SENSING NONLINEAR VISCOELASTIC CONSTITUTIVE PARAMETERS WITH A GEOMETRICALLY NONLINEAR BEAM: MODELING AND SIMULATION

Yanzhang Wu

Thesis submitted to the University of Ottawa

in partial Fulfillment of the requirements for the

MASTER OF APPLIED SCIENCE

in Mechanical Engineering

Department of Mechanical Engineering

Faculty of Engineering

University of Ottawa

Abstract

In this thesis, we present a sensor model comprised of a geometrically nonlinear beam coupled with a nonlinear viscoelastic Pasternak foundation via a distributed system of compliant elements. The governing equations of the system are obtained. By posing an inverse problem, the model is used to simulate the estimation of coupled substrates' material (constitutive) parameters. In the inverse problem, beam deformations are considered as measured parameters, and therefore an eventual hardware implementation would require measurements of these quantities. Different case studies are simulated to assess the robustness and applicability of this sensor model.

Nomenclature

ξ_1	horizontal coordinate of the undeformed beam
ξ_2	vertical coordinate of the undeformed beam
θ_0	rotation of the undeformed beam
u_1	horizontal Cartesian displacement
u_2	vertical Cartesian displacement
θ	rotation of the deformed cross section
χ	rotation of the cross section with respect to the neutral axis
Λ	relative stretching in the reference configuration
N	normal force
Q	shear force
M	bending moment
p_1	distributed normal force, Cartesian component
p_2	distributed shear force, Cartesian component
μ	distributed bending moment
ε	normal strain
γ	shear strain
κ	curvature
E	Young's modulus

G	shear elastic modulus
A	cross sectional area
I	moment of inertia
h	cross section thickness
p_η	reaction force induced by the substrate
η_0	initial profile of the substrate
η	displacement of the substrate
k_1	linear elastic parameter of the substrate
k_2	nonlinear elastic parameter of the substrate
k_3	shear coefficient of the substrate
k_c	viscosity of the substrate
$p_{u\eta}$	force transmitted by the coupling layer
g	current gap between the beam and the substrate
g_0	initial gap between the beam and the substrate
$\mu_{u\eta}$	distributed couple induced by $p_{u\eta}$

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor, Dr. Davide Spinello. He has displayed an enormous amount of patience in answering my questions. He trains me to think more, explore more and learn more. His help and inspiration s always encourage me to overcome difficulties and make this work possible. I appreciate his time and effort in reading and reviewing my thesis. His words have, and always will be, of great value to me.

I also would like to thank my committee members Dr. Michel Labrosse and Dr. Robin Chhabra who review and leave valuable comments.

I express my gratitude to Qiaoling Wu and Dr. Jin Yu for helping me and offering kind guidance of my life whenever I meet them. They are the reason why I come to Canada and start my graduate study.

I would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) for providing me financial support.

I also express my thanks to Faculty of Engineering, University of Ottawa, for supporting my research.

Last but not least, I would like to offer my greatest appreciation to my parents, Wende Wu and Jingxiu Wang. When I'm feeling down, they can comfort me like no one else. Without their unconditional endless support and encouragement, I cannot finish my graduate study.

CHAPTER 1
INTRODUCTION

1.1 Overview

Sensing technology can be broadly defined as to use sensors to detect quantifiable properties from measurable objects and to convert these properties into data that humans or machines can interpret. This thesis focuses on the modelling and simulation of a mechanical sensing system aimed at estimating material properties or, more generally, constitutive parameters of deformable solid materials. The sensing device is schematized as a deformable solid, eventually a beam, coupled with a substrate/terrain for which the material properties have to be estimated, see Fig.1.1. The context is that an autonomous robotic device deployed in some unstructured environment with the mission of exploring, navigating and, eventually, performing a set of tasks defining the mission.

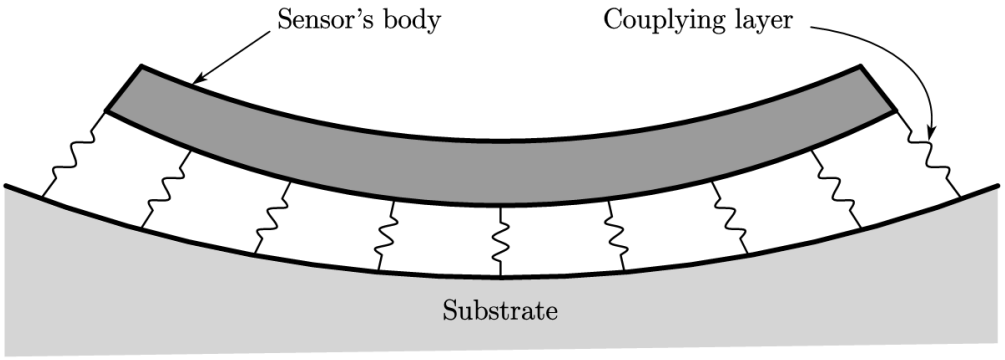


Fig. 1.1. Sensing system modelled in this thesis

A prototype of an autonomous mobile robot whose specific mechanical structures are designed for locomotion is being developed at the University of Ottawa, see Fig. 1.2. We are interested in expanding the functionalities of this mobile robot by using the existing mechanical structure without requiring extra components. Specifically, we want this mobile robot to also act as a sensor to detect the properties (material parameters) of the substrate. As the sketch in Fig. 1.1, the sensing system can be abstracted as a soft deformable beam (body of the mobile robot) interacting with the substrate through a coupling layer (legs of the mobile robot).

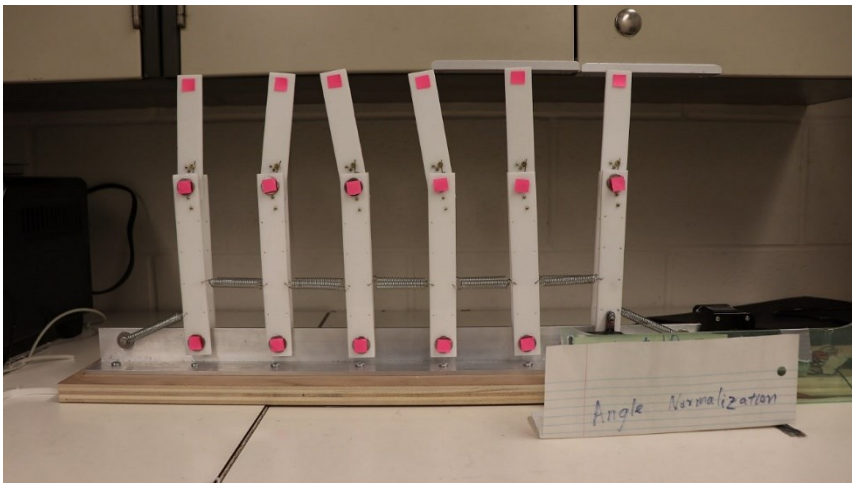


Fig. 1.2. An autonomous robot prototype at uOttawa in the same research group

This idea has been explored in [1], where the model of a mechanical sensing device has been proposed and simulated for the case in which the unknown solid terrain is nearly flat with respect to the characteristic length of the sensing device. In this case, the mechanical model of the sensing device can be based on a linearized beam theory in which large curvatures induced

by the coupling elements are neglected. However, in many applications of interest, as for example in the biomedical field, when estimating material parameters of soft tissues, it is crucial to be able to operate on large deflection and curvatures of the sensing device. This work is therefore focused on generalizing the model in [1] by including nonlinear geometric effects, which allows to improve the predictions of the sensor in soft substrate conditions.

The sensing system modeled in this thesis consists of a geometrically nonlinear beam based on Reissner model [2], a deformable substrate (terrain) with the inclusion of nonlinear and viscous coefficients modeled as a nonlinear Pasternak foundation, and a layer of coupling elements between the beam and the substrate, as schematized in Fig.1.1. The modle of the substrate will be discussed in Section 2.3. Material parameters of the terrain, which consist of linear stiffness, strain hardening stiffness, shear stiffness and viscosity are estimated by posing an inverse problem in which the inputs to the system are displacements and/or rotations along the axis of the beam, which induce a response of the substrate through the forces and/or moments exerted by the coupling layer. The response of the substrate, in terms of forces, is constitutively related to the deformation of the substrate, and the unknown material parameters are estimated through an iterative procedure that is based on the minimum least square minimization of a quadratic form of the distance between measured and estimated kinematics.

1.2 Background and Motivation

The sketch of the model in [1] (previous work) and its workflow are illustrated in Fig.1.3. The beam initially lays on the substrate through the coupling layer which is considered to be equivalent to a distribution of springs. When forces are applied, the beam bends. The beam's displacements which also account for response of reaction forces induced by the foundation are measured. Noting that reaction forces are ultimately related to the substrate's parameters, one can identify those parameters by solving the inverse problem (details in Section 2.5).

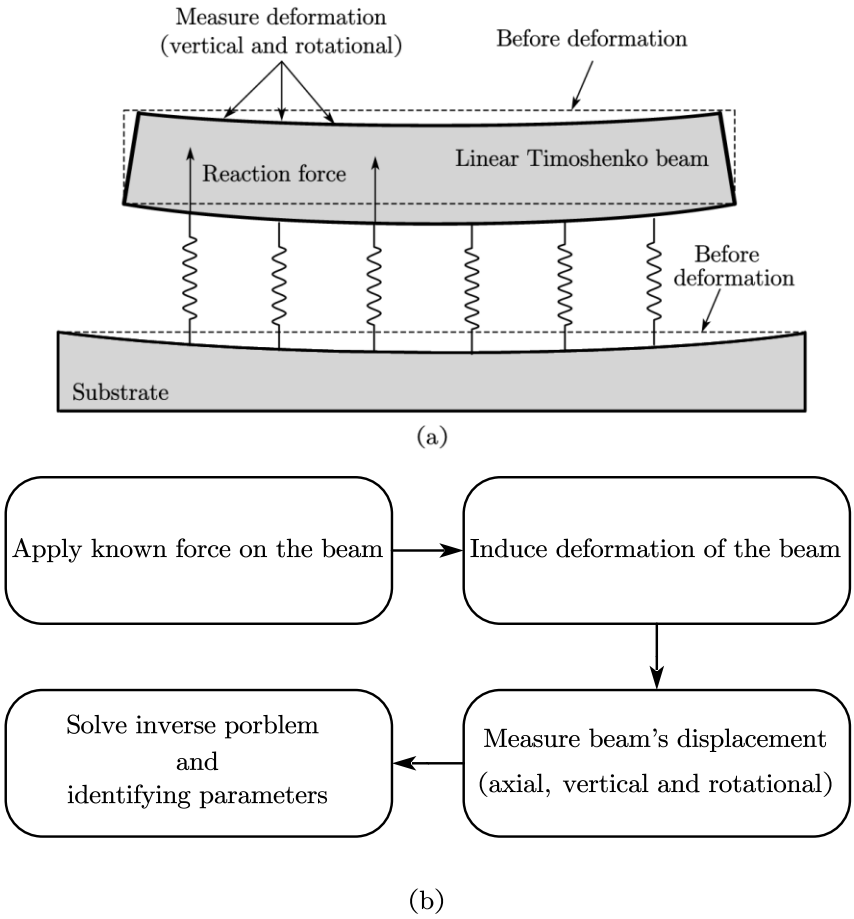


Fig. 1.3. (a) Sketch of the model in [1]. (b) The workflow

It is worth stressing again that the sensor model in [1] is based on a linear Timoshenko beam; therefore, the model is suitable for relatively stiff substrates and the beam model cannot predict and describe large deformation induced by large external forces or soft terrain coupling.

1.3 Aim and Objective

In the biomedical field, soft tissues are common materials, which typically undergo large deformations as part of their ordinary operation. In this case, using the simple linear beam theory may lead to large errors when the deflections are supposed to be large compared to the length of the beam (typically greater than 3%) [3]. In order to be able to extend the applicability of the sensor model to soft tissue, it is necessary to include nonlinear geometric effects in the kinematics of the mechanical sensor. This defines the main goal of the present work, which extends the work in [1] to generalize the sensor model in the sense just specified.

1.4 Potential Applications

As mentioned in Section 1.2, this work focus on the sensing capability equips on the mobile robot, which means the idealized model has the features of locomotion and sensing. Therefore, our sensor could be used to do environmental monitoring jobs like detecting the parameters of terrains. Moreover, with the locomotion of the mobile robot, our sensor is suitable for complicated and dangerous environment where engineers cannot finalize the

fieldwork, since sensor operators can remotely control the robot or the autonomous robot could be programmed to accomplish tasks automatically.

Despite of the monitoring environments, miniaturization of the robotic sensing system allows for healthy diagnosis related applications. For example, detecting parameters of skins and muscles by deploying the model on bodies. To go further, if the robot could be integrated as small as a capsule, testers can diagnose tissues of organs like stomach by swallowing it.

The mechanical structure of the sensor system in this work has distinct advantages in estimating soft material parameters, since traditional tests for soft tissues have some drawbacks. For example, it is not easy to cut a specimen to a particular shape, and cutting forces cause significant distortion. Besides, the process of cutting materials often significantly alters specimens' properties and causes trauma. Moreover, soft tissues are sensitive to the environment. Properties may change when they are in transport, in preparation, in storage and during the actual testing. Testing personnel may also be under radiation risk if touchless techniques are used. Compared to traditional ways of testing, our work overcomes the drawbacks above, and it is a safe and accurate model.

1.5 Contributions

The main contribution of this thesis is the formulation of a mechanical sensor model based on a geometrically nonlinear beam theory for the purpose of estimating material

parameters of coupled deformable solids. This work generalized the one in [1], addressing the technical difficulties associated with the inclusion of geometrically nonlinear effects. By generalizing the model in this way, we are able to extend the functionality of the sensing device by using the system on a substrate which may undergo large deformations. These features make the sensor system in this thesis suitable for estimating constitutive parameters of soft tissues in the biomedical field.

1.6 Research Methodology

This work includes four steps to build the theoretical model of the sensing system:

- Step 1: Formulating beam equations and substrate equations.
- Step 2: Modeling the interaction between the beam and the substrate.
- Step 3: Solving forward problems to generate simulated measurement data.
- Step 4: Solving the inverse problem to estimate the substrate's parameters by using the simulated measurement data in step 3.

The rest of thesis is organized as follows. In Chapter 2, the theoretical background of the sensing system is reviewed. In Chapter 3, the mechanical model of the sensing system is presented. In Chapter 4, we introduce the algorithm of the inverse problem, and by considering different cases, numerical simulations of sensor operation are conducted. In Chapter 5, conclusions are presented.

CHAPTER 2
THEORETICAL BACKGROUND AND LITERATURE REVIEW

The sensor system in this thesis is abstracted as a geometrically nonlinear beam resting on a foundation via coupling elements (Fig.1.1), and our goal is to estimate substrate's parameters from a given input. This chapter presents an overview of theoretical preliminaries of key elements in the sensor system. Also, the literature about inverse problem is reviewed.

2.1 Beam theories

Tracing back the long history of beam theories, Galileo Galilei is often recognized as the first who addressed the beam bending problem, but recent studies argues that Leonardo da Vinci preceded Galileo's work by more than 100 years [4]. Daniel Bernoulli and Leonhard Euler formulated what is modernly recognized as the linear beam theory, by structuring it into a rigorous mathematical framework that allows one to calculate deformations and stress resultant distributions. The Euler-Bernoulli beam theory is considered to be the backbone of beam theories and is still widely used in structural and mechanical engineering applications today.

In the early 1920s, Stepan Prokopovych Timoshenko proposed a beam theory developed

from the Euler-Benoulli theory [5]. In his model, lateral shear deformation and cross section moment of inertia are considered, which means the shear deformation caused by the transverse shear force in the beam is constitutively related to the deflection of the beam, and the cross section still remains plane but is no longer necessarily perpendicular to the longitudinal axis, see Fig. 2.1 and Table 2.1. Through the Timoshenko beam theory, the behaviour of relatively thick and short beams, sandwich composite beams, or beams subject to high-frequency response can be modelled with higher accuracy than with Euler's beam [6]. For this reason, this theory has been used in aerospace structural design.

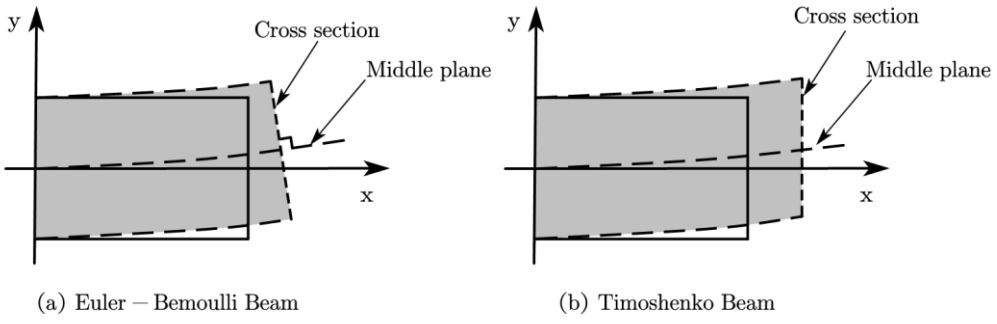


Fig. 2.1. Deformation of (a) Euler-Bernoulli Beam and (b) Timoshenko Beam

Table 2.1. Euler-Bernoulli beam theory and Timoshenko beam theory

Beam models	Bending moment	Lateral displacement	Shear deformation	Rotary inertia
Euler-Bernoulli	√	√	×	×
Timoshenko	√	√	√	√

It is worth noting that both the Euler-Benoulli beam theory and the Timoshenko beam

theory are limited to small deflections as they are linearized theories. To model nonlinear deformable systems, geometrically exact beam theories were proposed in the mid nineteenth century and developed in the twentieth century. In 1859, Kirchhoff derived a beam model that allows large deformations and initial curvatures in three dimensions [7]. Almost one-hundred years later, Love [8] extended Kirchhoff's beam theory by accounting for axial extension. In 1972, Reissner [2] proposed a two-dimensional shear deformable beam based on Kirchhoff-Love beam model and proposed his theory in [9] for the general three-dimensional case. Although the simplification of rotation matrix (neglecting second order rotations) in [9] is helpful to derive rotation strain measures, it decreased the geometric exactness of the theory. Inspired by Antman [10], Simo [11] extended Reissner's model by adopting a different approach. In [11], a spin rotation vector and the rotation tensor are employed to derive the beam's displacement. This beam theory, well known as Simo-Reissner theory is widely implemented nowadays to analyze structural behaviors like airplane wings, high-rise buildings and large span bridges.

Science this work aims at developing sensing of soft tissues, geometric nonlinearities should be included in the model (details in chapter 2.2). In future hardware implementations, the mobile robot should be relatively small for the purpose of easy operation. Since the sensing device body could be relatively thick as compared to its length, it is appropriate to use a beam theory that accounts for shear deformability. Under the hypothesis that the interaction between the sensor and the substrate mostly develops along a bending direction of the sensor's body,

we adopt a two dimensional kinematics, see Fig. 2.2.

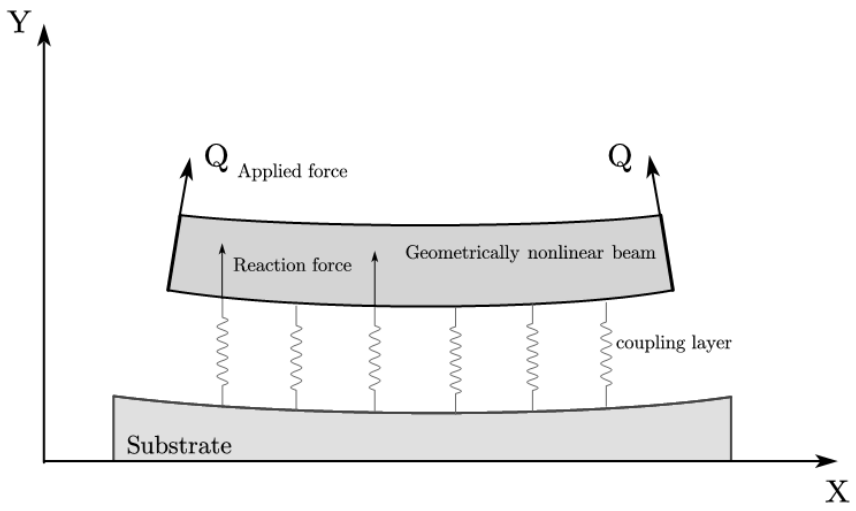


Fig. 2.2. Sensing model generalized in a 2D coordinate system

Therefore, the Reissner beam model in [2] is adopted since it allows to describe a system with all the features just discussed.

2.2 Geometric Nonlinearity

In engineering fields, many structural analyses are based on the assumption of small deformations, because the corresponding linear beam theory is a good tradeoff between mathematical manageability and range of applicability and, for many civil engineering applications, the hypothesis of small deformations is accurate. For example, when analyzing concrete structures, it is typically not necessary to use nonlinear theory if the beams spans are sufficiently large compared to their cross sections, but small enough to limit maximum

displacements with respect to the size of the cross section. However, in recent decades, with the development of computing hardware and sophisticated computational tools, the adoption of nonlinear beam theories has become more prominent. For example, in biomechanical engineering, structural elements may operate in large deflections regime. Geometric nonlinearities occur when strain-displacement relations need to include quadratic terms to accurately predict large deformations [12].

Reissner beam theory will be presented in detail in the next chapter, in the context of the model for the mechanical sensor. Here, we briefly consider the kinematics to illustrate the effect of geometric nonlinearities, by drawing a comparison with the predictions of the linearized version, the Timoshenko beam. The kinematics of Reissner's beam is schematized in Fig. 2.3.

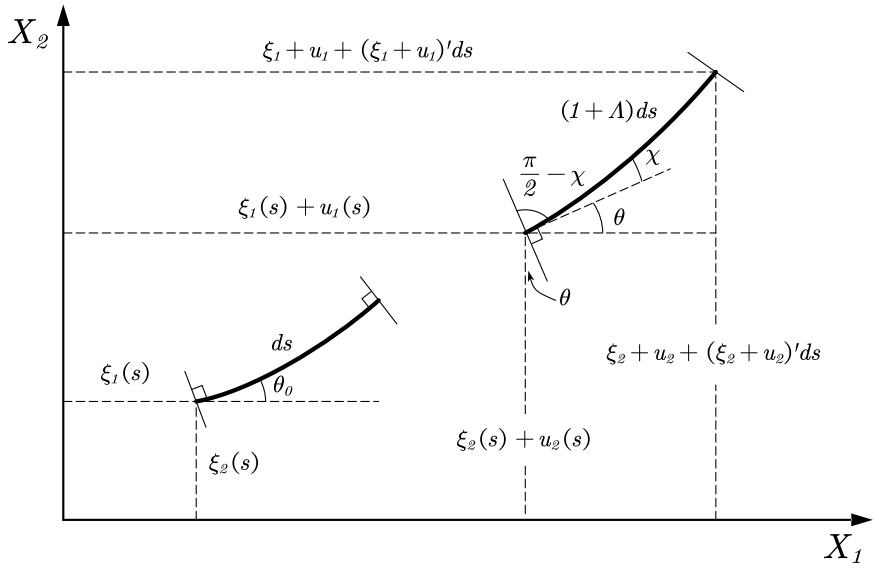


Fig. 2.3. Kinematics of Reissner's beam

Assume an initially flat Reissner beam undergoing bending. Axial and shear strains ε and γ (details in next chapter) can be written as

$$\begin{aligned} \varepsilon &= (u_1' + 1)\cos\theta + u_2'\sin\theta - 1 \\ \gamma &= u_2'\cos\theta - (u_1' + 1)\sin\theta \end{aligned} \tag{2.1}$$

where θ is the rotation angle of the cross section, and u_1, u_2 are the displacements of the neutral axis. For small deformations, we have $\sin\theta \approx \theta$ and $1 - \cos\theta \approx 0$. The linearized version of Equation (2.1) reduces to

$$\tilde{\varepsilon} = u_1' \qquad \tilde{\gamma} = u_2' - \theta \tag{2.2}$$

Equation (2.2) gives the strain-displacement relations of the Timoshenko beam. By assuming $u_1' = 0.5$ and $u_2' = 1$, we plot nonlinear strains ε, γ and linear strains $\tilde{\varepsilon}$, $\tilde{\gamma}$ with respect to rotation angle θ.

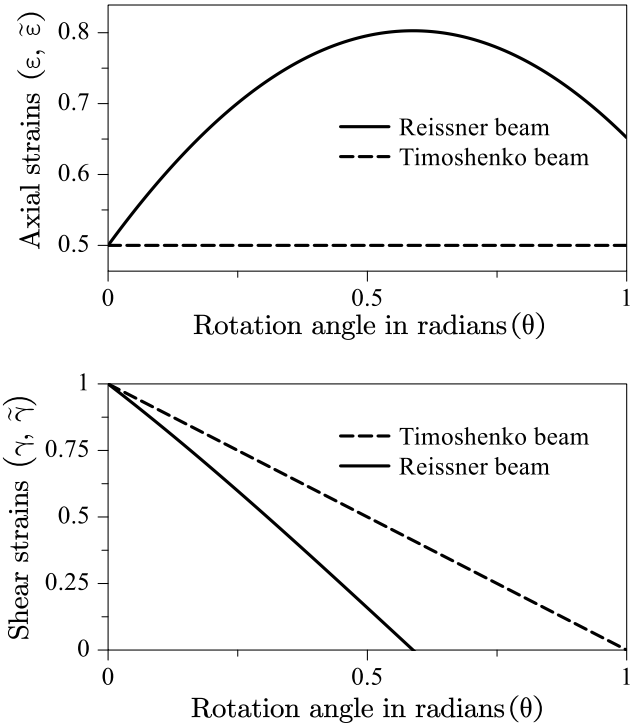


Fig. 2.4. Strains with respect to rotation θ

For the given u_1' and u_2', strains predicted by the two theories substantially differ. The axial strains ε in Reissner's beam changes for different rotation angles, while Timoshenko beam theory neglects θ effects on $\widetilde{\varepsilon}$. With the increase of deformations, errors between nonlinear beam theory and linear beam theory become no longer negligible. Therefore, for large displacements or large deformation problems, it is necessary to take geometric nonlinearity into consideration.

2.3 Substrate Models

Different substrate models have different properties and are used in various scenarios. Among those proposed in the literature, the simplest is the Winkler model (Fig. 2.5.(a)). It is the most widely used theoretical linear model of foundations, and it is the easiest for mathematical treatment. The hypothesis of the Winkler model is that the substrate at a given interface point responds as a linear spring. Therefore, within this model, the deformation of the foundation's surface is proportional to the stress at the same point. One of the most significant drawbacks is that displacements of adjacent springs are discontinuous due to the assumption of independence of each spring.

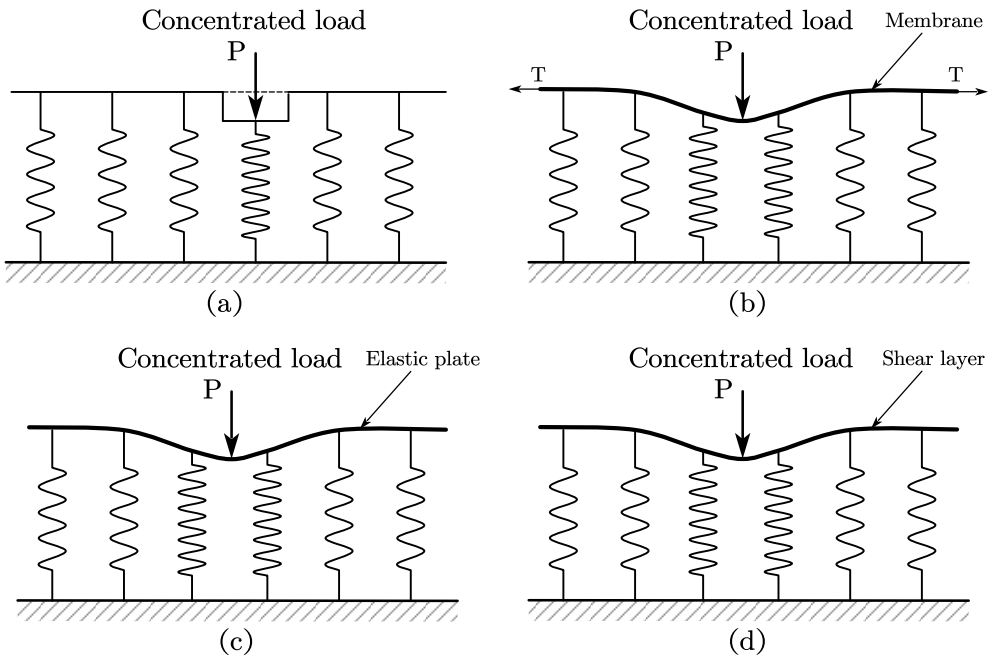


Fig. 2.5. Substrate model. (a) The Winkler model. (b) The Filonenko-Borodich model. (c) The Hetenyi model. (d) The Pasternak model.

Many substrate models have been proposed to overcome this limitation. Filonenko-Borodich, Hetenyi, Pasternak, and others made great contributions to the development of new models are named after them. Different from Winkler model, these new models all introduce local interactions between contiguous elements of the substrate, and therefore consider a continuos system. The continuity assumption translates into the introduction of additional constitutive parameters for the foundation.

In the 1940s, Filonenko-Borodich proposed a model in which an elastic membrane was added to the Winkler model (Fig. 2.5.(b)). At about the same time, Hetenyi presented a similar model in which the elastic membrane in the Filonenko-Borodich model is replaced by an elastic plate without any tension, which undergoes flexural deformation (Fig. 2.5.(c)). In 1954,

Pasternak also added an elastic plate to connect the individual springs. Unlike the Hetenyi model, the plate in the Pasternak model undergoes transverse shear deformation (Fig. 2.5.(d)). It was the first time that a shear behavior was taken into consideration in a substrate model [13].

In biomedical field, materials are usually soft and has distinct hardening, softening and hysteresis phenomena. The Pasternak model is suitable to describe soft materials because it accounts for local shear interaction between contiguous parts of the substrate, and therefore it allows one to model smooth deformed substrate profiles even under the action of localized forces[14]. Additionally, with the inclusion of nonlinear and viscous coefficients, one can model hardening, softening and viscous effects in the substrate. Therefore, nonlinear Pasternak foundation is modeled as the substrate of our sensing system. And a cubic term is considered in the nonlinear Pasternak model since it is sufficient to express nonlinear characteristics of common experimentally observed substrates [15]. Algebraic Equations of the substrate will be presented in Section 3.6.

The standard Pasternak model and the nonlinear viscoelastic Pasternak model adopted in this thesis are schematized in Fig. 2.6.

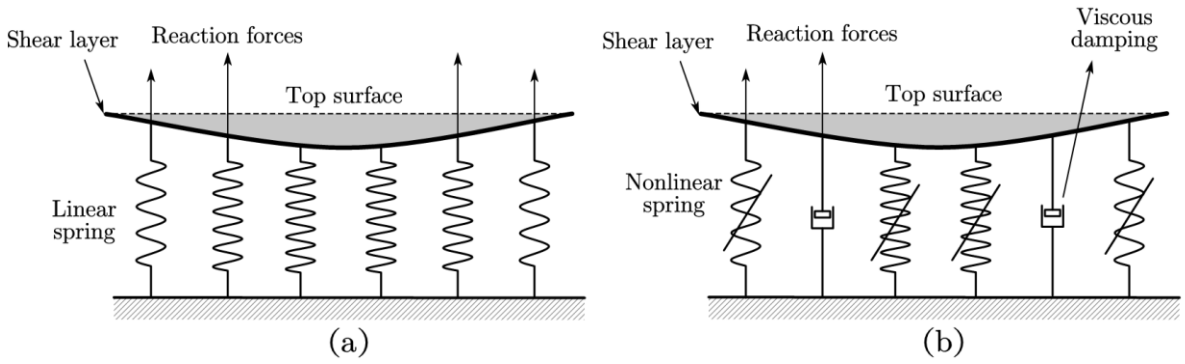


Fig. 2.6. (a) Standard Pasternak model. (b) Nonlinear viscoelastic Pasternak model

2.4 Coupling System

As described in chapter 1, our work is modeled in the framework of coupling system which has been studied in the past several years. For linear foundations, the dynamic behavior of beams is demonstrated in [16, 17], while the analysis of elastic beams on nonlinear foundations is carried out in [18]. The natural frequencies of a Timoshenko beam resting on a Pasternak-type foundation are calculated in [19]. The dynamic response for a Timoshenko beam resting on a nonlinear viscoelastic foundation is presented in [20]. In [21], a nonlinear dynamic analysis of a Timoshenko beam on a viscoelastic foundation and subjected to the motion of a travelling mass is analyzed. Although these researches may refer to alternative approaches of our work, they generalize the system as beams deploying on substrates directly. In our model, the sensing system should equip on an existent mobile robot (prototype) and use existent mechanical components to work. Therefore, a coupling layer which refers to the legs of the mobile robot is essential in this thesis. In [1], a sensor system is abstracted as a beam

deploying on a foundation via coupling elements, and our work extends its functionality by considering large deformation scenario in which evolved dynamics of the coupling layer is presented Section 3.5.

2.5 Inverse problem

As stated in Chapter 1, the theoretical framework of this thesis involves inverse problems which can be broadly defined as the process of using given observables to predict unknown properties of the model, see [22] for a general discussion. This framework is widely applied in science and engineering [23], for example: inverse problems for dynamical systems [24], identification of the unknown boundary of an object in thermal system [25], and the estimation of parameters in remote sensing field [26]. The work we present in this thesis belongs to a class in which the parameters are predicted in the continuum mechanics framework. One of the methods to solve inverse problems in elastoplastic solids is introduced in [27], and an inverse problem for estimating the elastic parameters is presented in [28]. In [29], a finite element method is used to estimate the material parameters from force and displacement measurements. In [30], advanced methods to estimate the parameters of substrate material are introduced in the civil engineering field. In this work, a deformable body coupled with a viscoelastic substrate through distributed coupling elements is modeled. Material parameters are predicted by analyzing the deformation of the body as well as the substrate's surface and

by minimizing the cost function encodes the distance between the measurements and the model predictions.

Inverse problems are solved using different mathematical methods. Textbook [22] introduces the general inverse problem theory and methods for model parameter estimation, and in textbook [31], methods for solving inverse problems in dynamic systems are presented, while the application of different methods to solve inverse problems can be found in [13, 14]. Solutions to specific inverse problems are also widely studied. Among all research, a discrete inverse vibration problem with uncertain parameters is studied in [31]; a variational approach for solving an inverse vibration problem is adopted in [34]; inverse problems with parameter uncertainty in vibrating systems are studied in [35]; a unique solution to an inverse electroencephalographic problem is proposed in [36], a fundamental solution method for an inverse problem of plates is derived in [37], and inverse problems and solutions in wave propagation are discussed in [38].

Considering the functionality of the prototype presented in Chapter 1, the autonomous robot locomotion problem is a direct problem since it establishes a relation between actuating inputs and the motion of the device, compatibly with a set of physical constraints posed by the environment. The problem of sensing the environment and estimating the material parameters of the substrate/terrain is, instead, posed as an inverse problem since, by assuming a material response of the terrain (constitutive model), one has to reconstruct the input to the device (force exerted on it by the terrain) that produces a measured output (deformation of the device).

In this inverse problem setting, once the material response is assumed, the set of material parameters in the constitutive model of the terrain are the unknowns and that have to be estimated.

In our sensor system, axial displacements, vertical displacements and rotational displacements of the beam are responses to a specific excitation of the system and the magnitude of the displacements can reflect certain properties of the substrate. For example, by applying certain forces to the beam, small deformations are expected if the substrate is hard; whereas if the substrate is soft, larger deformation of the beam should be expected. The parameters of the foundation are estimated using an iterative least square method, based on a metric that measures the distance between measured quantities and modelled ones, in which the substrate's material parameters are unknowns encoded in the model. This class of numerical optimization tools is used to solve inverse problems in linear and non-linear vibrating systems in [39 - 41], and in [23, 24], it is applied to estimate the material parameters. Additionally, building a realistic living tissue model, and estimating its properties, may lead to the development of diagnostic techniques. Inverse problems in soft tissue and material parameter identification are presented in book [44]. [45 - 48] show developments in treating human diseases in the biomechanics field. The theoretical model in this thesis can be implemented in engineering field for environmental monitoring and has the potentiality in diagnosis of diseases by detecting living tissues' parameters in biomechanics fields.

CHAPTER 3
MECHANICAL MODEL OF THE SENSOR SYSTEM

In this chapter, the mechanical model of the sensor system is presented. The model is based on a geometrically nonlinear deformable Reissner beam resting on a nonlinear viscoelastic foundation through a coupling layer. By modelling the beam's deformation induced by the material response of the substrate, we pose an inverse problem in which measurements of deformations are inputs, and the material parameters of the substrates are unknowns to be estimated. The governing equations of the system describe the deformation of the Reissner beam and of the substrate interface.

3.1 Kinematics of Sensor's Body Model

The work in [2] presents a shear deformable one-dimensional beam theory. The beam's reference configuration is parametrically defined in terms of a curvilinear abscissa $s \in [0, L]$ as

$$s \in [0, L] \rightarrow (s_1(s), s_2(s)) \in \mathbb{R}^2 \tag{3.1}$$

therefore describing a curve in $\mathbb{R}^2$. As illustrated in Fig. 3.1, a reference configuration is described by the curve $(\xi_1(s),\ \xi_2(s))$, and the field $\theta_0(s)$ representing the initial orientation

of the cross section in a global frame.

Due to deformation, the points $\xi_1 = \xi_1(s)$ and $\xi_2 = \xi_2(s)$ map to $\xi_1 = \xi_1(s) + u_1(s)$ and $\xi_2 = \xi_2(s) + u_2(s)$, in which u_1 and u_2 are the global Cartesian components of the displacement along X_1 and X_2 respectively. We assume that the cross section of the beam remains plain but not necessarily normal to the beam's axis. Therefore, χ is the rotation of the cross section with respect to the orthogonal configuration, and θ is the global orientation of the cross section.

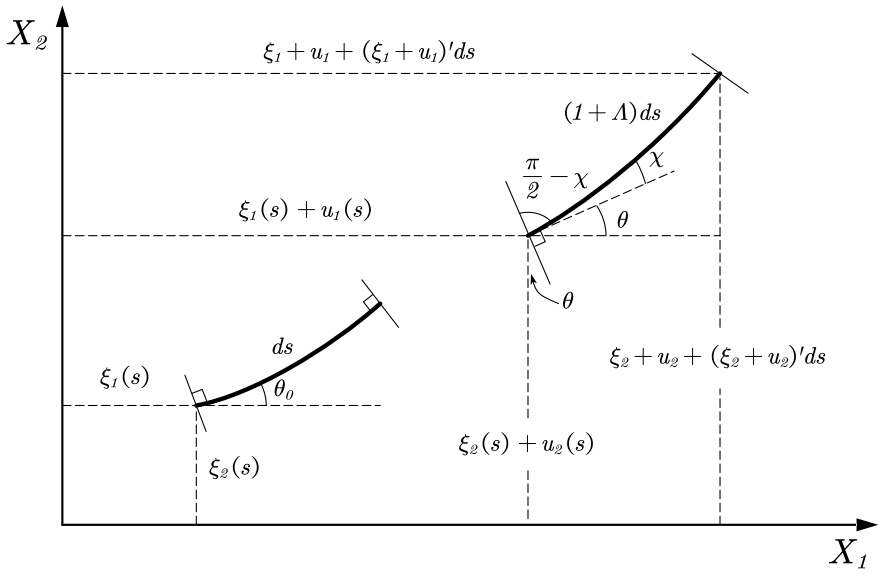


Fig. 3.1. Kinematics of the Reissner's beam that models the sensor's body

We then give the following relations

$$\begin{aligned}\tan(\theta+\chi) &= \frac{{\xi_2}' + {u_2}'}{{\xi_1}' + {u_1}'} \\ \cos(\theta+\chi) &= \frac{{\xi_1}' + {u_1}'}{1+\Lambda} \\ \sin(\theta+\chi) &= \frac{{\xi_2}' + {u_2}'}{1+\Lambda} \\ \Lambda &= \sqrt{({\xi_1}' + {u_1}')^2 + ({\xi_2}' + {u_2}')^2} - 1\end{aligned} \tag{3.2}$$

where $(\cdot)'$ means differentiation with respect to the curvilinear abscissa s, and Λ is the relative stretching, or change in length of an element of length *ds* in the reference configuration. In the reference configurations, relations (3.2) reduce to

$$\cos\theta_0 = {\xi_1}',\ \sin\theta_0 = {\xi_2}' \tag{3.3}$$

3.2 Dynamics of Sensor's Body Model

In order to obtain the equilibrium equation, we refer to Fig. 3.2 for a portion of deformed beam with normal force N, shear forces Q and a bending moment M in a frame rotated with the cross section. Forces per unit length p_1, p_2 and moment μ are distributed along the undeformed beam.

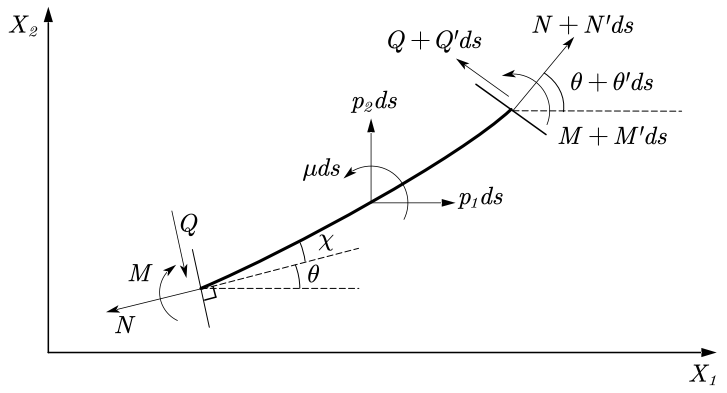


Fig. 3.2. Dynamics of the beam that models the sensor's body

To write the equilibria, we also consider the following first order approximations

$$\cos(\theta + \theta' ds) \approx \cos\theta - \theta' \sin\theta ds \tag{3.4}$$

$$\sin(\theta + \theta' ds) \approx \sin\theta + \theta' \cos\theta ds \tag{3.5}$$

The force equilibria along the global axis (X_1, X_2) gives

$$\begin{aligned} &-N\cos\theta + (N + N'ds)\cos(\theta + \theta' ds) \\ &+Q\sin\theta - (Q + Q'ds)\sin(\theta + \theta' ds) + p_1 ds = 0 \end{aligned} \tag{3.6}$$

$$\begin{aligned} &-N\sin\theta + (N + N'ds)\sin(\theta + \theta' ds) \\ &-Q\cos\theta + (Q + Q'ds)\cos(\theta + \theta' ds) + p_2 ds = 0 \end{aligned} \tag{3.7}$$

By substituting (3.4) (3.5) into (3.6) (3.7) and neglecting higher order terms, we obtain

$$(N\cos\theta - Q\sin\theta)' + p_1 = 0 \tag{3.8}$$

$$(N\sin\theta + Q\cos\theta)' + p_2 = 0 \tag{3.9}$$

The moment equilibrium, on the other hand, provide

$$M' + (1 + \Lambda)(Q\cos\chi - N\sin\chi) + \mu = 0 \tag{3.10}$$

We can also obtain a relationship between angles χ and θ using trigonometric summation formulas from (3.2)

$$\cos\chi = \frac{(\xi_1{}' + u_1{}')\cos\theta + (\xi_2{}' + u_2{}')\sin\theta}{1+\Lambda} \tag{3.11}$$

$$\sin\chi = \frac{(\xi_2{}' + u_2{}')\cos\theta - (\xi_1{}' + u_1{}')\sin\theta}{1+\Lambda} \tag{3.12}$$

Therefore, equation (3.10) can be rewritten in terms of u_1, u_2 and θ.

$$\begin{aligned} M' + Q\big((\xi_1{}' + u_1{}')\cos\theta + (\xi_2{}' + u_2{}')\sin\theta\big) \\ -N\big((\xi_2{}' + u_2{}')\cos\theta - (\xi_1{}' + u_1{}')\sin\theta\big) + \mu = 0 \end{aligned} \tag{3.13}$$

3.3 Strain-displacement Relations

In [2], strain-displacement relations are obtained from a variational argument by posing that strains ε, γ, and κ are the duals of the stress resultant N, Q, and M respectively. Therefore, the geometric interpretation of the three strain measures are respectively normal strain, shear strain, and curvature. The curvature of the neutral axis is given by the familiar expression

$$\kappa = \theta' - \theta_0{}' \tag{3.14}$$

By inverting (3.2) we obtain

$$\theta - \theta_0 = \tan^{-1}\left(\frac{\xi_2{}' + u_2{}'}{\xi_1{}' + u_1{}'}\right) - \tan^{-1}\left(\frac{\xi_2{}'}{\xi_1{}'}\right) - \chi \tag{3.15}$$

Therefore, the curvature can also be expressed as

$$\begin{aligned} \kappa = \frac{(\xi_1{}' + u_1{}')\,(\xi_2{}'' + u_2{}'') - (\xi_2{}' + u_2{}')\,(\xi_1{}'' + u_1{}'')}{(1+\Lambda)^{\,2}} \\ - (\xi_1{}'\xi_2{}'' - \xi_2{}'\xi_1{}'') - \chi' \end{aligned} \tag{3.16}$$

The set of strain-displacement relations is completed by [2]

$$\varepsilon = (1+\Lambda)\cos\chi - 1 \tag{3.17}$$

$$\gamma = (1+\Lambda)\sin\chi \tag{3.18}$$

We can also rewrite the moment equilibrium equation (3.10) using (3.17) and (3.18) in the form

$$M' + (1+\varepsilon)Q - \gamma N + \mu = 0 \tag{3.19}$$

3.4 Constitutive Relations

According to Reissner [2], for the class of nonlinear beams considered here, constitutive relations have to be determined by appropriate experiments. Following [49] and [14], it is possible to adopt linear isotropic material responses of the form

$$M = EI\kappa,\ N = EA\varepsilon,\ Q = GA\gamma \tag{3.20}$$

where E and G are isotropic elastic parameters, and A and I are cross sectional area and geometric moment of inertia. These relations respect the monotonicity condition in the sense of Antman and Rosenfeld [50]. It should be noted that when $\Lambda \to -1$, ε is close to -1 and the desired value of N should be $N \to -\infty$. However, the linear elastic constitutive law (3.20) cannot account for this property. Discussions can be found in [49] that the non-admissible region $\Lambda \leqslant -1$ can be considered as the limitation of the linear elastic material response, which also defines in the limitation of this thesis.

3.5 Coupling Layer: Force Between Beam and Substrate

By referring to Fig. 3.3, the coupling layer interfaces the body of the sensor with the substrate. The device modelled here is devised to be an autonomous robotic system with sensing capabilities, and therefore the coupling layer has the function of both providing locomotion and to provide coupling and actuation for sensing. Considering autonomous operation, it is desirable for the coupling to be as low power consumption as possible, and at the same time to have the level of complexity that allows one to generate the set of actions necessary for the tasks, especially locomotion.

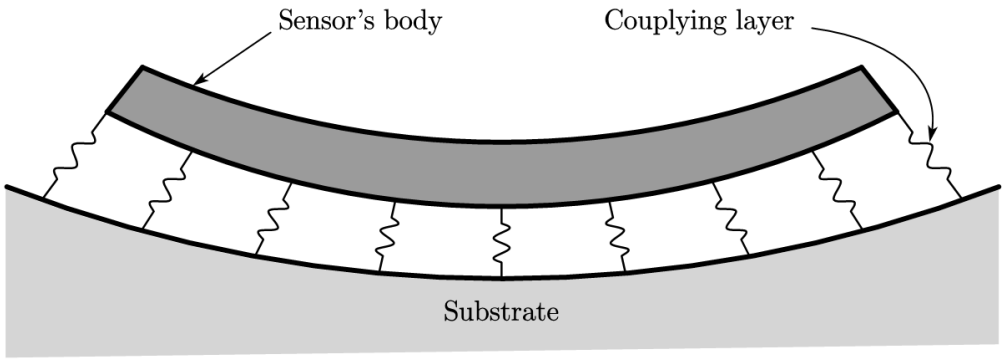


Fig. 3.3. Sensing system modelled in this thesis

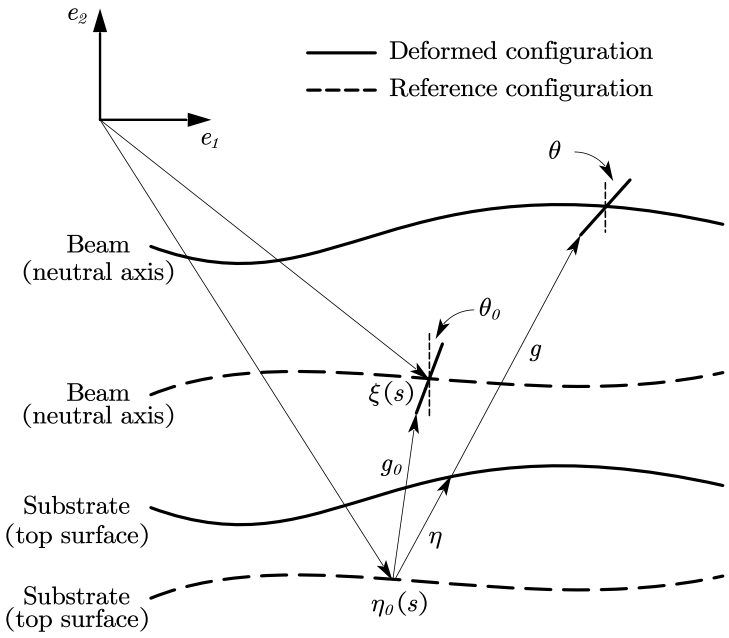


Fig. 3.4. Geometric relations between the beam and the substrate.

For the purpose of sensing, the coupling layer is considered to be equivalent to a distribution of springs, transmitting a force $\boldsymbol{p}$ along the local gap between the beam and the substrate. The geometry is schematized in Fig. 3.4, in which $\boldsymbol{g}$ is the actual gap and $\boldsymbol{\eta}_0 + \boldsymbol{\eta}$ is the substrate current profile. The coupling force provided by the layer is assumed to be directed along the current gap, and to have magnitude proportional to the change of gap magnitude:

$$\boldsymbol{p}_{u\eta} = -k(\|\boldsymbol{g}\| - \|\boldsymbol{g}_0\|)\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} = -k\left(1 - \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}\|}\right)\boldsymbol{g} = -k\lambda\boldsymbol{g} \tag{3.21}$$

where $\boldsymbol{g}_0$ is the gap in the reference configuration and

$$\lambda = 1 - \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}\|} \tag{3.22}$$

The current gap is given by

$$\boldsymbol{g} = \boldsymbol{\xi} + \boldsymbol{u} - \frac{h}{2}\boldsymbol{R}(\theta)\boldsymbol{e}_2 - (\boldsymbol{\eta}_0 + \boldsymbol{\eta}) \tag{3.23}$$

where h is the cross section thickness and $\boldsymbol{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ is the standard two dimensional rotation matrix. The gap in the reference configuration is the one corresponding to zero displacements, that is $\boldsymbol{g}_0 = \boldsymbol{\xi} - \frac{h}{2}\boldsymbol{R}(\theta_0)\boldsymbol{e}_2 - \boldsymbol{\eta}_0$. By accounting for the thickness, the coupling force $\boldsymbol{p}$ induces the distributed couple

$$\boldsymbol{\mu}_{u\eta} = -\frac{h}{2}\boldsymbol{R}(\theta)\boldsymbol{e}_2 \wedge \boldsymbol{p}_{u\eta} \tag{3.24}$$

where $\wedge$ is the cross product. Because of the assumption of plane motion, couple $\boldsymbol{\mu}_{u\eta}$ is always normal to the plane of the motion, and therefore it will be indicated with the magnitude $\mu_{u\eta}$ in the moment equation.

3.6 Substrate Model

The nonlinear viscoelastic Pasternak model is schematized in Fig. 3.5.

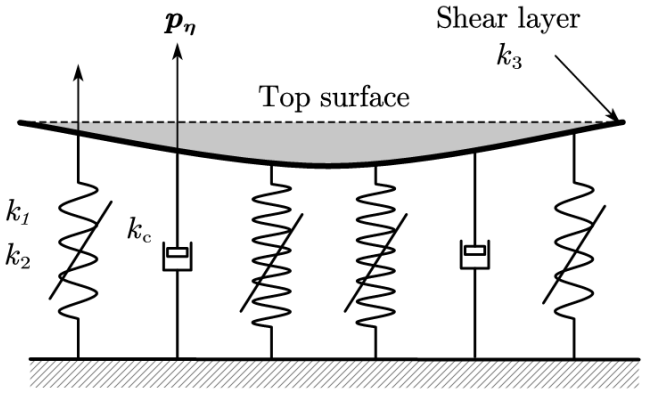


Fig. 3.5. Nonlinear viscoelastic Pasternak model

The pressure–deformation relationship of the nonlinear viscoelastic Pasternak model can be written as [20]

$$\boldsymbol{p}_{\boldsymbol{\eta}} = (k_1 + k_2(\boldsymbol{\eta} \cdot \boldsymbol{\eta}))\boldsymbol{\eta} - k_3\boldsymbol{\eta}'' + k_c\dot{\boldsymbol{\eta}} \tag{3.25}$$

where $\boldsymbol{p}_{\boldsymbol{\eta}}$ is the force induced by the substrate per unit length; $\boldsymbol{\eta}$ is the displacement of the top if substrate, and k_1, k_2 are linear and nonlinear elastic parameters, respectively. Furthermore, k_3 is the shear layer interaction stiffness, and k_c is the foundation's viscosity. If we set k_2 and k_c to zero, equation (3.24) reduces to the standard Pasternak model

$$\boldsymbol{p}_{\boldsymbol{\eta}} = k_1\boldsymbol{\eta} - k_3\boldsymbol{\eta}'' \tag{3.26}$$

3.7 Nondimensional Governing Equations

Following d'Alambert approach, we introduce the effective loads in equation (3.8), (3.9) and (3.19) to include the inertia:

$$\boldsymbol{p} = \boldsymbol{p}_{\boldsymbol{u\eta}} - \rho A\ddot{\boldsymbol{u}} - c_{\boldsymbol{u}}\dot{\boldsymbol{u}} \tag{3.27}$$

$$\mu = \mu_{\boldsymbol{u\eta}} - \rho I\ddot{\theta} - c_\theta\dot{\theta} \tag{3.28}$$

where ρ is the mass density assumed to be uniform, $c_{\boldsymbol{u}}$ and c_θ are linear damping coefficients, modeling structural damping effects as equivalent viscous damping [51]. Structural damping accounts for hysteresis in elastic material undergoing cyclic loading, and therefore it depends in general on the frequency content of the loading [51], [52]. In common equivalent damping models [52], the dependency on the dominant frequency of excitation $\overline{\omega}$ is expressed by the

inverse relation $c_{\boldsymbol{u}} = \overline{c}_{\boldsymbol{u}}/(\pi\overline{\omega})$ (similarly for c_θ), where the structural damping $\overline{c}_{\boldsymbol{u}}$ is independent of $\overline{\omega}$. Force balance at the interface between the substrate and the coupling layer gives

$$\boldsymbol{p}_{\boldsymbol{\eta}} = \boldsymbol{p}_{\boldsymbol{u\eta}} \tag{3.29}$$

Noting equation (3.24) and expressing the vector quantities in the global Cartesian coordinate frame, we obtain

$$(k_1 + k_2(\boldsymbol{\eta}_1 \cdot \boldsymbol{\eta}_1))\boldsymbol{\eta}_1 - k_3\boldsymbol{\eta}_1'' + k_c\dot{\boldsymbol{\eta}}_1 = \boldsymbol{p}_{\boldsymbol{\eta}1} = \boldsymbol{p}_{\boldsymbol{u\eta}1} = -k\lambda\boldsymbol{g}_1 \tag{3.30}$$

$$(k_1 + k_2(\boldsymbol{\eta}_2 \cdot \boldsymbol{\eta}_2))\boldsymbol{\eta}_2 - k_3\boldsymbol{\eta}_2'' + k_c\dot{\boldsymbol{\eta}}_2 = \boldsymbol{p}_{\boldsymbol{\eta}2} = \boldsymbol{p}_{\boldsymbol{u\eta}2} = -k\lambda\boldsymbol{g}_2 \tag{3.31}$$

with Cartesian coordinates indicated by subscripts 1 and 2.

By substituting equation (3.8), (3.9) and (3.19) into equation (3.26) and (3.27), and by combining equation (3.29) and equation (3.30), we obtain the governing equations

$$(N\cos\theta - Q\sin\theta)' - k\lambda g_1 - \rho A\ddot{u}_1 - c_{\boldsymbol{u}}\dot{u}_1 = 0 \tag{3.32a}$$

$$(N\sin\theta + Q\cos\theta)' - k\lambda g_2 - \rho A\ddot{u}_2 - c_{\boldsymbol{u}}\dot{u}_2 = 0 \tag{3.32b}$$

$$M' + (1+\varepsilon)Q - \gamma N + \mu_{\boldsymbol{u\eta}} - \rho I\ddot{\theta} - c_\theta\dot{\theta} = 0 \tag{3.32c}$$

$$(k_1 + k_2(\boldsymbol{\eta}_1 \cdot \boldsymbol{\eta}_1))\eta_1 - k_3\eta_1'' + k_c\dot{\eta}_1 + k\lambda g_1 = 0 \tag{3.32d}$$

$$(k_1 + k_2(\boldsymbol{\eta}_2 \cdot \boldsymbol{\eta}_2))\eta_2 - k_3\eta_2'' + k_c\dot{\eta}_2 + k\lambda g_2 = 0 \tag{3.32e}$$

We introduce the following nondimensional variables

$$\hat{s} = \frac{s}{L},\ \ \hat{\boldsymbol{u}} = \frac{\boldsymbol{u}}{L},\ \ \hat{\boldsymbol{\xi}} = \frac{\boldsymbol{\xi}}{L},\ \ \hat{\boldsymbol{\eta}} = \frac{\boldsymbol{\eta}}{L},\ \ \hat{t} = t\sqrt{\frac{G}{\rho L^2}} \tag{3.33}$$

and the nondimensional groups

$$\alpha_1 = \frac{E}{G},\quad \alpha_2 = \frac{I}{AL^2},\quad \alpha_3 = \frac{kL^2}{GA},\quad \bar{h} = \frac{h}{L} \tag{3.34}$$

$$\alpha_{c_u} = \frac{\bar{c}_{\boldsymbol{u}} L}{\pi A \bar{\omega}\sqrt{\rho G}},\quad \alpha_{c_\theta} = \frac{\bar{c}_{\theta} L}{\pi A \bar{\omega}\sqrt{\rho G}} \tag{3.35}$$

$$\alpha_4 = \frac{k_1 L^2}{GA},\quad \alpha_5 = \frac{k_2 L^4}{GA},\quad \alpha_6 = \frac{k_3 L}{GA},\quad \alpha_c = \frac{k_c L}{A\sqrt{\rho G}} \tag{3.36}$$

to obtain the nondimensionalized governing equations

$$(\alpha_1 \varepsilon \cos\theta - \gamma \sin\theta)' - \alpha_3 \lambda g_1 - \alpha_{c_u} \dot{u}_{\boldsymbol{1}} - \ddot{u}_{\boldsymbol{1}} = 0 \tag{3.37a}$$

$$(\alpha_1 \varepsilon \sin\theta + \gamma \cos\theta)' - \alpha_3 \lambda g_2 - \alpha_{c_u} \dot{u}_2 - \ddot{u}_2 = 0 \tag{3.37b}$$

$$\begin{aligned} \alpha_1\alpha_2(\theta - \theta_0)'' + \gamma\big((\xi_1{}' + u_1{}')\cos\theta + (\xi_2{}' + u_2{}')\sin\theta\big) \\ -\alpha_1\varepsilon\big((\xi_2{}' + u_2{}')\cos\theta - (\xi_1{}' + u_1{}')\sin\theta\big) \\ + \frac{\bar{h}}{2}\lambda(\boldsymbol{R}(\theta)\boldsymbol{e_2} \wedge \boldsymbol{g}) \cdot \boldsymbol{e}_3 - \alpha_{c_\theta}\dot{\theta} - \alpha_2\ddot{\theta} = 0 \end{aligned} \tag{3.37c}$$

$$(\alpha_4 + \alpha_5(\boldsymbol{\eta}_1 \cdot \boldsymbol{\eta}_1))\eta_1 - \alpha_6 \eta_1'' + \alpha_c \dot{\eta}_1 + \alpha_3 \lambda g_1 = 0 \tag{3.37d}$$

$$(\alpha_4 + \alpha_5(\boldsymbol{\eta}_2 \cdot \boldsymbol{\eta}_2))\eta_2 - \alpha_6 \eta_2'' + \alpha_c \dot{\eta}_2 + \alpha_3 \lambda g_2 = 0 \tag{3.37e}$$

where the hat on nondimensional variables has been dropped, thereby indicating nondimensional variables with the same symbols previously used for the dimensional ones. In (3.34), $\alpha_1\alpha_2 = \frac{EI}{GAL^2}$ measures the beam's bending stiffness versus its shear stiffness; α_3 measures the coupling layer's linear stiffness versus the beam's shear stiffness. In (3.36), α_4 is the measure of substrate's linear stiffness with respect to the beam's shear stiffness; α_5 is the measure of substrate's nonlinear stiffness with respect to the beam's shear stiffness and α_6 is the measure of substrate's shear stiffness with respect to the beam's shear stiffness.

3.8 Weak Form of The Governing Equations

With the aid of Galerkin projections, a reduced order model of the system (3.37) is derived. The kinematic fields are approximated by separation of variables as

$$u_1(s,t) = \bar{\boldsymbol{u}}_1^T(s)\boldsymbol{a}(t) \tag{3.38a}$$

$$u_2(s,t) = \bar{\boldsymbol{u}}_2^T(s)\boldsymbol{b}(t) \tag{3.38b}$$

$$\theta(s,t) = \bar{\boldsymbol{\theta}}^T(s)\boldsymbol{c}(t) \tag{3.38c}$$

$$\eta_1(s,t) = \bar{\boldsymbol{\eta}}_1^T(s)\boldsymbol{d}(t) \tag{3.38d}$$

$$\eta_2(s,t) = \bar{\boldsymbol{\eta}}_2^T(s)\boldsymbol{e}(t) \tag{3.38e}$$

where $\bar{\boldsymbol{u}}_1 = (\boldsymbol{u}_{1_1}\ \boldsymbol{u}_{1_2}\ ...\ \boldsymbol{u}_{1_n})^T$, $\bar{\boldsymbol{u}}_2 = (\boldsymbol{u}_{2_1}\ \boldsymbol{u}_{2_2}\ ...\ \boldsymbol{u}_{2_n})^T$, $\bar{\boldsymbol{\theta}} = (\boldsymbol{\theta}_1\ \boldsymbol{\theta}_2\ ...\ \boldsymbol{\theta}_n)^T$, $\bar{\boldsymbol{\eta}}_1 = (\boldsymbol{\eta}_{1_1}\ \boldsymbol{\eta}_{1_2}\ ...\ \boldsymbol{\eta}_{1_n})^T$ and $\bar{\boldsymbol{\eta}}_2 = (\boldsymbol{\eta}_{2_1}\ \boldsymbol{\eta}_{2_2}\ ...\ \boldsymbol{\eta}_{2_n})^T$ are spatial basis functions. $\boldsymbol{a} = (\boldsymbol{a}_1\ \boldsymbol{a}_2\ ...\ \boldsymbol{a}_n)^T$, $\boldsymbol{b} = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ ...\ \boldsymbol{b}_n)^T$, $\boldsymbol{c} = (\boldsymbol{c}_1\ \boldsymbol{c}_2\ ...\ \boldsymbol{c}_n)^T$, $\boldsymbol{d} = (\boldsymbol{d}_1\ \boldsymbol{d}_2\ ...\ \boldsymbol{d}_n)^T$ and $\boldsymbol{e} = (\boldsymbol{e}_1\ \boldsymbol{e}_2\ ...\ \boldsymbol{e}_n)^T$ are time-dependent amplitudes. Substituting (3.38) into (3.37) and pre-multiplying by $\bar{\boldsymbol{u}}_1$, $\bar{\boldsymbol{u}}_2$, $\bar{\boldsymbol{\theta}}$, $\bar{\boldsymbol{\eta}}_1$ and $\bar{\boldsymbol{\eta}}_2$ respectively, and by introducing the matrices,

$$\boldsymbol{M}_1 = \int_0^1 \bar{\boldsymbol{u}}_1\bar{\boldsymbol{u}}_1^T ds,\ \ \boldsymbol{M}_2 = \int_0^1 \bar{\boldsymbol{u}}_2\bar{\boldsymbol{u}}_2^T ds,\ \ \boldsymbol{M}_3 = \int_0^1 \bar{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}}^T ds \tag{3.39}$$

$$\boldsymbol{M}_4 = \int_0^1 \bar{\boldsymbol{\eta}}_1\bar{\boldsymbol{\eta}}_1^T ds,\ \boldsymbol{M}_5 = \int_0^1 \bar{\boldsymbol{\eta}}_1\left(\boldsymbol{d}\bar{\boldsymbol{\eta}}_1\bar{\boldsymbol{\eta}}_1^T\boldsymbol{d}^T\right)\bar{\boldsymbol{\eta}}_1^T ds,\ \boldsymbol{M}_6 = \int_0^1 \bar{\boldsymbol{\eta}}_1{}'\bar{\boldsymbol{u}}_1^T{}' ds \tag{3.40}$$

$$\boldsymbol{M}_7 = \int_0^1 \bar{\boldsymbol{\eta}}_2\bar{\boldsymbol{\eta}}_2^T ds,\ \boldsymbol{M}_8 = \int_0^1 \bar{\boldsymbol{\eta}}_2\left(\boldsymbol{e}\bar{\boldsymbol{\eta}}_2\bar{\boldsymbol{\eta}}_2^T\boldsymbol{e}^T\right)\bar{\boldsymbol{\eta}}_2^T ds,\ \boldsymbol{M}_9 = \int_0^1 \bar{\boldsymbol{\eta}}_2{}'\bar{\boldsymbol{\eta}}_2^T{}' ds \tag{3.41}$$

$$\boldsymbol{K}_1 = \int_0^1 \bar{\boldsymbol{u}}_1{}'(\alpha_1\varepsilon(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})cos\theta(\boldsymbol{c}) - \gamma(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\sin\theta(\boldsymbol{c}))ds \tag{3.42}$$

$$\boldsymbol{K}_2 = \int_0^1 \overline{\boldsymbol{u}}_2{}'(\alpha_1 \varepsilon(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}) \sin\theta(\boldsymbol{c}) + \gamma(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}) \cos\theta(\boldsymbol{c}))ds \tag{3.43}$$

$$\boldsymbol{K}_3 = \int_0^1 \overline{\boldsymbol{\theta}}'(\boldsymbol{\theta}' - \boldsymbol{\theta}_0{}')ds, \quad \boldsymbol{K}_4 = \int_0^1 \overline{\boldsymbol{\theta}}(1+\varepsilon)\gamma\, ds, \quad \boldsymbol{K}_5 = \int_0^1 \overline{\boldsymbol{\theta}}\varepsilon\gamma\, ds \tag{3.44}$$

$$\boldsymbol{F}_H = \alpha_3 \int_0^1 \overline{\boldsymbol{u}}_1 \lambda g_1 ds, \quad \boldsymbol{F}_v = \alpha_3 \int_0^1 \overline{\boldsymbol{u}}_2 \lambda g_2 ds, \quad \boldsymbol{F}_\theta = \frac{\overline{h}}{2} \int_0^1 \overline{\boldsymbol{\theta}} \lambda (\boldsymbol{R}(\theta)\boldsymbol{e}_2 \wedge \boldsymbol{g}) \cdot \boldsymbol{e}_3 ds \tag{3.45}$$

$$\boldsymbol{F}_{s1} = \alpha_3 \int_0^1 \overline{\boldsymbol{\eta}}_1 \lambda g_1 ds, \quad \boldsymbol{F}_{s2} = \alpha_3 \int_0^1 \overline{\boldsymbol{\eta}}_2 \lambda g_2 ds \tag{3.46}$$

and the boundary terms

$$\begin{aligned} \boldsymbol{F}_{b1} = {} & \overline{\boldsymbol{u}}_1(1)\,(N_1 \cos\theta(1) - Q_1 \sin\theta(1)) \\ & - \overline{\boldsymbol{u}}_1(0)\,(N_0 \cos\theta(0) - Q_0 \sin\theta(0)) \end{aligned} \tag{3.47}$$

$$\begin{aligned} \boldsymbol{F}_{b2} = {} & \overline{\boldsymbol{u}}_2(1)\,(N_1 \sin\theta(1) + Q_1 \cos\theta(1)) \\ & - \overline{\boldsymbol{u}}_2(0)\,(N_0 \sin\theta(0) + Q_0 \cos\theta(0)) \end{aligned} \tag{3.48}$$

$$\boldsymbol{F}_{b3} = \overline{\boldsymbol{\theta}}(1) M_1 - \overline{\boldsymbol{\theta}}(0) M_0 \tag{3.49}$$

$$\begin{aligned} \boldsymbol{F}_{b4} &= \alpha_6 \left(\overline{\boldsymbol{\eta}}_1(1) \overline{\boldsymbol{\eta}}_1^T{}'(1) - \overline{\boldsymbol{\eta}}_1(0) \overline{\boldsymbol{\eta}}_1^T{}'(0) \right) = 0 \\ \boldsymbol{F}_{b5} &= \alpha_6 \left(\overline{\boldsymbol{\eta}}_2(1) \overline{\boldsymbol{\eta}}_2^T{}'(1) - \overline{\boldsymbol{\eta}}_2(0) \overline{\boldsymbol{\eta}}_2^T{}'(0) \right) = 0 \end{aligned} \tag{3.50}$$

we obtain the reduced order model in the form of the following coupled ordinary differential equations for the amplitudes $\boldsymbol{a}(t)$, $\boldsymbol{b}(t)$, $\boldsymbol{c}(t)$, $\boldsymbol{d}(t)$ and $e(t)$

$$\boldsymbol{M}_1 \ddot{\boldsymbol{a}} + \alpha_{cu} \boldsymbol{M}_1 \dot{\boldsymbol{a}} + \boldsymbol{K}_1 + \boldsymbol{F}_H = \boldsymbol{F}_{b1} \tag{3.51a}$$

$$\boldsymbol{M}_2 \ddot{\boldsymbol{b}} + \alpha_{cu} \boldsymbol{M}_2 \dot{\boldsymbol{b}} + \boldsymbol{K}_2 + \boldsymbol{F}_V = \boldsymbol{F}_{b2} \tag{3.51b}$$

$$\alpha_2 \boldsymbol{M}_3 \ddot{\boldsymbol{c}} + \alpha_{cu} \boldsymbol{M}_3 \dot{\boldsymbol{c}} + \alpha_1 \alpha_2 \boldsymbol{K}_3 - \boldsymbol{K}_4 + \alpha_1 \boldsymbol{K}_5 + \boldsymbol{F}_\theta = \boldsymbol{F}_{b3} \tag{3.51c}$$

$$\alpha_4 \boldsymbol{M}_4 \boldsymbol{d} + \alpha_5 \boldsymbol{M}_5 \boldsymbol{d} + \alpha_6 \boldsymbol{M}_6 \boldsymbol{d} + \alpha_c \boldsymbol{M}_4 \dot{\boldsymbol{d}} + \boldsymbol{F}_{s1} = \boldsymbol{F}_{b4} \tag{3.51d}$$

$$\alpha_4 \boldsymbol{M}_7 \boldsymbol{e} + \alpha_5 \boldsymbol{M}_8 \boldsymbol{e} + \alpha_6 \boldsymbol{M}_9 \boldsymbol{e} + \alpha_c \boldsymbol{M}_7 \dot{\boldsymbol{e}} + \boldsymbol{F}_{s2} = \boldsymbol{F}_{b5} \tag{3.51e}$$

In (3.50), the boundary terms vanish since there is no force at the boundary of the substrate in contact with the sensor.

Forces and moments in (3.47-3.49) are considered to be the inputs for the sensing system. In future hardware implementation, these forcing elements can be generated by different kinds of actuators. One of the potential solutions is sketched as Fig. 3.6. A rotary actuator is set to generate torques and the horizontal linear actuator as well as the vertical linear actuator is used to generate shear force and normal force. These actuators are also idealized for the purpose of locomotion of the mobile robot.

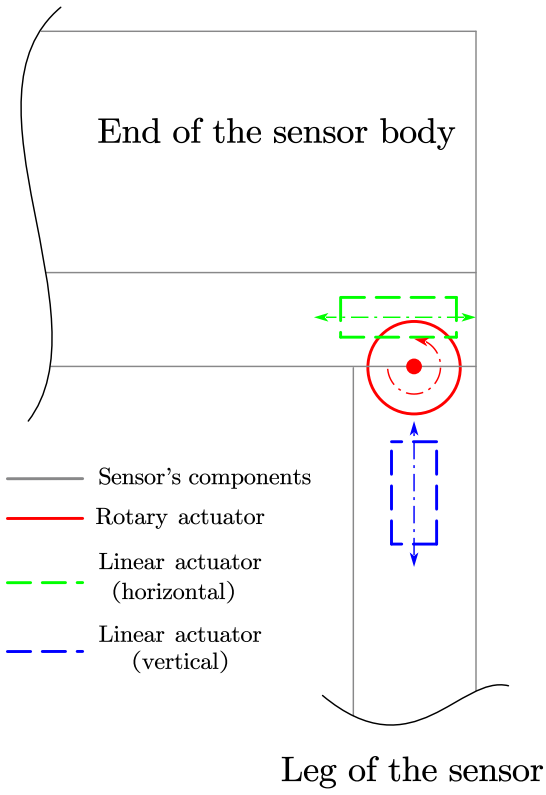


Fig. 3.6. Schematics of the actuators set at the end of the sensor

Displacements induced by the given forcing terms are measured at the observation points along the sensor beam, and substrate's material parameters are estimated. Details of the sensor's working method are presented in the next chapter.

3.9 Basis Functions for the Reduced Order Model

The reduced order model eliminates the spatial dependency by projecting the spatial part of the independent variables, and by integrating with respect to it. Generally, there are two different types of basis functions. One consists of functions defined on the whole domain (Fig. 3.7), the other consists of functions localized on sub-domains (Fig. 3.8). The latter is the approach of Finite Element Methods (FEM).

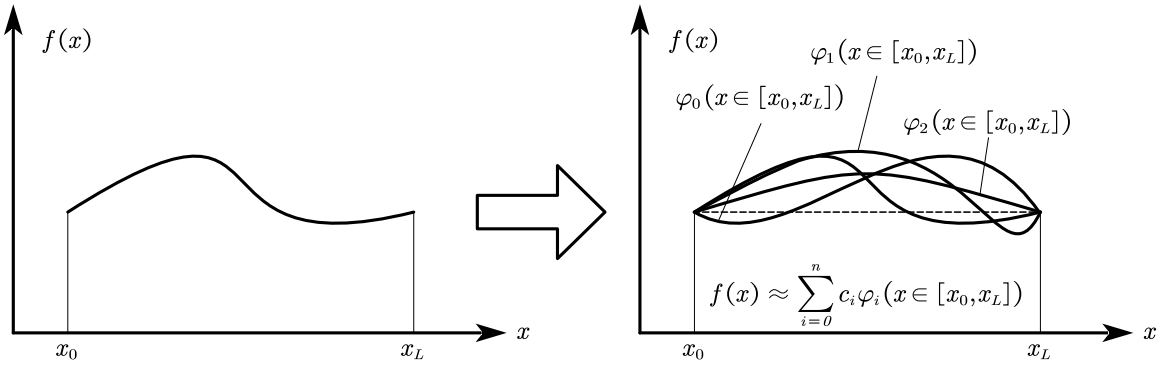


Fig. 3.7. Approximation based on $[x_0, x_L]$

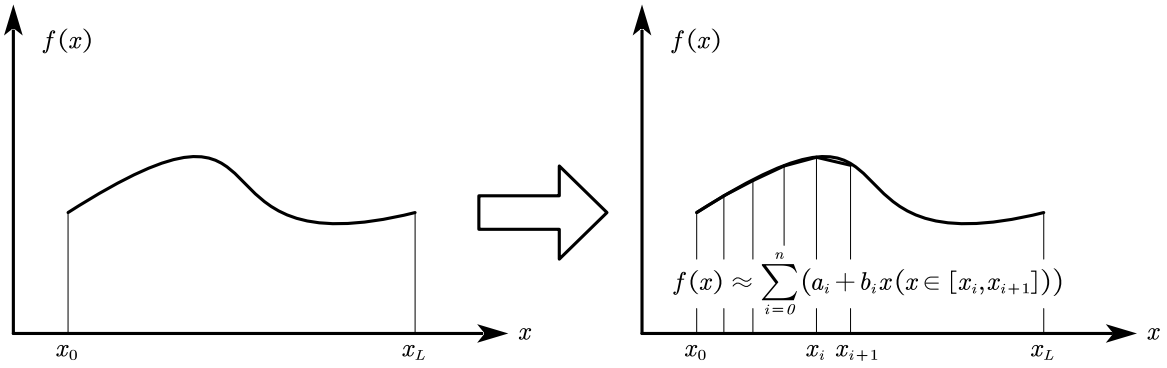


Fig. 3.8. Approximation based on $[x_i, x_{i+1}]$

Here, we adopt the Finite Element Method which is well established and reliable to solve nonlinear problems, since the related basis functions do not depend on the actual system to be solved, unlike modal basis functions. The advantage of using modal basis functions is that it is usually required to include a smaller number for the same accuracy, given that they are generated for the specific system; however, current computational tools typically allow to adopt Finite Element schemes with relatively high number of degrees of freedom, eliminating the steps of solving an eigenvalues problem that in this case requires some assumption on the substrate. Before adopting the finite element method, we have attempted to project on modal coordinates with generally poor results, likely due to the adoption of unsuitable shape functions for the substrate.

Let $0 = n_1 < n_2 < \cdots < n_i < n_{i+1} = 1$ be a partition of $[0, 1]$ into subintervals $I_i = (n_i,\ n_{i+1})$ of length $\Delta n_i = n_{i+1} - n_i$, and define the piecewise continuous functions in Fig. 3.8, which are suitable to be chosen as basis functions for $\bar{\boldsymbol{u}}_1$, $\bar{\boldsymbol{u}}_2$, $\bar{\boldsymbol{\theta}}$, $\bar{\boldsymbol{\eta}}_1$ and $\bar{\boldsymbol{\eta}}_2$ in

equation (3.38), since three criteria must be followed for finite elements basis functions: (1) Basis functions must be square integrable; (2) The first derivatives of basis functions must be integrable; (3) Function N_i must equal to 1 at node point n_i and equal to 0 at other node points. [53]

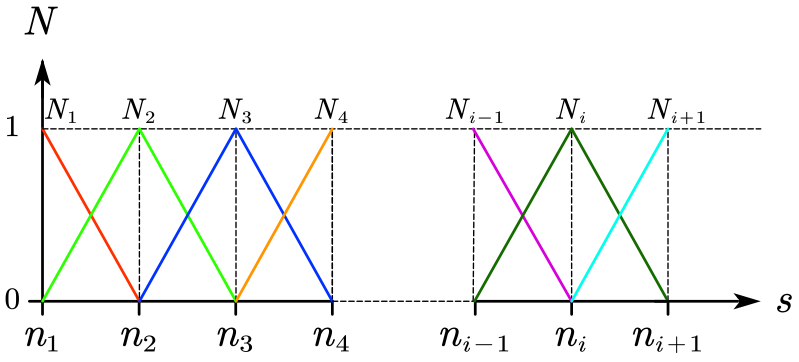


Fig. 3.9. Basis functions of the beam in global configuration

In algebraic form, these basis functions are defined as

$$N_i(s) = \begin{cases} \dfrac{s - n_{i-1}}{n_i - n_{i-1}}, & n_{i-1} \leqslant s \leqslant n_i \\ \dfrac{n_{i+1} - s}{n_{i+1} - n_i}, & n_i \leqslant s \leqslant n_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.52}$$

It is also worth noting that the basis functions also satisfy the following

$$N_j(n_i) = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases} \tag{3.53}$$

Examples of approximating beam shapes using FEM method are plotted in Fig. 3.10.

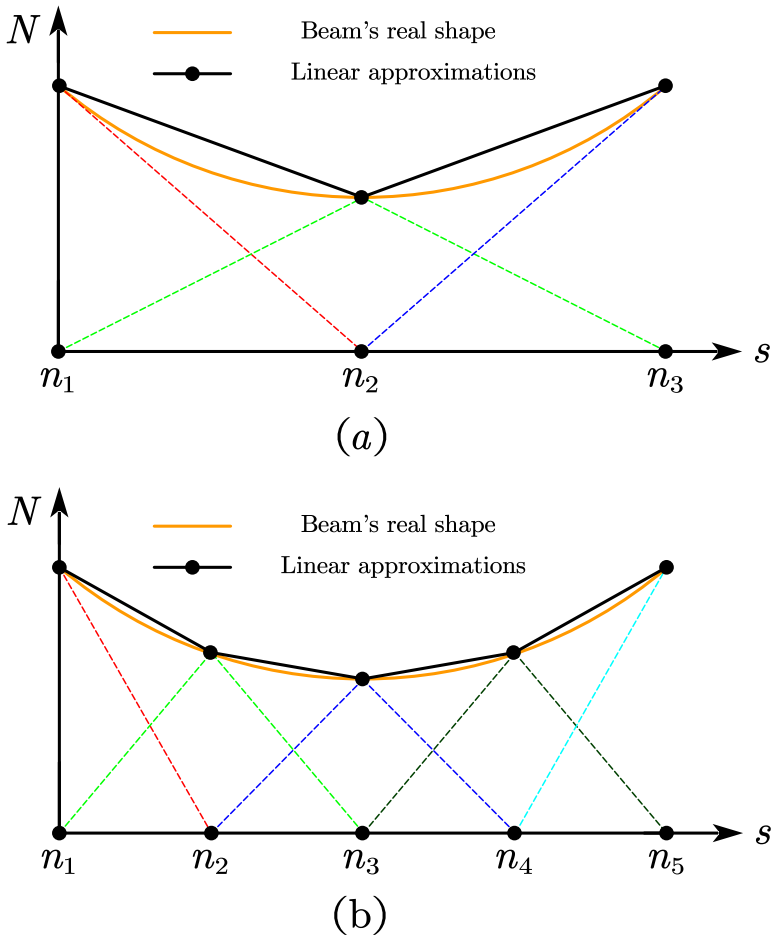


Fig. 3.10. FEM approximation. (a) 3 nodes. (b) 5 nodes

Approximation results are more accurate with more FEM nodes adopted even if the basis functions are linear. Linear basis functions do not conflict with geometric nonlinearity in the model. As stated in Section 2.2, nonlinear strain-displacement relations should be consider in large deformation scenario for modelling deformable body analytically. The FEM adopted here aims to analyze the deformed shapes as well but numerically. These are two approaches to describe the evolution of the deformable body with the same purpose, and errors generated by the numerical solution can be minimized with increasing FEM nodes adopted. Nonlinear

functions like piecewise quadratic shape functions can also be adopted as FEM basis functions; however, it would significantly increase computational workload when we solve the inverse problem since each iterative step requires a FEM solution (details in Section 4.2). To balance the accuracy of the FEM solutions and computational efficiency, linear basis functions are adopted to describe deformation of the beam and substrate.

Nonlinear terms in equation (3.51) cannot be integrated analytically. Therefore, we adopt numerical integration solutions. Gaussian quadrature, named after Carl Friedrich Gauss, is a numerical integration method that approximate an integral with weighted sum. The form of Gaussian quadrature to approximate a scalar integral is

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) = w_1 f(x_1) + w_2 f(x_2) + \cdots + w_n f(x_n) \tag{3.54}$$

Some quadrature rules over the standard interval $[-1,1]$ are listed below

Table 3.1. Number of quadrature points, point coordinates and weights

Number of quadrature points, n	Point coordinates, x_i	Weights, w_i
1	0	2
2	±0.57735	1
3	0	0.888889
	±0.774597	0.555556
4	±0.861136	0.347855
	±0.339981	0.652145

Other intervals $[a,b]$ can be mapped to $[-1,1]$ as

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t+\frac{a+b}{2}\right)dt \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}t+\frac{a+b}{2}\right) \tag{3.55}$$

Here is an example of how Gaussian quadrature is used to numerically integrate the nonlinear operators in governing equations (3.51). consider a one-dimensional domain discretized into two elements with three FEM nodes and in each subdomain, and we place two quadrature points on each element, see Fig. 3.11.

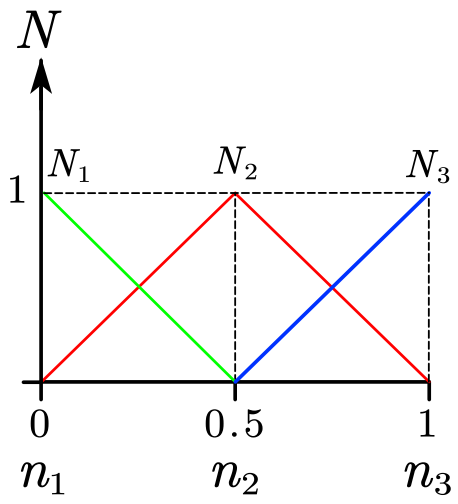


Fig. 3.11. Beam with 3 FEM nodes

For basis function N_1, N_2 and N_3, coordinates of the quadrature points and corresponding weights are listed in Table 3.2. Function value with respect to different quadrature points are evaluated in Table 3.3.

Table 3.2. Coordinates of the quadrature points and corresponding weights for basis functions

	Coordinates x_i	Weights w_i
N_1	0.105662	0.25
	0.394338	0.25
N_2	0.211325	0.5
	0.788675	0.5
N_3	0.605662	0.25
	0.894338	0.25

Table 3.3. Basis function value with respect to different quadrature points

Function value $N(x_i)$ / Coordinates x_i	$N_1(x_i)$	$N_2(x_i)$	$N_3(x_i)$
0.105662	0.788675	0.211325	0
0.394338	0.211325	0.788675	0
0.211325	0.57735	0.42265	0
0.788675	0	0.42265	0.57735
0.605662	0	0.788675	0.211325
0.894338	0	0.211325	0.788675

The nonlinear term $\boldsymbol{K}_1 = \int_0^1 \bar{\boldsymbol{u}}_1{}'(\alpha_1 \varepsilon(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}) cos\theta(\boldsymbol{c}) - \gamma(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}) \sin\theta(\boldsymbol{c}))ds$ in equation

(3.42) can be expanded as

$$\boldsymbol{K}_1 = \int_0^1 \bar{\boldsymbol{u}}_1'(\alpha_1 \varepsilon(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})cos\theta(\boldsymbol{c}) - \gamma(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\sin\theta(\boldsymbol{c}))ds$$

$$= \int_0^1 \bar{\boldsymbol{u}}_1'[\alpha_1((\boldsymbol{u}_1'+1)\cos\theta + \boldsymbol{u}_2'\sin\theta - 1)cos\theta - (\boldsymbol{u}_2'\cos\theta - (\boldsymbol{u}_1'+1)\sin\theta)\sin\theta]ds$$

$$= \int_0^1 \bar{\boldsymbol{u}}_1'\,[\alpha_1((\boldsymbol{u}_1'+1)\cos\theta + \boldsymbol{u}_2'\sin\theta)\,]ds - \int_0^1 \bar{\boldsymbol{u}}_1'\alpha_1 cos\theta ds$$

$$- \int_0^1 \bar{\boldsymbol{u}}_1'\,[\,(\boldsymbol{u}_2'\cos\theta - (\boldsymbol{u}_1'+1)\sin\theta)\sin\theta]ds \tag{3.56}$$

The simple nonlinear term

$$\int_0^1 \bar{\boldsymbol{u}}_1'\alpha_1 cos\theta ds \tag{3.57}$$

in equation (3.56) is used to demonstrate the calculations. Considering three FEM nodes, (3.57) can be expanded as

$$\int_0^1 \bar{\boldsymbol{u}}_1'\alpha_1 cos\theta ds = \alpha_1 \int_0^1 \begin{bmatrix} u_{1_1}' \\ u_{1_2}' \\ u_{1_3}' \end{bmatrix} cos[\theta_1 c_1 + \theta_2 c_2 + \theta_3 c_3]ds \tag{3.58}$$

where u_{1_j} and θ_j are j-th FEM basis functions in equation (3.38), and the time dependent variable c_1, c_2 and c_3 are unknow to this term. Therefore, (3.57) can be numerical integrated through the Gaussian quadrature rule

$$\int_0^1 \bar{\boldsymbol{u}}_1{}'\alpha_1 cos\theta ds = \alpha_1 \int_0^1 \begin{bmatrix} u_{1_1}{}' \\ u_{1_2}{}' \\ u_{1_3}{}' \end{bmatrix} cos[\theta_1 c_1 + \theta_2 c_2 + \theta_3 c_3] ds$$

$$= \begin{bmatrix} \alpha_1(w(0.11)u_{1_1}{}'(0.11)\cos[\theta_1(0.11)c_1 + \theta_2(0.11)c_2] + w(0.39)u_{1_1}{}'(0.39)\cos[\theta_1(0.39)c_1 + \theta_2(0.39)c_2]) \\ \alpha_1(w(0.21)u_{1_2}{}'(0.21)\cos[\theta_1(0.21)c_1 + \theta_2(0.21)c_2] + w(0.78)u_{1_2}{}'(0.78)\cos[\theta_2(0.78)c_2 + \theta_3(0.78)c_3]) \\ \alpha_1(w(0.61)u_{1_3}{}'(0.61)\cos[\theta_2(0.61)c_2 + \theta_3(0.61)c_3] + w(0.89)u_{1_3}{}'(0.89)\cos[\theta_2(0.89)c_2 + \theta_3(0.89)c_3]) \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_1(0.25\times(-2)\times\cos(0.79c_1 + 0.21c_2) + 0.25\times(-2)\times\cos(0.21c_1 + 0.79c_2)) \\ \alpha_1(0.5\times 2\times\cos(0.58c_1 + 0.42c_2) + 0.5\times(-2)\times\cos(0.42c_2 + 0.58c_3)) \\ \alpha_1(0.25\times 2\times\cos(0.79c_2 + 0.21c_3) + 0.25\times 2\times\cos(0.21c_2 + 0.79c_3)) \end{bmatrix}$$

(3.59)

Similarly, other terms in equation (3.56) can be evaluated, and other matrices in the governing equation (3.51) can be obtained through the same method.

To increase the accuracy of Gaussian quadrature results, integrations in equations (3.39) – (3.46) are performed by using Gauss quadrature rule by placing 10 quadrature points in each element. Computational software Mathematica is used to model the FEM basis functions and to calculate Gaussian quadrature. Mathematica codes are given in Appendix A.

CHAPTER 4
SIMULATION OF THE SENSOR MODEL

In this chapter, we present the simulation results of both the direct problem and the inverse problem. In Section 4.1, by solving the governing equation (3.51), solutions of the direct problem are given and discussed. Results for the direct problem are used to generate simulated measurement data. In Section 4.2, we introduce the workflow for the inverse problem, the iterative process of the optimization cost function, the optimization method and the numerical results for the simulated sensor.

4.1 Data Generation: Solutions of The Direct Problem

In this section, we address the direct problem by solving the governing equation (3.51) with a given set of parameters and forcing terms. Nondimensional physical and geometrical properties of the system used in simulations are listed in Table 5.1. The variables are defined in (4.34) - (4.36) and their physical meaning is discussed there.

Table 4.1. Properties of the system

Variable	α_1	α_2	α_3	α_{cu}	$\alpha_{c\theta}$	h	L	α_4	α_5	α_6	α_c
Dimensionless value	100	1/1000	5	10	10	0.1	1	2	0.5	0.5	10

We consider a shear deformable beam of length $L=1$, cross section's thickness $h=0.1$ and bending stiffness ten times smaller than shear stiffness ($\alpha_1\alpha_2=\frac{1}{10}$). The coupling layer's linear stiffness is five times greater than beam's shear stiffness ($a_3=5$), which means the coupling layer is relatively compressible with respect to the beam's stiffness. A parametric study on α_3 will be presented in the next section. α_4 representing the substrate's linear stiffness is twice the beam's shear stiffness and is four times greater than the nonlinear stiffness of the substrate α_5. By assuming $\alpha_4=2$ and $\alpha_5=0.5$, we describe a substrate that has a non-negligible hardening effect.

For the given parameters, we simulate the model with different force inputs. For the first case, we assume the input forces in (3.47 – 3.49) are $Q_1=-Q_0=Q$, while all the other forcing terms are zero. Therefore, the boundary terms (3.46 - 3.48) reduce to

$$\boldsymbol{F}_{b1}=-Q(\bar{\boldsymbol{u}}_1(1)\sin\theta(1)+\bar{\boldsymbol{u}}_1(0)\sin\theta(0)) \tag{4.1}$$

$$\boldsymbol{F}_{b2}=Q(\bar{\boldsymbol{u}}_2(1)\cos\theta(1)+\bar{\boldsymbol{u}}_2(0)\cos\theta(0)) \tag{4.2}$$

$$\boldsymbol{F}_{b3}=0 \tag{4.3}$$

The governing equation (3.51) is solved by considering boundary terms (4.1 - 4.3) and by using Finite Element Method (FEM). Sensor beam partitioned into finite elements denoted by Ω_i $(i=1,2,...,n-1)$ is schematized in Fig. 4.1.

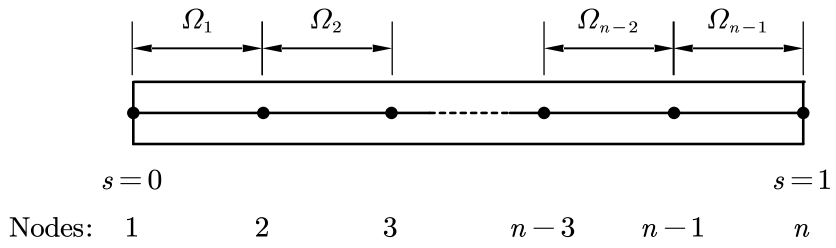


Fig. 4.1. Finite element partition of the sensor beam

As mentioned in Section 3.9, 10 quadrature points are considered within each element. Different numbers of finite element nodes n are used to simulate the deformation induced by this force input.

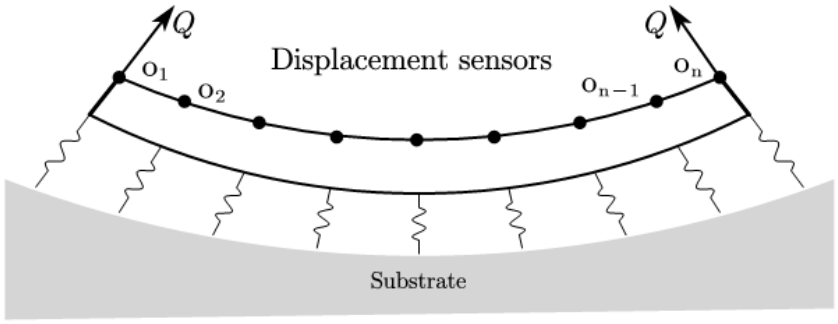


Fig. 4.2. Sensor system with sets of observation points with input Q

Results in Figs 4.3, 4.4 and 4.5 show the time histories of displacements u_1, u_2 and rotation θ at different points ($s=0, s=L/2=0.5, s=L=1$) along the beam's span with input $Q=0.5$ up to nondimensional time $t=100$. These three points are chosen due to the symmetry of the load and of the geometry, which makes the three points representative to sample the vertical and horizontal displacements and the cross section rotation field. The three

points coincide with FEM nodes where the numerical solutions are obtained; therefore with limited observation points and with limited FEM nodes, deformation of the beam can be analyzed with relatively high accuracy by adopting these three points.

For input Q in Fig. 4.2, the displacements $u_1(0.5,t), \theta(0.5,t)$ are 0. These curves will be used as simulated sensory data, that in the hardware implementation of the sensor would be generated by appropriate sensing devices, for example strain sensors, placed at specific points along the beam's span. These points are referred to as observation points in this thesis (see Fig. 4.2), to signify that are the points where sensors would have to be placed to generate displacement data. The number of observation points is not related to the number of nodes used to simulate the data with finite elements, and in general it is desirable to have a low number of observation points, since this would correspond to a robust sensor operation without requiring many measurements. The details of the inverse problem will be presented in the next section.

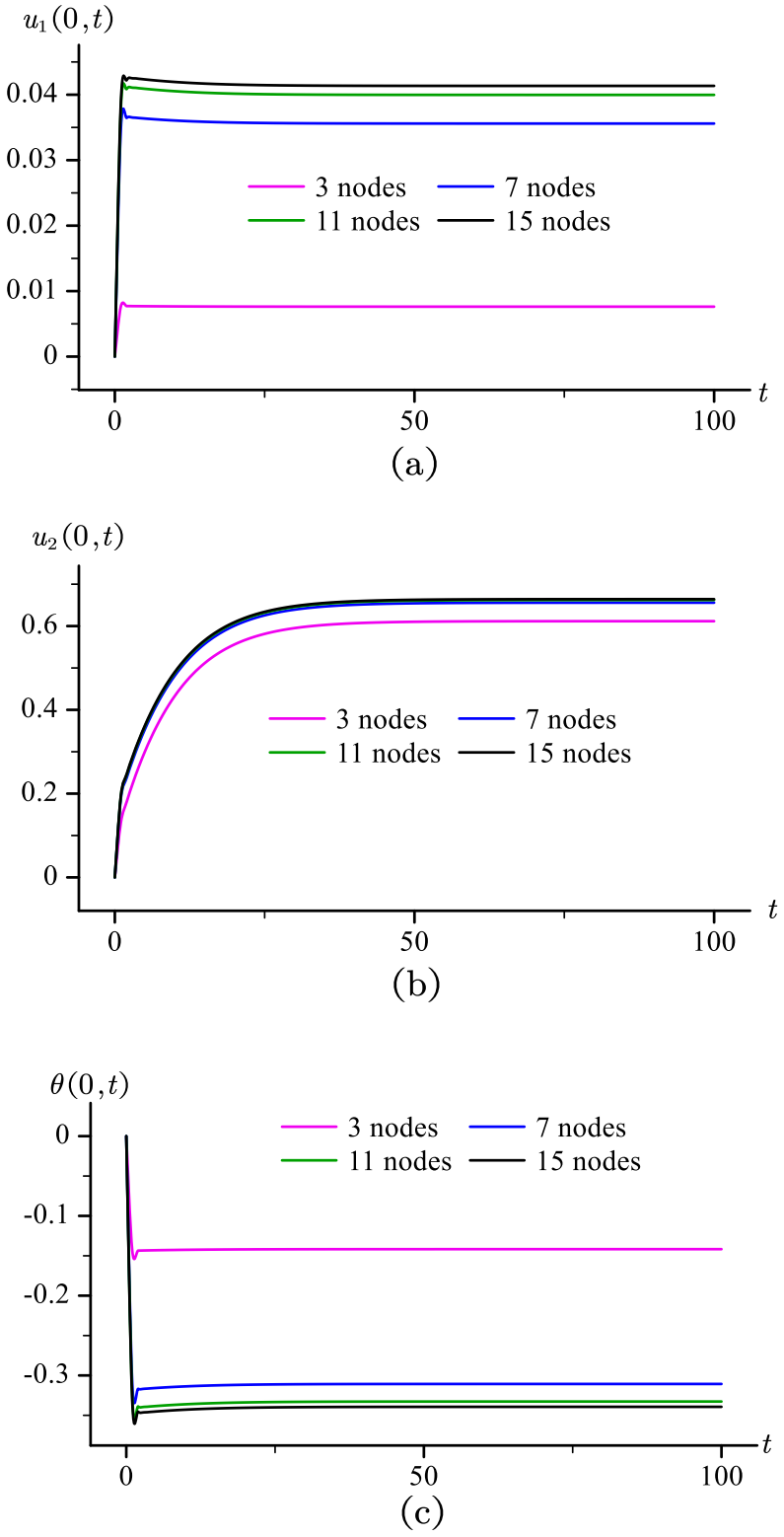


Fig. 4.3. Time histories of (a). $u_1(0,t)$, (b). $u_2(0,t)$ and (c). $\theta(0,t)$, for $Q = 0.5$

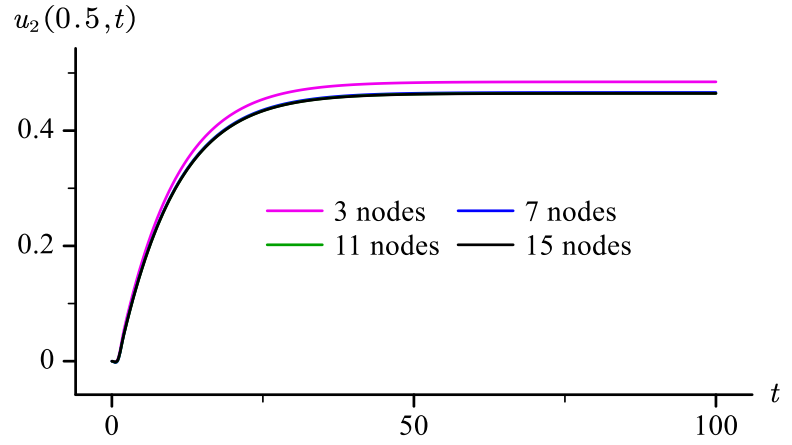
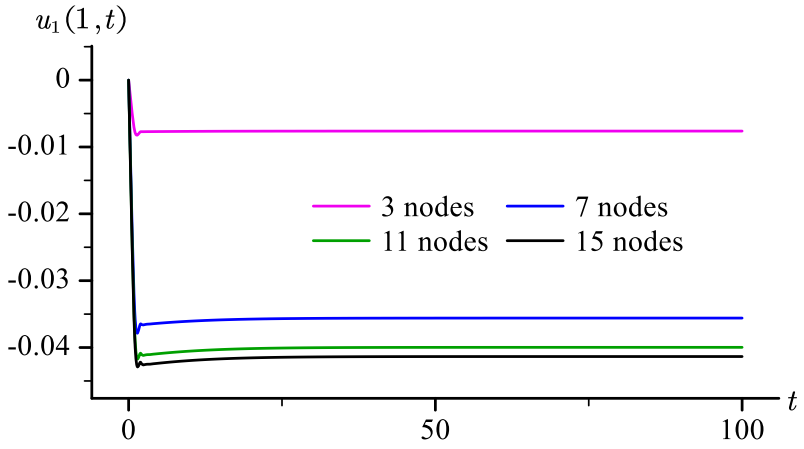
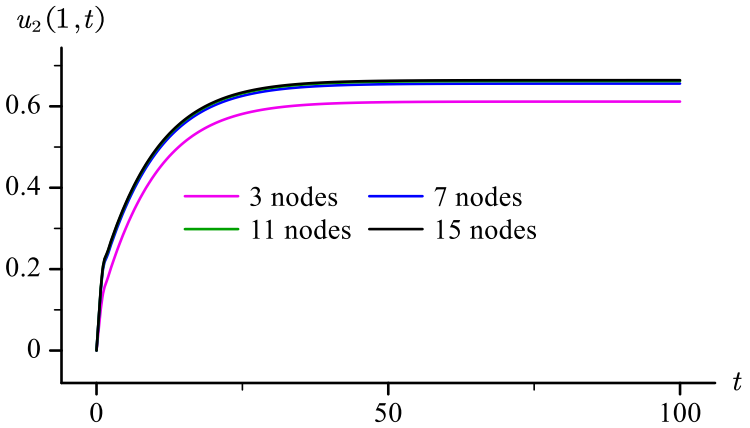


Fig. 4.4. Time histories of $u_2(0.5,t)$, for $Q = 0.5$



(a)



(b)

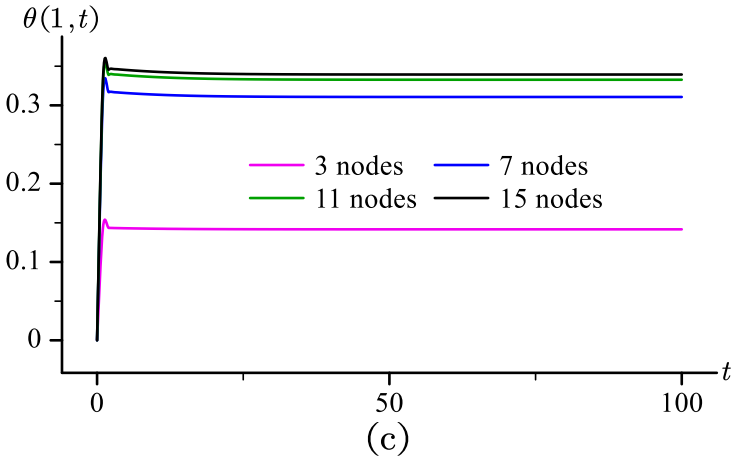


Fig. 4.5. Time histories of (a). $u_1(1,t)$, (b). $u_2(1,t)$ and (c). $\theta(1,t)$, for $Q = 0.5$

Table 4.2 lists deformation results with different number of nodes at the three observation points, when input $Q = 0.5$ and at the nondimensional time $t = 100$. Beam deformed shapes are plotted in Fig. 4.6.

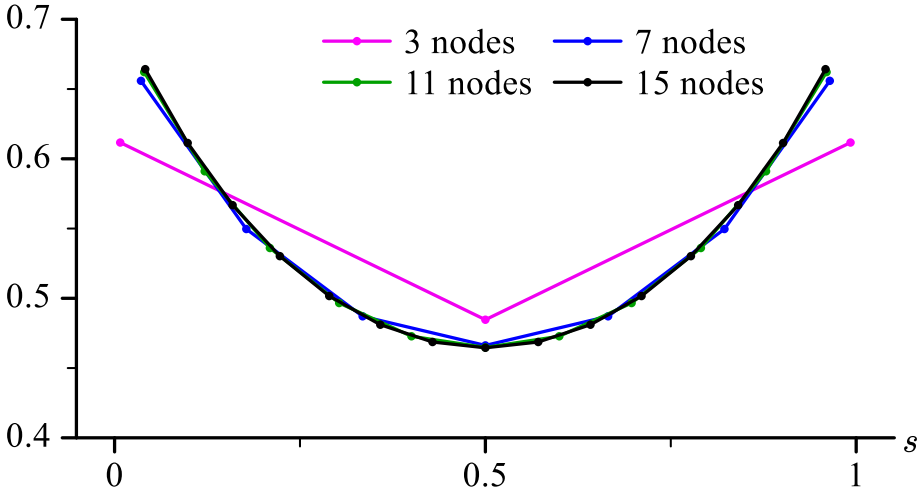


Fig. 4.6. Beam deformed shapes at $t = 100$, for $Q = 0.5$

Table 4.2 (a). Deformation results with different number of nodes, $s = 0$

Number of nodes	3	5	7	9	11	13	15
$u_1(0, 100)$	0.008	0.029	0.036	0.039	0.04	0.04	0.04
$u_2(0, 100)$	0.612	0.647	0.656	0.659	0.662	0.663	0.664
$\theta(0, 100)$	-0.142	-0.274	-0.311	-0.325	-0.333	-0.337	-0.339

Table 4.2 (b). Deformation results with different number of nodes, $s = L/2$

Number of nodes	3	5	7	9	11	13	15
$u_2(0.5, 100)$	0.485	0.469	0.466	0.465	0.465	0.465	0.465

Table 4.2 (c). Deformation results with different number of nodes, $s = L$

Number of nodes	3	5	7	9	11	13	15
$u_1(1, 100)$	-0.008	-0.029	-0.036	-0.039	-0.04	-0.04	-0.04
$u_2(1, 100)$	0.612	0.647	0.656	0.659	0.662	0.663	0.664
$\theta(1, 100)$	0.142	0.274	0.311	0.325	0.333	0.337	0.339

To investigate the relation between the results' accuracy and the number of nodes, we consider the percentage error ϵ

$$\epsilon = \left|1 - \frac{D_i(u_1, u_2, \theta)}{D_j(u_1, u_2, \theta)}\right| \times 100\% \tag{4.4}$$

where D is the displacements at a given observation point and the subscript represents the

number of nodes adopted, which means $i,j=3,5,7,9,11,13,15$. For example, at the observation point $s=0$ and by considering $u_1(0,100)$, the percentage error from using 3 nodes with respect to 15 nodes is given by

$$\epsilon(u_1)=\left|1-\frac{D_3(u_1)}{D_{15}(u_1)}\right|\times 100\%=\left|1-\frac{0.008}{0.04}\right|\times 100\%=80\% \tag{4.5}$$

Table 4.3 (a). Percentage error with respect to $j=15,\ s=0$

Number of nodes i	3	5	7	9	11	13	15
$\epsilon(u_1)$	80.00%	27.50%	10.00%	2.50%	0	0	0
$\epsilon(u_2)$	7.83%	2.56%	1.20%	0.75%	0.30%	0.15%	0
$\epsilon(\theta)$	58.11%	19.17%	8.26%	4.13%	1.77%	0.59%	0

Table 4.3 (b). Percentage error with respect to $j=15,\ s=L/2$

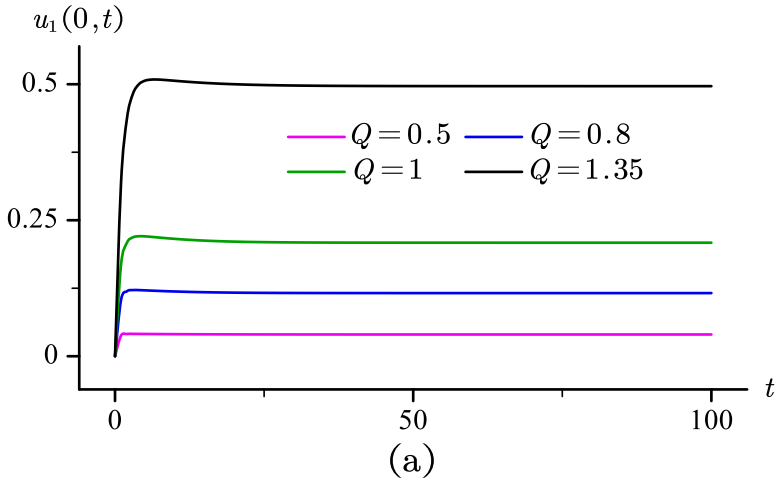
Number of nodes i	3	5	7	9	11	13	15
$\epsilon(u_1)$	4.30%	0.86%	0.22%	0	0	0	0

Table 4.3 (c). Percentage error with respect to $j=15,\ s=L$

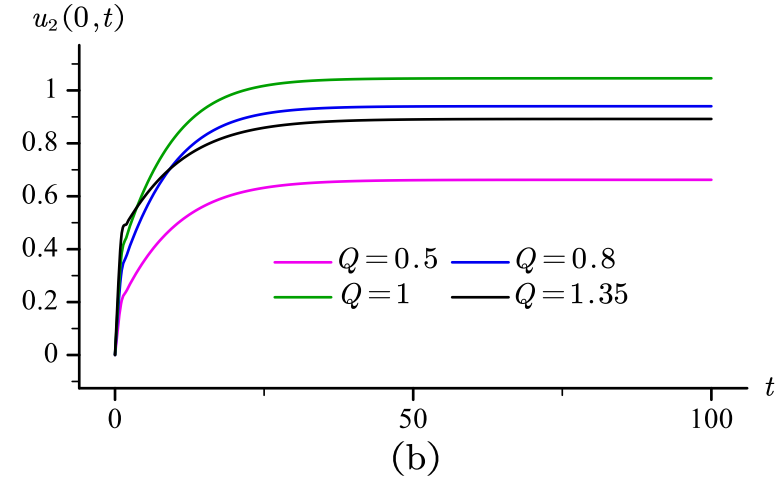
Number of nodes i	3	5	7	9	11	13	15
$\epsilon(u_1)$	80.00%	27.50%	10.00%	2.50%	0	0	0
$\epsilon(u_2)$	7.83%	2.56%	1.20%	0.75%	0.30%	0.15%	0
$\epsilon(\theta)$	58.11%	19.17%	8.26%	4.13%	1.77%	0.59%	0

As shown by the parametric study summarized in Table 4.3, for the three variables of interest (u_1, u_2 and θ), we obtain converged results for $j = 11$. Since there is no evident gain in using more nodes, we adopt 11 nodes for this set of simulations.

Time histories of u_1, u_2, θ at three different observation points ($s = 0, s = 0.5, s = 1$) by adopting 11 nodes with different input Q are plotted in Figs. 4.7 - 4.9, and corresponding deformed shapes of the beam and the substrate at $t = 100$ are shown in Fig. 4.10.



(a)



(b)

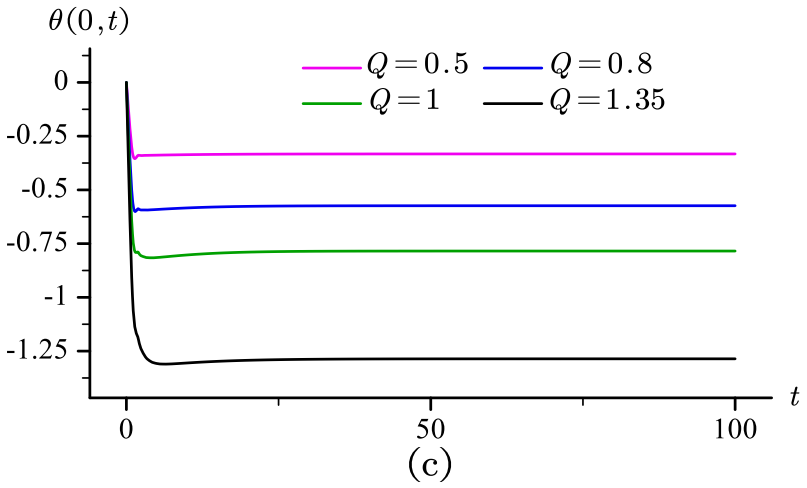


Fig. 4.7. Time histories of (a). $u_1(0,t)$, (b). $u_2(0,t)$ and (c). $\theta(0,t)$, for different inputs Q

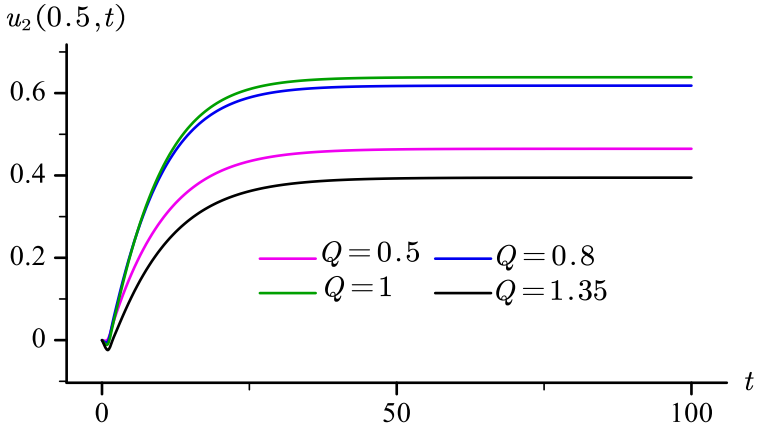
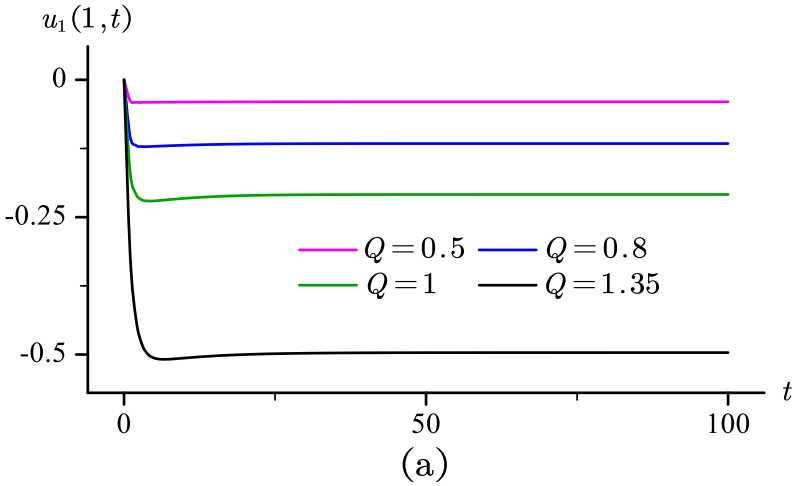


Fig. 4.8. Time histories of $u_2(0.5,t)$, for different inputs Q



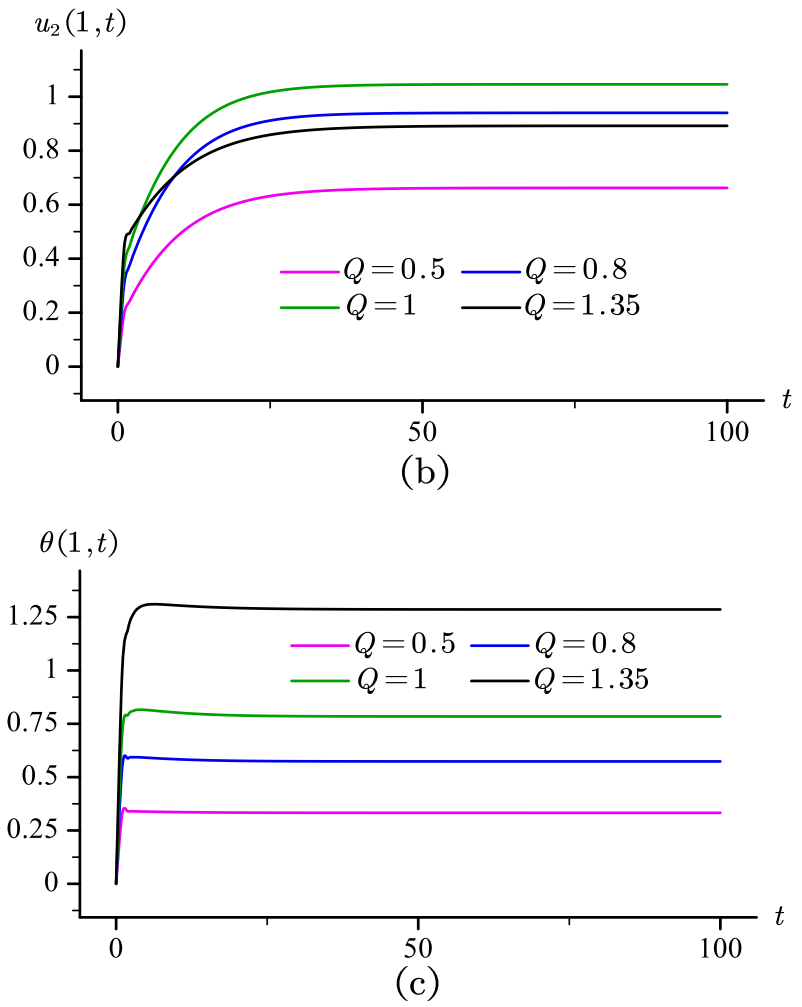


Fig. 4.9. Time histories of (a). $u_1(1,t)$, (b). $u_2(1,t)$ and (c). $\theta(1,t)$, for different inputs Q

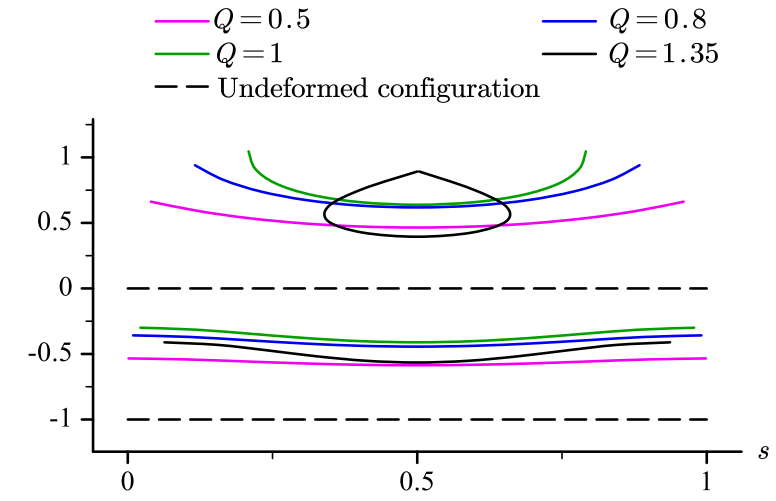


Fig. 4.10. Beam and substrate deformed shapes at $t = 100$, for different inputs Q

Larger deformations of the beam and the substrate results from the increasing force. When the input force $Q > 1$, the beam's extremities start to roll up. As shown in Fig. 4.10, when the input force reaches $Q = 1.35$, the two ends of the beam intersect at $s = 0.5$. Therefore, $Q = 1.35$ is considered as the maximum input for the beam undertaking pure shear force for the set of parameters adopted in Table 4.1.

A second data set is generated by considering a different input, namely bending moment at the boundary, see Fig. 4.11. This means that only $\boldsymbol{F}_{b3}$ exists in equation (3.49), while all other boundary terms in equation (3.47) and (3.48) are zero. By adopting $M_0 = M_1 = M$, equation (3.49) reduces to

$$\boldsymbol{F}_{b3} = M\left(\overline{\boldsymbol{\theta}}(1) - \overline{\boldsymbol{\theta}}(0)\right) \tag{4.6}$$

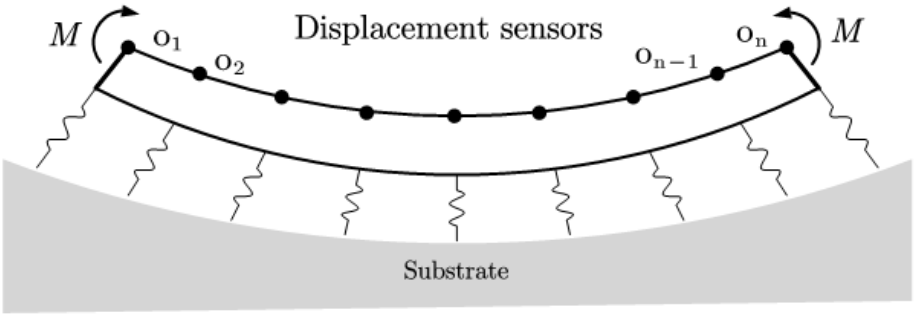


Fig. 4.11. Sensor system with sets of observation points with input M

By employing 11 nodes and nondimensional time $t = 100$, simulation results from different inputs M are presented in Figs. 4.12 - 4.15.

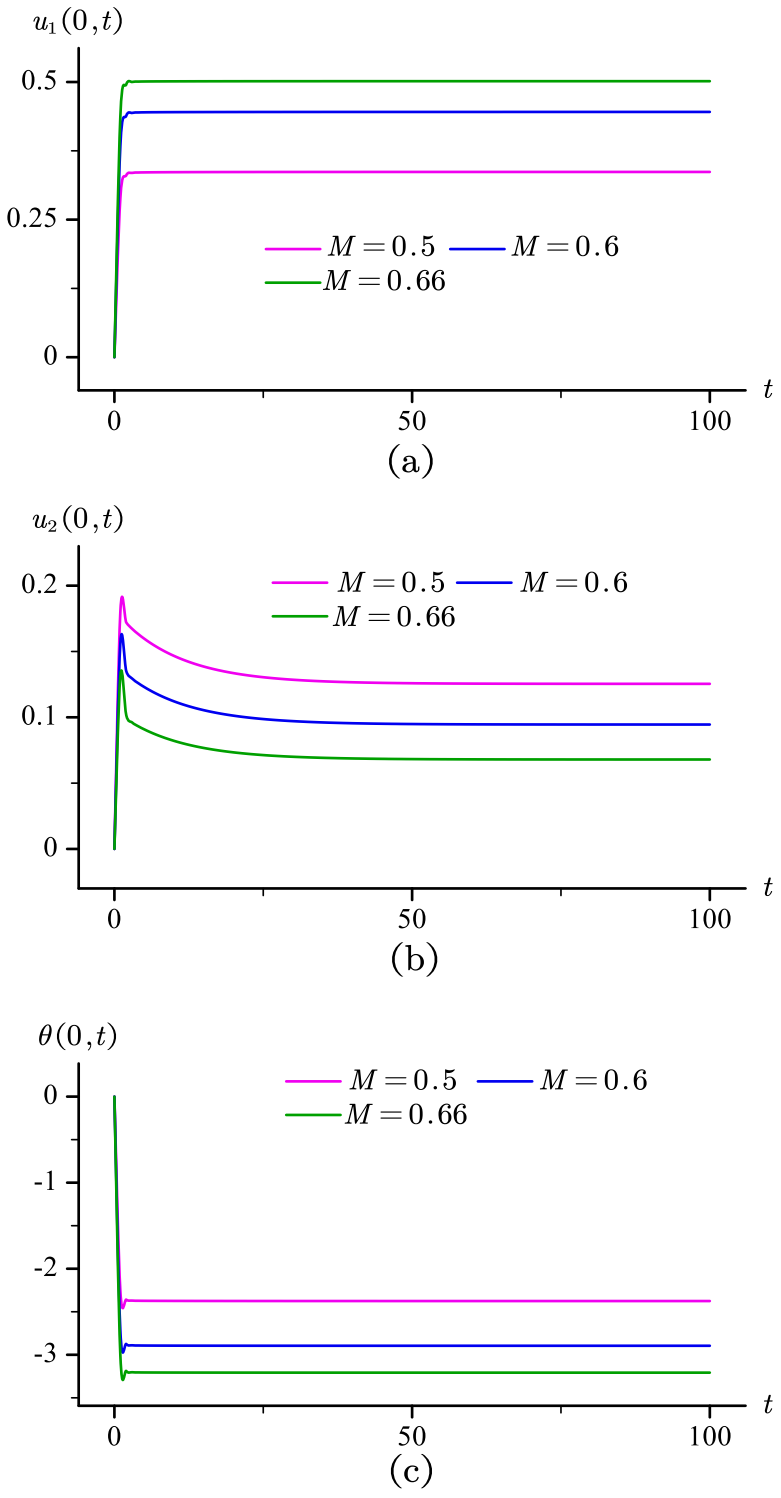


Fig. 4.12. Time histories of (a). $u_1(0,t)$, (b). $u_2(0,t)$ and (c). $\theta(0,t)$, for different inputs M

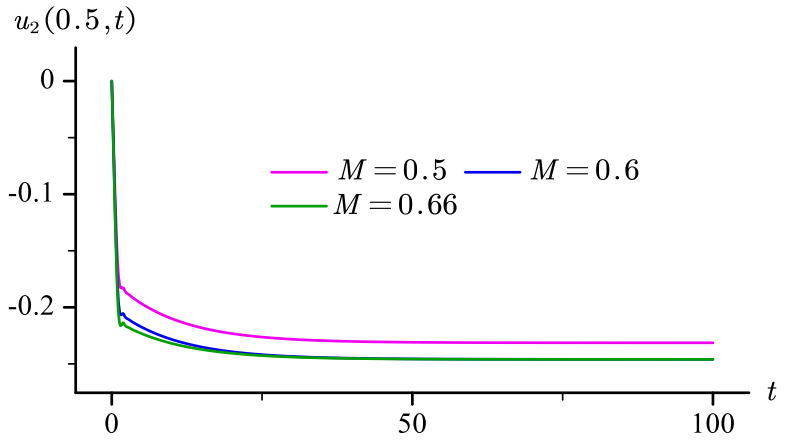
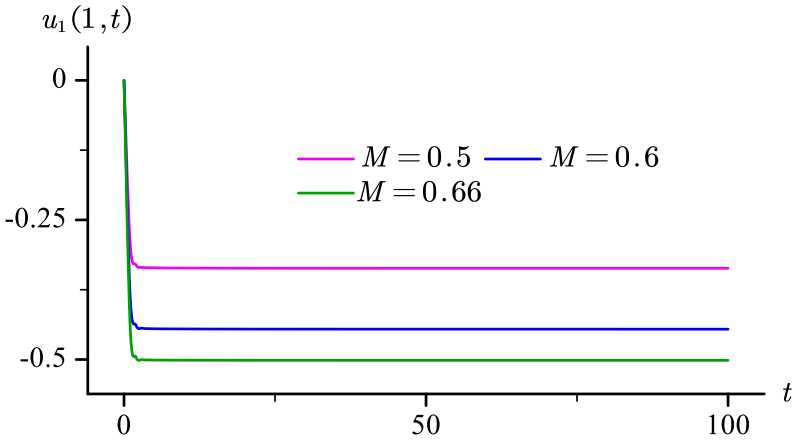
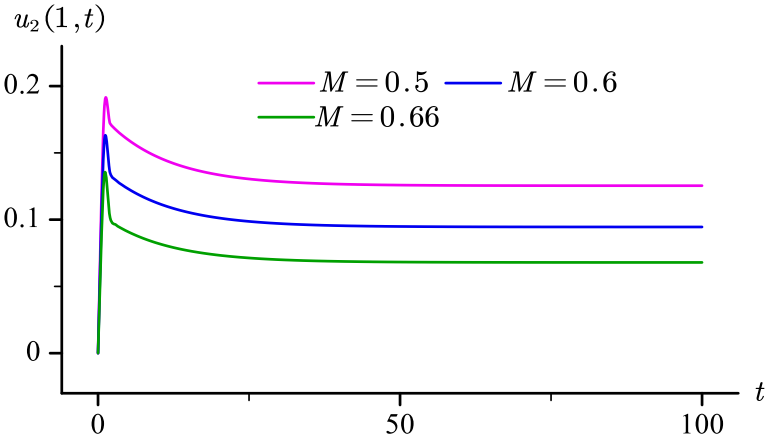


Fig. 4.13. Time histories of $u_2(0.5,t)$, for different inputs M



(a)



(b)

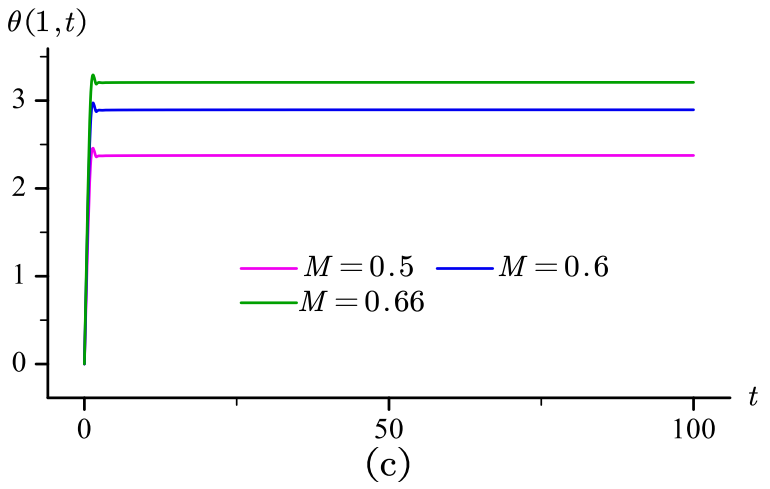


Fig. 4.14. Time histories of (a). $u_1(1,t)$, (b). $u_2(1,t)$ and (c). $\theta(1,t)$, for different inputs M

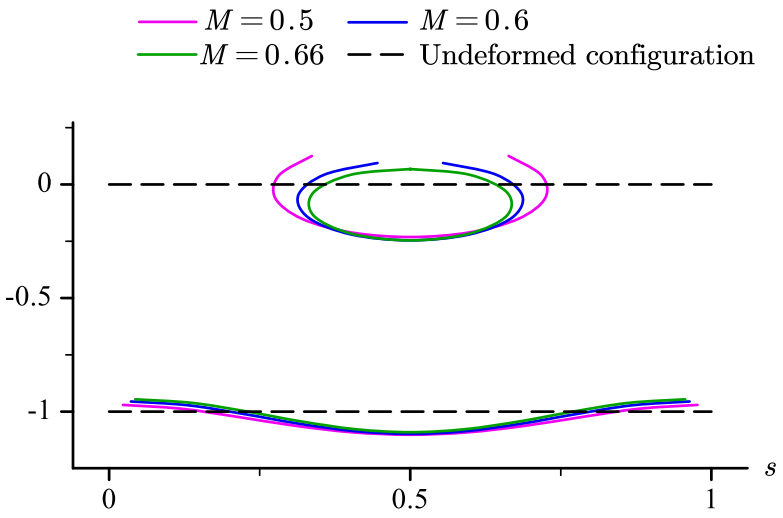


Fig. 4.15. Beam and substrate deformed shapes at $t = 100$, for different input M

When $M = 0.66$, the beam's two ends intersect at $s = 0.5$. Therefore, when the beam undertakes pure bending moment, nondimensional input M should not be greater than 0.66 for the set of parameters adopted here.

4.2 The Inverse Problem

In this section, we present an algorithm to estimate the material parameters of the substrate, and in Section 4.2.4, we use the data sets from Section 4.1 to simulate the sensor system.

4.2.1 Sensor Workflow

General methods for solving inverse problems are presented in [54]. Among different methods, the iterative least square has the advantage of being computationally easy [54]. The general structure of the inverse problem considered in this thesis is schematized in Fig. 4.16, where the least squares residual's cost function is built from measurements, and material parameters are selected as minimizers of the cost function

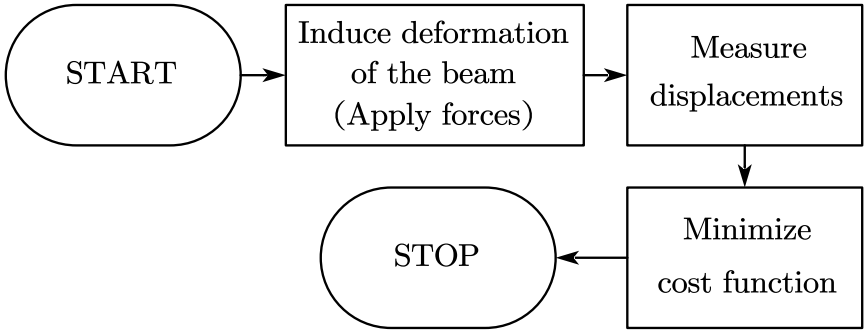


Fig. 4.16. General flowchart of the inverse problem

As illustrated in Fig. 4.2, A series of observation points $o_1,\ o_2,\ \cdots,\ o_n$ are distributed along the beam's body. Displacements are measured at these observation points.

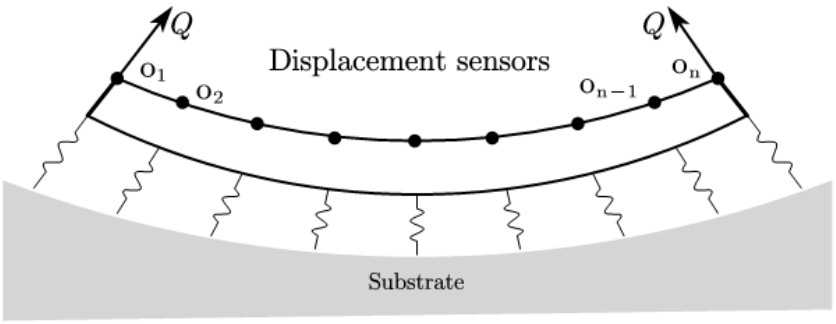


Fig. 4.2. Schematic for the sensor system with sets of observation points with input Q

Considering the time history of deformation fields, the measured displacements at time t_j and abscissa s are denoted by $\tilde{u}_1(s,t_j)$, $\tilde{u}_2(s,t_j)$ and $\tilde{\theta}(s,t_j)$.

The least square residual's cost function is defined as

$$\mathscr{L} = \frac{1}{2}\sum_{j=1}^{n}\left(\hat{u}_1(s,t_j) + \hat{u}_2(s,t_j) + \hat{\theta}(s,t_j)\right) \tag{4.7}$$

where

$$\hat{u}_1(s,t_j) = \left(\tilde{u}_1(s,t_j) - u_1^g(s,t_j)\right)^T\left(\tilde{u}_1(s,t_j) - u_1^g(s,t_j)\right) \tag{4.8a}$$

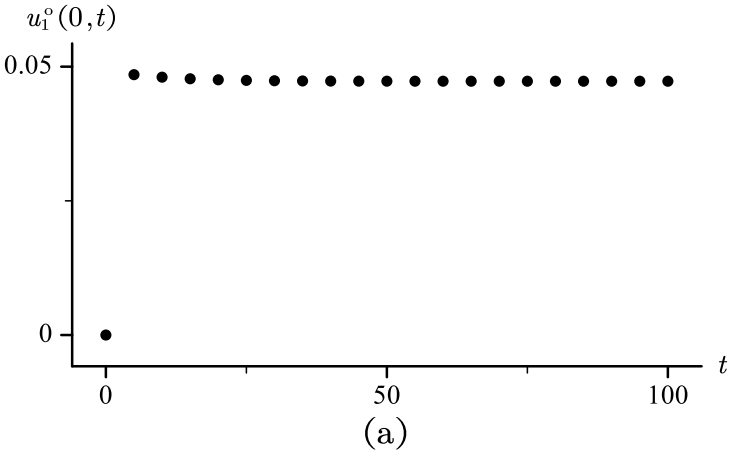
$$\hat{u}_2(s,t_j) = \left(\tilde{u}_2^s(s,t_j) - u_2^g(s,t_j)\right)^T\left(\tilde{u}_2^s(s,t_j) - u_2^g(s,t_j)\right) \tag{4.8b}$$

$$\hat{\theta}(s,t_j) = \left(\tilde{\theta}(s,t_j) - \theta^g(s,t_j)\right)^T\left(\tilde{\theta}(s,t_j) - \theta^g(s,t_j)\right) \tag{4.8c}$$

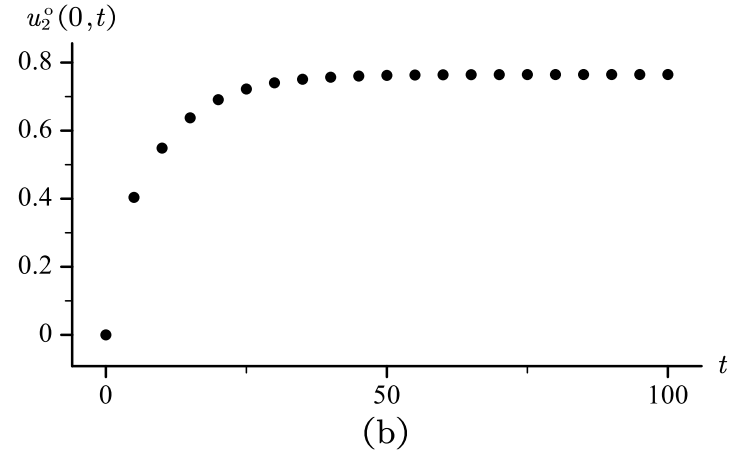
and u_1^g, u_2^g and θ^g are obtained from the solution of (3.51) when a set of estimated parameters $g = (\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4)$ are given.

4.2.2 Iterative Process for Root Search

Simulated displacement measurements are obtained from finite element solutions (11 nodes) of the direct problem presented in Section 4.1, sampled every 5 nondimensional time units and plotted again in Figs. 4.17 – 4.19. Simulated measurements at observation points are denoted with a superscript "o".



(a)



(b)

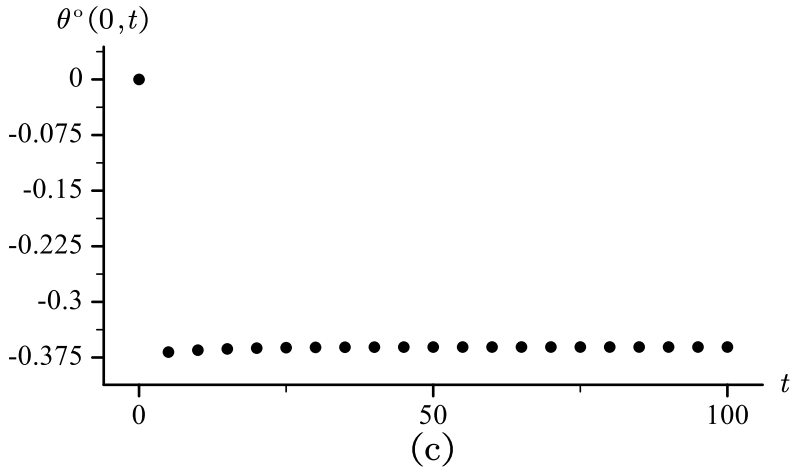


Fig. 4.17. Simulated measurements at $s = 0$, (a). $u_1^o(0,t)$, (b). $u_2^o(0,t)$ and (c). $\theta^o(0,t)$

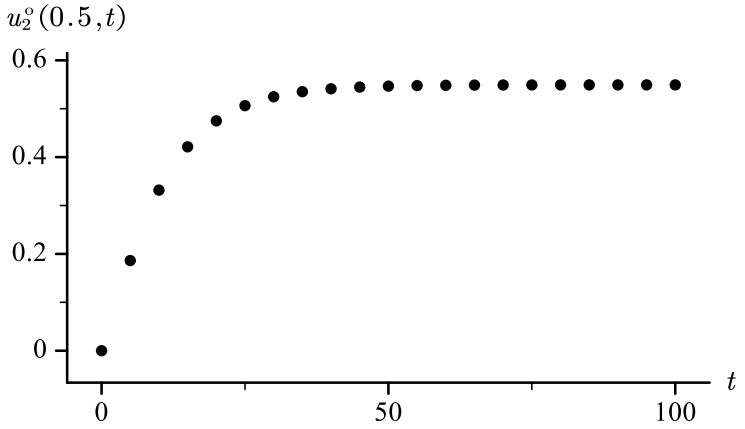
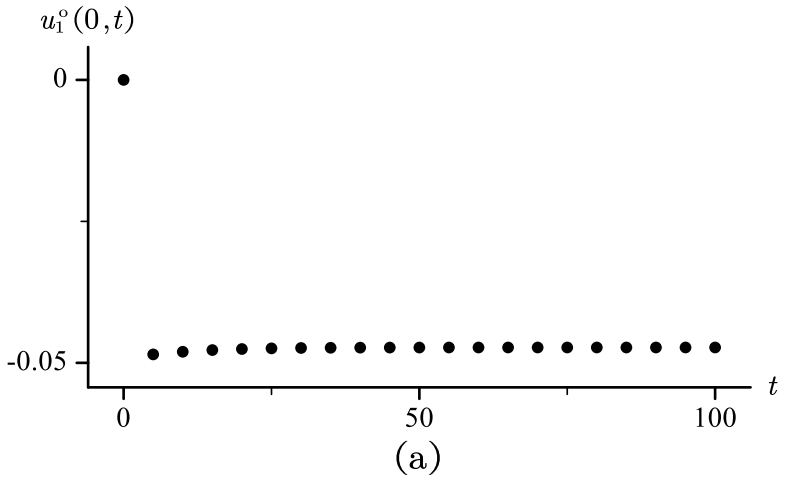
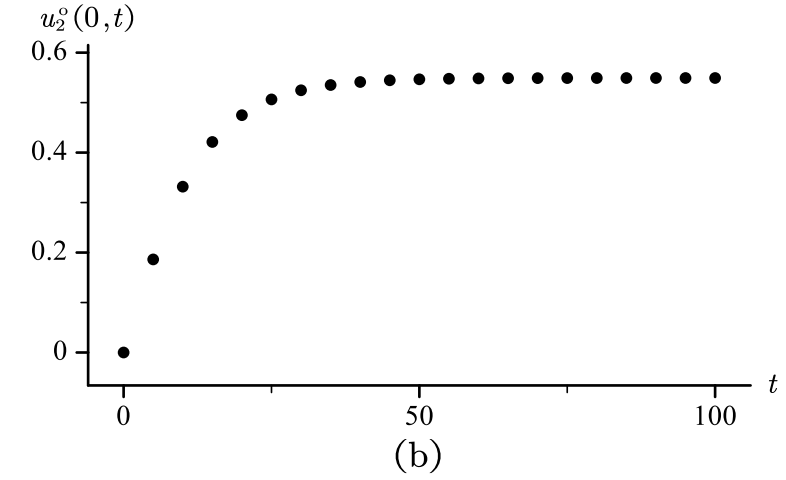


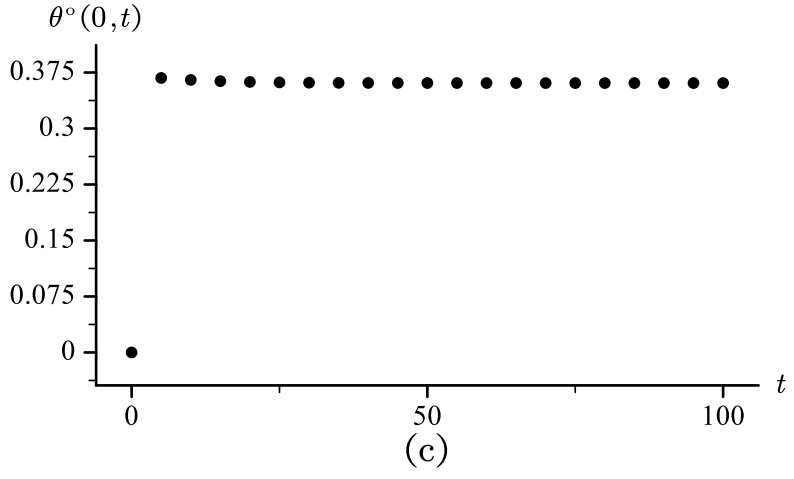
Fig. 4.18. Simulated measurements of $u_2^o(0.5,t)$ at $s = 0.5$



(a)



(b)



(c)

Fig. 4.19. Simulated measurements at $s = 1$, (a). $u_1^o(1,t)$, (b). $u_2^o(1,t)$ and (c). $\theta^o(1,t)$

By using the sampled simulated measurements, equation (4.7) can be rewritten as

$$\widetilde{u}_1(0,t) = (u_1^o(0,0), u_1^o(0,5) \cdots u_1^o(0,100))^T \tag{4.9a}$$

$$\widetilde{u}_1(1,t) = (u_1^o(1,0), u_1^o(1,5) \cdots u_1^o(1,100))^T \tag{4.9b}$$

$$\widetilde{u}_2(0,t) = (u_2^o(0,0), u_2^o(0,5) \cdots u_2^o(0,100))^T \tag{4.9c}$$

$$\widetilde{u}_2(0.5,t) = (u_2^o(0.5,0), u_2^o(0.5,5) \cdots u_2^o(0.5,100))^T \tag{4.9d}$$

$$\widetilde{u}_2(1,t) = (u_2^o(1,0), u_2^o(1,5) \cdots u_2^o(1,100))^T \tag{4.9e}$$

$$\widetilde{\theta}(0,t) = (\theta^o(0,0), \theta^o(0,5) \cdots \theta^o(0,100))^T \tag{4.9f}$$

$$\widetilde{\theta}(1,t) = (\theta^o(1,0), \theta^o(1,5) \cdots \theta^o(1,100))^T \tag{4.9g}$$

For a given set of trial parameters $g = (\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3, \widetilde{\alpha}_4)$, solutions u_1^g, u_2^g and θ^g from governing equation (3.51) can also be represented as

$$u_1^g(0,t) = (u_1^g(0,0), u_1^g(0,5) \cdots u_1^g(0,100))^T \tag{4.10a}$$

$$u_1^g(1,t) = (u_1^g(1,0), u_1^g(1,5) \cdots u_1^g(1,100))^T \tag{4.10b}$$

$$u_2^g(0,t) = (u_2^g(0,0), u_2^g(0,5) \cdots u_2^g(0,100))^T \tag{4.10c}$$

$$u_2^g(0.5,t) = (u_2^g(0.5,0), u_2^g(0.5,5) \cdots u_2^g(0.5,100))^T \tag{4.10d}$$

$$u_2^g(1,t) = (u_2^g(1,0), u_2^g(1,5) \cdots u_2^g(1,100))^T \tag{4.10e}$$

$$\theta^g(0,t) = (\theta^g(0,0), \theta^g(0,5) \cdots \theta^g(0,100))^T \tag{4.10f}$$

$$\theta^g(1,t) = (\theta^g(1,0), \theta^g(1,5) \cdots \theta^g(1,100))^T \tag{4.10g}$$

Therefore, equation (4.8) can be rewritten as

$$\begin{aligned} \widehat{u}_1(s,t_j) = {} & (\widetilde{u}_1(0,t_j) - u_1^g(0,t_j))^T (\widetilde{u}_1(0,t_j) - u_1^g(0,t_j)) \\ & + (\widetilde{u}_1(1,t_j) - u_1^g(1,t_j))^T (\widetilde{u}_1(1,t_j) - u_1^g(1,t_j)) \end{aligned} \tag{4.12a}$$

$$\begin{aligned}\widehat{u}_2(s,t_j) &= (\widetilde{u}_2^s(0,t_j) - u_2^g(0,t_j))^T(\widetilde{u}_2^s(0,t_j) - u_2^g(0,t_j)) \\ &+ (\widetilde{u}_2^s(0.5,t_j) - u_2^g(0.5,t_j))^T(\widetilde{u}_2^s(0.5,t_j) - u_2^g(0.5,t_j)) \\ &+ (\widetilde{u}_2^s(1,t_j) - u_2^g(1,t_j))^T(\widetilde{u}_2^s(1,t_j) - u_2^g(1,t_j))\end{aligned} \tag{4.12b}$$

$$\begin{aligned}\widehat{\theta}(s,t_j) &= \left(\widetilde{\theta}(0,t_j) - \theta^g(0,t_j)\right)^T\left(\widetilde{\theta}(0,t_j) - \theta^g(0,t_j)\right) \\ &+ \left(\widetilde{\theta}(1,t_j) - \theta^g(1,t_j)\right)^T\left(\widetilde{\theta}(1,t_j) - \theta^g(1,t_j)\right)\end{aligned} \tag{4.12c}$$

Substituting (4.12) into (4.7), we represent the cost function with the set of trial parameters g. Then we repeat the loop until convergence to a minimum of the cost function (4.8). Converged parameters g that result in the minimum cost function value are considered as the estimated parameters of the substrate. Since the cost function is quadratic in the displacements, there is a global minimum $\mathcal{L} = 0$ at $\widetilde{u}_1(s,t_j) = u_1^g(s,t_j)$, $\widetilde{u}_2(s,t_j) = u_2^g(s,t_j)$ and $\widetilde{\theta}(s,t_j) = \theta^g(s,t_j)$, corresponding to the estimated parameters $\widetilde{\alpha}_4 = \alpha_4$, $\widetilde{\alpha}_5 = \alpha_5$, $\widetilde{\alpha}_6 = \alpha_6$ and $\widetilde{\alpha}_c = \alpha_c$.

4.2.3 The Optimization Method

Optimization is the process of finding an extremum of the objective objective function. Generally, optimization problems are local or global. In local optimization problems, the objective function admits multiple extrema, and therefore a solution is optimal in a range that may depend on the search starting point and on other relevant parameters. In a global optimization problem, the objective function admits only one extremum, and therefore the solution of the optimization problem is unique and global. In a local optimization problem, it

is sometimes necessary to perform global searches to find the global optimum; this requires special techniques to classify solution points and continue the search after a local solution is found.

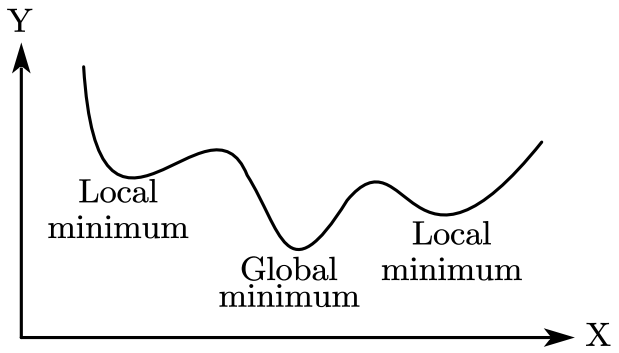


Fig. 4.20. Local minimum and global minimum

Search algorithms typically require an initial guess or a range for the unknowns. In a practical implementation scenario, this would have to be retrieved from some prior, coarse knowledge of the class of substrate materials. Ideally, we need a solver that is robust with respect to the starting points. If a local optimization is adopted, the solver can be easily trapped at local minimum unless good starting points are provided [55], which eventually leads to inaccurate estimations. Loops in Section 4.2.2 are completed by adopting the global optimization algorithm solver *NMinimize* [56] in *Mathematica* to find the global minimum of the cost function with the given constrains. Numerical simulation results are presented in the next section.

4.2.4 Numerical Simulation of the Sensor Operation

Here, we simulate the operation of the sensor system. The search boundaries of $\tilde{\alpha}$ for the search algorithm are given as in Table. 4.4.

Table 4.4. Constrains of the estimated parameters

Parameters	$\tilde{\alpha}_4$	$\tilde{\alpha}_5$	$\tilde{\alpha}_6$	$\tilde{\alpha}_c$
Constrains	$1 \leqslant \tilde{\alpha}_4 \leqslant 5$	$0.1 \leqslant \tilde{\alpha}_5 \leqslant 1$	$0.1 \leqslant \tilde{\alpha}_6 \leqslant 1$	$5 \leqslant \tilde{\alpha}_c \leqslant 15$

These bounds are not invariable, and they should be adjusted with respect to different substrate databases and the search algorithm. Table 4.5 presents the converged estimated parameters by adopting 3-node beam model and 5-node beam model in the material parameter estimation algorithm.

Table 4.5. Estimated parameters by adopting different number of nodes

	$\tilde{\alpha}_4$	$\tilde{\alpha}_5$	$\tilde{\alpha}_6$	$\tilde{\alpha}_c$	Computing time (seconds)
3-node beam	1.9996	0.5009	0.5107	10.0030	357
5-node beam	1.9982	0.5051	0.4973	10.0161	125500

Estimated nondimensional material parameters from the simulated sensor show good agreement with the ones used to simulate the measurements, reported in Table 4.1. However, adopting 5-node beam model is considerably more computationally expensive while there is

no obvious improvement in accuracy. Since the procedure to estimate the substrate material parameter is an iterative root searching with a nonlinear cost function quadratic in the measurements, each iteration requires a numerical solution (in this case a FEM solution) of the model, which iterates on the substrate parameters. Therefore the procedure is computationally expensive, which motivated the adoption of three nodes. The low accuracy FEM model is expected to be offset by the iterative nature of the root search.

To create a more realistic scenario, the algorithm is run with noisy sensory data, with the same set of inputs. Specifically, white noise sampled from Gaussian distributions is added to the simulated measurements. The measurements with noise are given by

$$\begin{aligned} U_1 &= \widetilde{u}_1^s(s,t) + \nu(0,\sigma^2) \\ U_2 &= \widetilde{u}_2^s(s,t) + \nu(0,\sigma^2) \\ \Theta &= \widetilde{\theta}^s(s,t) + \nu(0,\sigma^2) \end{aligned} \tag{4.12}$$

where ν is normally distributed with mean 0 and standard deviation σ. Here, we set the noise standard deviation σ to be a percentage of the max value of the variable.

Table 4.6 and Fig.4.21 show the estimated material parameters with different noise. Table 4.7 and Fig. 4.22 show the percentage error with respect to the noiseless measurements.

Table 4.6. Estimated material parameters with different noise percentage

Variables / Noise percentage	$\widetilde{\alpha}_4$	$\widetilde{\alpha}_5$	$\widetilde{\alpha}_6$	$\widetilde{\alpha}_c$
0	1.9996	0.5009	0.5107	10.0030
1%	2.0674	0.3329	0.5478	9.5882
2%	2.1385	0.3332	0.6243	9.0385
5%	2.2004	0.3137	0.7567	8.1194
8%	2.3583	0.3114	0.9424	6.9393

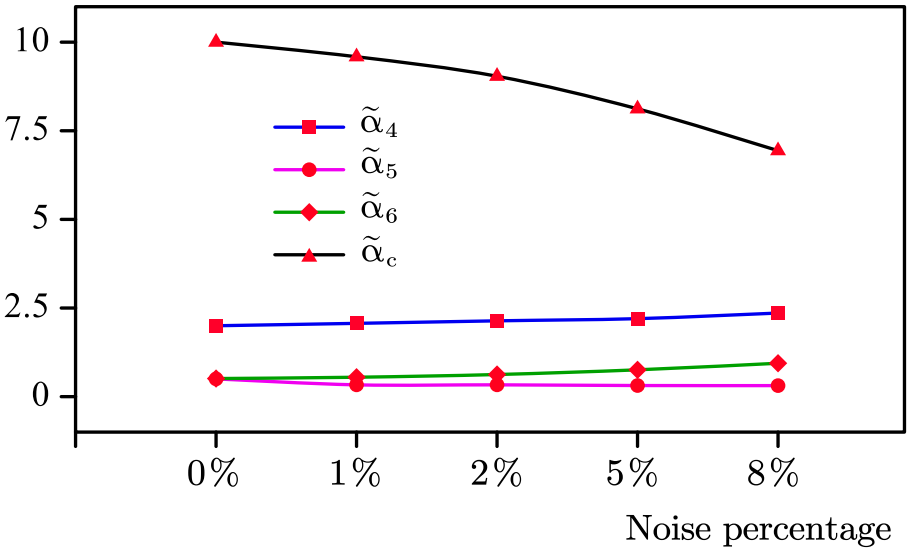


Fig. 4.21. Estimated material parameters with different noise percentage

Table 4.7. Percentage error with respect to $\sigma = 0$

Variables / Noise percentage	$\tilde{\alpha}_4$	$\tilde{\alpha}_5$	$\tilde{\alpha}_6$	$\tilde{\alpha}_c$
0	0	0	0	0
1%	3.39%	33.54%	7.26%	4.15%
2%	6.95%	33.48%	22.24%	9.64%
5%	10.04%	37.37%	48.17%	18.83%
8%	17.94%	37.83%	84.53%	30.63%

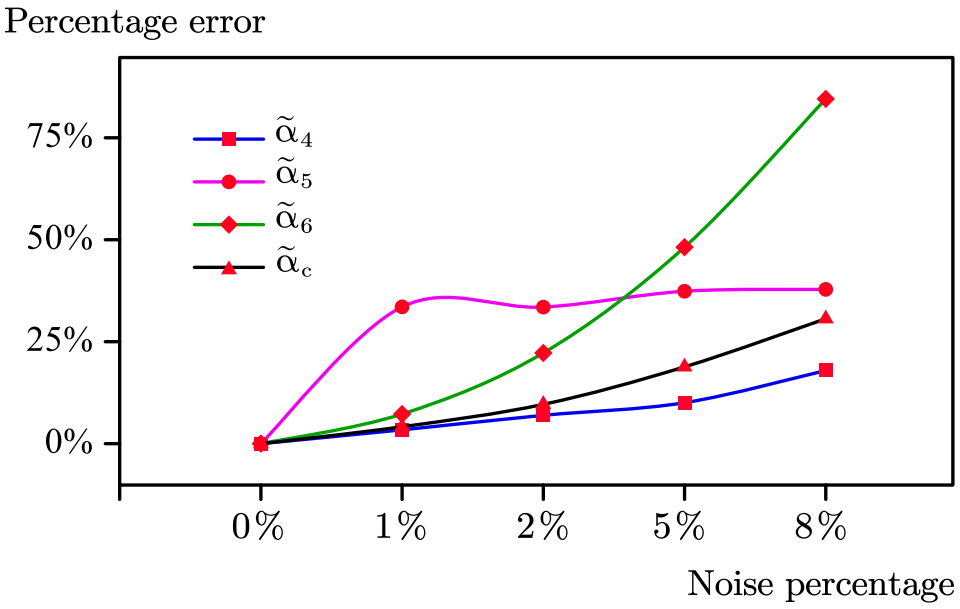
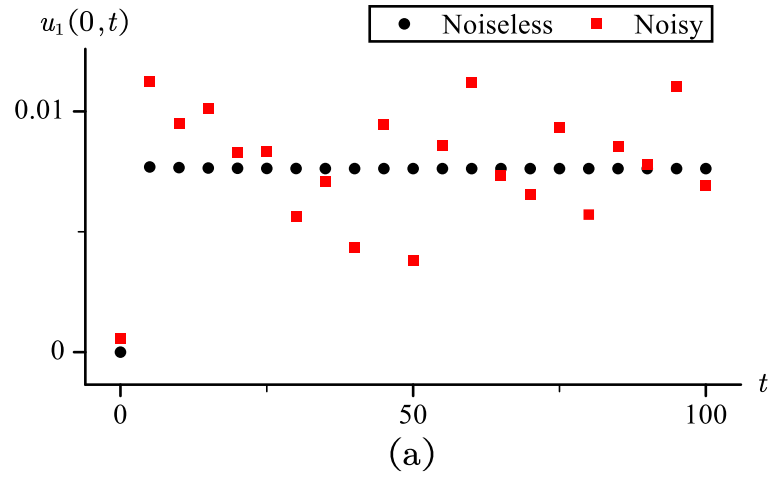
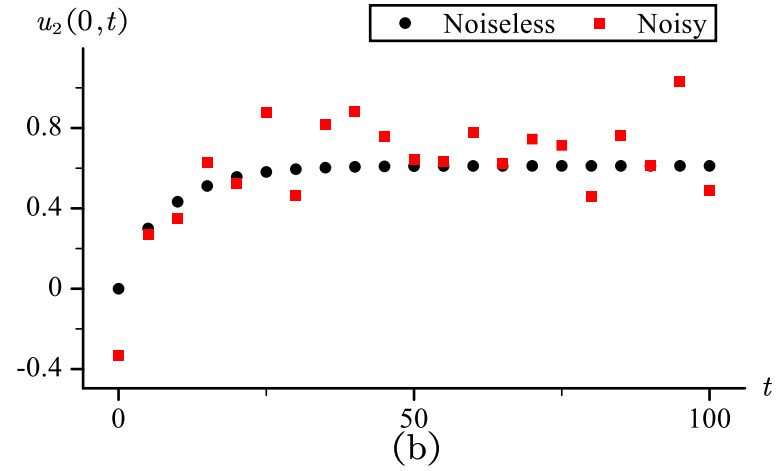


Fig. 4.22. Percentage error with respect to $\sigma = 0$

Time series of simulated 5% noisy measurements are shown in Figs. 4.23 – 4.25.





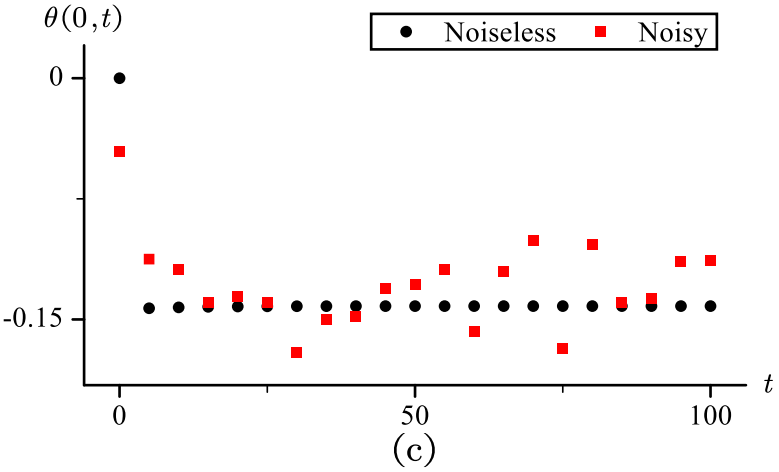


Fig. 4.23. Simulated noisy measurements (a). $u_1(0,t)$, (b). $u_2(0,t)$ and (c). $\theta(0,t)$, $\sigma = 5\%$

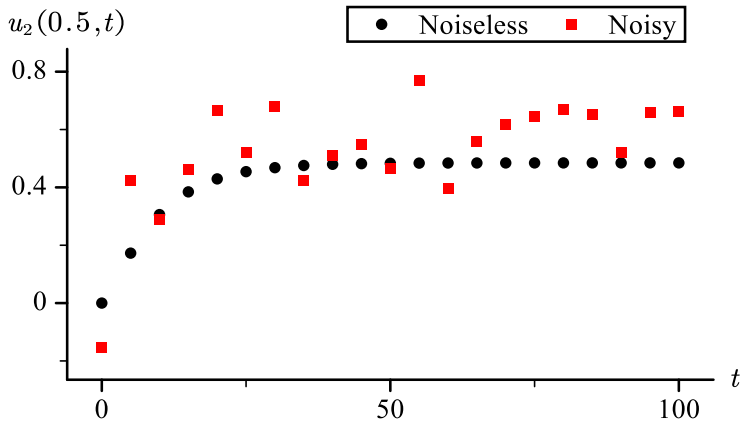
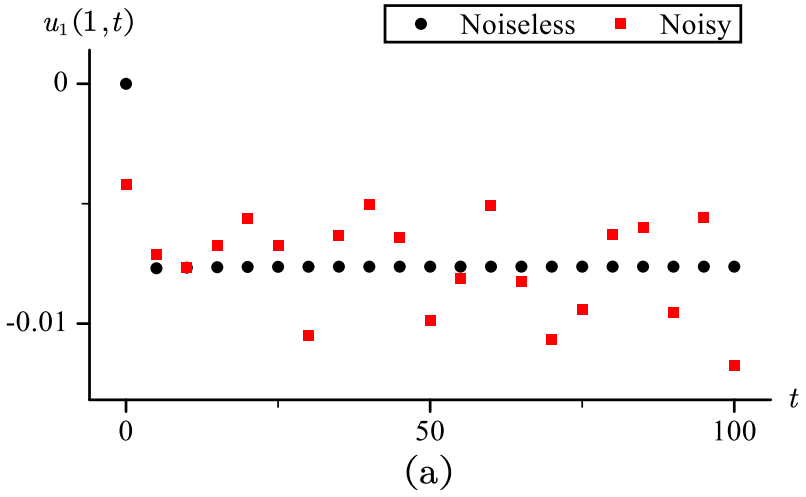
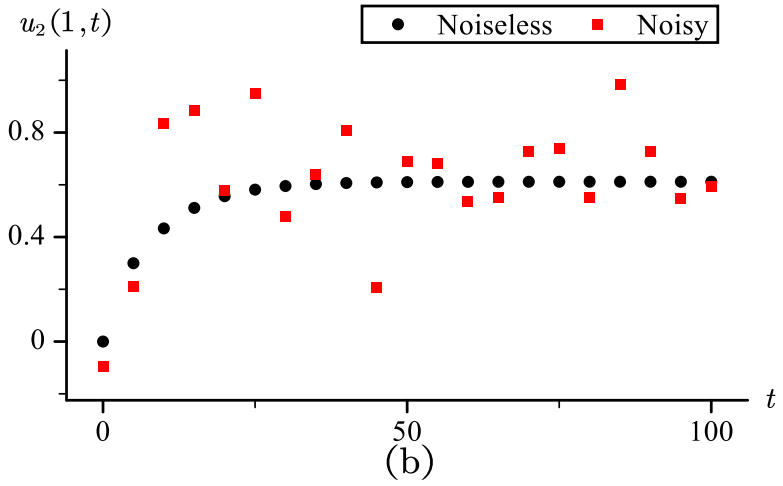


Fig. 4.24. Simulated noisy measurements $u_2(0.5,t)$, $\sigma = 5\%$



(a)



(b)

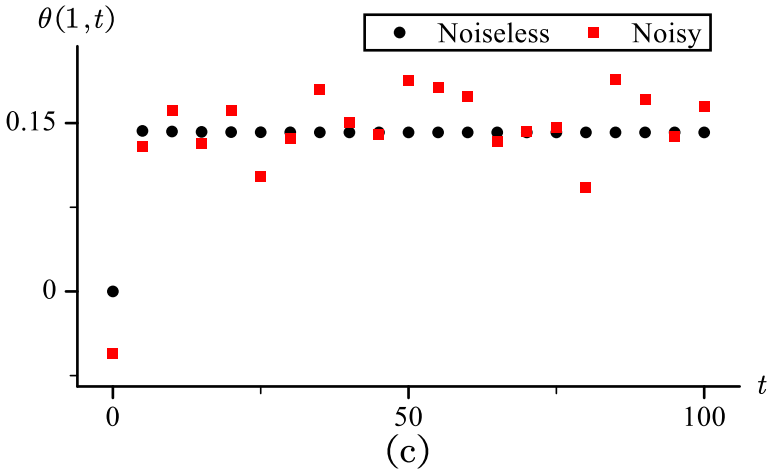


Fig. 4.25. Simulated noisy measurements (a). $u_1(1,t)$, (b). $u_2(1,t)$ and (c). $\theta(1,t)$, $\sigma = 5\%$

Estimated material parameters in Table 4.7 show a close agreement for $\sigma \leqslant 2\%$. When σ is greater than 5%, noise elimination techniques should be included to prevent large inaccuracies.

An increase or decrease in α_3 means increasing or decreasing the coupling layer's stiffness with respect to the stiffness of the beam, which may lead to different magnitudes of the sensor body's deflections, consequently changing the magnitude of the substrate's deformation. However, such change does not affect the parameters' prediction significantly (See Table 4.8). It is noted that for very stiff substrate and for very large α_3, the beam may not significantly deform for a suitable range of external loads. In this case, the sensor system may not operate as expected.

Table 4.8. Estimated parameters with input forces $Q = 0.5$ and different stiffness

Variables / α_3	$\widetilde{\alpha}_4$	$\widetilde{\alpha}_5$	$\widetilde{\alpha}_6$	$\widetilde{\alpha}_c$
5	1.9996	0.5009	0.5107	10.0030
10	2.0002	0.4958	0.4944	9.9862
15	2.0002	0.4995	0.5000	9.9988
20	2.0003	0.4994	0.5001	9.9986

CHAPTER 5
CONCLUSIONS

In this thesis we formulated and simulated the model of a mechanical sensor comprised of a deformable body coupled with a deformable solid substrate through an elastic layer that transmits actions between the sensor's body and the substrate. The deformable body is modelled through the geometrically nonlinear Reissner's beam theory with linear material response. The goal of the sensor is to estimate the material parameters of the substrate, which is modelled as a nonlinear viscoelastic Pasternak foundation, where the nonlinearity is a strain hardening effect.

In order to simulate the operation of the sensor, measurement data is generated by finite element solutions of the beam's deformations induced by different loading. From simulated measurement data with added Gaussian noise, an estimation algorithm that minimizes the distance between measurements and model predictions is used to estimate material parameters.

Simulated sensor outputs appear to be robust for relatively high noise levels, with accurate predictions using unfiltered measurement data. When the relative noise level increases the simulated sensor predictions deteriorate, suggesting that filtering may be necessary for input data with high noise.

Limitations of the current work mainly pertain to the search algorithm which require an initial guess in the range of material parameters to be estimated, and therefore some prior

knowledge of the material. Refinements should include the relaxation of this requirement, with the adoption of more robust search algorithmic solutions. Additionally, initially curved substrates should be included in the model as they reflect relevant scenarios for hardware implementation.

Future work is mainly directed towards the hardware implementation, which requires careful considerations to map modeling assumptions into implementable solutions. This will likely require iterations and modifications of the model as well. In addition, the sensing capabilities of the device are intended to be implemented on a mobile robotic device, with all the additional design complications associated with the fact that the coupling layer has also to have properties suitable for locomotion on unstructured terrains.

Reference

[1] J. Fattahi and D. Spinello, 'Sensing linear viscoelastic constitutive parameters with a Timoshenko beam on a multi-layer foundation: Modeling and simulation', *Sensing and Bio-Sensing Research*, vol. 4, pp. 78–89, Jun. 2015, doi: 10.1016/j.sbsr.2015.04.003.

[2] E. Reissner, 'On one-dimensional finite-strain beam theory: the plane problem', *Zeitschrift für angewandte Mathematik und Physik ZAMP*, vol. 23, no. 5, pp. 795–804, 1972.

[3] H. J. Halstead and D. A. Harris, 'Small Deflection of Beams', in *A Course in Pure and Applied Mathematics*, Palgrave, London, 1963, pp. 495–525.

[4] '"The Da Vinci-Euler-Bernoulli Beam Theory," ME Online Web Exclusive, April 18, 2003', Jun. 23, 2006. https://web.archive.org/web/20060623063248/http://www.memagazine.org/contents/current/webonly/webex418.html (accessed May 13, 2020).

[5] P. S. P. Timoshenko, 'LXVI. On the correction for shear of the differential equation for transverse vibrations of prismatic bars', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 245, pp. 744–746, May 1921, doi: 10.1080/14786442108636264.

[6] P. S. P. Timoshenko, 'X. On the transverse vibrations of bars of uniform cross-section', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 43, no. 253, pp. 125–131, Jan. 1922, doi: 10.1080/14786442208633855.

[7] G. Kirchhoff, 'Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes.', *Journal für die reine und angewandte Mathematik*, vol. 1859, no. 56, pp. 285–313, Jan. 1859, doi: 10.1515/crll.1859.56.285.

[8] A. E. H. Love, *A Treatise on the Mathematical Theory of Elasticity*. Cambridge University Press, 2013.

[9] E. Reissner, 'On finite deformations of space-curved beams', *Z. angew. Math. Phys.*, vol. 32, no. 6, pp. 734–744, Nov. 1981, doi: 10.1007/BF00946983.

[10] S. S. Antman, 'Kirchhoff's problem for nonlinearly elastic rods', *Quart. Appl. Math.*, vol. 32, no. 3, pp. 221–240, Oct. 1974, doi: 10.1090/qam/667026.

[11] J. C. Simo, 'A finite strain beam formulation. The three-dimensional dynamic problem. Part I', *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 1, pp. 55–70, May 1985, doi: 10.1016/0045-7825(85)90050-7.

[12] T. Tjahjowidodo, 'Characterization, modelling and control of mechanical systems comprising material and geometrical nonlinearities', PHD Thesis, 2006.

[13] A. P. Selvadurai, *Elastic analysis of soil-foundation interaction*. Elsevier, 2013.

[14] A. Humer, 'Exact solutions for the buckling and postbuckling of shear-

deformable beams', *Acta Mech*, vol. 224, no. 7, pp. 1493–1525, Jul. 2013, doi: 10.1007/s00707-013-0818-1.

[15] Y. Kobayashi, A. Kato, H. Watanabe, T. Hoshi, K. Kawamura, and M. G. Fujie, 'Modeling of viscoelastic and nonlinear material properties of liver tissue using fractional calculations', *Journal of Biomechanical Science and Engineering*, vol. 7, no. 2, pp. 177–187, 2012.

[16] M. Hetényi, *Beams on Elastic Foundation: Theory with Applications in the Fields of Civil and Mechanical Engineering*. Ann Arbor, 1946.

[17] C. Miranda and K. Nair, 'Finite beams on elastic foundations', *Journal of the Structural Division*, vol. 92, no. 2, pp. 131–142, 1966.

[18] F. W. Beaufait and P. W. Hoadley, 'Analysis of elastic beams on nonlinear foundations', *Computers & Structures*, vol. 12, no. 5, pp. 669–676, Nov. 1980, doi: 10.1016/0045-7949(80)90168-6.

[19] T. M. Wang and J. E. Stephens, 'Natural frequencies of Timoshenko beams on pasternak foundations', *Journal of Sound and Vibration*, vol. 51, no. 2, pp. 149–155, Mar. 1977, doi: 10.1016/S0022-460X(77)80029-1.

[20] Y. Yang, H. Ding, and L.-Q. Chen, 'Dynamic response to a moving load of a Timoshenko beam resting on a nonlinear viscoelastic foundation', *Acta Mech Sin*, vol. 29, no. 5, pp. 718–727, Oct. 2013, doi: 10.1007/s10409-013-0069-3.

[21] A. Mamandi and M. H. Kargarnovin, 'Nonlinear Dynamic Analysis of a Timoshenko Beam Resting on a Viscoelastic Foundation and Traveled by a Moving Mass', *Shock and Vibration*, 2014. https://www.hindawi.com/journals/sv/2014/242090/ (accessed Mar. 14, 2018).

[22] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[23] A. G. Ramm, *Inverse Problems: Mathematical and Analytical Techniques with Applications to Engineering*. Springer Science & Business Media, 2006.

[24] R. Sverdlove, 'Inverse problems for dynamical systems', *Journal of Differential Equations*, vol. 42, no. 1, pp. 72–105, Oct. 1981, doi: 10.1016/0022-0396(81)90034-6.

[25] K. Bryan and Jr. Caudill L., 'An Inverse Problem in Thermal Imaging', *SIAM J. Appl. Math.*, vol. 56, no. 3, pp. 715–735, Jun. 1996, doi: 10.1137/S0036139994277828.

[26] H. Kagiwada, R. Kalaba, S. Timko, and S. Ueno, 'Associate memories for system identification: Inverse problems in remote sensing', *Mathematical and Computer Modelling*, vol. 14, pp. 200–202, Jan. 1990, doi: 10.1016/0895-7177(90)90175-M.

[27] T. G. Faurholdt, 'Inverse modelling of constitutive parameters for elastoplastic problems', *The Journal of Strain Analysis for Engineering Design*, vol. 35, no. 6, pp. 471–478, 2000, doi: 10.1243/0309324001514233.

[28] B. Barabasz, E. Gajda-Zagórska, S. Migórski, M. Paszyński, R. Schaefer, and M. Smo\lka, ‘A hybrid algorithm for solving inverse problems in elasticity’, *International Journal of Applied Mathematics and Computer Science*, vol. 24, no. 4, pp. 865–886, 2014, doi: 10.2478/amcs-2014-0064.

[29] M. Tartibi, D. J. Steigmann, and K. Komvopoulos, ‘A reverse updated Lagrangian finite element formulation for determining material properties from measured force and displacement data’, *Comput Mech*, vol. 54, no. 6, pp. 1375–1394, Dec. 2014, doi: 10.1007/s00466-014-1064-7.

[30] T. Knabe, H. F. Schweiger, and T. Schanz, ‘Calibration of constitutive parameters by inverse analysis for a geotechnical boundary problem’, *Can. Geotech. J.*, vol. 49, no. 2, pp. 170–183, Jan. 2012, doi: 10.1139/t11-091.

[31] G. M. L. Gladwell, *Inverse problems in vibration*. Springer Netherlands, 1986.

[32] M. Yamamoto, ‘Inverse eigenvalue problem for a vibration of a string with viscous drag’, *Journal of Mathematical Analysis and Applications*, vol. 152, no. 1, pp. 20–34, Oct. 1990, doi: 10.1016/0022-247X(90)90090-3.

[33] H. T. Banks, R. K. Powers, and I. G. Rosen, ‘Inverse problems in the modeling of vibrations of flexible beams’, in *Distributed Parameter Systems*, Springer, Berlin, Heidelberg, 1987, pp. 1–22.

[34] L. D. Chiwiacowsky, H. F. D. C. Velho, and P. Gasbarri¶, ‘A variational approach for solving an inverse vibration problem’, *Inverse Problems in Science and Engineering*, vol. 14, no. 5, pp. 557–577, Jul. 2006, doi: 10.1080/17415970600574237.

[35] D. G. Moss and H. Benaroya, ‘A discrete inverse vibration problem with parameter uncertainty’, *Applied Mathematics and Computation*, vol. 69, no. 2, pp. 313–333, May 1995, doi: 10.1016/0096-3003(94)00140-Y.

[36] A. F. Collar, J. J. O. Oliveros, and A. I. Grebénnikov, ‘Uniqueness of Solution of the Inverse Electroencephalographic Problem’, in *Numerical Analysis and Its Applications*, Jun. 2000, pp. 207–213, doi: 10.1007/3-540-45262-1_25.

[37] Z. Gu and Y. Tan, ‘Fundamental solution method for inverse source problem of plate equation’, *Appl. Math. Mech.-Engl. Ed.*, vol. 33, no. 12, pp. 1513–1532, Dec. 2012, doi: 10.1007/s10483-012-1641-6.

[38] G. Chavent, G. Papanicolaou, P. Sacks, and W. Symes, *Inverse problems in wave propagation*, vol. 90. Springer Science & Business Media, 2012.

[39] R. Forke, D. Scheibner, K. Hiller, T. Gessner, W. Dötzel, and J. Mehner, ‘Fabrication and characterization of a force coupled sensor–actuator system for adjustable resonant low frequency vibration detection’, *Sensors and Actuators A: Physical*, vol. 145–146, pp. 245–256, Jul. 2008, doi: 10.1016/j.sna.2007.12.032.

[40] Y. Ching-Yu, ‘Solution of an inverse vibration problem using a linear least-squares error method’, *Applied Mathematical Modelling*, vol. 20, no. 10, pp. 785–788,

Oct. 1996, doi: 10.1016/0307-904X(96)00083-2.

[41] D. Hua and P. Lancaster, 'Linear matrix equations from an inverse problem of vibration theory', *Linear Algebra and its Applications*, vol. 246, pp. 31–47, Oct. 1996, doi: 10.1016/0024-3795(94)00311-4.

[42] W. Liu, D. Li, and J. Shen, 'Least-squares solutions of constrained inverse eigenproblem and associated optimal approximation problem', *International Journal of Computer Mathematics*, vol. 90, no. 3, pp. 641–650, Mar. 2013, doi: 10.1080/00207160.2012.735662.

[43] M. S. Gockenbach and A. A. Khan, 'An Abstract Framework for Elliptic Inverse Problems: Part 1. An Output Least-Squares Approach', *Mathematics and Mechanics of Solids*, vol. 12, no. 3, pp. 259–276, Jun. 2007, doi: 10.1177/1081286505055758.

[44] S. Avril and S. Evans, Eds., *Material Parameter Identification and Inverse Problems in Soft Tissue Biomechanics*. Springer International Publishing, 2017.

[45] M. Del Maestro, F. Cecchi, S. M. Serio, C. Laschi, and P. Dario, 'Sensing device for measuring infants' grasping actions', *Sensors and Actuators A: Physical*, vol. 165, no. 2, pp. 155–163, Feb. 2011, doi: 10.1016/j.sna.2010.08.016.

[46] M. E. Morris, F. Huxham, J. McGinley, K. Dodd, and R. Iansek, 'The biomechanics and motor control of gait in Parkinson disease', *Clin Biomech (Bristol, Avon)*, vol. 16, no. 6, pp. 459–470, Jul. 2001.

[47] G. Y. H. Lee and C. T. Lim, 'Biomechanics approaches to studying human diseases', *Trends Biotechnol.*, vol. 25, no. 3, pp. 111–118, Mar. 2007, doi: 10.1016/j.tibtech.2007.01.005.

[48] E. J. Weinberg, D. Shahmirzadi, and M. R. K. Mofrad, 'On the multiscale modeling of heart valve biomechanics in health and disease', *Biomech Model Mechanobiol*, vol. 9, no. 4, pp. 373–387, Aug. 2010, doi: 10.1007/s10237-009-0181-2.

[49] A. Magnusson, M. Ristinmaa, and C. Ljung, 'Behaviour of the extensible elastica solution', *International Journal of Solids and Structures*, vol. 38, no. 46, pp. 8441–8457, Nov. 2001, doi: 10.1016/S0020-7683(01)00089-0.

[50] S. S. Antman and G. Rosenfeld, 'Global Behavior of Buckled States of Nonlinearly Elastic Rods', *SIAM Rev.*, vol. 20, no. 3, pp. 513–566, Jul. 1978, doi: 10.1137/1020069.

[51] S. Timoshenko, *Vibration Problems in Engineering*. Read Books, 2008.

[52] L. Meirovitch, *Fundamentals of Vibrations*. McGraw-Hill, 2001.

[53] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, 2012.

[54] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.

[55] 'Comparison of Constrained Optimization Functions—Wolfram Language

Documentation’.
https://reference.wolfram.com/language/tutorial/ConstrainedOptimizationComparison.html (accessed Jun. 04, 2020).

[56] ‘NMinimize—Wolfram Language Documentation’. https://reference.wolfram.com/language/ref/NMinimize.html (accessed Apr. 20, 2020).

Appendix A

Mathematica Codes of Noise Generation and Direct Solution

Generate noise and Direct solution

This code is used for generating measuring noise. Noisy data should be saved and exported to the inverse problem

Strains and coupling forces

Kinematics

```
In[•]:= {cosχ, sinχ} = {Cos[χ], Sin[χ]} /.
  First[Solve[{(Cos[θ + χ] // TrigExpand) == (ξ1'[s] + u1'[s]) / (1 + Λ),
      (Sin[θ + χ] // TrigExpand) == (ξ2'[s] + u2'[s]) / (1 + Λ)},
    {Cos[χ], Sin[χ]}] // Simplify]
```

Define the strains

```
In[•]:= ϵs = ((1 + Λ) Cos[χ] - 1) /.
    First[Solve[{(Cos[θ + χ] // TrigExpand) == (ξ1'[s] + u1'[s]) / (1 + Λ),
      (Sin[θ + χ] // TrigExpand) == (ξ2'[s] + u2'[s]) / (1 + Λ)},
     {Cos[χ], Sin[χ]}]] /. θ → θ[s] // Simplify

In[•]:= γs = (1 + Λ) Sin[χ] /.
    First[Solve[{(Cos[θ + χ] // TrigExpand) == (ξ1'[s] + u1'[s]) / (1 + Λ),
      (Sin[θ + χ] // TrigExpand) == (ξ2'[s] + u2'[s]) / (1 + Λ)},
     {Cos[χ], Sin[χ]}]] /. θ → θ[s] // Simplify

In[•]:= κs = θ'[s] - θ0'[s]
```

Linear constitutive relations (non-dimensional)

```
In[•]:= NN = α1 ϵs;
Q = γs;
MM = α2 κs;

In[•]:= e1 = {1, 0};
e2 = {0, 1};
R = RotationMatrix[#1] &;
```

Distributed forces (non-dimensional)

```
In[•]:= g0 = {ξ1, ξ2} - h / 2 R[θ0].e2 - {η01, η02}; (*Initial gap;
it is related to the initial terrain profile η0*)
```

```
g = ({ξ1, ξ2} + {u1, u2} - h/2 R[θ].e2) - ({η01, η02} + {0, η2})
  (* Current gap; h is the nondimensional thickness *);
g = ({ξ1, ξ2} + {u1, u2} - h/2 R[θ].e2) - ({η01, η02} + {η1, η2})
  (* Current gap; h is the nondimensional thickness *)
```

Distributed loads along global directions ($\alpha 3$ is the non-dimensional stiffness)

```
p = -α3 (1 - Sqrt[g0.g0]/Sqrt[g.g]) g // Simplify
```

```
μ = Cross[Join[-h/2 R[θ].e2, {0}], Join[p, {0}]][[3]] //
  Simplify (*Nondimensional distributed torque;
  remebr that h is the nondimensional parameter h/L*)
```

Reference configuration

```
ξ[s_] = {s, 0};
Dξ[s_] = D[ξ[s], s];
θ0[s_] = 0;
η0[s_] = {s, -1};
```

FE basis functions

FE mesh

```
L = 1;
```

```
(*Number of nodes*)
Nn = 3;
```

```
Ne = Nn - 1; (*Number of elements*)
Nm = 2; (*Number of nodes for each element for Lagrange
  Polynomials (H¹ solutions). Set it to 2 so that the number of
  nodes in the global domain match the ones for the H² solution*)
```

```
n = Table[i, {i, 0, L, L/(Nn - 1)}] // N; (*Global nodes*)
```

Lagrange polynomials (H^1 weak solutions)

Grid generation

```
ne = Table[x, {x, 0, L, L / Ne}] // N;
```

```
nm = Table[s, {s, -1, 1, 2/(Nm - 1)}]
```

```
x[s_, i_] := (ne[[i]] + ne[[i + 1]])/2 + ((ne[[i + 1]] - ne[[i]])/2) s
```

```
In[ ]:= s[x_] = Table[s /. First[Solve[x[s, i] == x, s]], {i, Ne}] // Expand // Chop;

In[ ]:= Module[{loc}, n = Flatten[Table[x[nm[[i]], j], {j, Ne}, {i, Nm}]];
          loc = {n[[1]]};
          Do[If[n[[i]] > loc[[-1]], AppendTo[loc, n[[i]]]], {i, Length[n]}];
          n = loc];

In[ ]:= nn = Partition[n, Nm];
```

Lagrange shape functions in the master element

```
In[ ]:= NL[s_] = Table[Module[{loc}, loc = nm[[i]];
            Product[s - Delete[nm, i][[j]], {j, Length[nm] - 1}] / Product[
              nm[[i]] - Delete[nm, i][[j]], {j, Length[nm] - 1}]], {i, Length[nm]}];

In[ ]:= DNL[s_] = D[NL[s], s];

In[ ]:= Plot[NL[s] // Evaluate, {s, -1, 1}]

In[ ]:= Plot[DNL[s] // Evaluate, {s, -1, 1}]
```

Basis functions

Functions and functions derivatives definitions for u_1, u_2, θ, and η

Lagrange shape functions in the global domain

```
In[ ]:= ψA[x_] = Module[{ψL, ψR, ψC, loc1, loc2},
          ψL = Table[NL[s[x][[i]]][[-1]], {i, Ne}] // Expand // Chop;
          ψR = Table[NL[s[x][[i]]][[1]], {i, Ne}] // Expand // Chop;
          ψC = Table[NL[s[x][[i]]][[j]], {i, Ne}, {j, 2, Nm - 1}] // Expand // Chop;
          loc1 = Table[Evaluate /@ Which[ne[[i - 1]] ≤ x ≤ ne[[i]], ψL[[i - 1]],
              ne[[i]] ≤ x ≤ ne[[i + 1]], ψR[[i]], True, 0], {i, 2, Ne}];
          loc2 = Table[Evaluate /@ Which[ne[[i]] ≤ x ≤ ne[[i + 1]], ψC[[i, j]], True, 0],
            {i, 2, Ne - 1}, {j, Nm - 2}];
          Do[loc1 = Flatten[Insert[loc1, loc2[[i]], i Nm - i - Nm + 3]],
           {i, 1, Length[loc2]}];
          Join[Evaluate //@ {Which[ne[[1]] ≤ x ≤ ne[[2]], ψR[[1]], True, 0]},
           Table[Evaluate /@ Which[ne[[1]] ≤ x ≤ ne[[2]], Flatten[ψC][[i]], True, 0],
            {i, Nm - 2}], loc1,
           Table[Evaluate /@ Which[ne[[-2]] ≤ x ≤ ne[[-1]], Flatten[ψC][[i]], True, 0],
            {i, Length[Flatten[ψC]] - (Nm - 3), Length[Flatten[ψC]]}],
           Evaluate //@ {Which[ne[[-2]] ≤ x ≤ ne[[-1]], ψL[[-1]], True, 0]}]];

In[ ]:= Plot[ψA[x] // Evaluate, {x, 0, L}, PlotStyle → Thickness[0.005]]

In[ ]:= DψA[x_] = D[ψA[x], x];

In[ ]:= Plot[DψA[x][[-1]] // Evaluate, {x, 0, L}, PlotStyle → Thickness[0.005]]
```

Subdomains and gaussian quadrature points

Supports of the FE basis functions

```
Ωs = N[Join[{{n[[1]], n[[2]]}},
    Table[{n[[i - 1]], n[[i + 1]]}, {i, 2, Nn - 1}], {{n[[-2]], n[[-1]]}}]]
```

Gaussian Quadrature points

```
(* Numerical integration is performed by using Gausian quadrature rule*)
<< NumericalDifferentialEquationAnalysis`;
```

```
(*Number of Gaussian quadrature points*)
sp = 10;
```

```
(*Gaussian quadrature points and weights*)
gq = Table[GaussianQuadratureWeights[sp, Ωs[[i, 1]], Ωs[[i, 2]]], {i, Nn}];
```

Evaluation at quadrature points. ψQ[[i,k,j]] gives the j-th basis function evaluated at the k-th quadrature point of the i-th FE

```
ψQ = Table[ψA[gq[[i, k, 1]]], {i, Nn}, {k, sp}];
```

```
DψQ = Table[DψA[gq[[i, k, 1]]], {i, Nn}, {k, sp}];
```

Non-dimensional constitutive parameters

```
(* Beam's material properties in terms of nondimensional parameters *)
α1 = 100 (*Ratio E/G*);
α3 = 5(*Nondimensional stiffness of the coupling layer k L^2/(G A)*);
α2 = 1/1000; (* Ratio I/(A L^2) *);
h = 0.1(*Ratio h/L*);
αc = 10 (*Structural damping*);
```

```
(*Pasternak foundation parameters (from published work)*)
α4 = 2; (*linear spring k1 L^2/(G A)*)
α5 = 0.5;(*nonlinear spring*)
α6 = 0.5(*shear*);
αμ = 10;
```

Non-dimensional boundary forces and torques

```
In[•]:= N0 = 0;
        Q0 = -0.5;
        N1 = N0;
        Q1 = -Q0;
        M0 = 0;
        M1 = M0;
```

Reduced order model operators

Projected states and strains

```
In[•]:= (* Basis functions for the approximations*)
        u2n[s_] = ψA[s];
        θn[s_] = ψA[s];
        u1n[s_] = ψA[s];
        η1n[s_] = ψA[s];
        η2n[s_] = ψA[s];
        Du1n[s_] = Dψ A[s];
        Dθn[s_] = Dψ A[s];
        Du2n[s_] = Dψ A[s];
        Dη1n[s_] = Dψ A[s];
        Dη2n[s_] = Dψ A[s];
```

```
In[•]:= (*Coefficients*)
        au1[t_] = Table[ToExpression["au1" <> ToString[i] <> "[t]"], {i, Nn}];
        Dau1[t_] = Table[ToExpression["Dau1" <> ToString[i] <> "[t]"], {i, Nn}];
        au2[t_] = Table[ToExpression["au2" <> ToString[i] <> "[t]"], {i, Nn}];
        Dau2[t_] = Table[ToExpression["Dau2" <> ToString[i] <> "[t]"], {i, Nn}];
        aθ[t_] = Table[ToExpression["aθ" <> ToString[i] <> "[t]"], {i, Nn}];
        Daθ[t_] = Table[ToExpression["Daθ" <> ToString[i] <> "[t]"], {i, Nn}];
        aη1[t_] = Table[ToExpression["aη1" <> ToString[i] <> "[t]"], {i, Nn}];
        aη2[t_] = Table[ToExpression["aη2" <> ToString[i] <> "[t]"], {i, Nn}];
```

```
In[•]:= q[t_] = Join[au1[t], au2[t], aθ[t], aη1[t], aη2[t]];
        Dq[t_] = Join[Dau1[t], Dau2[t], Daθ[t]];
```

```
In[•]:= invq[t_] = Join[au1[t], au2[t], aθ[t], aη1[t], aη2[t]];
        invDq[t_] = Join[Dau1[t], Dau2[t], Daθ[t]];
```

```
(*Projected states*)
u1p[s_, t_] = u1n[s].au1[t];
u2p[s_, t_] = u2n[s].au2[t];
θp[s_, t_] = θn[s].aθ[t];
η1p[s_, t_] = η1n[s].aη1[t];
η2p[s_, t_] = η2n[s].aη2[t];
Du1p[s_, t_] = Du1n[s].au1[t];
Du2p[s_, t_] = Du2n[s].au2[t];
Dθp[s_, t_] = Dθn[s].aθ[t];
Dη1p[s_, t_] = Dη1n[s].aη1[t];
Dη2p[s_, t_] = Dη2n[s].aη2[t];
```

Replacement rules

```
rules = {u1[s] → u1p[s, t], u1 -> u1p[s, t], u1'[s] → Du1p[s, t],
    u2[s] → u2p[s, t], u2 -> u2p[s, t], u2'[s] → Du2p[s, t],
    θ[s] → θp[s, t], θ → θp[s, t], θ0 → θ0[s], θ'[s] → Dθp[s, t],
    η1 → η1p[s, t], η2 → η2p[s, t], ξ1[s] → ξ[s][[1]], ξ1 -> ξ[s][[1]],
    ξ1'[s] → D[ξ[s][[1]], s], ξ2[s] → ξ[s][[2]], ξ2 → ξ[s][[2]],
    ξ2'[s] → D[ξ[s][[2]], s], η01 → η0[s][[1]], η02 → η0[s][[2]]};
```

Projected strains

```
εp[s_, t_] = εs /. rules // Chop;
γp[s_, t_] = γs /. rules // Chop;
κp[s_, t_] = κs /. rules // Chop;
```

Projected loads

```
pp[s_, t_] = p /. rules;
μp[s_, t_] = μ /. rules;
```

Projected internal forces and moments

```
Pp[s_, t_] = RotationMatrix[θp[s, t]].{α1 εp[s, t], γp[s, t]} // Chop;
Mp[s_, t_] = Cross[Join[Expand[(1 + Λ) {cosχ, sinχ}], {0}],
      Join[{α1 εp[s, t], γp[s, t]}, {0}]][[3]] /. rules;
```

Operators evaluated at quadrature points

Mass matrices

```
Mu1 = Table[Sum[gq[[i, k, 2]] × u1n[gq[[i, k, 1]]][[i]] × u1n[gq[[i, k, 1]]][[j]],
     {k, sp}], {i, Nn}, {j, Nn}];
Mu2 = Table[Sum[gq[[i, k, 2]] × u2n[gq[[i, k, 1]]][[i]] × u2n[gq[[i, k, 1]]][[j]],
     {k, sp}], {i, Nn}, {j, Nn}];
Mθ = α2 Table[Sum[gq[[i, k, 2]] × θn[gq[[i, k, 1]]][[i]] × θn[gq[[i, k, 1]]][[j]],
      {k, sp}], {i, Nn}, {j, Nn}];
Mη1 = Table[Sum[gq[[i, k, 2]] × η1n[gq[[i, k, 1]]][[i]] ×
      η1n[gq[[i, k, 1]]][[j]], {k, sp}], {i, Nn}, {j, Nn}];
Mη2 = Table[Sum[gq[[i, k, 2]] × η2n[gq[[i, k, 1]]][[i]] ×
      η2n[gq[[i, k, 1]]][[j]], {k, sp}], {i, Nn}, {j, Nn}];
```

Stiffness operators

```
Ku1 = Table[Sum[gq[[i, k, 2]] × Pp[gq[[i, k, 1]], t][[1]] ×
      Du1n[gq[[i, k, 1]]][[i]], {k, sp}], {i, Nn}];

Ku2 = Table[Sum[gq[[i, k, 2]] × Pp[gq[[i, k, 1]], t][[2]] ×
      Du2n[gq[[i, k, 1]]][[i]], {k, sp}], {i, Nn}];

Kθ =
  Table[Sum[gq[[i, k, 2]] (α1 α2 κp[gq[[i, k, 1]], t] × Dθn[gq[[i, k, 1]]][[i]] -
       Mp[gq[[i, k, 1]], t] × θn[gq[[i, k, 1]]][[i]]), {k, sp}], {i, Nn}];

Kη1 =
  Table[Sum[gq[[i, k, 2]] (α4 η1n[gq[[i, k, 1]]][[i]] × η1n[gq[[i, k, 1]]][[j]] +
       α6 Dη1n[gq[[i, k, 1]]][[i]] × Dη1n[gq[[i, k, 1]]][[j]]),
     {k, sp}], {i, Nn}, {j, Nn}];
Kη1NL = Table[Sum[gq[[i, k, 2]] α5 ({η1p[gq[[i, k, 1]], t],
       η1p[gq[[i, k, 1]], t]}.{η1p[gq[[i, k, 1]], t], η1p[gq[[i, k, 1]], t]})
     η1n[gq[[i, k, 1]]][[i]] × η1n[gq[[i, k, 1]]][[j]],
    {k, sp}], {i, Nn}, {j, Nn}];
Kη2 = Table[Sum[gq[[i, k, 2]] (α4 η2n[gq[[i, k, 1]]][[i]] ×
       η2n[gq[[i, k, 1]]][[j]] + α6 Dη2n[gq[[i, k, 1]]][[i]] ×
       Dη2n[gq[[i, k, 1]]][[j]]), {k, sp}], {i, Nn}, {j, Nn}];
Kη2NL = Table[Sum[gq[[i, k, 2]] α5 ({η2p[gq[[i, k, 1]], t],
       η2p[gq[[i, k, 1]], t]}.{η2p[gq[[i, k, 1]], t], η2p[gq[[i, k, 1]], t]})
     η2n[gq[[i, k, 1]]][[i]] × η2n[gq[[i, k, 1]]][[j]],
    {k, sp}], {i, Nn}, {j, Nn}];
```

Distributed force operators

```
Fu = Table[Sum[gq[[i, k, 2]] (-pp[gq[[i, k, 1]], t] * {u1n[gq[[i, k, 1]]][[i]],
       u2n[gq[[i, k, 1]]][[i]]}), {k, sp}], {i, Nn}] // Transpose;

Fθ = Table[Sum[gq[[i, k, 2]] (-μp[gq[[i, k, 1]], t] * θn[gq[[i, k, 1]]][[i]]),
     {k, sp}], {i, Nn}] // Chop;
(*Fθ=Table[0,Nn];*)

Fη1 = Table[Sum[gq[[i, k, 2]]
     (pp[gq[[i, k, 1]], t][[1]] × η1n[gq[[i, k, 1]]][[i]]), {k, sp}], {i, Nn}];

Fη2 = Table[Sum[gq[[i, k, 2]]
     (pp[gq[[i, k, 1]], t][[2]] × η2n[gq[[i, k, 1]]][[i]]), {k, sp}], {i, Nn}];
```

Boundary operators

```
Fub = -({N1, Q1} * {u1n[1], u2n[1]} - {N0, Q0} * {u1n[0], u2n[0]}) // Chop
  (* Global boundary forces *);
Fub = -(RotationMatrix[θp[1, t]].{N1, Q1} * {u1n[1], u2n[1]} -
      RotationMatrix[θp[0, t]].{N0, Q0} * {u1n[0], u2n[0]}) //
   Chop (* Rotated boundary forces *);

Fθb = -(M1 θn[1] - M0 θn[0]) // Chop;
```

Direct solution (generate data for inverse

problem)

```
(*
eqD=Join[Mu1.au1''[t]+αc Mu1.au1'[t]+Ku1+Fu[[1]]+Fub[[1]],
  Mu2.au2''[t]+αc Mu2.au2'[t]+Ku2+Fu[[2]]+Fub[[2]],
  Mθ.aθ''[t]+αc Mθ.aθ'[t]+Kθ+Fθ+Fθb,
  αμ Mη1.aη1'[t]+Kη1.aη1[t]+Fη1,αμ Mη2.aη2'[t]+Kη2.aη2[t]+Fη2]
*)
```

```
eqD =
  Join[au1'[t] - Dau1[t], Mu1.Dau1'[t] + αc Mu1.Dau1[t] + Ku1 + Fu[[1]] + Fub[[1]],
   au2'[t] - Dau2[t], Mu2.Dau2'[t] + αc Mu2.Dau2[t] + Ku2 + Fu[[2]] + Fub[[2]],
   aθ'[t] - Daθ[t], Mθ.Daθ'[t] + αc Mθ.Daθ[t] + Kθ + Fθ + Fθb,
   αμ Mη1.aη1'[t] + (Kη1 + Kη1NL).aη1[t] + Fη1,
   αμ Mη2.aη2'[t] + (Kη2 + Kη2NL).aη2[t] + Fη2];
```

```
icD = Join[q[0], Dq[0]];
```

```
tf = 100;
qD[t_] = q[t] /. First[NDSolve[Thread[Join[eqD, icD] == 0], Join[q[t], Dq[t]],
      {t, 0, tf}, Method -> {"EquationSimplification" -> "Residual"}]];
```

```
ListPlot[Transpose[{n, Take[qD[tf], {Nn + 1, 2 Nn}]}], Joined → True,
 PlotRange → All, AxesLabel → {"s", "u₂(s," <> ToString[tf] <> ")"}]
```

```
ListPlot[Transpose[{n, Take[qD[tf], {3 Nn + 1, 4 Nn}]}], Joined → True,
 PlotRange → All, AxesLabel → {"s", "η₁(s," <> ToString[tf] <> ")"}]
```

```
Plot[Take[qD[t], {Nn + 1, 2 Nn}].u2n[L / 2],
 {t, 0, tf}, PlotRange → All, AxesLabel → {"t", "u₂(L/2,t)"}]
```

```
Plot[Take[qD[t], {2 Nn + 1, 3 Nn}].θn[0.75 L],
 {t, 0, tf}, PlotRange → All, AxesLabel → {"t", "θ(3L/4,t)"}]
```

```
ListPlot[Transpose[{n, Take[qD[tf], {2 Nn + 1, 3 Nn}] / Degree}],
 Joined → True, PlotRange → All]
```

```
Plot[Take[qD[t], {4 Nn + 1, 5 Nn}].η2n[L / 2],
 {t, 0, tf}, PlotRange → All, AxesLabel → {"t", "η₂(L/2,t)"}]
```

```
Plot[Take[qD[t], {3 Nn + 1, 4 Nn}].η1n[L / 2],
 {t, 0, tf}, PlotRange → All, AxesLabel → {"t", "η₁(L/2,t)"}]
```

```
Plot[Take[qD[t], {1, Nn}].u2n[0], {t, 0, tf},
 PlotRange → All, AxesLabel → {"t", "u₁(L/2,t)"}]
```

```
(*Solutions from direct problem*)
sol = NDSolveValue[Thread[Join[eqD, icD] == 0], {q[t]}, {t, 0, tf},
   Method -> {"EquationSimplification" -> "Residual"}];(*Results from eqD*)
(*u1 @ 0, L/2, L*)
zu10 = Table[Take[First[sol], {1, Nn}].u2n[0], {t, 0, tf, 1}];(*u1 @ 0*)
(*zu1L2=Table[Take[First[sol],{1,Nn}].u2n[L/2],{t,0,tf,1}];(*u1 @ L/2*)*)
zu1L = Table[Take[First[sol], {1, Nn}].u2n[L], {t, 0, tf, 1}];(*u1 @ L*)

(*u2 @ 0, L/2, L*)

zu20 = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[0], {t, 0, tf, 1}];(*u2 @ 0*)
zu2L2 = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[L / 2], {t, 0, tf, 1}];
(*u2 @ L/2*)
zu2L = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[L], {t, 0, tf, 1}];(*u2 @ L*)

(*θ @ 0, L/2, L*)

zθ0 = Table[Take[First[sol], {2 Nn + 1, 3 Nn}].θn[0], {t, 0, tf, 1}];(*θ @ 0*)
(*zθL2=Table[Take[First[sol],{2Nn+1,3Nn}].θn[L],{t,0,tf,1}];(*θ @ L/2*)*)
zθL = Table[Take[First[sol], {2 Nn + 1, 3 Nn}].θn[L], {t, 0, tf, 1}];(*θ @ L*)
```

```
(*Measurement, no noise*)
```

```
zu10nn = Take[zu10, {1, 101, 5}];(*No noise (u1@0)*)
zu1Lnn = Take[zu1L, {1, 101, 5}];(*No noise (u1@L)*)

zu20nn = Take[zu20, {1, 101, 5}];(*No noise (u2@0)*)
zu2L2nn = Take[zu2L2, {1, 101, 5}];(*No noise (u2@L/2)*)
zu2Lnn = Take[zu2L, {1, 101, 5}];(*No noise (u2@L)*)

zθ0nn = Take[zθ0, {1, 101, 5}];(*No noise (θ@0)*)
zθLnn = Take[zθL, {1, 101, 5}];(*No noise (θ@L)*)
```

Generate noise

```
zt = Table[t, {t, 0, tf, 5}];
```

```
σ = 0.08;
ν := RandomVariate[NormalDistribution[0, Sqrt[σ]]]

(* @ 0*)
```

```
zu10N = Table[{zt[[i]], (zu10nn + ν * zu10nn[[21]])[[i]]}, {i, 1, 21}];
Show[ListPlot[zu10N, PlotStyle → Black],
 ListPlot[Table[{zt[[i]], (zu10nn)[[i]]}, {i, 1, 21}], PlotStyle → Red],
 PlotRange → All]
zu1LN = Table[{zt[[i]], (zu1Lnn + ν * zu1Lnn[[21]])[[i]]}, {i, 1, 21}];
```

```
In[ ]:=
(* @ L/2*)
zu20N = Table[{zt[[i]], (zu20nn + ν * zu20nn[[21]])[[i]]}, {i, 1, 21}];
zu2L2N = Table[{zt[[i]], (zu2L2nn + ν * zu2L2nn[[21]])[[i]]}, {i, 1, 21}];
zu2LN = Table[{zt[[i]], (zu2Lnn + ν * zu2Lnn[[21]])[[i]]}, {i, 1, 21}];

(* @ L*)
zθ0N = Table[{zt[[i]], (zθ0nn + ν * zθ0nn[[21]])[[i]]}, {i, 1, 21}];
zθLN = Table[{zt[[i]], (zθLnn + ν * zθLnn[[21]])[[i]]}, {i, 1, 21}];


(*Export data files*)

In[ ]:= SetDirectory[NotebookDirectory[]];
Export["zu10N.dat", zu10N];
Export["zu1LN.dat", zu1LN];
Export["zu20N.dat", zu20N];
Export["zu2L2N.dat", zu2L2N];
Export["zu2LN.dat", zu2LN];
Export["zTheta0N.dat", zθ0N];
Export["zThetaLN.dat", zθLN];

SetOptions[SelectedNotebook[],
 PrintingStyleEnvironment → "Printout", ShowSyntaxStyles → True]
```

Appendix B

Mathematica Codes of Inverse Problem

Inverse Problem

The folded parts are the same as the parts in Direct problem. Noisy data should be imported for repeatability

Strains and coupling forces

Reference configuration

FE basis functions

FE mesh

```
In[•]:= L = 1;

In[•]:= (*Number of nodes*)
        Nn = 3;

In[•]:= Ne = Nn - 1; (*Number of elements*)
        Nm = 2; (*Number of nodes for each element for Lagrange
         Polynomials (H¹ solutions). Set it to 2 so that the number of
         nodes in the global domain match the ones for the H² solution*)

In[•]:= n = Table[i, {i, 0, L, L / (Nn - 1)}] // N; (*Global nodes*)
```

Lagrange polynomials (H^1 weak solutions)

Basis functions

Subdomains and gaussian quadrature points

Non-dimensional constitutive parameters

```
(* Beam's material properties in terms of nondimensional parameters *)
α1 = 100 (*Ratio E/G*);
α3 = 5 (*Nondimensional stiffness of the coupling layer k L^2/(G A)*);
α2 = 1/1000; (* Ratio I/(A L^2) *);
h = 0.1 (*Ratio h/L*);
αc = 10 (*Structural damping*);
```

```
(*Pasternak foundation parameters (from published work)*)
α4 = 2; (*linear spring k1 L^2/(G A)*)
α5 = 0.5; (*nonlinear spring*)
α6 = 0.5 (*shear*);
αμ = 10;
```

Non-dimensional boundary forces and torques

```
N0 = 0;
Q0 = -0.5;
N1 = N0;
Q1 = -Q0;
M0 = 0;
M1 = M0;
```

Reduced order model operators

Projected states and strains

Operators evaluated at quadrature points

Inverse problem

```
invKη1 = Table[
   Sum[gq[[i, k, 2]] (x1 η1n[gq[[i, k, 1]]][[i]] × η1n[gq[[i, k, 1]]][[j]] +
       x3 Dη1n[gq[[i, k, 1]]][[i]] × Dη1n[gq[[i, k, 1]]][[j]]),
    {k, sp}], {i, Nn}, {j, Nn}];
invKη1NL = Table[Sum[gq[[i, k, 2]] x2
     ({η1p[gq[[i, k, 1]], t], η2p[gq[[i, k, 1]], t]}.
       {η1p[gq[[i, k, 1]], t], η2p[gq[[i, k, 1]], t]})
    η1n[gq[[i, k, 1]]][[i]] × η1n[gq[[i, k, 1]]][[j]],
   {k, sp}], {i, Nn}, {j, Nn}];
invKη2 = Table[Sum[gq[[i, k, 2]]
     (x1 η2n[gq[[i, k, 1]]][[i]] × η2n[gq[[i, k, 1]]][[j]] +
       x3 Dη2n[gq[[i, k, 1]]][[i]] × Dη2n[gq[[i, k, 1]]][[j]]),
    {k, sp}], {i, Nn}, {j, Nn}];
invKη2NL = Table[Sum[gq[[i, k, 2]] x2
     ({η2p[gq[[i, k, 1]], t], η2p[gq[[i, k, 1]], t]}.
       {η2p[gq[[i, k, 1]], t], η2p[gq[[i, k, 1]], t]})
    η2n[gq[[i, k, 1]]][[i]] × η2n[gq[[i, k, 1]]][[j]], {k, sp}],
  {i, Nn}, {j, Nn}];(*Terms with unknown parameters*)
```

```
inveqD = Join[au1'[t] - Dau1[t], Mu1.Dau1'[t] +
    αc Mu1.Dau1[t] + Ku1 + Fu[[1]] + Fub[[1]], au2'[t] - Dau2[t],
   Mu2.Dau2'[t] + αc Mu2.Dau2[t] + Ku2 + Fu[[2]] + Fub[[2]],
   aθ'[t] - Daθ[t], Mθ.Daθ'[t] + αc Mθ.Daθ[t] + Kθ + Fθ + Fθb,
   x4 Mη1.aη1'[t] + (invKη1 + invKη1NL).aη1[t] + Fη1,
   x4 Mη2.aη2'[t] + (invKη2 + invKη2NL).aη2[t] + Fη2];
 (*Equation with unknown parameters*)
```

```
invicD = Join[invq[0], invDq[0]];
```

```
invsol = ParametricNDSolve[Thread[Join[inveqD, invicD] == 0],
    invq[t][[All, 0]], {t, 0, tf}, {x1, x2, x3, x4},
    Method -> {"EquationSimplification" -> "Residual"},
    SolveDelayed → True][[All, 2]];(*Parametric Function*)
```

```
eqD = Join[au1'[t] - Dau1[t], Mu1.Dau1'[t] +
    αc Mu1.Dau1[t] + Ku1 + Fu[[1]] + Fub[[1]], au2'[t] - Dau2[t],
   Mu2.Dau2'[t] + αc Mu2.Dau2[t] + Ku2 + Fu[[2]] + Fub[[2]],
   aθ'[t] - Daθ[t], Mθ.Daθ'[t] + αc Mθ.Daθ[t] + Kθ + Fθ + Fθb,
   αμ Mη1.aη1'[t] + (Kη1 + Kη1NL).aη1[t] + Fη1,
   αμ Mη2.aη2'[t] + (Kη2 + Kη2NL).aη2[t] + Fη2];
```

```
icD = Join[q[0], Dq[0]];
```

```
tf = 100;
```

```
sol = NDSolveValue[Thread[Join[eqD, icD] == 0], {q[t]}, {t, 0, tf}, Method ->
    {"EquationSimplification" -> "Residual"}];(*Results from eqD*)
```

```
(*Solutions from direct problem*)

(*u1 @ 0, L/2, L*)

zu10 = Table[Take[First[sol], {1, Nn}].u2n[0], {t, 0, tf, 1}];(*u1 @ 0*)
(*zu1L2=Table[Take[First[sol],{1,Nn}].u2n[L/2],{t,0,tf,1}];
(*u1 @ L/2*)*)
zu1L = Table[Take[First[sol], {1, Nn}].u2n[L], {t, 0, tf, 1}];(*u1 @ L*)

(*u2 @ 0, L/2, L*)

zu20 = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[0], {t, 0, tf, 1}];
(*u2 @ 0*)

zu2L2 = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[L / 2], {t, 0, tf, 1}];
(*u2 @ L/2*)

zu2L = Table[Take[First[sol], {Nn + 1, 2 Nn}].u2n[L], {t, 0, tf, 1}];
(*u2 @ L*)

(*θ @ 0, L/2, L*)

zθ0 = Table[Take[First[sol], {2 Nn + 1, 3 Nn}].θn[0], {t, 0, tf, 1}];
(*θ @ 0*)

(*zθL2=Table[Take[First[sol],{2Nn+1,3Nn}].θn[L],{t,0,tf,1}];
(*θ @ L/2*)*)

zθL = Table[Take[First[sol], {2 Nn + 1, 3 Nn}].θn[L], {t, 0, tf, 1}];
(*θ @ L*)

(*Measurement, no noise*)

zu10nn = Take[zu10, {1, 101, 5}];(*No noise (u1@0)*)
zu1Lnn = Take[zu1L, {1, 101, 5}];(*No noise (u1@L)*)

zu20nn = Take[zu20, {1, 101, 5}];(*No noise (u2@0)*)
zu2L2nn = Take[zu2L2, {1, 101, 5}];(*No noise (u2@L/2)*)
zu2Lnn = Take[zu2L, {1, 101, 5}];(*No noise (u2@L)*)

zθ0nn = Take[zθ0, {1, 101, 5}];(*No noise (θ@0)*)
zθLnn = Take[zθL, {1, 101, 5}];(*No noise (θ@L)*)
```

Adding noise

```
zt = Table[t, {t, 0, tf, 5}];
```

```
(*σ=0.1*)

SetDirectory["C:\Users\Woo\Desktop\FEM\MMA\Noise\0.02"]
(*Directory should be changed with respect to the value of σ*)


(*Import Data*)

(* @ 0*)
zu10N = Transpose[Import["zu10N.dat"]][[2]];
zu1LN = Transpose[Import["zu1LN.dat"]][[2]];

(* @ L/2*)
zu20N = Transpose[Import["zu20N.dat"]][[2]];
zu2L2N = Transpose[Import["zu2L2N.dat"]][[2]];
zu2LN = Transpose[Import["zu2LN.dat"]][[2]];

(* @ L*)
zθ0N = Transpose[Import["zTheta0N.dat"]][[2]];
zθLN = Transpose[Import["zThetaLN.dat"]][[2]];


(*Results from the inverse problem EqD *)
z10 = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[1 ;; Nn]]).u1n[0],
    {t, 0, tf, 5}];(*u1@0*)
z1L = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[1 ;; Nn]]).u1n[L],
    {t, 0, tf, 5}];(*u1@L*)

z20 = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[Nn + 1 ;; 2 Nn]]).u2n[0],
    {t, 0, tf, 5}];(*u2@0*)
z2L2 = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[Nn + 1 ;; 2 Nn]]).u2n[0.5 L],
    {t, 0, tf, 5}];(*u2@L/2*)
z2L = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[Nn + 1 ;; 2 Nn]]).u2n[L],
    {t, 0, tf, 5}];(*u2@L*)

z30 = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[2 Nn + 1 ;; 3 Nn]]).θn[0],
    {t, 0, tf, 5}];(*θ@0*)
z3L = Table[(#[x1, x2, x3, x4][t] & /@ invsol[[2 Nn + 1 ;; 3 Nn]]).θn[L],
    {t, 0, tf, 5}];(*θ@L*)
```

Monitor process and get final results

```
(*Cost Function of clean measurements*)
CF =
  0.5 (#.# && (zu10nn - z10) + #.# && (zu1Lnn - z1L) + #.# && (zu20nn - z20) +
                  #.# && (zu2L2nn - z2L2) + #.# && (zu2Lnn - z2L) +
            #.# && (zθ0nn - z30) + #.# && (zθLnn - z3L));
```

```
(*Cost Function of noisy measurements*)
CFN = 0.5
    (#.# &@ (zu10N - z10) + #.# &@ (zu1LN - z1L) + #.# &@ (zu20N - z20) + #.# &@
                (zu2L2N - z2L2) + #.# &@ (zu2LN - z2L) +
            #.# &@ (z00N - z30) + #.# &@ (z0LN - z3L));

(*Inverse noisy case*)

t = {};
Monitor[NMinimize[{CFN, 1 ≤ x1 ≤ 5, 0.1 ≤ x2 ≤ 1, 0.1 ≤ x3 ≤ 1, 5 ≤ x4 ≤ 15},
    {x1, x2, x3, x4}, Method → {"NelderMead", "ReflectRatio" → 0.1},
    EvaluationMonitor :→ (AppendTo[t, {x1, x2, x3, x4, CF}])],
  t // Column] // AbsoluteTiming

(*Inverse without adding noise*)

t = {};
Monitor[NMinimize[{CF, 1 ≤ x1 ≤ 5, 0.1 ≤ x2 ≤ 1, 0.1 ≤ x3 ≤ 1, 5 ≤ x4 ≤ 15},
    {x1, x2, x3, x4}, Method → {"NelderMead", "ReflectRatio" → 0.8},
    EvaluationMonitor :→ (AppendTo[t, {x1, x2, x3, x4, CF}])],
  t // Column] // AbsoluteTiming

SetOptions[SelectedNotebook[],
 PrintingStyleEnvironment → "Printout", ShowSyntaxStyles → True]
```