [8]

$K_{**}$ represent the stiffnesses of the Hertz contacts and are schematized in figure 1.7.
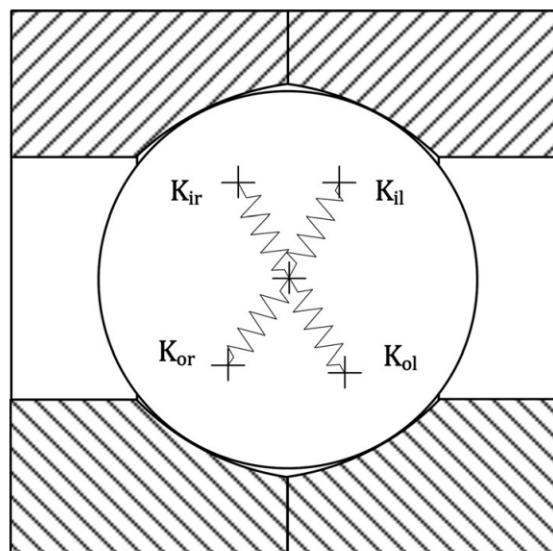


Figure 1.7: Four-point contact stiffness diagram, [8]

Moreover $F_c$ represents the centrifugal force acting at the center of each sphere and which

depends on the rotation speed.

It is possible to rewrite the previous system 1.18 in the following way:

$$\begin{cases} G(V, W, \boldsymbol{q}) = 0 \\ H(V, W, \boldsymbol{q}) = 0 \end{cases} \tag{1.20}$$

In which, once $\boldsymbol{q}$ has been defined, it is possible to solve it to obtain $V$ and $W$, that is the coordinates of the center of the given sphere.

Since the system is made up of implicit nonlinear equations, it is solved by using Newton-Raphson's method.

Also in figure 1.6 we observe the initial position of the sphere, i.e. the position in the previous integration step of the multibody system with respect to the final position of the sphere, i.e. the position obtained by solving the system 1.20.

At this point it is possible to obtain the contact forces between the spheres and the rings $Q_{**}$ simply by substituting the values just obtained of $V$ and $W$ in the appropriate equations that define the various $\delta_{**}$ and then using the 1.19.

Once all the contact forces are known, it is possible to calculate the reaction forces between the two rings by summing them:

$$F_1 = \sum_{n=1}^{z} [Q_{il,n} sin\beta_{il,n} - Q_{ir,n} sin\beta_{ir,n}] \tag{1.21}$$

$$F_2 = \sum_{n=1}^{z} [Q_{il,n} cos\beta_{il,n} + Q_{ir,n} cos\beta_{ir,n}] cos\varphi \tag{1.22}$$

$$F_3 = \sum_{n=1}^{z} [Q_{il,n} cos\beta_{il,n} + Q_{ir,n} cos\beta_{ir,n}] sin\varphi \tag{1.23}$$

$$F_4 = \sum_{n=1}^{z} -R_i [Q_{il,n} sin\beta_{il,n} - Q_{ir,n} sin\beta_{ir,n}] sin\varphi \tag{1.24}$$

$$F_5 = \sum_{n=1}^{z} -R_i [Q_{il,n} sin\beta_{il,n} - Q_{ir,n} sin\beta_{ir,n}] cos\varphi \tag{1.25}$$

Where $z$ represents the number of balls in the bearing, $R_i$ represents the radius from bearing spin axis to inner raceway curvature center at initial free contact, and $\varphi$ represents the ball station clocking angle.

Obviously $F_6$ is not defined as the inner ring is free to rotate with respect to the outer ring around the bearing axis.

The bearing is integrated into the overall multibody formulation as a nonlinear force element. The nonlinear implicit equations, as described above, are solved per global iteration step when solving the multibody system equations, see figure 1.8.

$$\boldsymbol{q}_k \implies \text{Solve system (1.20)} \implies Q_{**,n}$$

$$\boldsymbol{q}_{k+1} \impliedby \text{Solve system (1.7)} \impliedby F_{1,2,3,4,5}$$
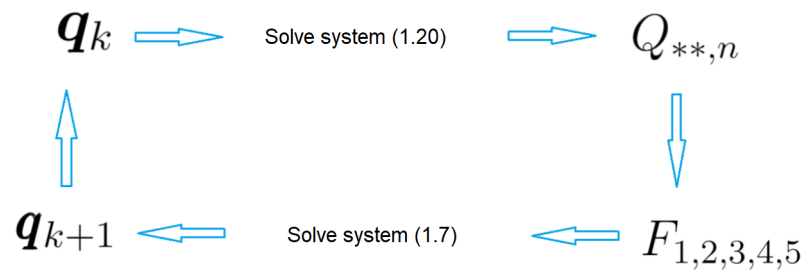
Figure 1.8: Solver scheme

## 1.5.   Extended Kalman Filter

In many engineering applications it is necessary to evaluate the internal, typically unmeasured, states of a system given a limited set of measurements of specific quantities taken at specific locations. Solutions to this kind of problems can be found by making use of a dynamic model of the system under investigation in combination with the limited set of measurements available. Such system model represents a dynamic relationship between the latter and the internal system states and forms the basis of any so-called model based estimation technique. The main part of such model is typically given by a differential equation which governs the time evolution of the system internal states, which is also referred to as the process equation. In many application cases this equation takes the following form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1.26}$$

i.e. of a continuous-time explicit state space equation.

The previous equation is a first order explicit Ordinary Differential Equation (ODEs) in the state vector $\boldsymbol{x} \in \mathbb{R}^{n_S}$, which contains the system internal states, $\boldsymbol{u} \in \mathbb{R}^{n_I}$ is instead the input vector, which contains all the external actions that affect the system behavior, e.g. the external forces and moments acting on a structure, $\boldsymbol{f}$ is a real vector-valued non-linear function of dimension $n_S$. This is the form for which most of the state estimation methods have been initially developed.

Next to the process equation a so-called measurement equation is defined as:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{1.27}$$

which relates the internal states to a specific measurement set taken on the real system and collected in the measurement vector $\boldsymbol{y} \in \mathbb{R}^{n_M}$. Model inaccuracies are then always present as well as measurement errors and noise.

For these reasons modeling uncertainties are typically introduced to take into account for model inaccuracies by adding a so-called process noise vector $\boldsymbol{w} \in \mathbb{R}^{n_S}$ to the right hand side of the process equation. Analogously a so-called measurement noise vector $\boldsymbol{v} \in \mathbb{R}^{n_O}$ is added to the right hand side of the measurement equation to take into account for measurement errors and noise. The noise covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are associated to the noise vectors $\boldsymbol{w}$ and $\boldsymbol{v}$, respectively.

The objective of any model based state estimation method can now be stated as estimating the sequence of state vectors $\boldsymbol{x}_k$ given the sequence of noisy measurements $\boldsymbol{y}_k$ and deterministic inputs $\boldsymbol{u}_k$. After an estimation is carried out, the sequence of retrieved state vectors $\boldsymbol{x}_k$ can be used to estimate any quantity at any location in the model as specified by an output equation as follows:

$$\boldsymbol{y}_k^O = \boldsymbol{h}^O(\boldsymbol{x}_k) \tag{1.28}$$

where $\boldsymbol{y}_k^O \in \mathbb{R}^{n_O}$ is the output vector.

The Kalman filter is an algorithm that basically allows to perform the estimation of the state vector, however, in the case of non-linear systems, it is necessary to linearize the system in a series of certain points as will be explained in more detail in this paragraph, this filter is called Extended Kalman Filter, for more information see Simon's book [21].

Suppose there is the following nonlinear system in explicit form:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \end{cases} \tag{1.29}$$

Whose discretized form is expressed by:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_{k+1}) \\ \boldsymbol{y}_{k+1} = \boldsymbol{h}_d(\boldsymbol{x}_{k+1}) \end{cases} \tag{1.30}$$

Several discretization methods can be found on the book [23].

The key in solving the problem on the application of the Kalman filter in non-linear systems consists in linearizing the system at the point given by the corrected estimate due to the measurements (called "a posteriori" estimate) of the previous step:

$$\boldsymbol{A} = \left. \frac{\partial \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_{k+1})}{\partial \boldsymbol{x}_k} \right|_{\hat{\boldsymbol{x}}_k^+} \tag{1.31}$$

where:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{v} \end{bmatrix} \tag{1.32}$$

On the other hand, as regards the measurement equations, the linearization takes place around the "a priori" estimate of the current integration step:

$$H = \left.\frac{\partial h_d(x_{k+1})}{\partial x_k}\right|_{\hat{x}_{k+1}^-} \tag{1.33}$$

The following equation represents the prediction of the state error covariance:

$$P_{k+1}^- = A P_k^+ A^T + Q \tag{1.34}$$

Where $Q$ represents the process covariance matrix, that is a covariance matrix that expresses the grade of accuracy of the system model.

The following equation represents the Kalman gain:

$$K_{k+1} = P_{k+1}^- H^T \left(H P_{k+1}^- H^T + R\right)^{-1} \tag{1.35}$$

$R$ represents the covariance matrix of the measurements, which is an indicator of how much the measurements are corrupted by noise.

Note that in the setup of a Kalman filter, impose $Q > R$ means to say that we are more uncertain in the model than the measurements, this means that the estimates will tend to follow the measurements rather than rely on the model, the final result will be estimates affected by noise if the measurements are characterized by noise, on the contrary, if $Q < R$ means that we are confident in the accuracy of our model, therefore the estimates will tend to follow more the output of the model that the measurements themselves, the final result will be estimates characterized by less noise than in the previous case, however it is necessary to pay particular attention to this case since, if the model is not correct, the estimates will tend to diverge from the measurements leading to incorrect results.

The following equation represents the correction of the state prediction in light of the knowledge of the new measurements:

$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + K_{k+1}(y_{k+1} - h_d(\hat{x}_{k+1}^-)) \tag{1.36}$$

Where $y_{k+1}$ represents the measurement of the current integration step.

The following equation represents the correction of the state error covariance matrix:

$$\boldsymbol{P}^+_{k+1} = (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H})\,\boldsymbol{P}^-_{k+1} \tag{1.37}$$

## 1.6.   Augmented Extended Kalman Filter - State Input Estimation

It is possible to estimate the states $\boldsymbol{x}$ and the unknown inputs $\boldsymbol{u}^{UK}$ simultaneously using a Kalman filter, simply considering the inputs as if they were new states and adding them to the state vector itself, so generating the augmented state vector, [2], [4], [5], [18], [19]:

$$\boldsymbol{x}^* = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u}^{UK} \end{bmatrix} \tag{1.38}$$

In order to define the Augmented Kalman Filter, the dynamics of the input has also to be modeled, so a zeroth-order random walk model for the vector of unknown inputs is introduced:

$$\boldsymbol{u}_{k+1}^{UK} = \boldsymbol{u}_k^{UK} + \boldsymbol{w}_k^{UK} \tag{1.39}$$

where $\boldsymbol{w}_k^{UK}$ is a white, zero mean, uncorrelated random process with the associate covariance matrix $\boldsymbol{Q}_u$.

The selection of $\boldsymbol{Q}_u$ is done using the same approach used by Cumbo et al. [4]:

$$\boldsymbol{Q}_u = (\Delta t \cdot \bar{\omega}_u \cdot \bar{a}_u)^2 \tag{1.40}$$

Where $\bar{\omega}_u$ and $\bar{a}_u$ represent indicative values chosen by the user and have the same order of magnitude of the input frequency and amplitude respectively.

Clearly, the matrices that characterize the dynamic system in the form of a discretized state will be modified as follows:

$$\boldsymbol{A}^* = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_u \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{1.41}$$

$$\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \tag{1.42}$$

Usually the output does not depend on the input, so $\boldsymbol{D} = \boldsymbol{0}$.

$$
\begin{cases}
\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_{k+1}^{UK} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_u \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k^{UK} \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{w}_k^{UK} \end{bmatrix} \\
\boldsymbol{y}_k = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k^{UK} \end{bmatrix} + \boldsymbol{v}_k
\end{cases}
\tag{1.43}
$$

Where $\boldsymbol{w}_k$ is the Gaussian process noise and $\boldsymbol{v}_k$ is the Gaussian measurement noise.

Written in compact form:

$$
\begin{cases}
\boldsymbol{x}_{k+1}^* = \boldsymbol{A}^* \boldsymbol{x}_k^* + \boldsymbol{w}_k^* \\
\boldsymbol{y}_k = \boldsymbol{H}^* \boldsymbol{x}_k^* + \boldsymbol{v}_k
\end{cases}
\tag{1.44}
$$

Therefore, the augmented covariance matrix $\boldsymbol{Q}^*$ will be written as:

$$
\boldsymbol{Q}^* = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_u \end{bmatrix}
\tag{1.45}
$$

where:

$$
\boldsymbol{Q}_u >> \boldsymbol{Q}
\tag{1.46}
$$

The pseudo-algorithm of the Augmented Extended Kalman Filter for state-input estimation is shown below:

---

**Algorithm 1** Augmented Extended Kalman Filter - Input estimation

---

1: Starting from the system of implicit non linear equations expressed in the GGL formulation:

$$
\begin{cases}
\frac{1}{h}(\boldsymbol{q}_{k+1} - \boldsymbol{q}_k) - \boldsymbol{v}_{k+1} + \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\mu}_{k+1} = 0 \\
\boldsymbol{M}(\boldsymbol{q}_{k+1}) \frac{1}{h}(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k) + \boldsymbol{f}(\boldsymbol{q}_{k+1}, \boldsymbol{v}_{k+1}, \boldsymbol{u}_{k+1}) - \boldsymbol{B}(\boldsymbol{q}_{k+1})^T \boldsymbol{\lambda}_{k+1} = 0 \\
\boldsymbol{B}(\boldsymbol{q}_{k+1}) \boldsymbol{v}_{k+1} = 0 \\
\boldsymbol{\phi}(\boldsymbol{q}_{k+1}) = 0 \\
\boldsymbol{u}_{k+1}^{UK} = \boldsymbol{u}_k^{UK}
\end{cases}
\tag{1.47}
$$

where the input vector is split in two subvectors, one defining the known input vector, $\boldsymbol{u}_{k+1}^K$ and another one defining the unknown input vector $\boldsymbol{u}_{k+1}^{UK}$, which is the one we need to estimate:

$$
\boldsymbol{u}_{k+1} = \begin{bmatrix} \boldsymbol{u}_{k+1}^K \\ \boldsymbol{u}_{k+1}^{UK} \end{bmatrix}
\tag{1.48}
$$

for simplicity we define $\boldsymbol{u}_{k+1}^{UK} = \hat{\boldsymbol{u}}_{k+1}^-$.

The 1.47 is written in compact form as:

$$
\boldsymbol{g}_d(\hat{\boldsymbol{x}}_{k+1}^-, \hat{\boldsymbol{x}}_k^+, \hat{\boldsymbol{u}}_{k+1}^-) = 0
\tag{1.49}
$$

so we find an explicit formulation in the form of, [2]:

$$
\hat{\boldsymbol{x}}_{k+1}^- = \boldsymbol{f}_d(\hat{\boldsymbol{x}}_k^+, \hat{\boldsymbol{u}}_{k+1}^-)
\tag{1.50}
$$

where:

$$
\hat{\boldsymbol{x}}_{k+1}^- = \begin{bmatrix} \boldsymbol{q}_{k+1} \\ \boldsymbol{v}_{k+1} \end{bmatrix}
\tag{1.51}
$$

$$
\hat{\boldsymbol{x}}_k^+ = \begin{bmatrix} \boldsymbol{q}_k \\ \boldsymbol{v}_k \end{bmatrix}
\tag{1.52}
$$

2: Knowing $\hat{\boldsymbol{x}}_k^+$ and $\hat{\boldsymbol{u}}_{k+1}^-$, compute a linearization of the state matrix A:

$$
\boldsymbol{A} = \left. \frac{\partial \boldsymbol{f}_d(\hat{\boldsymbol{x}}_k^+, \hat{\boldsymbol{u}}_{k+1}^-)}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}_k^+}
\tag{1.53}
$$

3: Compute also a linearization of the system with respect to the unknown input vector in the "a priori" input prediction of the current step $\hat{\boldsymbol{u}}_{k+1}^-$:

$$\boldsymbol{B}_u = \left.\frac{\partial \boldsymbol{f}_d(\hat{\boldsymbol{x}}_k^+, \hat{\boldsymbol{u}}_{k+1}^-)}{\partial \boldsymbol{u}^{UK}}\right|_{\hat{\boldsymbol{u}}_{k+1}^-} \tag{1.54}$$

4: Find the "a priori" state prediction solving the implicit system of equations 1.47, obtaining the "a priori" state prediction $\hat{\boldsymbol{x}}_{k+1}^-$.

5: Assemble the augmented matrices:

$$\boldsymbol{A}^* = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}_u \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{1.55}$$

$$\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \tag{1.56}$$

6: Compute the "a priori" state error covariance matrix:

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{A}^* \, \boldsymbol{P}_k^+ \, \boldsymbol{A}^{*T} + \boldsymbol{Q}^* \tag{1.57}$$

7: Compute the Kalman gain:

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^- \, \boldsymbol{H}^{*T} \left( \boldsymbol{H}^* \, \boldsymbol{P}_{k+1}^- \, \boldsymbol{H}^{*T} + \boldsymbol{R} \right)^{-1} \tag{1.58}$$

8: Compute the "a posteriori" augmented states using the available measurements, note that the method on which the constraint are taken in to account is referred to Adducci et al. [1]:

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{k+1}^+ \\ \hat{\boldsymbol{u}}_{k+1}^+ \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_{k+1}^- \\ \hat{\boldsymbol{u}}_{k+1}^- \end{bmatrix} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{h}_d(\hat{\boldsymbol{x}}_{k+1}^-)) \tag{1.59}$$

9: Compute the "a posteriori" state error covariance matrix:

$$\boldsymbol{P}_{k+1}^+ = (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}^*) \, \boldsymbol{P}_{k+1}^- \tag{1.60}$$

Some important observations coming from Capalbo et al. [2] need to be made:

- To have the maximum theoretical observability, a number of strain sensors must be used at least equal to the total number of quantities to be estimated (inputs and parameters).

- Mass parameters cannot be observed in static conditions, therefore in the estimation, for example, of the mass of a body it is necessary to use a dynamic input. Moreover, the input must be characterized by a frequency more or less similar to the dynamic excitation frequency of the component.

## 1.7.    Augmented Extended Kalman Filter - State Parameter Estimation

In the case of estimating the parameters, the procedure is similar to that used for the estimation of the input, so the parameter is added to the vector of states and considered as if it were a new state, [2], [16].

## 1.8.    Measurement Equation for Strain Gauges

The sensors that will be used in the work of this thesis are strain sensors known as Strain Gauges, consequently, in this paragraph the method that allows to obtain the measurement equations is shown by referring to Appendix B of Risaliti's Phd Thesis [17].

# 2 | Input Load Estimation - Numerical Validation

The measurements are not taken from real sensors placed in the real test bench, a model of the test bench it self is instead created to virtually collect measurements.

In this thesis, the measurements were simulated by creating a model described in the previous paragraphs, but using real input forces, i.e. measured in the real test bench, we therefore see that it is of a hybrid approach, virtual-real, see figure 2.1.
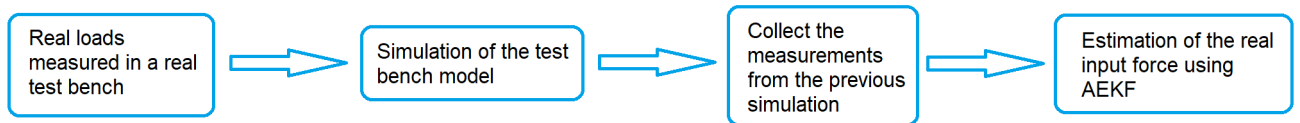


Figure 2.1: General procedure for input load estimation

In the following plot we will see the results of the AEKF with data without noise. As can be seen, good results have been obtained since the estimated load seems to be almost perfectly matched with the real one.
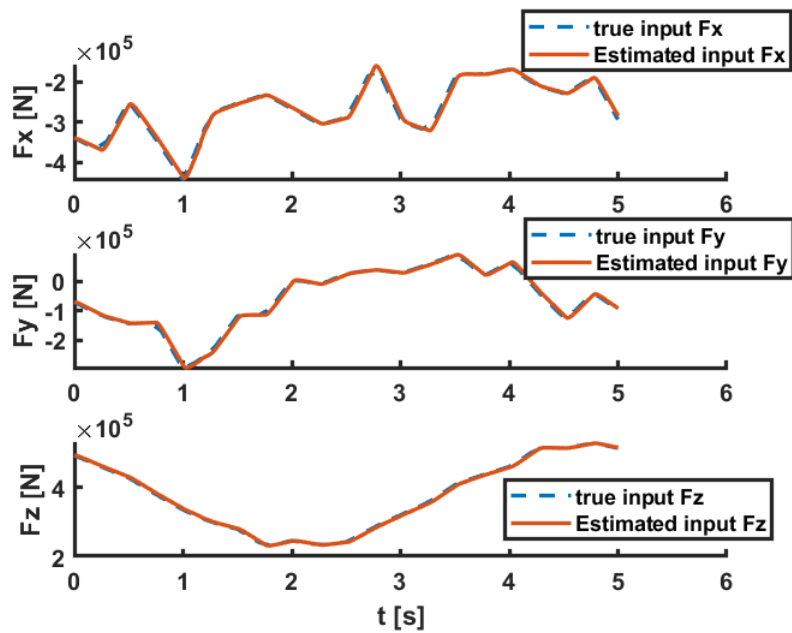
Figure 2.2: Estimated input forces, case without noise

Also the estimated strain gages measurements are well estimated as can be seen in the following figure.
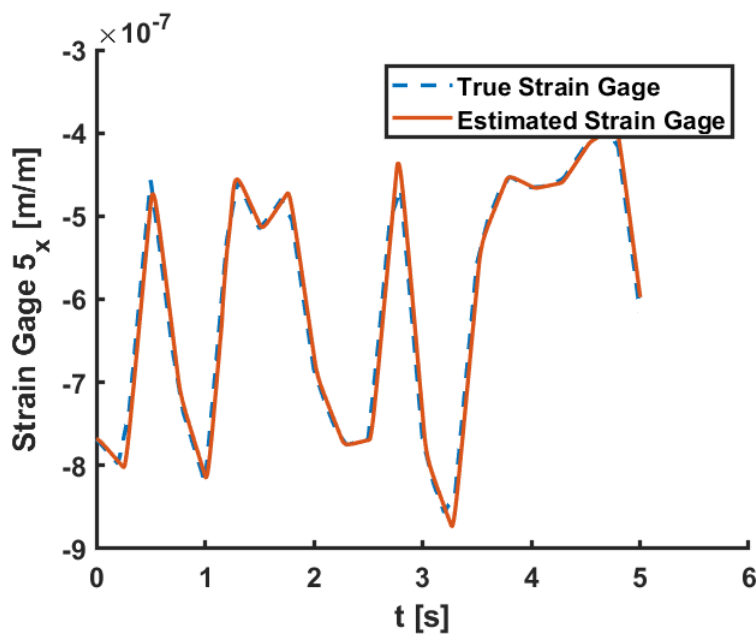


Figure 2.3: Strain gage measurement taken from the hub, case without noise

A sensitivity study has been conducted to set the Kalman filter parameters. The influence of the time step size, ranging from 0.001 to 0.3 second, given a fixed plant noise covariance

value for the augmented states, is shown in 2.4:
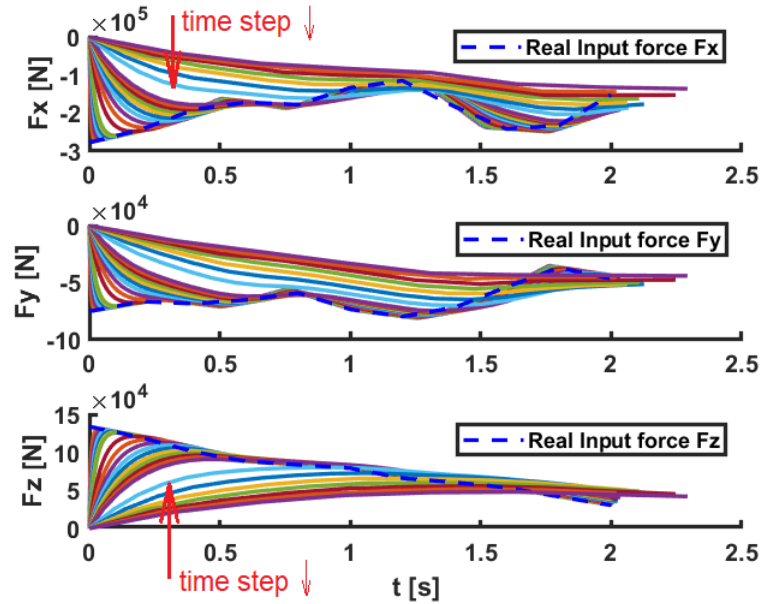


Figure 2.4: Input load estimation for different values of time step, case without noise

In the following figure, a comparison between different noise level cases is shown for the input estimation, with the following standard deviations for the zero-mean white Gaussian noise values added to the measurements:

Table 2.1: Noise levels

| Noise level | value |
|:---:|:---:|
| low noise | 1e-8 |
| regular noise | 1e-7 |
| high noise | 5e-7 |
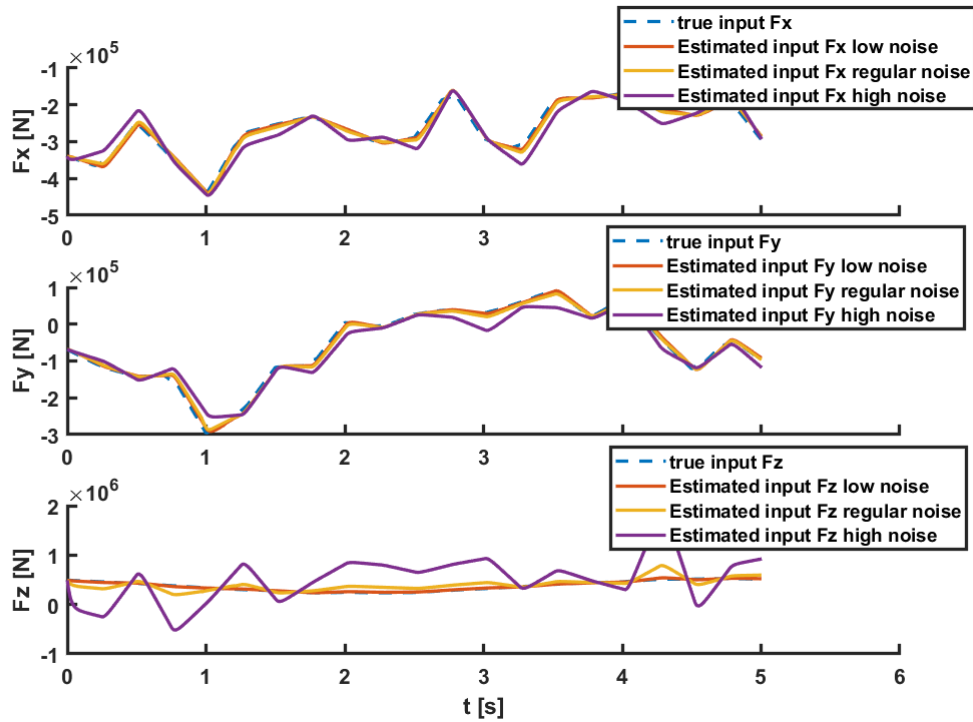
Figure 2.5: Estimated input forces with noisy data

It is observed that the noise in the measurements plays an important role, in fact we have a worse estimate of the loads, especially for the component along the z axis.
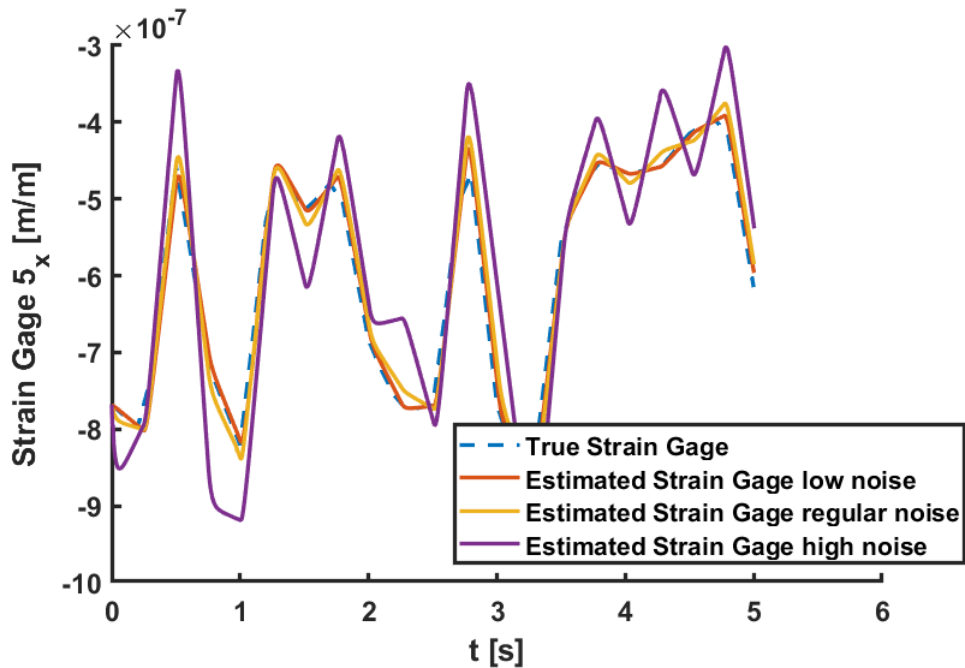


Figure 2.6: Strain gage measurement taken from the hub for the different noise levels

Again in the case of input estimation, it is interesting to note how, while imposing a model with incorrect bearing parameters with in particular 10% higher values with respect to correct one, the algorithm is still able to estimate the three components of the input, this because the bearing acts as a force element in the multibody system, and will always be in equilibrium with the applied and reaction forces at its interfaces. As strain measurements taken from the hub are used, the estimated force will therefore be correct. The internal state variables of the bearing itself will be wrong however. The noise level in this case is regular:
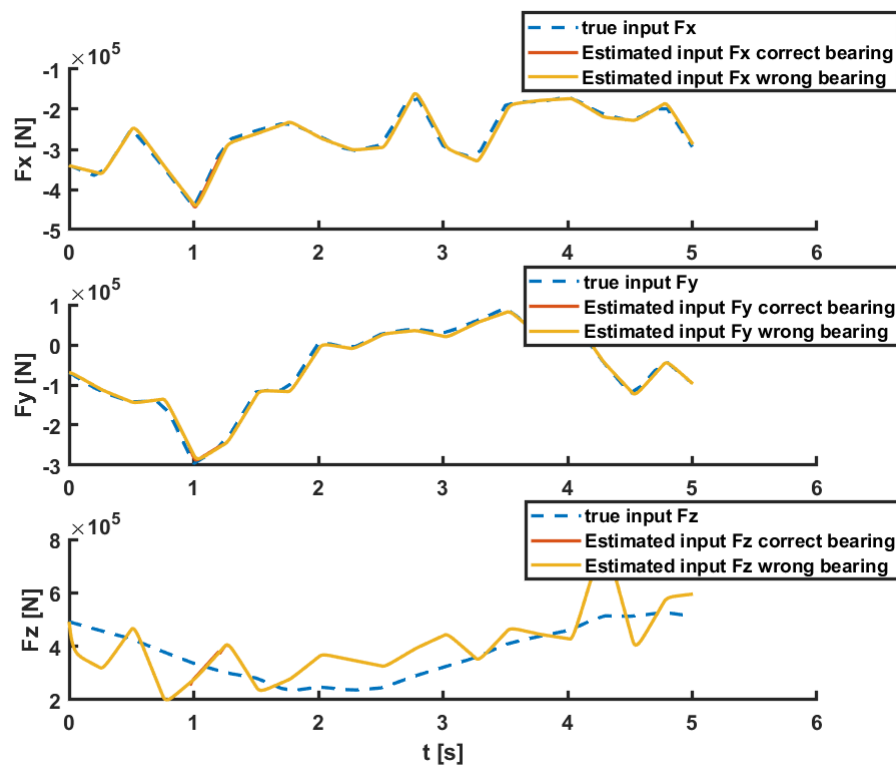


Figure 2.7: Comparison between estimated input forces with correct and wrong bearing parameters, regular noise level

If, on the other hand, the hub model is wrong and in particular with a 10% higher Young's modulus than the real one, we notice an OK estimate of the y component of the input, a very incorrect z component of the input and a component x of larger magnitude than the real one, in these estimations, the noise level is regular:
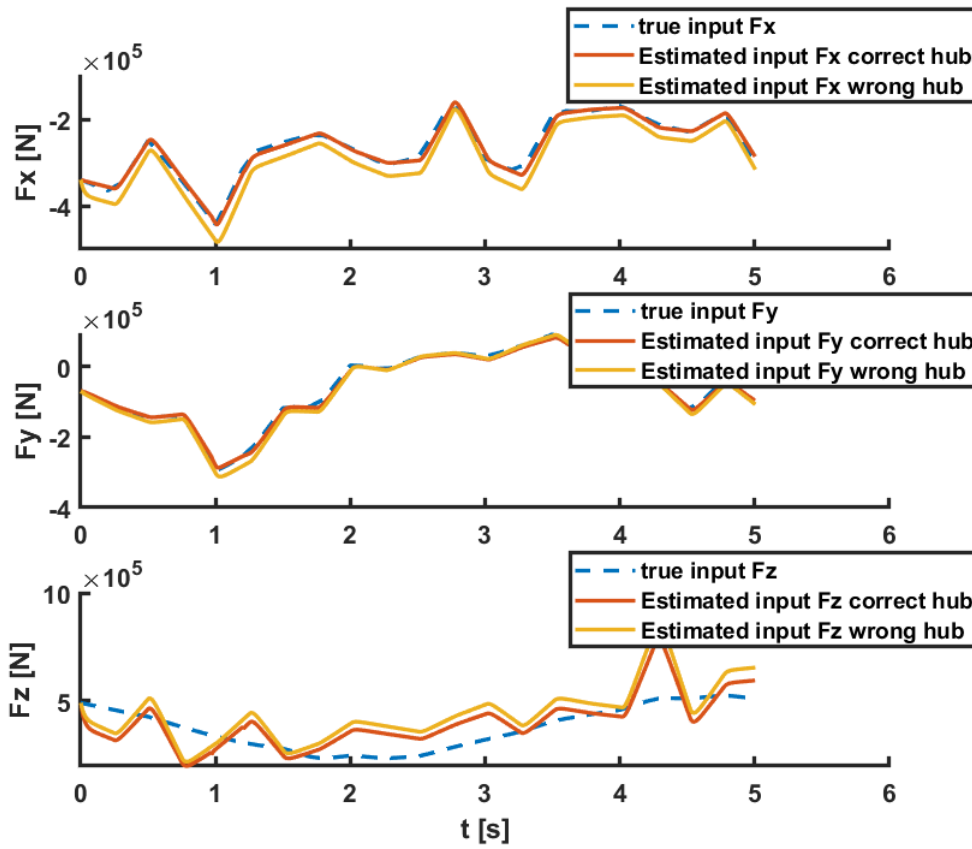
Figure 2.8: Comparison between estimated input forces with correct and wrong hub Young Modulus, regular noise level

Note that, in this case, larger force component values are expected, as the wrong Young's modulus requires larger forces to be applied to obtain the same strain values as the supplied measured ones.

# 3 | Parameter Estimation - Numerical Validation

Some parameters that characterize the system play a fundamental role in the modeling of the system itself and often, some of them are not easily measurable even using the best state-of-the-art instrumentation, consequently having a method that allows their estimation can result a big plus.

The results for the estimation of some parameters that characterize the bearing and which can not be measured in the laboratory are presented below.

In particular, the diameter of the balls and the raceway radius are estimated. In the adopted model, four independent radii are considered.
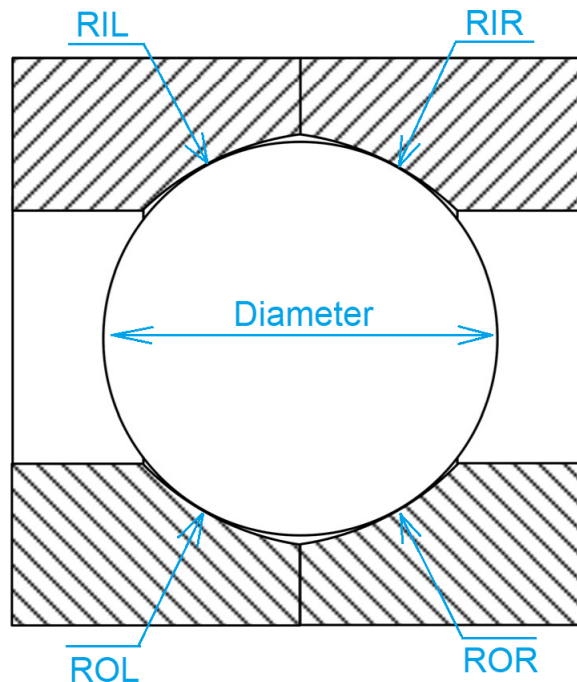


Figure 3.1: Rolling element parameters, [8]

The following figure shows the result of the estimation of the diameter of the spheres in the case of ideal measurements without noise.

The estimate was performed for different time step values ordered in the table 3.1 to show that its reduction allows for a faster convergence, also characterized by more dampened numerical oscillations, on the contrary, an increase in the time step allows to reach the convergence more slowly, the computational time is however shorter in the case of the larger time step.

Table 3.1: Time step sizes

$$0.11 \ [s]$$
$$0.15 \ [s]$$
$$0.2 \ [s]$$
$$0.2513 \ [s]$$
$$0.3 \ [s]$$
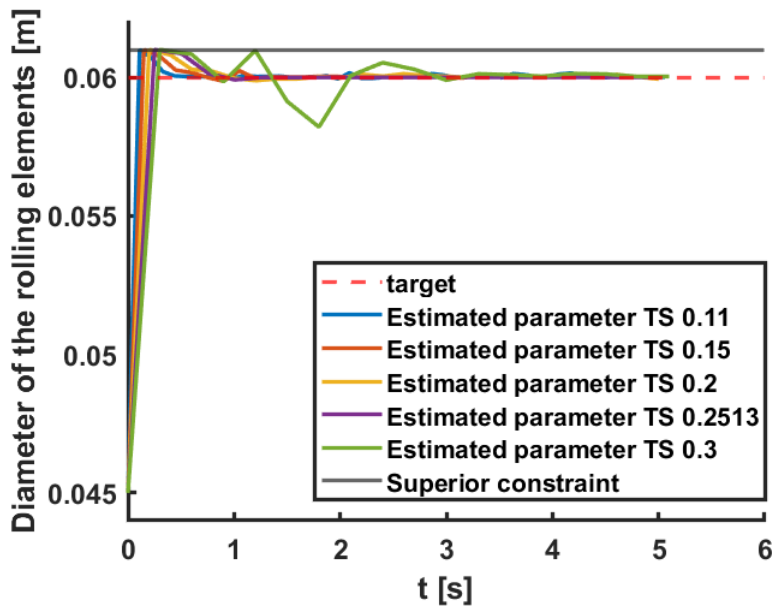


Figure 3.2: Estimated diameter of the rolling elements, for different time step sizes, case without noise

Note that too large a time step could lead to the divergence of the estimate and therefore a stop of the algorithm.

The following figure shows that the Kalman filter is able to estimate the measurements of the sensors used well.

The relative sensor is a sensor that allows to measure the displacement of the bearing inner ring with respect to the outer ring.
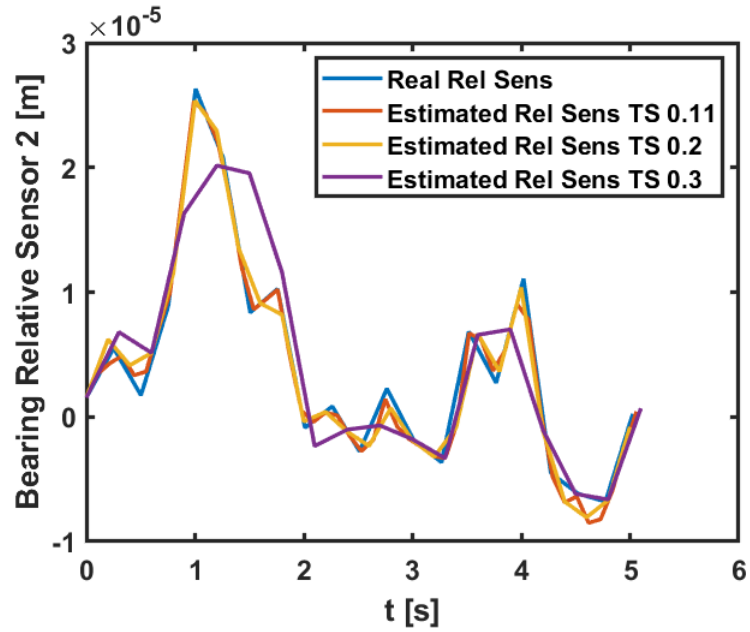


Figure 3.3: Estimated bearing relative sensor, for different time step sizes, case without noise

The following figures show the results for the Raceway Inner Radius (RIL) estimation with different values of plant covariance.

Table 3.2: Plant noise covariance values

5e0
1e1
5e1
1e2
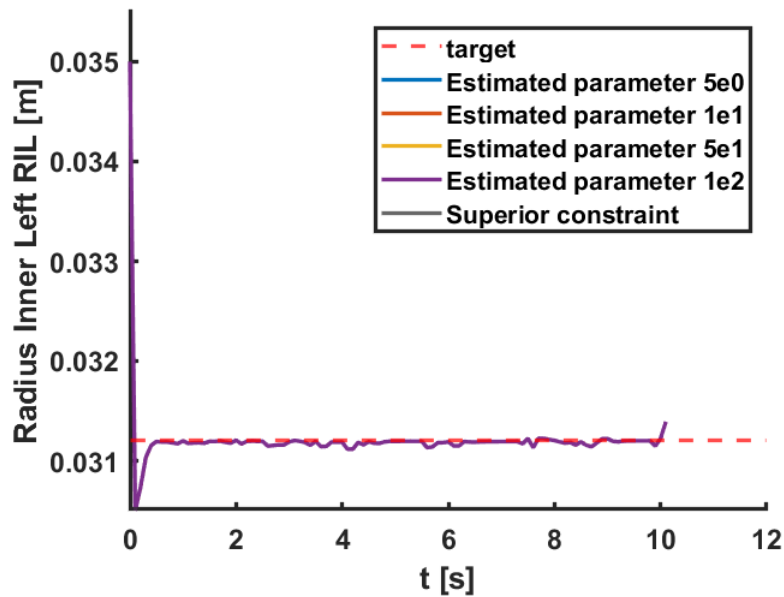
Figure 3.4: Estimated RIL, for different augmented state plant noise covariance values, case without noise
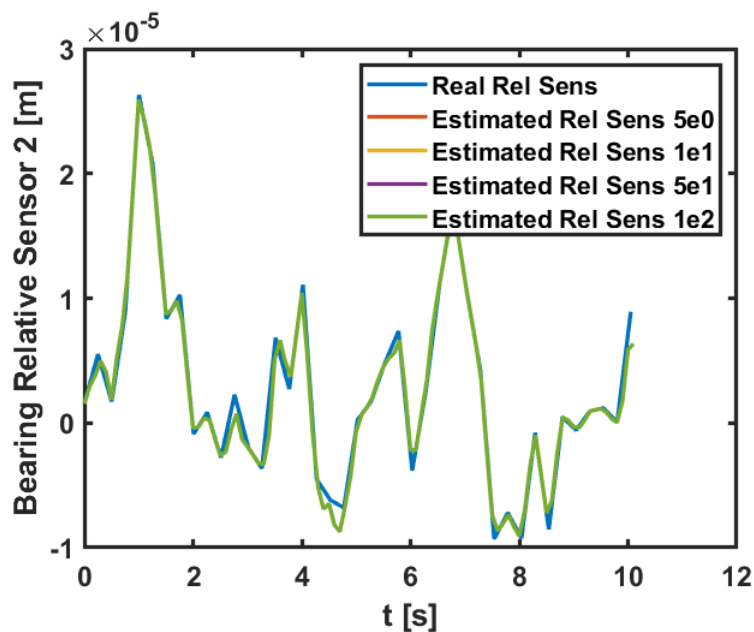


Figure 3.5: Estimated bearing relative sensor, for different augmented state plant noise covariance values, case without noise

From the two previous figures, it can be noted that the value of the plant covariance noise does not play a relevant role in the result as the same result of the estimates is observed for different values of the plant covariance noise.

Given that the estimated parameters have constant values, this behavior is to be expected, Risaliti et al. [17].

As in the case of the input estimate, also in the case of parameter estimation, the presence of noise in the measurements makes the parameter estimate worse, as can be seen in the following figures.
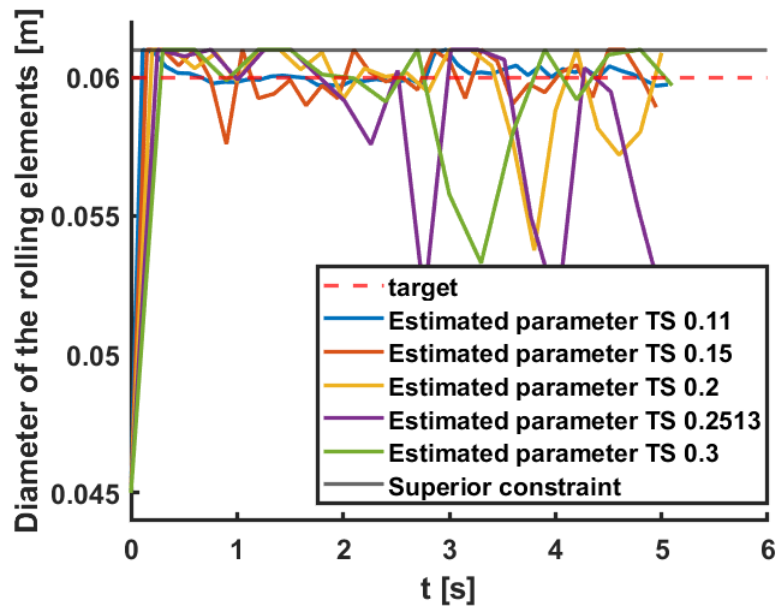


Figure 3.6: Estimated diameter of the rolling elements, regular noise level
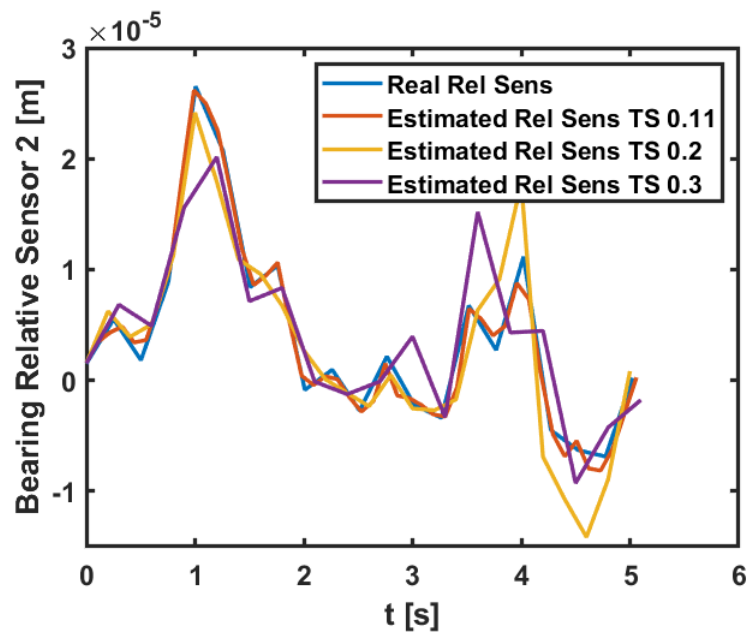


Figure 3.7: Estimated bearing relative sensor, regular noise level

Note that in the previous two figures, a comparison of the estimation for different time step sizes has been done as in the diameter of the bearing rolling element estimation without noise to show again, how the time step size can play an important role in the estimation of the parameter.

The following figures show the results for the Raceway Inner Radius (RIL) estimation in the case of noisy measurements for different values of the plant noise covariance.
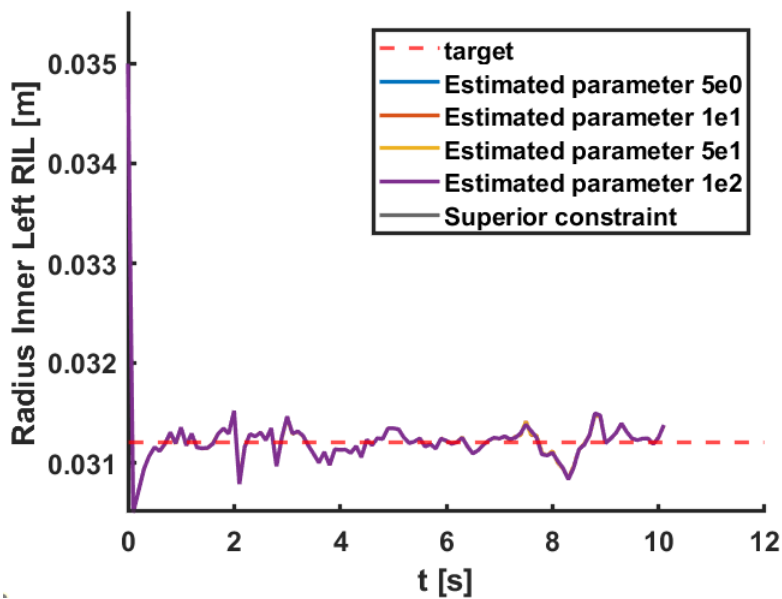


Figure 3.8: Estimated RIL, regular noise level
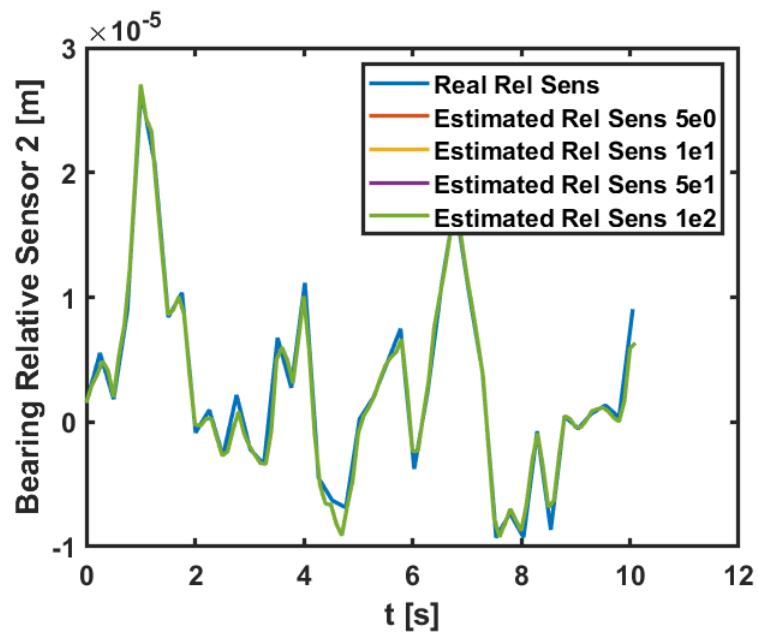
Figure 3.9: Estimated bearing relative sensor, regular noise level

From the last two figures it can be seen that, as in the case of the RIL parameter without noise, the estimation results are not so sensitive with the plant covariance noise.

# 4 | Test Bench Down Scaling

For example, suppose that we have a specific wind turbine of a certain size, in order to carry out test campaigns, it is necessary to create a test bench that has the same size as the turbine being analyzed, however, there are different types of wind turbines on the market of different size, it would therefore be economically inconvenient to create a different test bench for each one of them, then having a tool that allows you to create a single test bench of reduced dimensions and simply scale the external loads, appropriately estimated on the basis of the type of turbine that we are testing, could help a lot in reducing costs for testing campaigns.

In particular, starting from a test bench model of a certain size associated with the 3.4 MW wind turbine, we want to create a scaled test bench model, i.e. of reduced size, so that it allows to reproduce the same KPIs, like e.g. displacements, of the ball bearing using a particular estimated load:

1. Simulate the full-scale test bench model by applying a known static or dynamic load and collect sensor measurements.

2. Create a test bench model that is scaled, i.e. of smaller size with respect to the previous model.

3. Apply the AEKF algorithm to the scaled model to estimate the input forces needed to have the same displacement state measured in the previous simulation of the full scale model.

In figure 4.1 we can observe the trend of the forces estimated for the scaled test bench, in the case of measurements with and without noise compared with the forces applied in the original test bench.

These estimated loads thus have to be applied to the scaled test bench to obtain the same KPIs as the full-scale testbench, like e.g. bearing displacements.
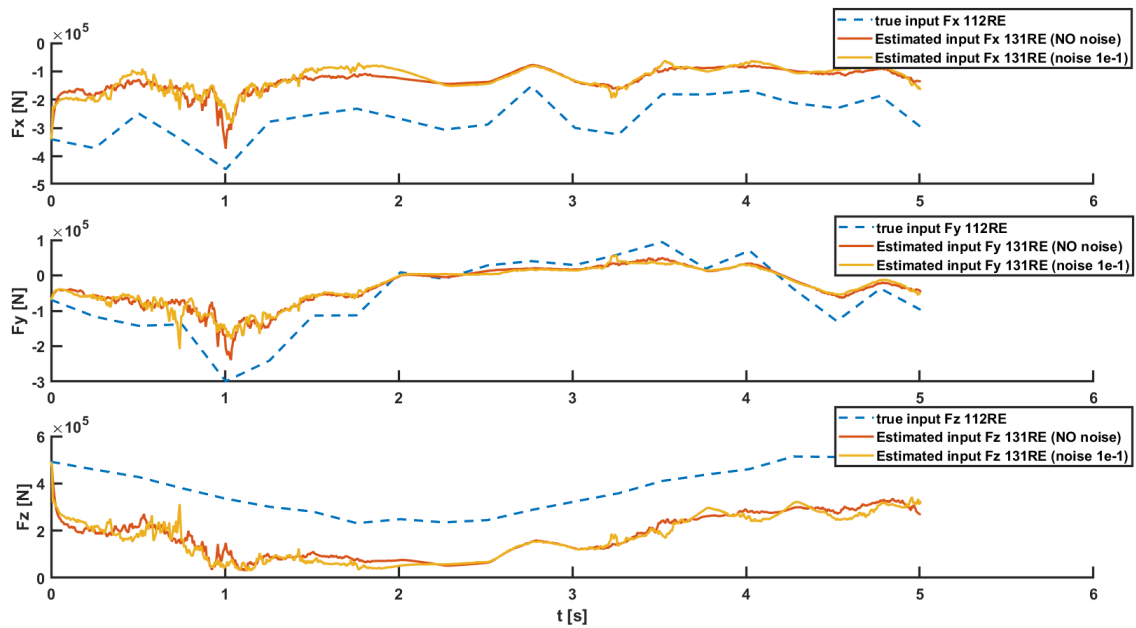
Figure 4.1: Estimated input forces in the scaled test bench model

In figure 4.2 we observe the trend of the measurements of the relative sensor in the bearing with respect to those estimated by the Kalman filter for measurements without noise, we can see a good matching between the two.
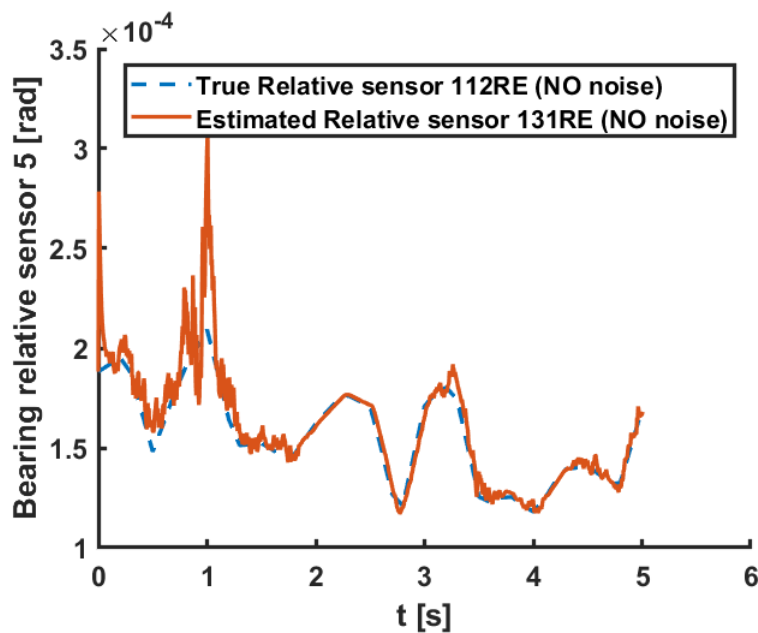


Figure 4.2: Estimated state of the relative displacement sensor of the bearing (case without noise)

Note that the 112RE is the full-scale testbench and the 131RE is the scaled testbench.

In the realistic case where the measurements are affected by noise we still have a matching between the real signal and the estimated one but with more random behavior, see figure 4.3
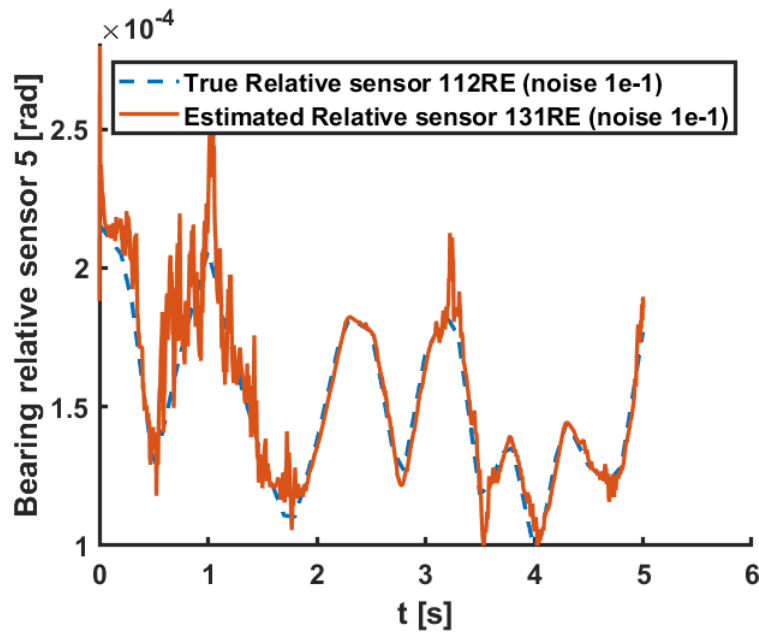


Figure 4.3: Estimated state of the relative displacement sensor of the bearing (case with noise)

In the following figures we observe the measurements of one of the different strain sensors placed in the hub, the comparison between that obtained in the original non-scaled system and those estimated for the scaled system show that the latter have an absolute value smaller magnitude, which is what we expected since the forces estimated for the scaled system are also lower in magnitude.
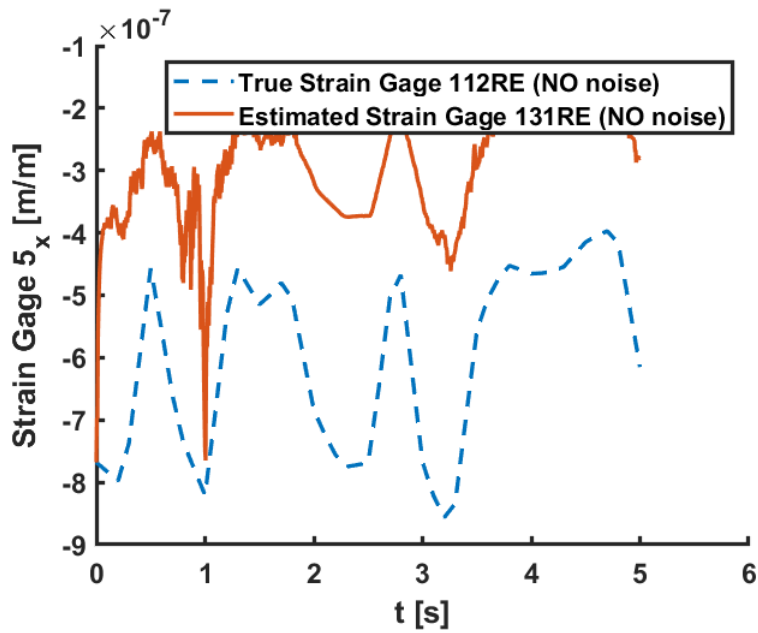
Figure 4.4: Estimated state of a strain gage measurement taken from the hub (case without noise)
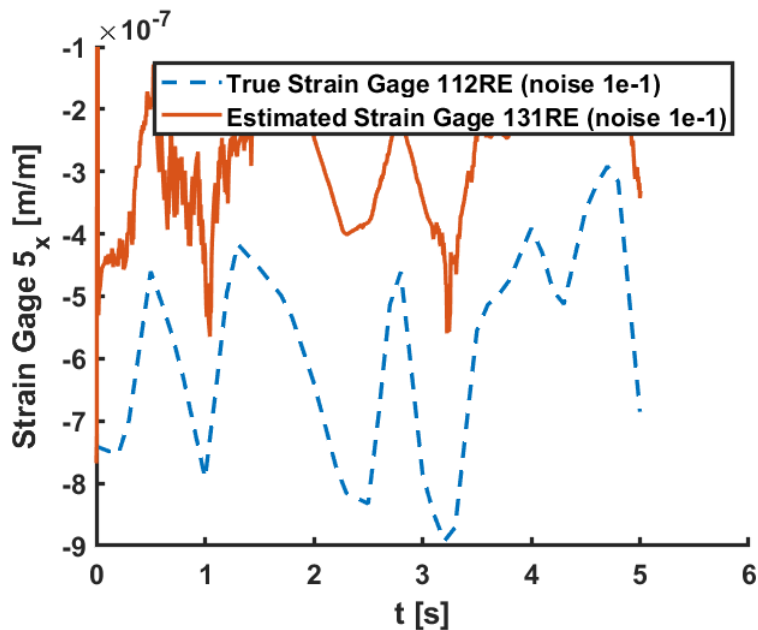


Figure 4.5: Estimated state of a strain gage measurement taken from the hub (case with noise)

In the following plots it is possible to observe the trend of the ratio between the dynamic load applied in the original test bench and that estimated in the scaled test bench, and as was expected, this ratio is highly non-linear, thus confirming that it is not possible to
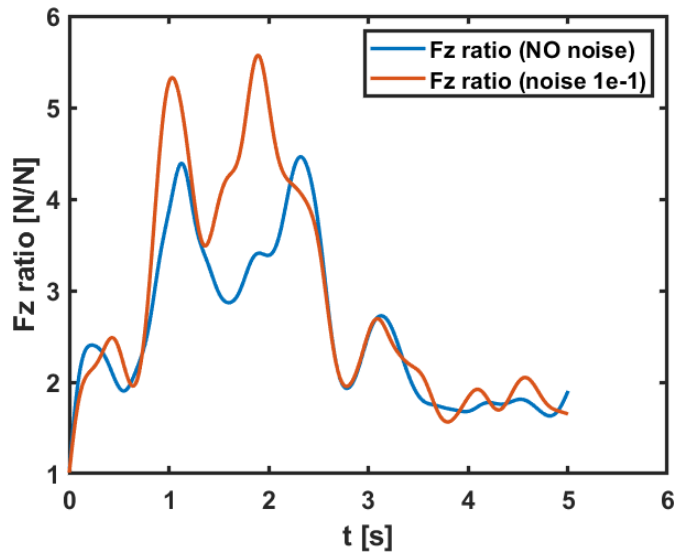
reduce the magnitude of the loads by the same percentage of reduction as the test bench and thus showcasing the need for an estimator as proposed in this thesis.
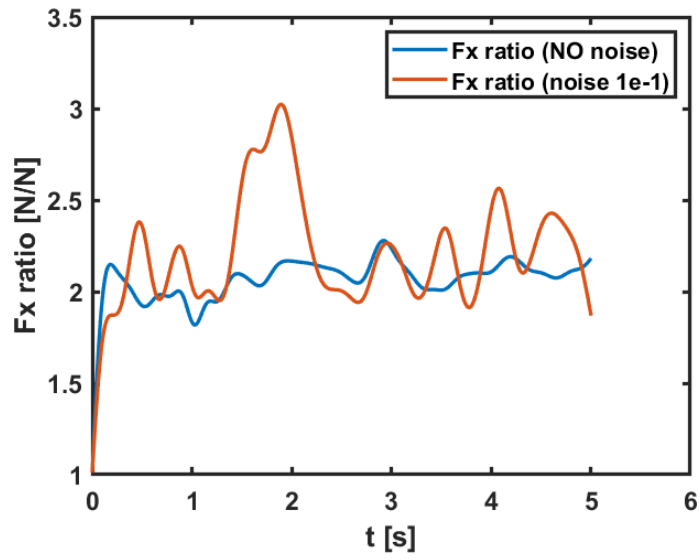
(a) Fx ratio.
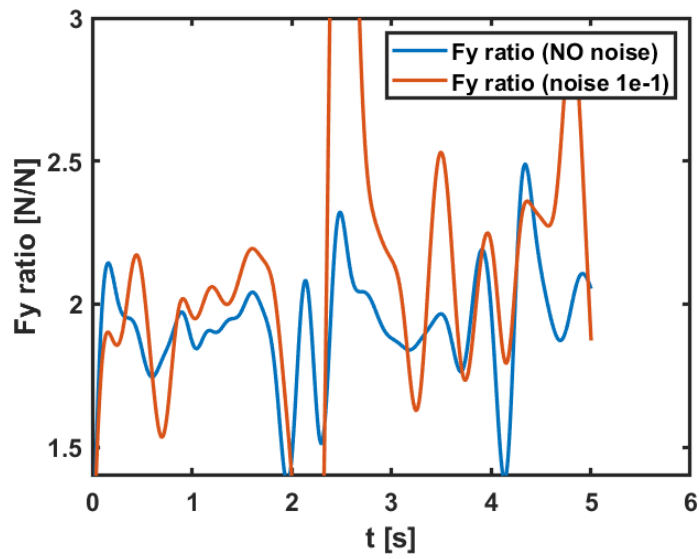


(b) Fy ratio.



(c) Fz ratio.

# Part II

# Optimal Sensor Placement

# 5 | Theoretical Background

An important aspect that needs to be taken into account is whether the measurements acquired on the real system contain enough information to retrieve the states of the system. This is related to the concept of observability and to the question of the sensors selection.

In this chapter we try to understand if there is a method that allows to position the sensors in an optimal way in a flexible statically constrained structure in such a way as to have the maximum observability of the system and therefore to obtain better estimations of the input and parameters.

The approach that will be used on the optimization of the position of the sensors uses a sorting algorithm developed by Cumbo et al. in which more details can be found in [6], [22] and [4], in which the Steady State Error Covariance, calculated by solving the Continuous Algebraic Riccati Equation (CARE) is used as a performance metric.

The system, that is a flexible body, is modeled using the finite element method, however, given the high number of degrees of freedom that such a model presents and the very high consequent computational cost, the application of a method of reducing the number of degrees of freedom of the system is necessary, so a parametric model order reduction will be applied as explained in the paper of Capalbo et al. [2].

## 5.1.   Parametric Model Order Reduction

In this application a Parametric Model Order Reduction method will be used, parametric in the sense that the dependence of the system on some structural parameters such as Young's modulus and density is retained.

From now on we will indicate the Parametric Model Order Reduction technique with "pMOR" and the Parametric Reduced Order Model with "pROM".

For linear parametric finite element models, the equations of motion can be written as:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{z}} + \boldsymbol{C}(\boldsymbol{\theta})\dot{\boldsymbol{z}} + \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{z} = \boldsymbol{S}\boldsymbol{u} \tag{5.1}$$

where: $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ represents the vector containing the parameters such as Young's modulus $E$ and density $\rho$, $n_p = number\ of\ parameters$, $\boldsymbol{z} \in \mathbb{R}^{n_{dof}}$ is the vector containing the nodal degrees of freedom of the FE model, $n_{dof} = number\ of\ nodal\ degrees\ of\ freedom$, $\boldsymbol{M} \in \mathbb{R}^{n_{dof} \times n_{dof}}$ is the mass matrix, $\boldsymbol{C} \in \mathbb{R}^{n_{dof} \times n_{dof}}$ is the damping matrix, $\boldsymbol{K} \in \mathbb{R}^{n_{dof} \times n_{dof}}$ is the stiffness matrix, $\boldsymbol{S} \in \mathbb{R}^{n_{dof} \times n_i}$ is the booelian input shape matrix, $n_i = number\ of\ inputs$, $\boldsymbol{u} \in \mathbb{R}^{n_i}$ is the input vector.

We also assume that the materials are isotropic.

The matrices just described can reach considerable dimensions due to the high number of degrees of freedom that characterize the finite element models, for this reason, as previously mentioned, it is necessary to reduce their dimensions by means of suitable approximation techniques, as explained above for the case of flexible multibody systems.

In this regard, the reduction method based on the vibration modes of a structure is shown.

That is, if we project the degrees of freedom vector $\boldsymbol{z}$ into a subspace of lower dimension using the global base $\boldsymbol{\Psi}$:

$$\boldsymbol{z} \approx \boldsymbol{\Psi}\boldsymbol{q} \tag{5.2}$$

then we can rewrite the system of equations of motion in the reduced version:

$$\boldsymbol{M}^r\ddot{\boldsymbol{q}} + \boldsymbol{C}^r\dot{\boldsymbol{q}} + \boldsymbol{K}^r\boldsymbol{q} = \boldsymbol{S}^r\boldsymbol{u} \tag{5.3}$$

where:

$$M^r = \Psi^T M \Psi \tag{5.4}$$

$$C^r = \Psi^T C \Psi \tag{5.5}$$

$$K^r = \Psi^T K \Psi \tag{5.6}$$

$$S^r = \Psi^T S \tag{5.7}$$

Note that the reduction base $\Psi$ is calculated using the Singular Values Decomposition (SVD) described in Capalbo et al. [2].

The size of $q$ is equal to the number of vibrating modes we are using in the modal reduction process, therefore we notice a considerable reduction in the number of degrees of freedom, which goes from hundreds of thousands for finite elements models, to the tens for reduced order models.

At this point, we can observe that the matrices of the system, and consequently also the reduced ones, show an affine dependence on the aforementioned parameters of the form:

$$X(\theta) = X_0 + \sum_i X_i f_i(\theta) \tag{5.8}$$

Where $X_0$ is a constant term, $X_i$ are the affine components and $f_i(\theta)$ are the affine functions.

In our specific case, considering the system as consisting of a single isotropic material, the affine relationship between the reduced stiffness matrix $K^r$ and the Young's modulus $E$ is given by:

$$K^r(E) = K_0 + K^E E \tag{5.9}$$

Recognizing in $K_0$ the constant term of the affine relation, in $K^E$ the affine component and in $E$ the affine function.

Similarly for the reduced mass matrix $M^r$, the affine relationship that binds it to the density $\rho$ is expressed by:

$$\boldsymbol{M}^r(\rho) = \boldsymbol{M}_0 + \boldsymbol{M}^\rho \rho \tag{5.10}$$

The method by which the affine components are calculated is beyond the scope of this thesis, but it can be deepened in Capalbo et al. [2].

At this point it is possible to represent the dynamic system in its explicit state form:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{cases} \tag{5.11}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{r-1}\boldsymbol{K}^r & -\boldsymbol{M}^{r-1}\boldsymbol{C}^r \end{bmatrix} \tag{5.12}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{r-1}\boldsymbol{S}^r \end{bmatrix} \tag{5.13}$$

From now on, the superscript $()^r$ will be omitted to make the formulation cleaner, meaning the arrays still in their reduced version.

## 5.2.    OSP - Training

An Optimal Sensor Placement strategy aims to select the optimal position of the sensors needed to solve an input and parameter estimation problem, however, this thesis focuses mainly on optimizing the positioning of strain sensors, called Strain Gauges, [6], [22], [4].

Before proceeding with the description of the method, an important assumption must be made, that is, in the case of estimating the input, the knowledge of the direction of the input and the point of application of the same, are known, in any case, in the event that it is not possible to know the direction of the load, it would be sufficient to estimate the three Cartesian components of the load and obtain the direction of the resulting load by means of a simple vector operation.

An outline of the first part of the method, the Training, is represented in the figure below.

The training aims to reduce the number of sensors while keeping a large enough number before applying the second step of the method, that is the Steady State Error Covariance metric.
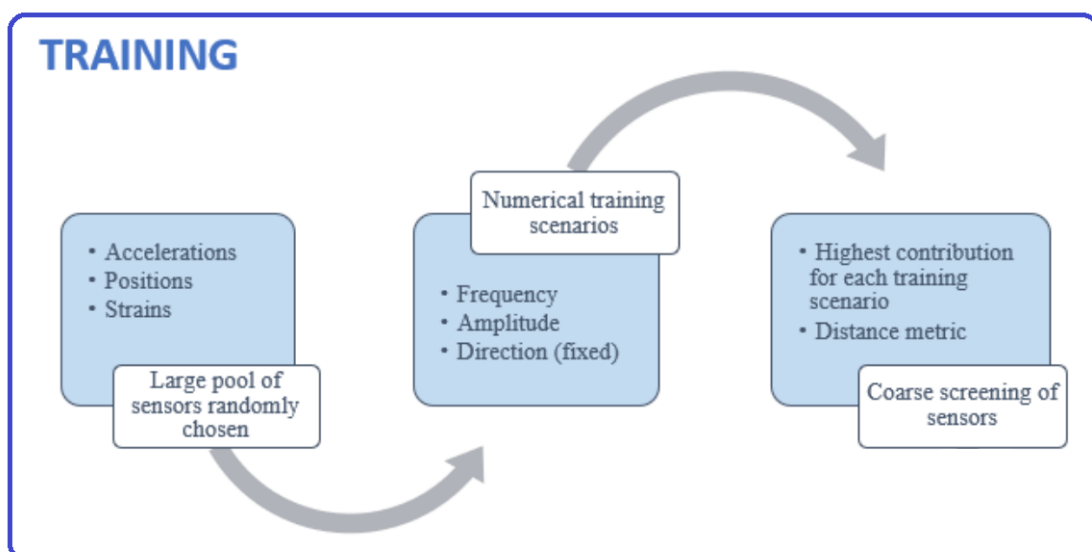


Figure 5.1: Training of the system for the first coarse screening, this figure is taken from [6]

So we start from the generation of a pool of potential sensors, in particular we select a set of elements in the finite element model.

Then the algorithm starts with the creation of the output matrix $\boldsymbol{H}$ by placing a Strain Gauge sensor in the outer surface of each element contained in the starting element set.

At this point we proceed with the Coarse Screening, which consists in simulating the system with a training input (usually sinusoidal), the measurements of the initial pool of sensors are collected and those with the lowest signal-to-noise ratio are removed, i.e. those less sensitive to training input.

A further removal of sensors takes place on the basis of the proximity between them, in particular, all those nearby sensors are removed that create clusters in the surface of the model.

In the end, a set of sensors that represent the basis on which the Steady State Error Covariance metric begins to work is obtained.

## 5.3.   OSP - Steady State Error Covariance Metric

Starting from the pool of sensors obtained after coarse screening, this algorithm is based on the solution of the Continuous Algebraic Riccati Equation (CARE) for different sensor configurations, [6]:

$$\boldsymbol{F}^*\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^{*T} + \boldsymbol{Q}^* - (\boldsymbol{P}\boldsymbol{H}^{*T}\boldsymbol{R}^{-1})\boldsymbol{R}(\boldsymbol{P}\boldsymbol{H}^{*T}\boldsymbol{R}^{-1})^T = 0 \tag{5.14}$$

Where $\boldsymbol{F}^*$ represents the augmented linearised state matrix, while $\boldsymbol{H}^*$ represents the augmented measurement matrix.

The term $\boldsymbol{Q}^*$ represents the plant covariance matrix expressing the uncertainty in the augmented model, that is, the uncertainty in the finite element model, and in the input and parameter dynamical model.

An important assumption is that $\boldsymbol{Q} << \boldsymbol{Q}_u, \boldsymbol{Q}_p$, this is because we assume that the finite element structural model is orders of magnitude more accurate than the zeroth-order hold model of the input and parameter.

So the following approximation of $\boldsymbol{Q}^*$ is proposed:

$$\boldsymbol{Q}^* = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_u & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_p \end{bmatrix} \tag{5.15}$$

$\boldsymbol{Q}_u$ and $\boldsymbol{Q}_p$ are calculated on the basis of the type of estimate we want to make, in fact they take on a different form if the goal is to estimate only the input, only the parameters or the set of inputs and parameters.

The term $\boldsymbol{Q}_u$ is set using the following rule of thumb, [2], [6]:

$$Q_{u_i} = (h\omega_{u_i}a_{u_i})^2 \tag{5.16}$$

where $\omega_{u_i}$ and $a_{u_i}$ represent a guess on the frequency and amplitude of the input, the index $i$ represents the $i$-th element that constitutes the vector of the inputs to be estimate.

This same rule is applied in the definition of $\boldsymbol{Q}_p$, [2]:

$$Q_{p_i} = (h\delta_{p_i}a_{p_i})^2 \tag{5.17}$$

where $\delta_{p_i}$ represents a scale factor that helps in the setup of the covariance of the parameters to be estimated by decoupling it from the absolute value of the parameter itself as well as from the time step of the integration scheme, while $a_{p_i}$ is the first attempt value of the parameter to be estimated.

Finally, $\boldsymbol{R}$ represents the covariance matrix of the measurements.

The formulations for the different cases of augmentation are shown below:

### 5.3.1.    State - Input Augmentation

Starting from the state representation of the linear dynamic system under analysis, augmenting the state vector $\boldsymbol{x}$ with the input $\boldsymbol{u}$ means creating the following state vector:

$$
\boldsymbol{x}^* = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \boldsymbol{u} \end{bmatrix}
\tag{5.18}
$$

Consequently the new system becomes:

$$
\begin{cases} \dot{\boldsymbol{x}}^* = \boldsymbol{F}^* \boldsymbol{x}^* \\ \boldsymbol{y} = \boldsymbol{H}^* \boldsymbol{x}^* \end{cases}
\tag{5.19}
$$

Where:

$$
\boldsymbol{F}^* = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} & \boldsymbol{M}^{-1}\boldsymbol{S} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}
\tag{5.20}
$$

$$
\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{D} \end{bmatrix}
\tag{5.21}
$$

### 5.3.2.    State - Young's Modulus Augmentation

In the case when the augmentation is done with the Young's modulus parameter, the dynamic system in state form is no longer linear, because inside the matrix $\boldsymbol{A}$ the stiffness matrix $\boldsymbol{K}$ is a function of Young's modulus, therefore consequently also the matrix $\boldsymbol{A}$, since in the augmented version, the parameter is seen by the Kalman filter as simply a new state, it means that $\boldsymbol{A}$ is dependent on the states (on the parameter) and therefore

it is no longer linear, that is $\boldsymbol{A}$ is no longer constant but varies over time.

The following system is still linear because it is not the augmented version:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}(E)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases} \tag{5.22}$$

with:

$$\boldsymbol{A}(E) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K}(E) & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \tag{5.23}$$

But if we augment the parameter $E$ with the states we get the following non-linear system:

$$\begin{cases} \dot{\boldsymbol{x}}^* = \boldsymbol{A}^*(\boldsymbol{x}^*)\boldsymbol{x}^* + \boldsymbol{B}^*\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{H}^*\boldsymbol{x}^* + \boldsymbol{D}\boldsymbol{u} \end{cases} \tag{5.24}$$

where:

$$\boldsymbol{x}^* = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ E \end{bmatrix} \tag{5.25}$$

$$\boldsymbol{A}^*(\boldsymbol{x}^*) = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{x}^*) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{5.26}$$

$$\boldsymbol{B}^* = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} \tag{5.27}$$

$$\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{0} \end{bmatrix} \tag{5.28}$$

So all that remains is to linearize the system with respect to an augmented reference state.

$$\boldsymbol{F}^* = \frac{\partial(\boldsymbol{A}^*\boldsymbol{x}^* + \boldsymbol{B}^*\boldsymbol{u})}{\partial\boldsymbol{x}^*} \tag{5.29}$$

$$\boldsymbol{F}^* = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} & -\boldsymbol{M}^{-1}\frac{\partial\boldsymbol{K}}{\partial E}\boldsymbol{q} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.30}$$

$$\frac{\partial\boldsymbol{K}}{\partial E} = \boldsymbol{K}^E \tag{5.31}$$

$\boldsymbol{F}^*$ and $\boldsymbol{H}^*$ are the matrices that will actually be used in solving (5.14).

### 5.3.3. State - Density Augmentation

Following a procedure similar to the previous case, we can derive the matrices that will be used to solve equation (5.14):

$$\boldsymbol{F}^* = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} & -\frac{\partial(\boldsymbol{M}^{-1})}{\partial\rho}(\boldsymbol{K}\boldsymbol{q} + \boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{S}\boldsymbol{u}) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{5.32}$$

$$\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{H} & \mathbf{0} \end{bmatrix} \tag{5.33}$$

where:

$$\frac{\partial(\boldsymbol{M}^{-1})}{\partial\rho} = -\frac{1}{\rho^2}(\boldsymbol{M}^\rho)^{-1} \tag{5.34}$$

It can be seen in the 5.34 that an explicit dependence on the density parameter $\rho$ appears, so some might wonder what value we have to assign to that parameter if we don't know it yet since it still has to be estimated.

Then an important observation comes out, since the final goal is to estimate the parameters (in our case $E$ and $\rho$), it therefore means that we do not know them a priori, otherwise we would not need to estimate them, or at least typically reference values are available from material libraries, but these typically differ (slightly) from the real properties, so the only thing we can do is to use values that we believe to be likely with the real ones, remember, are not yet known.

From now on, therefore, when the terms of $E$ and $\rho$ appear in the calculations of the matrices for solving the (5.14), it will be understood that they are the reference values.

Finally, note that even the affine components $\boldsymbol{K}^E$ and $\boldsymbol{M}^\rho$ are based on the use of reference parameters and not real ones.

So in the end, the entire Optimal Sensor Placement procedure is affected by how likely the reference parameters that are initially chosen are compared to the real ones and this can therefore be seen as a disadvantage of the procedure.

In the matrix 5.32, the terms $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $u$ appear in the last column, the method for their selection is based on a random choice of the index of the time vector of the training scenario simulation (i.e. the instance $t^*$), so we select $\boldsymbol{q}(t^*)$, $\dot{\boldsymbol{q}}(t^*)$ and $u(t^*)$.

For the other augmentation cases refer to the Appendix A.

### 5.3.4. OSP Algorithm

Then, once the case of interest has been defined and the appropriate $\boldsymbol{F}^*$ and $\boldsymbol{H}^*$ matrices have been selected based on the kind of augmentation is used, we proceed with the application of the OSP algorithm based on the solution of equation (5.14).

The pseudo algorithm is shown below:

---
**Algorithm 2** Optimal sensor placement - pseudo algorithm, [6]

---
1: Execute the iterative process to find the $\bar{n}_s$ optimal sensors:
2: **for** $k = 0 : (n_s^0 - \bar{n}_s)$ **do**
3:     given the measurement matrix $\boldsymbol{H}^{n_s^0 - k}$ evaluate CARE
4:     $\boldsymbol{P}_{uu}^k = care(\boldsymbol{H}^{n_s^0 - k})$
5:     Iterate on the subset of sensors $n_s^0 - k - 1$:
6:     **for** $g = 1 : (n_s^0 - k - 1)$ **do**
7:         remove sensor g from $\boldsymbol{H}^{n_s^0 - k}$ to obtain $\boldsymbol{H}^{n_s^0 - k - 1}$, with $g \notin (n_s^0 - k - 1)$
8:         $\boldsymbol{P}_{uu}^{k,g} = care(\boldsymbol{H}^{n_s^0 - k - 1})$
9:     **end for**
10:     Evaluate the sensor $\bar{g}$ which gives the lowest contirbution to the covariance matrix
11:     $\bar{g} = min(trace(\boldsymbol{P}_{uu}^k) - trace(\boldsymbol{P}_{uu}^{k,g}))$
12:     remove the sensor $\bar{g}$ from $\boldsymbol{H}^{n_s^0 - k}$ to obtain $\boldsymbol{H}^{n_s^0 - \bar{k}}$, with $\bar{k} = k + 1$
13: **end for**

---

# 6 | Optimal Sensor Placement - Numerical Validation

In this chapter, a numerical validation is carried out using a benchmark case, i.e. a cantilever beam model, and the subsequent numerical test is done using the hub of the test bench.

The procedure consists of applying the OSP algorithm for the type of estimate to be made, subsequently, once the optimal sensors have been obtained, a dynamic simulation of the system is performed by collecting the measurements of the optimal sensors previously obtained and finally the estimation of the parameter or input is performed using an Augmented Extended Kalman Filter.

# 6.1. Cantilever Beam Numerical Validation

### 6.1.1. OSP - Young's Modulus Estimation

The first OSP method we apply is the one related to the estimate of the Young's Modulus parameter, in particular the training load for the coarse screening that is applied is a sinusoidal input with a frequency of 20 Hz and an amplitude equal to 1e4 Newton along the global Z-direction, i.e. orthogonal to the cantilever beam, the signals generated in the training simulation are not affected by noise.

The number of sensors that is selected after the coarse screening is set to 100, it has been avoided to increase this number to avoid too long computational times.

The two figures below show the results of the algorithm, in particular the image on the left shows where the 4 final sensors are located in the surface of the beam, while the image on the right shows the trend of the Parameter Error Covariance (PEC) metric used in the OSP:
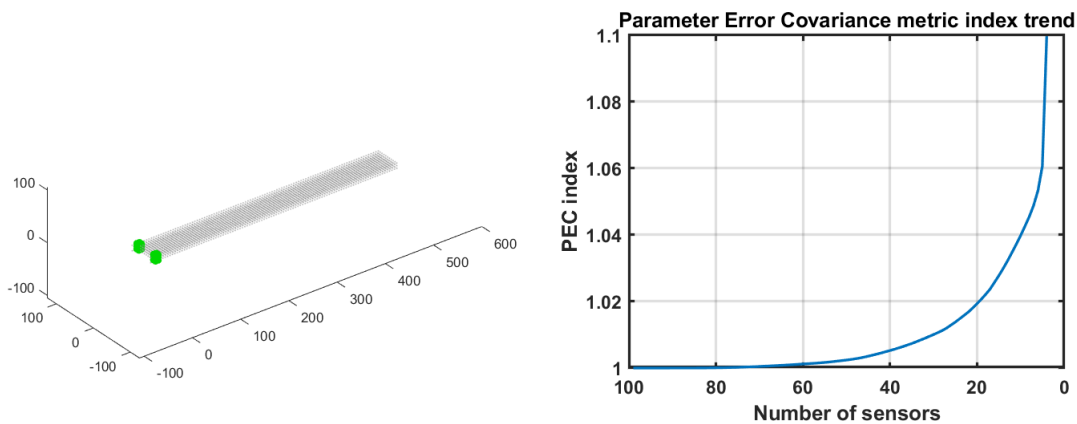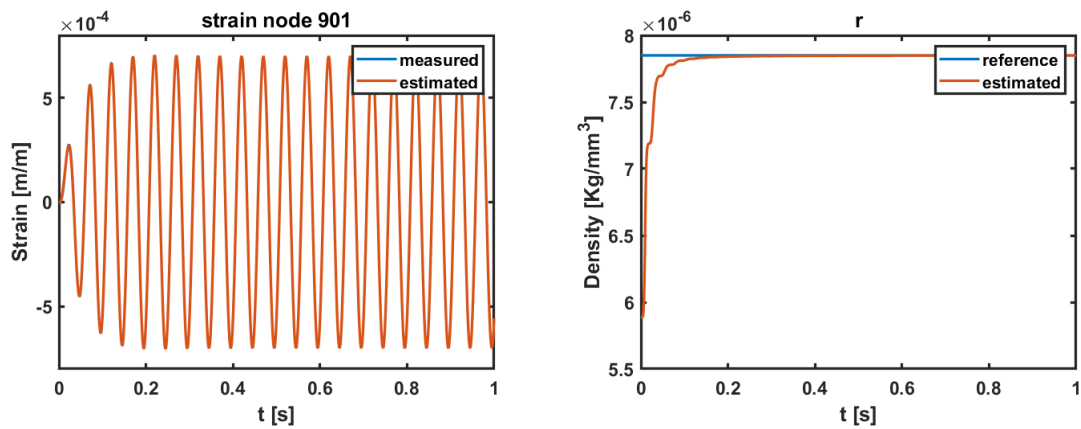


(a) Optimal sensors obtained with PEC metric for Young modulus estimation

(b) Trend of the PEC metric

Figure 6.1: OSP - Young's Modulus Estimation Result

It can be seen from the graph on the left that two clusters of two sensors each have been created, suggesting that some of them are redundant, in fact sensors close enough to each other will give the same information, obviously with the assumption that the strain field doesn't change too much throughout the structure/that region, this implies that the Young's modulus could be estimated using just two sensors.

At this point, the next step consists in simulating the dynamic system by applying an

input equal to that used in the training scenario of the OSP and the measurements coming from the four optimal sensors previously obtained with the OSP are collected.

Zero-mean, white Gaussian noise with a standard deviation that has been chosen based on the amplitude of the measurements is also added to the measurements itself to make them as realistic as possible.

Finally, the AEKF algorithm is used to estimate the strains in the elements of the OSP set and the Young's modulus parameter.

The following figures show the results of the estimation of the parameter, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the Young's modulus parameter.



(a) Estimation of one of the four sensors

(b) Young's Modulus estimation

Figure 6.2: Young's Modulus Estimation Result

As can be seen from the figures, the estimate of the sensor coincides almost perfectly with the real one, while the estimate of the parameter converges to the real value almost instantaneously.

In the following figure, an interesting comparison between the OSP set of sensors and many different random sensors selected from the coarse screening set is made for the Young's modulus estimation in order to demonstrate that the OSP set gives the most consistent and therefore most optimal solution:
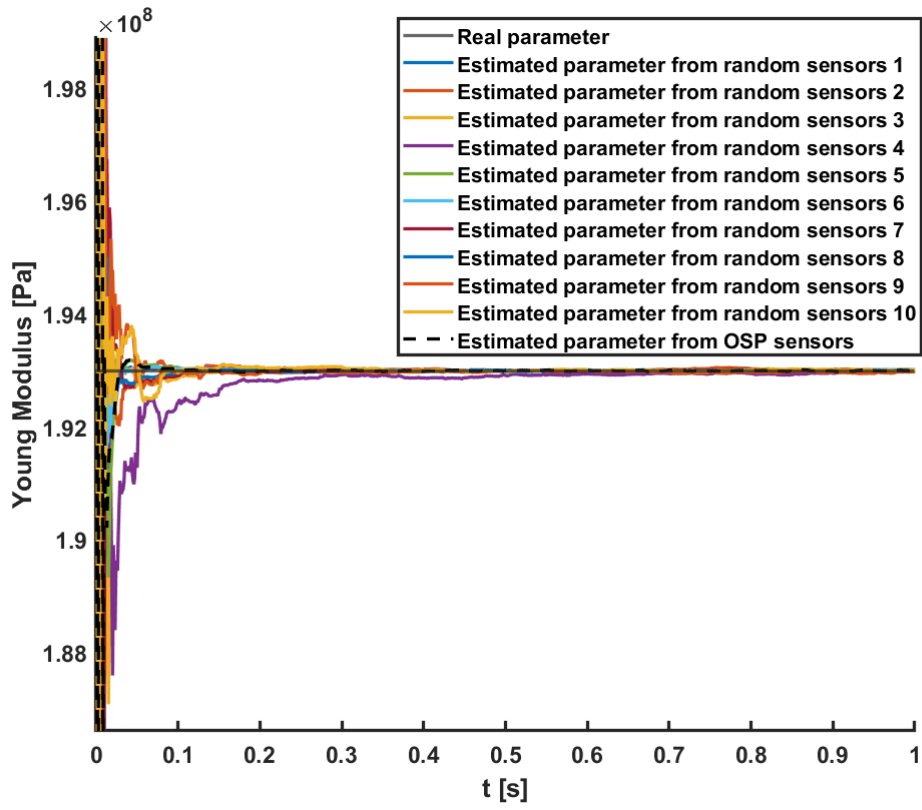
Figure 6.3: Comparison of the Young's modulus estimation between the OSP set of sensors and the 10 random set of sensors
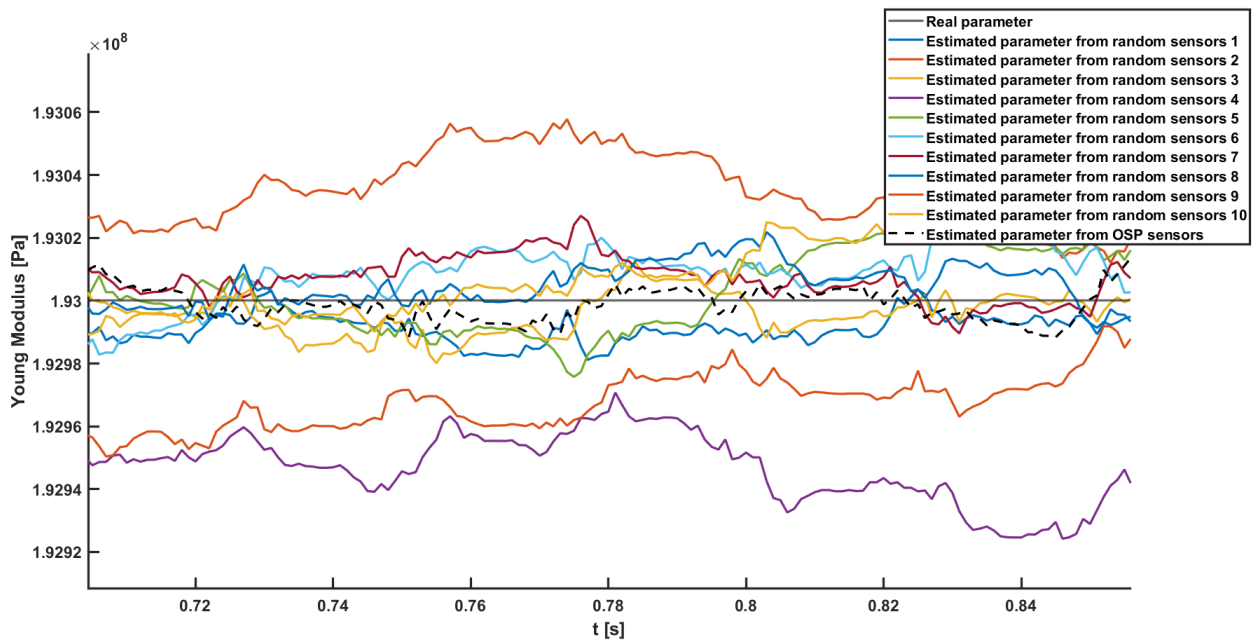


Figure 6.4: Zoom of the previous figure

Indeed, choosing random sets of sensors does not guarantee consistency and corresponding robustness, whereas the sets selected by the proposed OSP approaches perform in a very consistent and robust manner, this explains the reasons of why is it better to use an OSP algorithm.

### 6.1.2. OSP - Density Estimation

In this case, the OSP we apply is that relating to the estimate of the density parameter, in particular the training load that is applied is a sinusoidal input with a frequency equal to 20 Hz and an amplitude equal to 1e4 Newton along the global Z-direction, i.e. orthogonal to the cantilever beam, the signals generated in the training simulation are not affected by noise.

The number of sensors that is selected after the coarse screening is also in this case equal to 100.

The two figures below show the results of the algorithm, in particular the image on the left shows where the 4 final sensors are located in the surface of the beam, while the image on the right shows the trend of the Parameter Error Covariance (PEC) metric used in the OSP:



(a) Optimal sensors obtained with PEC metric for density estimation

(b) Trend of the PEC metric

Figure 6.5: OSP - Density Estimation Result

The following figures show the results of the estimate of the parameter, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the density parameter.

(a) Optimal sensors obtained with PEC metric for density estimation

(b) Trend of the PEC metric

Figure 6.6: OSP - Density Estimation Result

## 6.1.3.    OSP - Input Estimation

Once the parameters of interest have been estimated, we have a more accurate model of the system available, so it is now possible to estimate a generic input.

The following figures show the results of the OSP for the estimation of the input.



(a) Optimal sensors obtained with IEC metric for input estimation

(b) Trend of the IEC metric
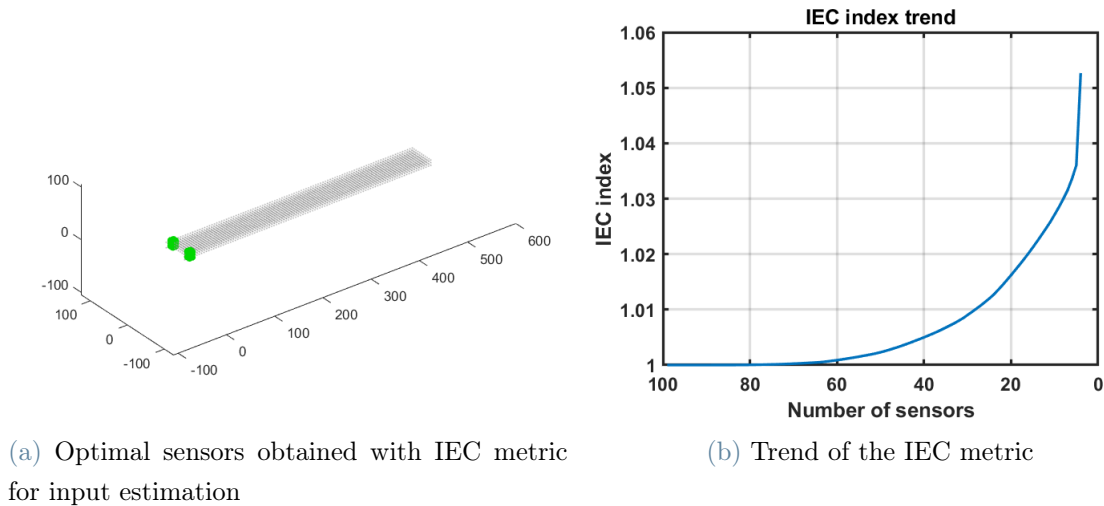
Figure 6.7: OSP - Input Estimation Result

The following figures show the results of the input estimate, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the input.



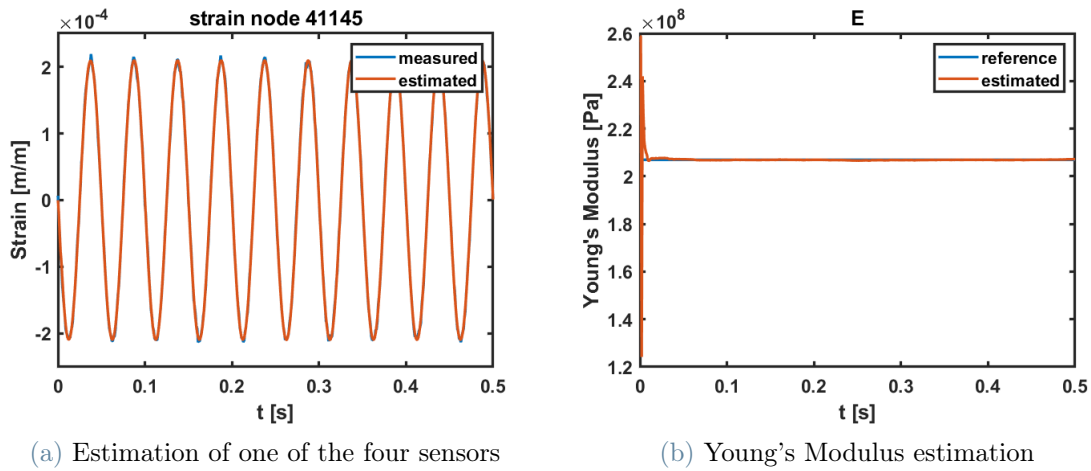(a) Optimal sensors obtained with IEC metric for input estimation

(b) Trend of the IEC metric
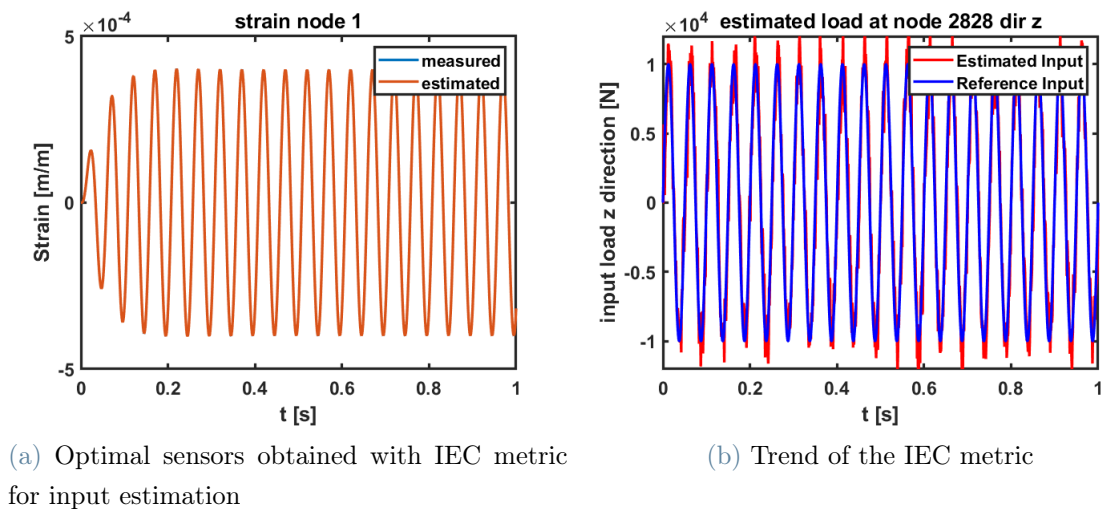
Figure 6.8: OSP - Input Estimation Result

## 6.2. Hub Test Bench Numerical Testing

The numerical test of the OSP is performed on the hub of the Test Bench, with the difference however that the load (the input) is applied directly in node ID 2 rather than in the blade, for more details on the numerical model consult chapter 2 of this thesis.

The OSP will be applied only to the following cases:

Table 6.1: Observable cases

OSP-E
OSP-rho
OSP-input

while, due to observability problems, it is not possible to find a set of optimal sensors that allows to obtain good estimates in the remaining cases:

Table 6.2: Non observable cases

OSP-E-rho
OSP-input-E
OSP-input-rho
OSP-input-E-rho

The demonstration of the non-observability of the system in the cases listed above will take place by using the Popov-Belevitch-Hautus (PBH) observability test [22], where in particular the following expression is used to define the metric for the observability of the system:

$$\boldsymbol{O}_m = cond(\mathbf{PBH}(0)) \tag{6.1}$$

which is the condition number of the **PBH** matrix evaluated at 0 Hz which is the frequency related to the random walk model.

The **PBH** matrix is defined according to Capalbo et al. [2].

It will be noted that in the unobservable cases of the table 6.2, this metric assumes values of several orders of greater magnitude with respect to the cases in which the system is observable, table 6.1, see figure 6.9.
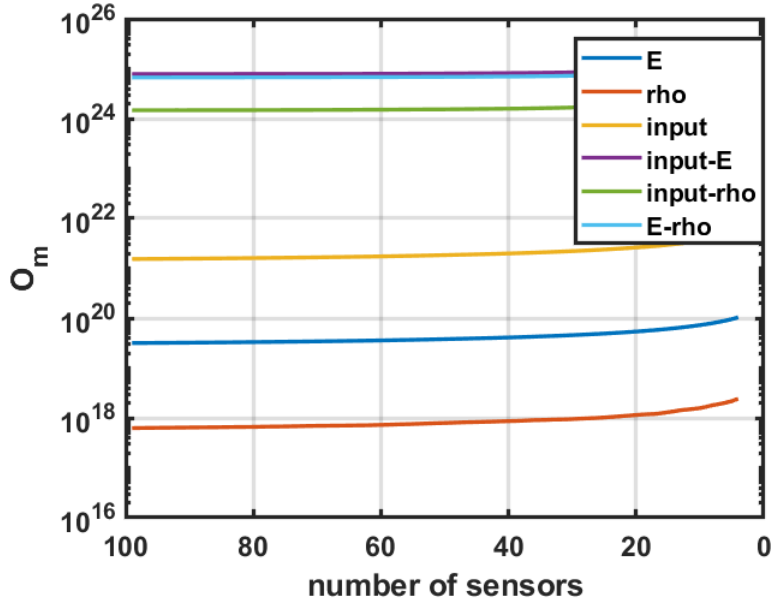
Figure 6.9: Comparison of the PBH between the different cases of augmented estimation

In the previous figure it can be observed that the cases in which it was possible to perform an acceptable estimate, correspond to a relatively lower PBH value compared to the cases in which the estimate did not give good results, thus confirming that the less observable system is, the more difficult the estimation is.

### 6.2.1.   OSP - Young's Modulus Estimation

The first OSP method we apply is the one related to the estimate of the Young's Modulus parameter, in particular the training load for the coarse screening that is applied is a sinusoidal input with a frequency of 20 Hz and an amplitude equal to 1e10 Newton along the global X-direction, the signals generated in the training simulation are not affected by noise.

The number of sensors that is selected after the coarse screening is set to 100, it has been avoided to increase this number to avoid that the algorithm arrives at the solution in too long computational times.

The two figures below show the results of the algorithm, in particular the image on the left shows where the 4 final sensors are located in the surface of the hub, while the image on the right shows the trend of the Parameter Error Covariance (PEC) metric used in the OSP:
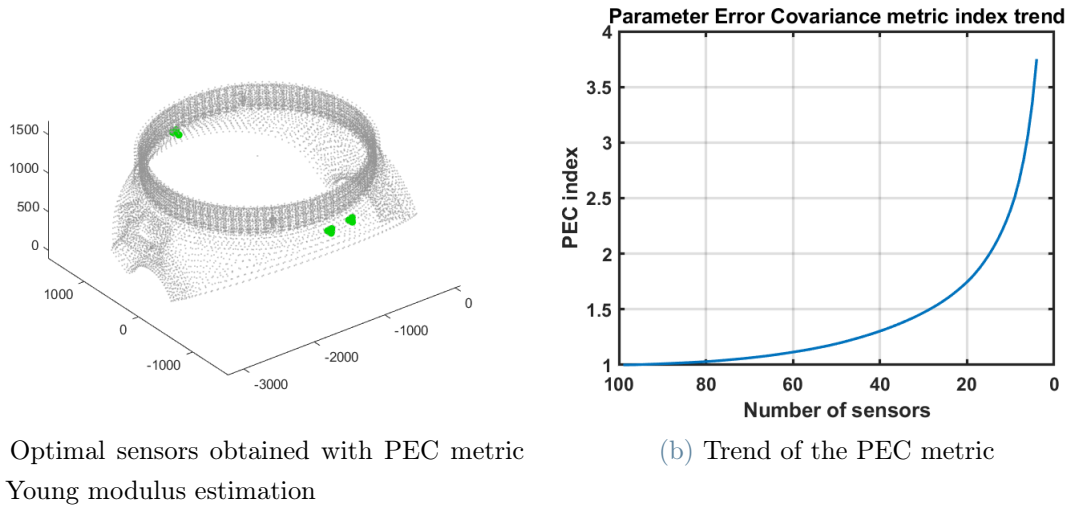
(a) Optimal sensors obtained with PEC metric for Young modulus estimation

(b) Trend of the PEC metric

Figure 6.10: OSP - Young's Modulus Estimation Result

It can be seen from the graph on the left that two clusters of two sensors each have been created, suggesting that some of them are redundant, in fact sensors close enough to each other will give the same information, obviously with the assumption that the strain field doesn't change too much throughout the structure/that region, this implies that the Young's modulus could be estimated using just two sensors.

Furthermore, it can be seen from the graph on the trend of the metric that in passing from the initial 100 sensors obtained from the first coarse screening to the final 4 sensors of the OSP, the metric quadruples.

At this point, the next step consists in simulating the dynamic system by applying an input equal to that used in the training scenario of the OSP and the measurements coming from the four optimal sensors previously obtained with the OSP are collected.

Zero-mean, white Gaussian noise with a standard deviation that has been chosen based on the amplitude of the measurements is also added to the measurements itself to make them as realistic as possible.

Finally, the AEKF algorithm is used to estimate the strains in the elements of the OSP set and the Young's modulus parameter.

The following figures show the results of the estimation of the parameter, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the Young's modulus parameter.

(a) Estimation of one of the four sensors



(b) Young's Modulus estimation

Figure 6.11: Young's Modulus Estimation Result

As can be seen from the figures, the estimate of the sensor coincides almost perfectly with the real one, while the estimate of the parameter converges to the real value almost instantaneously.

## 6.2.2. OSP - Density Estimation

In this case, the OSP we apply is that relating to the estimate of the density parameter, in particular the training load that is applied is a sinusoidal input with a frequency equal to 600 Hz and an amplitude equal to 1e10 Newton along the direction global x, the signals generated in the training simulation are not affected by noise, so we are in the deterministic environment.

Note that the reason why the frequency has increased in this case is due to the fact that, since density is a parameter associated with mass and therefore associated with inertia, excite this system, which we also specify to be very rigid, with a low frequency input with respect to the first eigenfrequency (which is 622 Hz), it would mean not completely exciting the dynamic response of the system, that is, it would simply mean exciting the static part which has nothing to do with density, thus giving incorrect results.

The number of sensors that is selected after the coarse screening is also in this case equal to 100.

The two figures below show the results of the algorithm, in particular the image on the left shows where the 4 final sensors are located in the surface of the hub, while the image on the right shows the trend of the Parameter Error Covariance (PEC) metric used in the OSP:
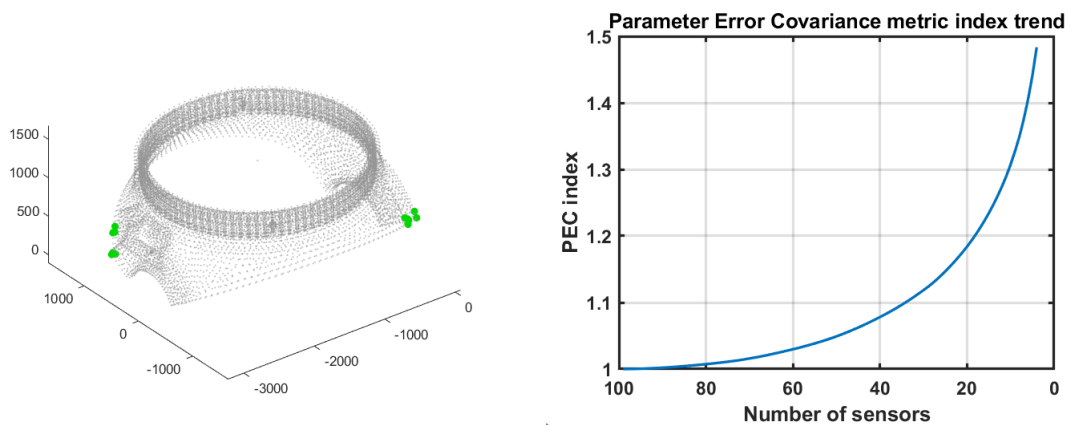


(a) Optimal sensors obtained with PEC metric for density estimation

(b) Trend of the PEC metric

Figure 6.12: OSP - Density Estimation Result

The following figures show the results of the estimate of the parameter, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the density parameter.
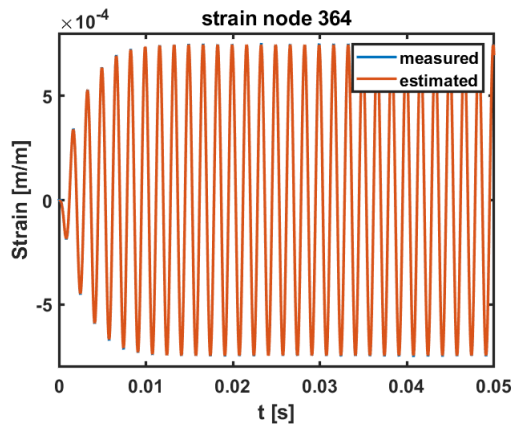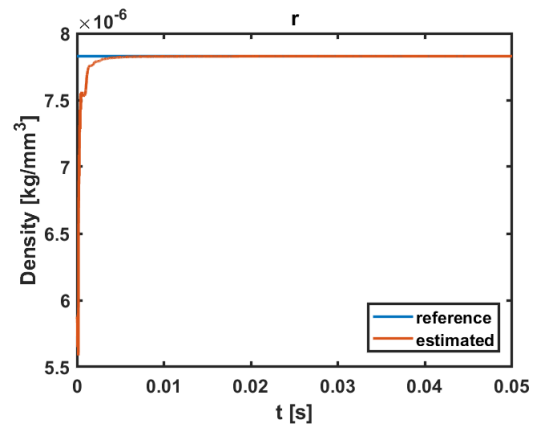
(a) Estimation of one of the four sensors       (b) Density estimation

Figure 6.13: Density Estimation Result

### 6.2.3. OSP - Input Estimation

Once the parameters of interest have been estimated, we have a more accurate model of the system available, so it is now possible to estimate a generic input.

The following figures show the results of the OSP for the estimation of the input.



(a) Optimal sensors obtained with IEC metric for input estimation

(b) Trend of the IEC metric
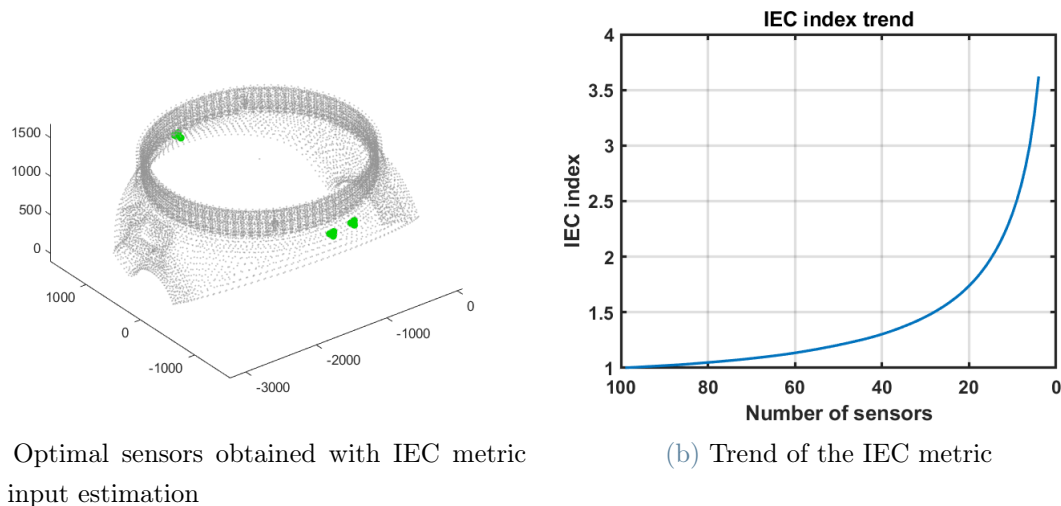
Figure 6.14: OSP - Input Estimation Result

Note that the estimated input is the same as the one estimated in chapter 2.

The following figures show the results of the input estimate, in particular on the left we observe the estimate of one of the four strain sensors while in the figure on the right we observe the estimate of the input.



(a) Estimation of one of the four sensors
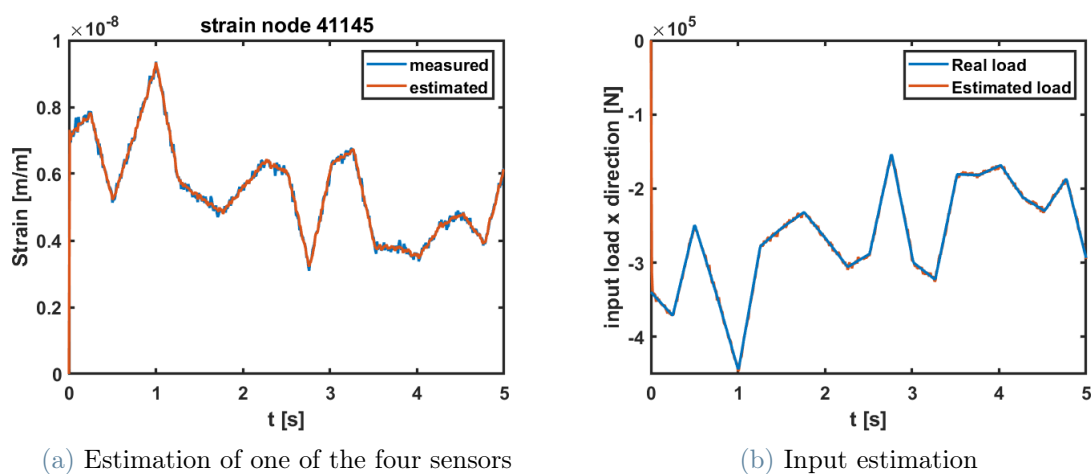
(b) Input estimation

Figure 6.15: Input Estimation Result

# 7 | Conclusions and Future Developments

The proposed Augmented Extended Kalman Filter (AEKF) based state-input-parameter estimator has been applied to estimate loads, strain fields, and parameters on a Wind Turbine Test Bench. The estimator effectively handles mechanical systems defined by implicit equations of motion, commonly found in flexible multibody models.

The use of a flexible multibody model representation allows for accurate reconstruction of time domain signals of external loads and strain fields, even with a minimal set of measured strains. This approach provides a promising alternative to direct measurement methods, which are often expensive and time-consuming.

The Kalman filter, particularly the AEKF, is a practical and widely used state estimator for systems with nonlinear dynamics like wind turbines. It accounts for uncertainties in measurements and models, making it suitable for the development of a virtual sensing strategy.

The estimation of joint state-input-parameters in wind turbine systems involves the selection of an optimal set of sensors. The Optimal Sensor Placement (OSP) strategy, based on the resolution of the Continuous Algebraic Riccati Equation (CARE), helps determine the best sensor configuration in terms of system observability and accurate load-parameter estimation.

This thesis highlights the importance of accurate physical models and the estimation of unknown parameters in wind turbine systems. Virtual sensing, which combines information from affordable sensors with system behavioral information from physical models, can provide cost-effective alternatives to direct measurements.

Based on the current state of the art, some potential future developments that can be explored are:

1. Further investigation and refinement of the AEKF-based state-input-parameter estimator for more complex mechanical systems and different applications beyond wind turbines

should be conducted. Its applicability in other fields, such as automotive, aerospace, or robotics, needs to be explored.

2. Advanced techniques for sensor placement optimization should be developed, considering additional factors such as measurement noise, system dynamics, and parameter identifiability. The use of machine learning algorithms or optimization methods should be investigated to enhance the accuracy and robustness of sensor selection.

3. The integration of other estimation techniques, such as Unscented Kalman Filters or Particle Filters, should be explored to compare their performance with the AEKF in terms of accuracy and computational efficiency. Their suitability for specific wind turbine applications or other mechanical systems should be evaluated.

4. The virtual sensing strategy should be extended to consider system-level analysis, including the estimation of higher-level parameters or performance indicators. The integration of data-driven modeling techniques, such as system identification or neural networks, should be explored to improve the fidelity of virtual prototypes.

5. Further experimental validations should be conducted on real wind turbine systems or other mechanical systems to assess the robustness and accuracy of the proposed state-input-parameter estimation method. The estimation results should be compared with direct measurements, and the practical implementation challenges should be evaluated.

# A | Appendix

## A.1. State - Input - Young's Modulus Augmentation

$$F^* = \begin{bmatrix} 0 & I & 0 & 0 \\ -M^{-1}K & -M^{-1}C & M^{-1}S & -M^{-1}\frac{\partial K}{\partial E}q \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.1}$$

$$H^* = \begin{bmatrix} H & D & 0 \end{bmatrix} \tag{A.2}$$

## A.2. State - Input - Density Augmentation

$$F^* = \begin{bmatrix} 0 & I & 0 & 0 \\ -M^{-1}K & -M^{-1}C & M^{-1}S & -\frac{\partial(M^{-1})}{\partial \rho}(Kq + C\dot{q} - Su) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.3}$$

$$H^* = \begin{bmatrix} H & D & 0 \end{bmatrix} \tag{A.4}$$

## A.3. State - Density - Young's Modulus Augmentation

$$F^* = \begin{bmatrix} 0 & I & 0 & 0 \\ -M^{-1}K & -M^{-1}C & -M^{-1}\frac{\partial K}{\partial E}q & -\frac{\partial(M^{-1})}{\partial \rho}(Kq + C\dot{q} - Su) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.5}$$

$$H^* = \begin{bmatrix} H & 0 & 0 \end{bmatrix} \tag{A.6}$$

## A.4.   State - Input - Density - Young's Modulus Augmentation

$$F^* = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ -M^{-1}K & -M^{-1}C & M^{-1}S & -M^{-1}(\frac{\partial K}{\partial E}q + C) & -\frac{\partial(M^{-1})}{\partial \rho}(Kq + C\dot{q} - Su) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tag{A.7}$$

$$H^* = \begin{bmatrix} H & D & 0 & 0 \end{bmatrix} \tag{A.8}$$

# Bibliography

[1] R. Adduci, M. Vermaut, F. Naets, J. Croes, and W. Desmet. A discrete-time extended kalman filter approach tailored for multibody models: State-input estimation. *Sensors*, 21(13):4495, 2021.

[2] C. E. Capalbo, D. De Gregoriis, T. Tamarozzi, H. Devriendt, F. Naets, G. Carbone, and D. Mundo. Parameter, input and state estimation for linear structural dynamics using parametric model order reduction and augmented kalman filtering. *Submitting*, 2022.

[3] R. R. Craig Jr and M. C. Bampton. Coupling of substructures for dynamic analyses. *AIAA journal*, 6(7):1313–1319, 1968.

[4] R. Cumbo, T. Tamarozzi, K. Janssens, and W. Desmet. Kalman-based load identification and full-field estimation analysis on industrial test case. *Mechanical Systems and Signal Processing*, 117:771–785, 2019.

[5] R. Cumbo, T. Tamarozzi, P. Jiranek, W. Desmet, and P. Masarati. State and force estimation on a rotating helicopter blade through a kalman-based approach. *Sensors*, 20(15):4196, 2020.

[6] R. Cumbo, L. Mazzanti, T. Tamarozzi, P. Jiranek, W. Desmet, and F. Naets. Advanced optimal sensor placement for kalman-based multiple-input estimation. *Mechanical Systems and Signal Processing*, 160:107830, 2021.

[7] C. W. Gear, B. Leimkuhler, and G. K. Gupta. Automatic integration of euler-lagrange equations with constraints. *Journal of Computational and Applied Mathematics*, 12:77–90, 1985.

[8] J. D. Halpin and A. N. Tran. An analytical model of four-point contact rolling element ball bearings. *Journal of Tribology*, 138(3):031404, 2016.

[9] B. J. Hamrock and W. J. Anderson. *Arched-Outer-Race Ball-Bearing Analysis Considering Centrifugal Forces*, volume 6765. National Aeronautics and Space Administration, 1972.

[10] A. Leblanc and D. Nelias. Ball motion and sliding friction in a four-contact-point ball bearing. 2007.

[11] P. Masarati. *Dynamics and Control of Flexible Aircraft*. Politecnico di Milano, 2020.

[12] P. Masarati. *Multibody System Dynamics*. Politecnico di Milano, 2020.

[13] S. Miller, T. Soares, Y. Van Weddingen, and J. Wendlandt. Modeling flexible bodies with simscape multibody software. *An Overview of Two Methods for Capturing the Effects of Small Elastic Deformations, MathWorks.–2017*, 2017.

[14] P. E. Nikravesh. *Computer-aided analysis of mechanical systems*. Prentice-Hall, Inc., 1988.

[15] P. E. Nikravesh. *Planar Multibody Dynamics: Formulation, Programming with MAT-LAB®, and Applications*. CRC Press, 2018.

[16] G. L. Plett. Dual and joint ekf for simultaneous soc and soh estimation. In *Proceedings of the 21st Electric Vehicle Symposium (EVS21), Monaco*, pages 1–12, 2005.

[17] E. Risaliti. Model based virtual sensors for wheel center loads and full strain field on vehicle suspension components. 2019.

[18] E. Risaliti, M. Vermaut, J. Croes, B. Cornelis, and W. Desmet. Force estimation on a mcpherson system by means of a state estimator and a multibody model. In *proc. of the ECCOMAS Thematic Conference on Multibody Dynamics*, 2017.

[19] E. Risaliti, T. Tamarozzi, M. Vermaut, B. Cornelis, and W. Desmet. Multibody model based estimation of multiple loads and strain field on a vehicle suspension system. *Mechanical Systems and Signal Processing*, 123:1–25, 2019.

[20] B. Simeon. Computational flexible multibody dynamics. In *A Differential-Algebraic Approach. Differential-Algebraic Equations Forum. Springer, Heidelberg*. Springer, 2013.

[21] D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[22] T. Tamarozzi, E. Risaliti, W. Rottiers, W. Desmet, et al. Noise, ill-conditioning and sensor placement analysis for force estimation through virtual sensing. In *In International Conference on Noise and Vibration Engineering (ISMA2016),*, pages 1741–1756. Katholieke Univ Leuven, Dept Werktuigkunde, 2016.

[23] Y. Yang, G. I. Webb, and X. Wu. Discretization methods. *Data mining and knowledge discovery handbook*, pages 101–116, 2010.

# List of Figures

# List of Tables