Real time simulation in non linear dynamics : application in soft robots

Santiago Montagud

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGENIEUR

SPÉCIALITÉ : MECANIQUE

Par Santiago MONTAGUD PEREZ DE LIS

SIMULATION TEMPS REEL EN DYNAMIQUE NON LINEAIRE. APPLICATION A LA ROBOTIQUE SOUPLE

Sous la direction de : Pierre JOYOT
(Co-directeur : Francisco Chinesta)

Soutenue le 13 de décembre de 2018

Membres du jury :

AZAIEZ, Mejdi	Professeur des Universités	Bordeaux INP	Président
AMMAR, Amin	Professeur	ENSAM Angers	Rapporteur
Gonzalez, David	Professeur	I3A	Rapporteur
AGUADO, José V.	Ingénieur de recherche	Ecole Centrale de Nantes	Examinateur
CHINESTA, Francisco	Professeur des Universités	ENSAM Paris	Co-Directeur
JOYOT, Pierre	MCF	ESTIA	Directeur

Abstract

Title : Real time simulation of nonlinear structural dynamics. Application in soft robots.

The use of soft robots in industry has considerably increased thanks to its adaptability, a better interaction with humans and control techniques development. Soft robots development, comprehending energy absorption in collisions, improving tasks as grasping or allowing operations in confined places, needs new control strategies. Computational control requires structure deformation consideration and new methods are needed in order to simulate real time soft structures deformations. In this work we propose the use of parametric solutions in the resolution of the nonlinear problems in structural dynamics. Parametric solutions allow the pre-computation of a general solution, reducing the computational time. These solutions can be used in real time structural dynamics computations, which can be not affordable by classic methods.

Keywords : [Real time simulation, structural dynamics, nonlinear modeling, inverse problems]

Résumé

Titre : Simulation en temps réel de la dynamique des structures non linéaires. Application à la robotique souple.

L'utilisation de robots dans l'industrie a considérablement augmenté en raison de leur polyvalence, d'une meilleure interaction avec les utilisateurs et le développement de nouvelles techniques de contrôle. Le développement de robot à structure souple pouvant absorber des chocs, améliorer des manœuvres, prendre des objets ou permettre des opérations dans des espaces confinés nécessite de nouvelles stratégies de contrôle. En effet, les calculs nécessaires pour le contrôle demandent de considérer la déformation de la structure et donc de développer des méthodes permettant de simuler en temps réel la dynamique de ces structures déformables. Dans ce travail on propose l'utilisation de solutions paramétriques pour la résolution de problèmes de structures en dynamique non linéaire. Ces solutions paramétriques permettent le pré-calcul d'une solution générale. Celle-ci est ensuite utilisée pour calculer le comportement dynamique en temps réel, ce qui est impossible avec des techniques classiques.

Mots clés : [Simulation en temps réel, dynamique des structures, modélisation non linéaire, problèmes inverses]

ESTIA-Recherche, I2M

[ESTIA, Technopôle Izarbel, 64210 Bidart, France]

[I2M, UMR CNRS 5295, Talence, France]

Contents

List of Figures

List of Tables

Sommaire de la thèse

Le calcul de déplacements et forces en temps réel est toujours un défi dans le domaine de la mécanique. Différentes applications industrielles où la dynamique de structures est importante demandent actuellement des méthodes de calcul plus efficaces.

Le développement de la robotique souple ces dernières années a permis l'apparition des nouvelles applications innovantes. Des nouveaux matériaux et des nouvelles techniques de conception permettent la fabrication de robots souples. Cette possibilité permet une interaction avec les personnes réduisant le risque de dommage par l'absorption de l'énergie lors des impacts ou l'adaptation aux endroits confinés. Les domaines comme la médecine ou l'industrie sont deux exemples où cet avantage est profitable.

Le concept de laboratoires hybrides est un nouveau type de démarche où la simulation et l'expérimentation sont couplées et réalisées en même temps. L'idée derrière cette approche est de réduire le coût des expériences, simulant ce qui est raisonnable ou possible, et de coupler et enrichir cette simulation par des expériences pour conserver la richesse des phénomènes. Le couplage entre simulation et expérimentation est faite à l'aide d'actuateurs.

Le point commun entre ces applications industrielles comme la robotique et le concept de laboratoires hybrides se trouve dans la nécessité d'utiliser des algorithmes de calcul rapide, précis et adaptable aux possibles changements structurels. Les déplacements et les forces sont requis en temps réel pour surveiller et contrôler les structures. Les méthodes classiques ne sont pas adaptées aux nouvelles demandes, d'où la motivation d'approfondir le développement de nouveaux algorithmes.

La problématique principale à affronter est le calcul en temps réel des forces et des déplacements de structures dans des scénarios changeants. Deux défis peuvent être extraits de cette problématique : des calculs rapides qui permettent de suivre les structures en temps réel et des algorithmes capables de s'adapter aux changements dans les systèmes structurels.

Pour aborder la problématique de l'adaptabilité aux changements du système, une méthode de réduction de modèles est appliquée. Cette méthode, la Proper Generalized Decomposition (PGD), est très efficace dans l'obtention de solutions paramétriques en forme séparée, en étant très peu affectée par la difficulté de la multidimensionalité. Le principal avantage de cette méthode est la possibilité de construire l'approximation de la solution sans explorer tout le domaine paramétrique grâce à l'implantation d'un algorithme d'enrichissement itératif. Les solutions en forme séparée permettent des réponses rapides face aux changements des systèmes analysés. Les caractéristiques comme le module de Young ou l'amortissement peuvent être considérés comme paramètres, et les solutions prenant en compte ces paramètres peuvent être pré-calculés pour leur utilisation en temps réel. La condition pour que la PGD réussisse est la séparabilité paramétrique du problème. Pour cette raison, le problème est abordé dans l'espace fréquentiel, qui permet d'écrire l'équation de la dynamique de façon à ce que le paramètre *fréquence* soit en forme séparée, contrairement à sa version équivalente dans l'espace temporel, où le paramètre temps n'est pas séparable. La Théorie de Fourier peut être appliquée pour transformer la solution dans l'espace temporel ou fréquentiel selon la convenance.

Pour aborder la deuxième problématique, le calcul en temps réel, la théorie de la réponse impulsionnelle est considérée. La réponse impulsionnelle est la fonction équivalente dans l'espace temporel à la fonction de transfert, qui est définie dans l'espace fréquentiel. Cette théorie nous dit qu'un système peut être caractérisé par une réponse impulsionnelle, c'est-à-dire, si on connaît la réponse du système à une entrée unitaire on peut calculer la solution pour n'importe quelle entrée. Comme les réponses sont calculées par superposition, uniquement des systèmes linéaires invariants sont considérés. En outre, cette méthode permet de découpler en espace les calculs par degré de liberté, ce qui permet, par la suite, de réduire le temps de calcul et de faciliter le choix de l'intervalle de temps à utiliser.

Le calcul de déplacements et de forces est traité de facon séparée dans ce travail. Les deux variables sont obtenues à l'aide de la théorie de la réponse impulsionnelle, mais l'obtention de ces réponses est différente dans chaque cas. Par cohérence avec la bibliographie, le problème du calcul de déplacements dans ce travail est appelé problème direct et le problème du calcul de forces est appelé problème inverse.

Le calcul de déplacements en temps réel est abordé à l'aide de l'obtention de la réponse impulsionnelle directe (DIR par ses sigles en anglais). Pour obtenir

la DIR, la fonction de transfert est obtenue dans un premier temps par l'équation de la dynamique dans l'espace fréquentiel. Dans un deuxième temps, la Théorie de Fourier est appliquée pour transformer la fonction de transfert dans l'espace temporel. Pour calculer les forces en temps réel, la théorie de la réponse impulsionnelle est aussi appliquée en utilisant la réponse impulsionnelle inverse (IIR par ses sigles en anglais). Par définition des propriétés dans l'intégrale de convolution, une fonction est l'inverse de l'autre, donc un problème inverse doit être résolu.

Le problème inverse à résoudre est connu mathématiquement comme l'intégrale de Fredholm du premier type. Ce type de problème est mal-posé, c'est-à-dire, ne satisfait pas un des points suivants:

- La solution existe.
- La solution est unique.
- La solution dépend de façon continue des données.

De plus, l'opérateur résultant de la discrétisation temporelle de l'équation est généralement mal conditionné. Cet opérateur est une matrice de Toeplitz avec certaines caractéristiques qui doivent être prises en compte pour le calcul de l'inverse, et donc des méthodes spécifiques doivent être développées.

Les problèmes inverses apparaissent régulièrement dans le monde physique. L'obtention des images par tomographie, la prospection géologique ou l'astronomie sont trois exemples de domaines où des problèmes inverses apparaissent. Deux sous-problèmes peuvent être différenciés dans les problèmes inverses: l'identification de systèmes et les problèmes de déconvolution. Dans le premier cas, on connaît les entrées et sorties d'un système et on essaie de trouver le modèle représentatif. Dans le deuxième cas, la sortie et le noyau sont connus, et on essaie de calculer l'entrée.

Plusieurs méthodes sont appliquées pour résoudre ces deux problématiques, bien que conceptuellement différentes, elles peuvent être traitées numériquement d'une façon similaire. La régularisation de Tikhonov est une famille de méthodes qu'utilise la méthode de moindres carrées pour approximer la solution en ajoutant un paramètre de régularisation qui permet de contrôler le mauvais conditionnement de l'opérateur. Cette méthode, bien que très efficace, connaît certains problèmes lors de la manipulation de grands systèmes de matrices. Les méthodes itératives, comme la méthode de Landweber, sont mieux adaptées quand de grandes matrices interviennent. En utilisant un algorithme récursif pour approximer la solution,

l'inversion de matrices est évitée, et les variations entre itérations sont contrôlées au moyen d'un paramètre appelé paramètre de relaxation. Le point commun entre les deux méthodes précédentes vient de l'ajout du paramètre de régularisation ou de relaxation. Ce paramètre doit être testé à posteriori. C'est pourquoi certaines résolutions du problème doivent être réalisées pour trouver la bonne valeur de ce paramètre. La Troncation de la Décomposition par Valeur Singuliers (TSVD) est basée sur la décomposition de l'opérateur par un produit de trois matrices. Cette technique permet de tronquer la décomposition en éliminant les termes qui peuvent affecter la bonne solution. D'autres méthodes utilisent une approche statistique en ajoutant certaines connaissances sur la distribution probabiliste des données du problème. Les filtres de Wiener et de Kalman sont deux approches les plus utilisées pour leur simplicité d'implantation et leur efficacité.

Dans ce travail on utilise une approche de minimisation de moindres carres. Avec la méthode PGD, on construit une approximation de la solution par une procédure d'enrichissement où les premiers termes sont associés à une valeur plus élevée d'énergie. La troncation de l'enrichissement évite d'ajouter des termes qui pourraient contaminer la solution. Pour obtenir une meilleure approximation de la solution, certaines forces et déplacements sont créés synthétiquement au moyen de la DIR obtenue par le problème direct, et introduites dans la minimisation.

Pour la validation de la méthode une plaque de deux dimensions a servi de référence pour le calcul de la DIR et l'IIR en forme paramétrique. Le module de Young a été considéré comme paramètre, ainsi que la fréquence. Les résultats montrent que l'estimation de la force appliquée peut être faite de façon très précise.

La résolution de problèmes non linéaires est possible aussi en appliquant la théorie de la réponse impulsionnelle. Bien que la théorie impulsionnelle est généralement limitée aux systèmes linéaires invariants, il est possible de l'utiliser dans certains cas non linéaires. En effet, certains problèmes non linéaires peuvent être divisés en deux comportements : un comportement linéaire et un comportement non linéaire. Les réponses impulsionnelles peuvent être calculées dans un premier temps (phase *offline*) dans la subdivision avec un comportement linéaire. Dans ce travail, toutes les forces qui interviennent sur la structure, autant internes que externes, sont considérées dans les calculs des réponses impulsionnelles. La partie non linéaire est résolue dans un deuxième temps en temps réel (phase *online*). Cette approche du problème permet de minimiser les calculs de la phase *online*, permettant des plus rapides itérations pour résoudre les non linéarités.

Introduction

Real time control and monitoring structural dynamics has been an issue for the last decades. While the industry usually requires fast and accurate solutions, the solution of the equation of motion is computationally expensive. New numerical methods and technological development have reduced to some extent the existing problems, while new industrial demands continuously arise.

Soft robots [91] or hybrid laboratories [6] [23] are two examples of new industrial applications whose requirements are still a challenge. Hybrid laboratories are a new generation of laboratories where experiments are carried out coupled with real time simulations, saving the construction cost of the simulated part. The main idea is to compute the simulation in the part of the structure whose behaviour is well-established, and make the experiment in the part of the structure whose behaviour is complex or widely unknown. A coupling device, called actuator, is in charge of putting in communication the experiment and the simulation, transforming displacements in forces, and *vice versa*, forces into displacements. Soft robots require in many applications the control of the forces to achieve a certain space configuration, which involves the solution of nonlinear dynamic problems in real time.

Those are two examples of the challenges in structural dynamics nowadays: real time control and monitoring of forces and displacements, considering nonlinear behaviour. These requirements lead to the search of a resolution method that should allow fast computation, adaptability to changes in the system and integration of nonlinearities.

Time requirements in many industrial applications are in the order of a kHz, while the accuracy, depending on the application field, can vary from micrometers in robotics to millimeters in civil engineering applications. Classical methods, although their wide applicability, are not always suitable for real time computation. Time integration schemes may not be fast enough, and modal methods require complete solutions if some parameters of the analysis change. In this scenario, transfer

functions and impulse responses, while generally constrained to linear problems, can provide a fast scheme for the computation of both forces and displacements. Transfer functions are widely used in structural dynamics taking advantage of the Fourier space. Their application offers a simple and fast procedure to obtain the displacements when an external force is applied, and relevant information can be extracted from the obtained frequency spectrum. Its equivalence in the time domain is called impulse response, and allows processing the data stream as it is collected, thus offering a framework for fast computation, and avoiding a continuous transformation of the data stream to the Fourier's domain.

The adaptability to changes in the characteristics of the analysed structure is an interesting issue for the industry. Parametric or pre-computed solutions are a powerful tool when some features of the structure may change, or some optimization is searched among some structural parameters configuration. When a parametric solution is computed, changes in the structure only require a parametrization of the general solution. There exists some model order reduction techniques as *Proper Orthogonal Decomposition* (POD) [17] or *Reduced Basis Method* (RBM) [18] that produces good results solving parametric models. *Proper Generalized Decomposition* (PGD) is another model order reduction technique which has some advantages compared to the aforementioned [28].

The resolution of the equation of motion to monitor physical magnitudes in real time has been successfully applied in thermal problems [16]. In structural dynamics, the application of the PGD method to compute a parametric solution is also useful: a generalized transfer function (GTF) and a generalized impulse response (GIR) can be computed considering additional parameters, e.g. the Young's modulus, the damping factor or some boundary conditions. The GIR can be applied to real time monitoring of displacements or optimization problems, as the parametrization of the generalized solution is computationally inexpensive.

A more complicated scenario is the monitoring of the forces. The GIR allows to compute the displacements caused by an external applied force. The generalized inverse impulse response (GIIR) is the GIR counterpart: it allows the computation of the external applied force from measured displacements. Its calculation involves the solution of an inverse problem.

Inverse problems appear commonly in many fields of physics and engineering problems, where the problem is modelled by a *kernel* or a black box that relates inputs and outputs. In these problems, the output is available, and the input or the *kernel* are to be computed. When the relation is constructed by the convolution

integral, the problem of inferring the input or the *kernel* is called deconvolution. Among the different existing methods to solve the deconvolution problems, one can find the Truncated Singular Values Decomposition, Tikhonov regularization methods, Landweber iterative methods. Statistical approaches for the solution of more general inverse problems, based on filtering theory (e.g. Kalman filters), are also possible.

Impulse response theory is usually applied on linear invariant systems, but under certain conditions this kind of problems can be split in a linear behaviour part and a nonlinear behaviour part. Then, classic linear analysis can be applied on the linear part as an *offline* work, and the nonlinear computations can be done in the *online* phase.

The objective of this work is to take advantage of separated representations to improve the computational cost in structural dynamics applications. The combination of both generalized direct and inverse impulse responses is an interesting option for real time control and monitoring of structures. In this work is provided a method to compute displacements and forces in real time. This is possible by means of both GIR and GIIR, where the coupling between impulse response theory and generalized solutions allow a model order reduction of the structural system, and, in consequence, the reduction of the time computation cost. In this work, it is also provided an application of the method in some nonlinear contexts, where impulse responses can be applied to simulate nonlinear behaviours.

The document is organized as follows: Chapter 1 provides the state of the art of inverse problems, and more specifically a particular case that is the deconvolution problem. Chapter 2 deals with the solution of the dynamic problem in a parametric form. The proposed solution for the monitoring of displacements is a GIR built in parametric form that allows the computation of displacements in real time and fast solutions to face changes in the structural parameters. Chapter 3 addresses the problem of monitoring forces by the GIIR, where the theoretical approach is similar than in the Chapter 2, but an inverse problem must be solved. The PGD is proposed then to solve the deconvolution problem. Chapter 4 shows some applications of the GIR in nonlinear problems. Finally, conclusions are drawn in the last chapter.

Chapter 1

State of the art in inverse problems. The problem of deconvolution

In physics, an inverse problem appears when determining the input or the internal characteristics of a physical system from external measurements. This kind of problems appears in many physical systems where the output depends on the properties of the system and an input. Trying to infer the external applied force from measured displacements, or the structure of the soil from known electromagnetic inputs and outputs are two examples of inverse problems. The main characteristic of inverse problems is that the small errors introduced in the measurements can affect considerably the solution. In most of the problems, the analytic solution is not available, and the direct resolution of the problem can lead to a solution that is far from the exact one. By definition, inverse problems suffer of lack of information making impossible the reconstruction of the input, and numerical methods must be considered to find an approximation of the solution. Even if perfect measures (i.e. noiseless) could be done, one would still need to perform computations in infinite precision to avoid noise amplification. Therefore, approaches based on each problem characteristics must be considered. From the mathematical analysis to the numerical solution of the problem, numerous authors have contributed to the analysis of inverse problems.

The mathematical standard form for inverse problems in many physical systems can be formulated as the Fredholm integral equation of the first kind, which

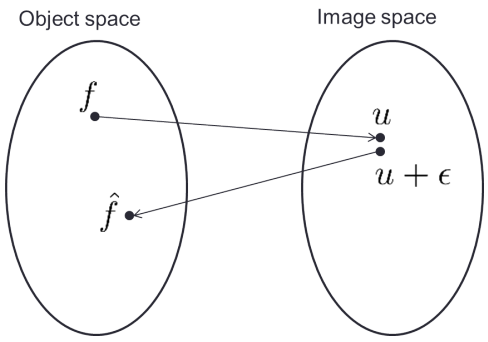

Figure 1.1: Graphic representation of the effect on the object space of small perturbations in the image space [49]

is written as [44]:

$$u(x) = \int_a^b h(x,y)f(y)\mathrm{d}y, \tag{1.1}$$

where $h(x,y)$ is called the *kernel*, and $f(y)$ and $u(x)$ are the input and the output functions respectively. The function $f(y)$ is to be determined, $h(x,y)$ is known in the interval $a \leq x, y \leq b$ and $u(x)$ is known in the interval $a \leq x \leq b$. Consider a physical system, e.g. a tomograph, which can be modelled and considered the *kernel*. The obtained images will be then the outputs $u(x)$, which will be probably obtained with a certain amount of errors due to the precision of the machine, defects, noise, etc. One wants to find the input $f(y)$, i.e. the function that represents the body organ, with the best possible accuracy.

A general idea of the inverse problems can be obtained from the consideration of the mapping between the input space S_i and the output space S_o: small perturbations in the object space lead to small perturbations in the image space. However, in the inverse mapping small perturbations in the image space can lead to big perturbations in the object space [49], leading to solutions far from expected. This feature is shown graphically in figure 1.1. Therefore, computation of $f(y)$ could lead to an erroneous representation of the body organ.

A classification can be made in the inverse problems field [35]. When the known variables are the input and the output the problem is classified as a system identification problem. When the known variables are the *kernel* and the output, the problem is classified as a deconvolution problem (the word *unfolding* can be also found in some textbooks). The previous definitions are summarized in table

	System design	**Convolution**	**System identif.**	**Deconvolution**
Input:	Known	Known	Known	Unknown
Kernel:	To be designed	Known	Unknown	Known
Output:	Predefined	Unknown	Known	Known

Table 1.1: Direct and Inverse Problems classification [35]

1.1, extracted from [35].

The use of system identification methods started to be important since 1960s. The augment of data collection and measurement techniques allowed new possibilities for physicians. New techniques were developed taking profit of the data to obtain models for unknown systems. These models considers the system as black boxes that relates inputs and outputs. The identification of this black box is the objective of the system identification techniques [57]. [86, 87] are two examples where system identification techniques are used to estimate unknown parameters of the models. The exact composition and materials of the earth layers is not usually accessible, or there not exist a model on a complex physic process. System identification techniques use the collected data, inputs and outputs, to try to identify the hidden physical system or the unknown parameters of the model.

Deconvolution problems appear in physical systems where the *kernel* is known and the problem can be modelled by the convolution theorem. Wave or heat equations allow to model with accuracy dynamics or heat transfer in structures, and are usually applied to compute displacements and temperatures.

Both system identification and deconvolution share numerical methods for resolution procedures. When the *kernel* is a convolution operator, the properties of the integral allow to interchange the kernel and the input roles, which makes no difference between both fields. In this work, despite this equivalence, the inverse problem will be referred as a deconvolution problem. The reason can be found on the structure of this thesis: a direct problem will be first solved by using the convolution operator, so for the sake of simplicity, is called deconvolution to the computation of the inverse problem which is treated subsequently, even if it is the kernel what will be computed.

The convolution operation appears in many physical systems. It is a special case of the Fredholm equation of the first kind where the *kernel* is a convolution

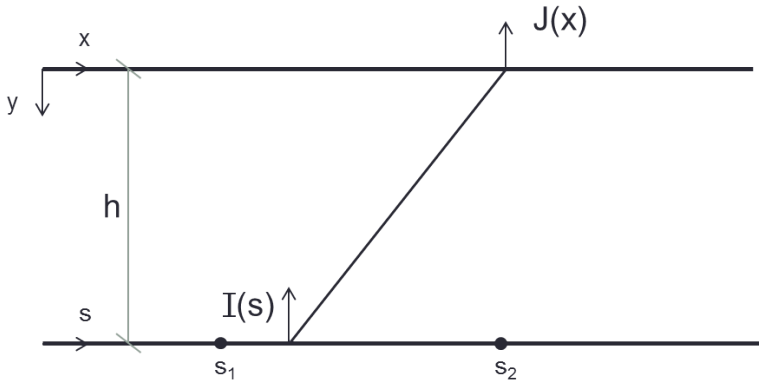

Figure 1.2: Graphical definition of the geomagnetic prospecting problem

operator. The integral in this case can be written as:

$$u(t) = \int_0^t h(t-\tau)f(\tau)\mathrm{d}\tau, \tag{1.2}$$

where the time parameter $t \in I_t$ is introduced and a range of $[0, T]$ is considered. For the range $[-\infty, 0]$ the input is zero, so it is the output, and causality holds. This equation is also known as Volterra equation of the first kind.

One of the most known problems in the community, used as an example of deconvolution problems, is the geomagnetic prospecting problem [44]. Consider that a deposit is situated under the surface of the earth in a plane stratum at a known distance h from the surface, as shown in figure 1.2.

This deposit emits a magnetic signal whose intensity determines its volume, then its economic value. Measures of the vertical component $J(x)$ of the magnetic field intensity at the surface $S(x, 0)$ are available. The field is created by a mass whose distribution $I(x)$ is required. The value of $J(x)$ in the surface S due to an infinitesimal part ds of the mass is:

$$\frac{sin\theta I(s)ds}{(\sqrt{h^2+(x-s)^2})^2} = \frac{hI(s)ds}{(h^2+(x-s)^2)^{\frac{3}{2}}}. \tag{1.3}$$

The deposit occupies a space from x_1 to x_2, then the total contribution is:

$$J_s(x) = \int_{x_1}^{x_2} \frac{I(s)ds}{(h^2+(x-s)^2)^{\frac{3}{2}}}. \tag{1.4}$$

Obtaining $I(s)$ from (1.4) is a deconvolution problem. Many other physical

systems require the resolution of the deconvolution problem, among them one can find astronomy [73], biology [71] or tomography [74].

Deconvolution is a non well-posed problem in the sense of Hadamard definition [41]. Hadamard defined the term well-posed problem when it fulfils the following conditions:

- The solution exists.
- The solution is unique.
- The solution depends continuously on the data.

When a problem does not fulfill at least one of the conditions, the problem is called ill-posed problem. Inverse problems are generally ill-posed problems, and consequently, the deconvolution problem is generally ill-posed. Furthermore, it is usually ill-conditioned, which means that small variations in the data produce big variations in the solution, so special theoretical and numerical approaches must be applied to solve the problem. The convolution problems are usually described as:

$$\tilde{u}(t) = \int_0^t h(t-\tau)f(\tau)\mathrm{d}\tau, \tag{1.5}$$

where $\tilde{u}(t) = u(t) + e$. This means that an error has been added to the output, and it is generally unknown. These errors come principally from measurement errors and truncation (identified as noise in signal treatment). The introduction of these errors can cause unacceptable deviations in the solution, even if they are negligible from a physical point of view.

The theory about the Fredholm equations of the first kind states that the integration of f with h is a smoothing operation, what means that high frequencies are damped. In fact, the convolution of f and h maximizes the differentiability of the functions f and h [36], in other words, the function u is smoother than f. By following this property, the deconvolution operation will undo the smoothing transformation. If in the convolution operation the high frequencies are damped, in the deconvolution the high frequencies are expected to be augmented. Thus, the relation between the errors and the high frequencies is a key point in the deconvolution procedure.

1.1 Discrete convolution and deconvolution. Properties of the Toeplitz matrix

The integral of convolution (1.2) can be written in discrete form as:

$$u(k) = \sum_{i=0}^{k-1} h(k-i) f(i) \tag{1.6}$$

The equivalent version in matrix form is:

$$\mathbf{u} = \mathbf{H}\mathbf{f}, \tag{1.7}$$

where $\mathbf{H}$ is the Toeplitz matrix of the impulse response $\mathbf{h}$, where the entries of the matrix have been added until a time step n, and the corresponding values in the range of time $[-\infty, 0]$ are set to zero. The Toeplitz matrix takes the form:

$$\mathbf{H} = \begin{bmatrix} h(1) & 0 & ... & 0 & 0 \\ h(2) & h(1) & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h(n-1) & h(n-2) & ... & h(1) & 0 \\ h(n) & h(n-1) & ... & h(2) & h(1) \end{bmatrix}. \tag{1.8}$$

This Toeplitz matrix is a $n \times n$ matrix, where only $2n-1$ elements are different and there are placed with a characteristic form: each element of h occupies one diagonal of the matrix. This feature makes Toeplitz matrices ill-conditioned, but the repetitive structure makes possible to derive appropriate methods to improve the resolution of the problems involving Toeplitz matrices. One of the main characteristics is the persymmetry, which means that they are symmetric across the anti-diagonal.

The previous Toeplitz matrix is a special case of the more general Toeplitz matrix definition [63]:

$$\mathbf{H} = \begin{bmatrix} h(1) & h(-1) & ... & h(-n-1) & h(-n) \\ h(2) & h(1) & ... & h(-n-2) & h(-n-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h(n-1) & h(n-2) & ... & h(1) & h(-1) \\ h(n) & h(n-1) & ... & h(2) & h(1) \end{bmatrix}. \tag{1.9}$$

A Toeplitz matrix is then symmetric when $h(j) = h(-j)$ for all j.

The direct resolution $\mathbf{f} = \mathbf{H}^{-1}\mathbf{u}$ by inverting the Toeplitz matrix $\mathbf{H}$ is usually precluded because it is usually ill-conditioned [43]. The relative error of the solution $\mathbf{u}$ is related to the condition number of $\mathbf{H}$. How large a condition number is acceptable depends on the resolution accuracy or the tolerance associated to the problem.

Let us consider the discrete form of the deconvolution problem to obtain $\mathbf{f}$, which can be easily deduced from equation (1.6):

$$f(k) = \frac{u(k)}{h(1)} - \frac{\sum_{i=1}^{k-1} u(i+1)h(k-i+1)}{h(1)}. \tag{1.10}$$

It can be observed the importance of the errors in measured $\mathbf{u}$. Errors in the input $\mathbf{u}$ are inherent to discrete calculus, and they are reintroduced in the subsequent computations, leading to an unstable resolution. The equation (1.10) shows that the importance of the errors in the solution grows proportionally to the factorial of the number of computations k.

1.2 Numerical solution methods

The existence of inherent errors in the inverse problems can be considered as a lack of information, and this lack of information can lead to a lack of uniqueness of the solution. The lack of information can never be avoided, so the approaches are based on the addition of information to the problem: statistical knowledge of the noise, cost function energy control, etc.

Different techniques exist in the literature to solve the lack of information, which try in some way to add information or constraint in the resolution process to reduce the subspace of possible solutions. Ivanov, Tikhonov and Phillips [50] proposed a minimization problem, which is actually known as regularization method. Iterative variants of the regularization methods, where the solution is sought in an iterative procedure until a certain prescribed error is achieved, also exist [47]. The *Truncated Singular Value Decomposition* can also be used to deal with the ill-conditioning of the Toeplitz matrices [63]. Another kind of approaches comes from the statistical analysis of the problem by introducing the theory of probabilities to cover the lack of information [48]. There has been also attempts to solve the deconvolution problem in real time, where some methods can be found in [54][55].

Fourier based analysis

The resolution of the problem in the frequency domain is based in the Fourier's theory [8]. With the development of the Fast Fourier Transform, the computational cost of the transformation between the time domain and the frequency domain was considerably reduced. Consequently, a wide family of methods and applications has emerged in the last decades taking profit of its benefits. Signal processing, seismography, tomography, etc. are some of the fields where it is applied [21].

The resolution of the problem in the frequency domain eliminates the bad conditioning which arises when inverting the Toeplitz matrix form of $\mathbf{H}$. The equations are transformed to the frequency domain, where the convolution becomes a multiplication [49]:

$$\hat{\mathbf{u}} = \hat{\mathbf{h}}\hat{\mathbf{f}}, \tag{1.11}$$

where the symbol $(\hat{})$ denotes that the function lives in the frequency domain, so $\hat{\mathbf{u}}, \hat{\mathbf{h}}$ and $\hat{\mathbf{f}} \in \mathbb{C}$. To obtain $\hat{\mathbf{f}}$, a simple division is made, which means that the deconvolution in the frequency domain is transformed into a division between the functions $\hat{\mathbf{u}}$ and $\hat{\mathbf{h}}$ as follows:

$$\hat{\mathbf{f}} = \frac{\hat{\mathbf{u}}}{\hat{\mathbf{h}}}. \tag{1.12}$$

The existence of a solution depends on the r.h.s. of the equation: it must be well-defined, which is not always the case. Consider the case when $\hat{\mathbf{h}}$ is zero or near zero for one or more values of ω: then the solution in frequency will have singularities and the inverse Fourier transform may not exist.

Fourier analysis gives a useful view about the deconvolution problem. Remember that the acquired data is usually obtained in discrete form, and the analytic solution is not available. Errors in the data is unavoidable as the precision of the machines is not infinite. If no errors were added to the output data, the deconvolution would give back the exact searched function f, but usually is not the case. Consider then the equation (1.12) and add the noise errors $\hat{\mathbf{u}}_\epsilon$ in the acquired data:

$$\hat{\mathbf{f}} = \frac{\hat{\mathbf{u}} + \hat{\mathbf{u}}_\epsilon}{\hat{\mathbf{h}}} = \frac{\hat{\mathbf{u}}}{\hat{\mathbf{h}}} + \frac{\hat{\mathbf{u}}_\epsilon}{\hat{\mathbf{h}}}. \tag{1.13}$$

Physically motivated transfer functions tend usually to zero when the frequency tends to infinity. On the other hand, transfer functions related to noise usually show a nearly flat spectrum (e.g. white noise has equal intensity at all frequencies). The division of the high frequency zone of the data spectrum by the high frequency

zone of the transfer function spectrum will amplify the effect of the noise, and will dramatically affect the solution.

Truncated SVD

The singular value decomposition is a factorization technique for matrices. Over the field of real numbers, the SVD decomposition is defined as:

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{1.14}$$

where $\mathbf{U}$ is the matrix of the left-singular vectors, $\mathbf{D}$ is a pseudo-diagonal matrix containing the singular values σ and $\mathbf{V}$ is a matrix that contains the right-singular vectors. The singular vectors are stored by columns in $\mathbf{U}$ and $\mathbf{V}$, so their columns are orthonormal:

$$\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}, \tag{1.15}$$

where $\mathbf{I}$ is the identity matrix. The decomposition can be also written as:

$$\mathbf{H} = \sum_{i=1}^{n} \boldsymbol{\chi}_i \sigma_i \boldsymbol{\eta}_i, \tag{1.16}$$

where n is the rank of $\mathbf{H}$.

Then, the right and left singular vectors are stored in matrices $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{U} = (\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_n), \qquad \mathbf{V} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n), \tag{1.17}$$

and the right and singular vectors are orthogonal following:

$$\boldsymbol{\chi}_i^T \boldsymbol{\chi}_j = \boldsymbol{\eta}_i^T \boldsymbol{\eta}_j = \delta_{ij}. \tag{1.18}$$

The singular values reveal some important properties of $\mathbf{H}$:

- The singular values of $\mathbf{H}$ decay with a certain slope until it reaches the machine precision times σ_1, where the slope tends to be horizontal.
- The previous statement means that the condition number is related to the machine precision.
- There is no gap in the singular values spectrum, and it usually follows a harmonic or a geometric progression.

- The number of sign changes of the singular vectors is inversely proportional to the value of the corresponding singular value.

The number of non-zero singular values is the rank of $\mathbf{H}$, and the condition number of $\mathbf{H}$ can be computed from:

$$cond(\mathbf{H}) = \frac{\sigma_1}{\sigma_n}, \tag{1.19}$$

where σ_1 is the largest singular value and σ_n is the minimum non-zero singular value.

The SVD permits to stablish an important relation between the matrix $\mathbf{H}$ and the singular values and singular vectors:

$$\mathbf{H}\boldsymbol{\eta}_i = \sigma_i \boldsymbol{\chi}_i, \tag{1.20}$$

where the $\mathcal{L}_2$ norm of $\mathbf{H}\boldsymbol{\eta}_i$ is:

$$||\mathbf{H}\boldsymbol{\eta}_i|| = \sigma_i. \tag{1.21}$$

These properties can be exploited to analyse the problem of deconvolution. Consider then the problem of the equation (1.7):

$$\mathbf{H}\mathbf{f} = \mathbf{u}. \tag{1.22}$$

By using (1.20), the terms $\mathbf{u}$ and $\mathbf{f}$ can be expanded as:

$$\mathbf{f} = \sum_{i=1}^{n} (\boldsymbol{\eta}_i^T \mathbf{f}) \boldsymbol{\eta}_i \,; \qquad \mathbf{u} = \sum_{i=1}^{n} (\boldsymbol{\chi}_i^T \mathbf{u}) \boldsymbol{\chi}_i, \tag{1.23}$$

and substituting (1.23) in (1.20), is straight to write the equation:

$$\mathbf{H}\mathbf{f} = \sigma_i \sum_{i=1}^{n} (\boldsymbol{\eta}_i^T \mathbf{u}) \boldsymbol{\chi}_i. \tag{1.24}$$

Finally, the approximated solution can be written as a decomposition in singular values and singular right and left vectors as:

$$\mathbf{f} = \sum_{i=1}^{n} \frac{\boldsymbol{\chi}_i^T \mathbf{u}}{\sigma_i} \boldsymbol{\eta}_i. \tag{1.25}$$

This decomposition allows to compute an approximation of the solution by truncating the sum. The truncation can be made by controlling the singular values

and the corresponding left and right eigenvectors. In fact, the solution is governed by the relation between $\boldsymbol{\chi}_i^T\mathbf{u}$ and the singular values. The Discrete Picard Condition [64] states that the quantity $|\boldsymbol{\chi}_i^T\mathbf{u}|$ decays faster than the singular values σ_i until they arrive to the level of the machine precision. In this point, the round noise starts to govern over the SVD components, and the information obtained in the decomposition is no longer useful. It is evident that the terms $\boldsymbol{\chi}_i^T\mathbf{u}$ and the singular values must be monitored to avoid the introduction of spurious terms on the approximation of the solution. If the quantity $\boldsymbol{\chi}_i^T\mathbf{u}$ does not decay faster than the singular values, there is not any function square integrable f that is a solution of the problem such that:

$$\int_{-\infty}^{\infty} |f(t)|^2 \mathrm{d}t < \infty, \tag{1.26}$$

and any effort solving the problem should be avoided.

Therefore, useful information can be extracted from the SVD decomposition of matrix $\mathbf{H}$. The first singular values and singular vectors still carry the most important information: the effort must be then in the separation of the valuable information and the spurious information, by applying a truncation:

$$\tilde{\mathbf{f}} = \sum_{i=1}^{N} \frac{\boldsymbol{\chi}_i^T\mathbf{u}}{\sigma_i}\boldsymbol{\eta}_i, \tag{1.27}$$

where $N < n$. This approximation is known as Truncated Singular Values Decomposition. The election of the truncation number N is quite simple: a SVD decomposition must be done on matrix $\mathbf{H}$, and the quantities $\boldsymbol{\chi}_i^T\mathbf{u}$ must be computed. The number N corresponds to the change of the slope of the quantities $\boldsymbol{\chi}_i^T\mathbf{u}$. There exists methods that reduce the computational cost of the previous method [65].

The main problem in the TSVD method is the computational cost: the decomposition of large matrices can be unaffordable for the actual computational power.

Regularization method

At the same time, Ivanov and Tikhonov published similar techniques to solve the Fredholm equation of the first kind [44][49], and their theories can be also applied to the deconvolution problem. These methods are known as Tikhonov regularization method or just regularization method. Those techniques use a least squares approximation with a prescribed energy weighted by a parameter μ to resolve the

problem. As the solution is not unique, a least squares technique is applied to find a unique solution, which will be the best approximation to the solution in the constrained problem. In the discrete form of the regularization method, $\mathbf{f}$ is sought in order to minimize the functional Π:

$$\Pi(\mathbf{f}) = ||\mathbf{Hf} - \mathbf{u}||^2, \tag{1.28}$$

with a prescribed energy for $\mathbf{f}$:

$$E^2(\mathbf{f}) = ||\mathbf{f}||^2 \leq E^2. \tag{1.29}$$

The bound E^2 of $\mathbf{f}$ must be in the order of the known $\mathbf{u}$. A bound for $\mathbf{u}$ is not usually a simple guess. Consider a structural dynamic problem, where the structure characteristics are known. In most of the engineering problems, an order of magnitude of the displacements is usually known, so an order of magnitude of the displacements is also guessed. This added information is what can in some way substitute the lack of information.

This problem can be solved by applying the method of Lagrange multipliers, where the condition is introduced in the same functional of the problem as:

$$\Omega = ||\mathbf{Hf} - \mathbf{u}||_2^2 + \mu^2||\mathbf{f}||_2^2. \tag{1.30}$$

where μ is an arbitrary positive real number. The main problem of the method is the estimation of the parameter μ. Great values of this parameter will lead to a solution which may differ significantly from the real solution, while very small values will lead to the ill-conditioned problem, where the solution exists but can be far from what expected. Several methods exist on the research of the optimal value of μ [61][62][60]. In the middle of the extreme values, there exists infinite possible solution for the problem.

The election of the regularization parameter is still a challenge for the research [67]. The different possible solutions are related to the information one has about the problem. The distribution of the error, or the assumptions that can be made about the correlation with the solution leads to different methods. The discrepancy principle searches the regularization parameter μ as a function of the error norm, which must be known or assumed known. The cross validation method considers that the noise is uncorrelated with the solution, and therefore some previous information is needed. One of the most known methods is the L-curve criterion [34][66], which consists in computing the regularized solution for several different

values of the parameter $\mu > 0$. The plot of the $\mathbf{f}_\mu$ versus the norm of the residual is a curve in a more or less *L* shape, where a minimum can be found along the curve. [34][66] recommend using the corresponding value to the minimum of the plot, which is a good choice in the balance between the solutions influenced by the ill-conditioning of the problem and the solutions influenced by the noise.

The SVD decomposition can be used to better understand the properties of the regularization method. The approximation of the solution can be written as:

$$\mathbf{f}_\mu = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \mu^2} \frac{\boldsymbol{\chi}_i^T \mathbf{u}}{\sigma_i} \boldsymbol{\eta}_i. \tag{1.31}$$

The elements in the quotient $Q_i = \frac{\sigma_i^2}{\sigma_i^2+\mu^2}$ are called Tikhonov filter factors, and they satisfy the condition $0 \leq Q_i \leq 1$. Comparing (1.31) with (1.25), is directly deduced that the quotients Q_i control the behaviour of the SVD solution $\frac{\boldsymbol{\chi}_i^T \mathbf{u}}{\sigma_i} \boldsymbol{\eta}_i$. For a fixed value of the regularization parameter μ between the maximum σ_1 and minimum σ_n values of the singular values σ_i, the quotient tends to 1 when $\sigma_i \gg \mu$; by the other hand, when $\sigma_i \ll \mu$ the quotient tends to zero. That means that the larger singular values and vectors have more contribution to the solution than the smaller ones, expecting that the useful information is kept in the computation of the solution and the noise contribution is damped. From this statement is deduced that for values of μ near σ_i, both TSVD and Tikhonov solutions should be very close.

The numerical resolution of equation (1.28) is not suited for numerical computations. It is usually done by two different formulations: the normal equations for the least squares problem:

$$(\mathbf{H}^T \mathbf{H} + \mu^2 \mathbf{I}) \mathbf{f} = \mathbf{H}^T \mathbf{u}, \tag{1.32}$$

and extended matrix version for the least squares problem:

$$min \left|\left| \begin{bmatrix} \mathbf{H} \\ \mu \mathbf{I} \end{bmatrix} \mathbf{f} - \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \right|\right|_2, \tag{1.33}$$

where $\mathbf{I}$ is the identity matrix.

Another interesting method can be applied when $\mathbf{H}$ is symmetric and positive definite that allow to work with a smaller system of equations:

$$(\mathbf{H} + \mu \mathbf{I}) \mathbf{f} = \mathbf{u}, \qquad \mu \geq 0. \tag{1.34}$$

A Cholesky decomposition can be applied on $\mathbf{H}$, and then the solution can be computed as:

$$\mathbf{f} = \sum_{i=1}^{n} \frac{\psi_i \mathbf{u}}{\phi_i + \mu} \psi_i, \tag{1.35}$$

where ψ_i and ϕ_i are the components of the eigenvalue decomposition:

$$\mathbf{H} = \sum_{i=1}^{n} \psi_i^T \phi_i \psi_i. \tag{1.36}$$

The equation presented in (1.28) it is not the best option for deconvolution problems. For a better approximation of the solution is recommended to introduce in the formulation the first or second derivative. Therefore the minimization is done on the $\mathcal{L}_2$ norm of the derivative weighted by the regularization parameter. When using the first or second derivatives, the corresponding matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively, have a form:

$$\mathbf{S}_1 = n^{-1} \begin{Bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & -1 & 1 & 0 \\ 0 & \dots & 0 & -1 & 1 \end{Bmatrix} \tag{1.37}$$

$$\mathbf{S}_2 = n^{-2} \begin{Bmatrix} 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \end{Bmatrix}, \tag{1.38}$$

which are not square: S_1 dimension is $(n-1) \times n$ and S_2 dimension is $(n-2) \times n$. Nevertheless, the introduction of these matrices in the minimization equation:

$$\min(||\mathbf{Hf} - \mathbf{u}||_2^2 + \mu^2 ||\mathbf{Sf}||_2^2), \tag{1.39}$$

leads to a matrices product $S_1^T S_1$ and $S_2^T S_2$ which result is a square matrix.

Iterative methods

Several iterative methods have been developed in order to solve inverse problems. Landweber method [52] is possibly the reference on the subject, but many others can also be found in the literature as Richardson method [45].

Landweber method is an iterative method to solve the Fredholm equation of the first kind, and hence for the deconvolution problem, based on the transformation of the integral equation in a fixed point iteration method. The main advantage in this method is that any matrix inversion is avoided, therefore it is suitable for large or sparse systems. On the other hand, an *a priori* parameter controls the variation of the subsequent iterations, and bad choices could enlarge the time computation.

The iterative Landweber equation takes the form:

$$\mathbf{f}^k = \mathbf{f}^{k-1} + \lambda H^T(\mathbf{u} - \mathbf{H}\mathbf{f}^{k-1}), \tag{1.40}$$

where λ is the relaxation parameter and k defines the iteration step. The relaxation parameter must satisfy:

$$0 < \lambda < \frac{2}{||\mathbf{H}^t\mathbf{H}||_2}. \tag{1.41}$$

An interesting analysis of the method can be made from the SVD decomposition. The iterative equation can be written as:

$$\mathbf{u}^k = \sum_{i=1}^{n}(1 - (1 - \lambda\sigma_i^2)^k)\frac{\boldsymbol{\chi}_i^T\mathbf{u}}{\sigma_i}\boldsymbol{\eta}_i. \tag{1.42}$$

Looking at (1.42) can be verified that the iteration number k is the equivalent in Landweber to the regularization parameter in Tikhonov's method. In the first iterations the largest components have more weight in the equation than the smaller ones, which are incorporated to the iteration scheme with the increase of k. The convergence of the method can also be deduced from (1.42): the increase of the number of iterations adds practically the half of the small filter factors $(1 - (1 - \lambda\sigma_i^2)^k)$.

In the family of Landweber methods can be found the *Conjugate Gradient method*. Hestenes and Stiefel [68] proposed a method for the resolution in an iterative scheme to the inverse problem associated with a least squares problem, which comprehends the following equations:

$$\begin{aligned} a_k &= \frac{||H^T r^{(k-1)}||_2^2}{||H s^{(k-1)}||_2^2} \\ f^k &= f^{k-1} + a_k s^{k-1} \\ r^k &= r^{k-1} - a_k H s^{k-1} \\ u_k &= \frac{||H^T r^{(k)}||_2^2}{||H^T r^{(k-1)}||_2^2} \end{aligned}$$

$$s^k = H^T r^k + u_k s^{k-1}. \tag{1.43}$$

In the group of equations (1.43) the two vectors r and s are defined by:

$$r^0 = u - Hf^0 \qquad ; \qquad s^0 = H^T r^0. \tag{1.44}$$

This algorithm can be applied both in the formulation described in (1.33) and in the standard form $\mathbf{Hf} = \mathbf{u}$. For the first formulation a good pre-conditioner is suitable to be used, and an adequate special conditioner is still a challenge for the researchers. By the other hand, the second approach is interesting as it preserves the original form of the equation, and the regularization is provided by the iteration number k. The convergence of the CG method is faster than in the Landweber method, but is not exempt of troubles: some conditions are more suitable to be fulfilled before applied the method to improve the success of the regularization, but they not guaranty the desired result:

- The singular values should decay gradually to zero
- The slope should tend to vertical shape
- The quantities Q, as defined in 1.2, should decay faster than the singular values.

Methods based on statistical information

As explained before, the deconvolution problem lacks of information to reconstruct the original requested function. There is a family of methods that address the problem from a statistical point of view [72], where the lack of information is in some way substituted by a statistical knowledge of the variables of the problem.

The Bayesian approach to resolve inverse problems relies on conditional probabilities [81]. Considering the problem modelled by $HF = U + \epsilon$, ϵ defines an additive noise with continuous probability distribution, F and U are random values and f and u are samples. Using Bayes formula, a *posterior distribution* can be defined as:

$$p_{F|U}(f|u) \sim p_F(f) p_{U|F}(u|f), \tag{1.45}$$

where the *probability density function* p_F is defined by:

$$p_F(f) \geq 0 \qquad \text{for all } \ f \in \mathbb{R},$$

$$\int_{-\infty}^{\infty} p_F(f)df = 1, \tag{1.46}$$

and where $p_{F|U}(f|u)$ is the conditional probability. The density $p_{U|F(u|f)}$ is called likelihood distribution and is related to data misfit. For a given u, the density $p_{F|U(f|u)}$ specifies the probability of the measured u to be caused by f. The objective of statistical inversion is to compute the posterior distribution $p_{F|U}(f|u)$ which is the solution of the inverse problem. The difficulties to compute the density functions can be approached by two methods:

- *Maximum a posteriori* estimation, which implies an optimization problem:

$$\text{arg max} p_{U|F}(u|f). \tag{1.47}$$

- The conditional mean estimate, which implies n-dimensional space integrations:

$$\int x p_{U|F}(u|f). \tag{1.48}$$

The Wiener filter is one of the simplest examples of Bayesian methods. By considering Gaussian processes, the following assumptions are made:

- The probability distribution of the noise is approached by a Gaussian distribution.
- Vector u is also considered to be a Gaussian vector.
- Noise and vector u are considered independent variables, so the cross-variance matrix is zero.

Kalman filter is one of the most applied methods in the Bayesian framework, concretely in the *Maximum a posteriori* methods [82]. The method is based on estimations of the unknown variables by inferring probability information from measures. It is a two steps algorithm, where first the estimation of the variable is computed, and when the next time step values arrive, it updates the prediction. Kalman filter is optimal in the sense that if the noise has a Gaussian distribution,

the filter minimizes the mean square error of the estimated parameters. If the noise is not Gaussian, Kalman filter is the best linear estimator. When the noise is not considered to be Gaussian some other approaches are possible [76]. The Kalman filter is related to a recursive weighted least squares method [83].

Let us consider the dynamics equation, which is explained in detail in 2.1. The transformation into a state-space representation, the following substitution is made [72]:

$$\alpha_1(t) = u(t),$$

$$\alpha_2(t) = \dot{u}(t). \tag{1.49}$$

Considering the derivatives of the state vector and substituting in the dynamics equation (2.3) one arrives to a linear state-space model or Kalman filter model:

$$\dot{\alpha}(t) = A\alpha(t) + Bf(t),$$

$$u(t) = M\alpha(t) + Nf(t). \tag{1.50}$$

The Kalman filter estimates the state of α in the linear process defined by:

$$\alpha_{k+1} = A\alpha_k + Bf_k + \eta_k,$$

$$u_{k+1} = M_k\alpha_k + \xi_k, \tag{1.51}$$

where the subscript k the corresponding time step of $t_k = k\Delta t$, and η_k and ξ_k are prior estimate error and the measurement error respectively. These two equations summarize the concept of the algorithm: first an estimate is given and then it is corrected.

1.3 Numerical methods for Toeplitz matrices

Some special algorithms have been developed by taking in account the special structure of the Toeplitz matrix that is detailed in 1.1. One of the most known are the Levinson algorithms, which are developed for standard Toeplitz matrices as Levinson-Trench-Zohar [70] and symmetric Toeplitz as Levinson's algorithm [69].

This algorithm improves the methods based on Tikhonov's regularization, which can be computationally complex for large matrices.

Consider the case of an upper triangular Toeplitz matrix H and a matrix L with dimensions $p \times n$ where $p \leq n$ and zeros under the diagonal, which is representative of the scheme detailed in 1.2. Considering $n = 4$ the matrix is written as:

$$\begin{bmatrix} \mathbf{H} \\ \lambda L \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ 0 & h_1 & h_2 & h_3 \\ 0 & 0 & h_1 & h_2 \\ 0 & 0 & 0 & h_1 \\ l_1 & l_2 & l_3 & l_4 \\ 0 & l_1 & l_2 & l_3 \\ 0 & 0 & l_1 & l_2 \end{bmatrix}. \tag{1.52}$$

The previous matrix can be converted into a diagonal matrix by applying n Givens transformations [77]. The algorithm starts by applying a rotation to rows 1 and $n + 1$, to eliminate the element l_1. As this operation changes all the elements in both two rows, a superscript indicates the number of operations that an element has been submitted to. The same operation is applied to the row pairs $(j, n + j)$ where $j = 2, \ldots, p$. Notice that, regarding the Toeplitz matrix structure, new rows are versions of rows 1 and $n + 1$. The following matrix T_2 is obtained:

$$H^2 = \begin{bmatrix} h_1^1 & h_2^1 & h_3^1 & h_4^1 \\ 0 & h_1^1 & h_2^1 & h_3^1 \\ 0 & 0 & h_1^1 & h_2^1 \\ 0 & 0 & 0 & h_1^1 \\ 0 & l_2^1 & l_3^1 & l_4^1 \\ 0 & 0 & l_2^1 & l_3^1 \\ 0 & 0 & 0 & l_2^1 \end{bmatrix}, \tag{1.53}$$

where the first diagonal of the submatrix L has been eliminated. Applying the same

operations to all rows except the first one, the following structure is obtained:

$$H^2 = \begin{bmatrix} h_1^2 & h_2^2 & h_3^2 & h_4^2 \\ 0 & h_1^2 & h_2^2 & h_3^2 \\ 0 & 0 & h_1^2 & h_2^2 \\ 0 & 0 & 0 & h_1^2 \\ 0 & 0 & l_3^2 & l_4^2 \\ 0 & 0 & 0 & l_3^2 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{1.54}$$

The procedure is repeated until an upper diagonal matrix is obtained. The n steps of the algorithm involve around $8n^2$ operations.

If the matrix H is not upper diagonal, a more complex algorithm, developed by Bojanczyk [78] and [79], can be applied in order to obtain the matrix R of the QR factorization:

$$\begin{bmatrix} \mathbf{H} \\ \lambda L \end{bmatrix} = QR. \tag{1.55}$$

Considering general L matrices, a partition is done as:

$$\begin{bmatrix} \mathbf{H} \\ \lambda L \end{bmatrix} = \begin{bmatrix} h_0 & u^T \\ v & \bar{H} \\ \lambda & 0 \\ 0 & \lambda I \end{bmatrix} = \begin{bmatrix} \bar{H} & \bar{u} \\ \bar{v}^T & h_{m-n} \\ \lambda I & 0 \\ 0 & \lambda \end{bmatrix}. \tag{1.56}$$

where $\bar{H}$ is a submatrix taking from the beginning or from the end the range of elements $[m-1, n-1]$, I is the identity matrix, and the vectors $u, v, \bar{u}$ and $\bar{v}$ must be taken in order to fulfill H. Matrix R must be also be partitioned as:

$$R = \begin{bmatrix} r_{11} & z^T \\ 0 & R_b \end{bmatrix} = \begin{bmatrix} R_t & \bar{z} \\ 0 & r_{mn} \end{bmatrix}. \tag{1.57}$$

By applying:

$$\begin{bmatrix} \mathbf{H} \\ \lambda L \end{bmatrix}^T \begin{bmatrix} \mathbf{H} \\ \lambda L \end{bmatrix} = (QR)^T QR, \tag{1.58}$$

and remembering the orthogonality $Q^T Q = I$, the nonzero blocks of R can be computed from the following deduced relations:

$$r^2 = h_0^2 + v^T v + \lambda^2,$$

$$z^T = \frac{h_0 u^T + v^T \bar{H}}{11},$$

$$R_b^T R_b = R_t^T R_t + u u^T - \bar{v}\bar{v}^T - z z^T. \tag{1.59}$$

The first row of R can be computed by taking the first two equations, and the submatrix R_b can be then computed as it depends on R_t. Once the first row of R_t is computed, the first row of R_b is calculated: a Givens rotation G^{1a} is applied on the first row of R_t and u^T to eliminate the first element of u:

$$G^{1a} \begin{bmatrix} R_t \\ u^T \end{bmatrix} = \begin{bmatrix} \bar{R}_t \\ (u^1)^T \end{bmatrix}, \tag{1.60}$$

where $u^1(1) = 0$.

Matrix $\bar{R}_t$ is the same as R_t but the first row has been changed. Determining now a second Givens rotation G^{1b} to eliminate the first element of z, one arrives to:

$$G^{1b} \begin{bmatrix} \bar{v}^t \\ z^T \end{bmatrix} = \begin{bmatrix} \hat{v}^T \\ (z^1)^T \end{bmatrix}, \tag{1.61}$$

and finally a hyperbolic rotation H^1 is applied to the first row of $\bar{R}$ and $\hat{v}$ to eliminate the first element of $\hat{v}^T$:

$$H^1 \begin{bmatrix} \bar{R}_t \\ \hat{v}^T \end{bmatrix} = \begin{bmatrix} R_t^1 \\ (\bar{v}^1)^T \end{bmatrix}, \tag{1.62}$$

where $\bar{v}^1(1) = 0$. Notice that the first row of R_b has been computed, and therefore the second row of R. Applying now the algorithm to the second row of R_t^1 to eliminate the second element of z^1, Givens rotations are again applied:

$$G^{2a} \begin{bmatrix} R_t^1 \\ (u^1)^T \end{bmatrix} = \begin{bmatrix} \bar{R}_t^1 \\ (u^2)^T \end{bmatrix}, \tag{1.63}$$

and

$$G^{2b} \begin{bmatrix} \bar{v^1}^t \\ (z^1)^T \end{bmatrix} = \begin{bmatrix} (\hat{v}^1)^T \\ (z^2)^T \end{bmatrix}, \tag{1.64}$$

where $z^2(1) = 0$, and then the hyperbolic rotation H^2 is applied in order to eliminate the first element of $\bar{v}^2$:

$$H^2 \begin{bmatrix} (\bar{R})_t^1 \\ (\hat{v}^1)^T \end{bmatrix} = \begin{bmatrix} R_t^2 \\ (\bar{v}^2)^T \end{bmatrix}. \tag{1.65}$$

The first three rows of matrix R have been computed at this step, and the procedure can be continued to achieve the complete triangularization. The Tikhonov regularization must be computed by applying:

$$R^T R u_\lambda = H^T f. \tag{1.66}$$

As λ is chosen to stabilize the condition number of the least square minimization, the solution of (1.66) is stable. If H is symmetric and positive definite then the Franklin's method [80] can be used by solving the system:

$$(H + \lambda I)u = f, \tag{1.67}$$

which can be solved by applying Levinson's recursion [70].

Chapter 2

Monitoring of displacements

Structural dynamics appears in various different fields as civil engineering, industrial mechanics, robotics, aeronautics, etc. The particularities of each domain and the final application define the hypothesis, methods, accuracy and several other parameters to consider. Frequency analysis advantages are different from time integration methods ones; the accuracy in medical applications maybe different than in seismic analysis, etc.

Civil engineering structural dynamics appears mainly in bridges and buildings design phase. Wind, seisms and vehicles are the principal dynamic loads to consider during the design phase (which must comprehend the construction and the service phase). In the last years, dynamic effects are also being considered in real time: control devices are being placed in high buildings to reduce seismic effects [12], which leads to consider real time computations of the dynamics of the structure in order to compute the counter-effects to the seismic actions.

Industrial mechanics dynamic issues are related to undesired vibrations in machines and damage detection [1]. Undesired vibrations waste energy or may be harmful, and also industries spend lots of money reducing the vibration in transport vehicles for their clients comfort. Vibrations can also be induced to find or reveal structural problems [2].

Aeronautic structures are designed by considering high external forces due to the service efforts: great changes in velocities, temperatures and pressures. The analysis of the structure state during the fabrication and service is exhaustive, and among other techniques, vibration based methods are widely used thanks to a low cost application [3][4].

Hybrid laboratories are a new type of laboratories where experiments and simulations are carried out together [5] [6] [7]. Some physical complex process are

difficult to simulate, therefore the experiment must be done in the real physical scenario. At the same time, maybe this complex process affects only a small part of the structure, while the behaviour of the rest of the structure is known and easily represented by simulation. The idea behind hybrid laboratories is to build only the part of the structure that will suffer the complex physical process and simulate the behaviour of the rest of the structure. The benefit of the method is the savings in the construction cost of the simulated part.

Robotics is one of the engineering fields that has more increased the number of publications and investment in the last decade. The appearance of new materials and technologies as sensors or actuators has made possible new applications in numerous fields, as industry or medicine [91][13], requiring new methods to deal with the arising problems and challenges.

The common point of all the previous fields is the importance of the dynamics, it is, the appearance of inertial and damping forces which play an important role in the computation of strains and stresses. In fact, strains and stresses values can be quite different between a static or a dynamic point of view, even when applying forces of the same magnitude when the frequency of the force is near the natural frequency of the structure.

The knowledge of strains and stresses is the main information in structural dynamics. There are always design values that can not be exceeded, and designers must know their relevant information, e.g. maximum possible values. For that end, a comprehensive analysis of the structure is done by carrying out one or more of the methods explained in section 2.1.

A slightly more complicated is the computation of displacements (or strains or stresses) in real time. Real time resolution of structural dynamics problems requires fast and accurate computation, while the speed of computation and the accuracy are usually inversely proportional. Here is where model order reduction techniques appear to reduce the computation cost trying to lose the minimum possible information. The key in model order reduction is the treatment of information: the applied method must be capable of extract the relevant information.

In structural dynamics is widely known the concept of modal analysis 2.1.3, a method to solve the dynamics equation and also used as model order reduction technique. The method uses the projection of the system of n equations into a new basis to transform it in a set of uncoupled equations. The solution of each equation gives a mode of vibration, and the general solution is decomposed as the sum of each equation solution. Each mode of vibration has associated an energy, and

usually the energy of the modes decrease with the frequency, so high frequency modes are related to less energy. Modal analysis is used as a MOR technique when the sum of the general solution is truncated in a mode $N << n$. In many structural cases, a small number of modes can represent sufficiently the dynamics of the structure. This is how modal analysis has been used as a model order reduction technique for decades, by just considering a few modes to compute the whole dynamic solution, it is, extracting the relevant information. Other model order reduction techniques are available, very close in some aspects to modal analysis, e.g. Proper Orthogonal Decomposition or Reduced Basis Method.

Some kinds of problems needs the evaluation of several possibilities or combinations to find the best solution. When the involved parameters exceed a certain number, the classic solutions as Monte-Carlo simulations are unaffordable. The exploration of high dimensional spaces can lead to a number of computations that, with the actual computer power, could take months. Consider a problem with a number of dimensions or variables $\xi = (\xi_1, \ldots, \xi_n)$ and a number m of discretization nodes for each dimension. The number of numerical simulations will be m^n. It is evident that the problem can be unaffordable with just a few considered parameters. In this scenario the *Proper Generalized Decomposition* (PGD) is a suitable method to face the curse of dimensionality. The method splits the high dimensional problem into several low dimension problems, whose computational cost is acceptable [28].

In this Chapter a method to compute the GIR is provided. The GIR allows to monitor the displacements in real time. Classic methods in structural dynamics are revisited in order to enlighten the reasons for what the harmonic analysis has been chosen as the approach for the parametric resolution of the dynamic problem, and some MOR techniques are explained to give a general view of the approach in multi-dimensional problems.

2.1 Structural dynamics

2.1.1 Model problem

Consider a linear viscoelastic body occupying an open bounded domain $\delta\Omega \subset \mathbb{R}^{d\leq 3}$, which will be used as an example for the application of the methods explained in this section. The boundary is divided in two subdomains Neumann Γ_N and Dirichlet Γ_D such that $\delta\Omega = \Gamma_N \cup \Gamma_D$. Assuming small perturbations theory, the computation of the displacements $\mathbf{u}$ in the time interval $t \in I_t$ follows the

equation:

$$\begin{cases} \rho\ddot{\mathbf{u}} - \nabla\boldsymbol{\sigma} = 0 & \text{in} \quad \Omega \times I_t, \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} & \text{on} \quad \Gamma_N \times I_t, \\ \mathbf{u} = 0 & \text{on} \quad \Gamma_D \times I_t, \end{cases}$$

where ρ is the mass density of the material, $\boldsymbol{\sigma}$ is the stress, $\mathbf{n}$ is the outward unit normal to the boundary $\delta\Omega$ and $\mathbf{t}$ is the surface traction. The constitutive low follows the Kelvin-Voigt linear visco-elastic model:

$$\boldsymbol{\sigma} = \mathbf{D} : (\epsilon + \tau\dot{\epsilon}), \tag{2.1}$$

where $\mathbf{D}$ is the fourth-order Hooke elasticity tensor and the parameter τ controls the viscous behaviour. The relation between strains and displacements is given by the Green-Cauchy strain tensor:

$$\epsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla^t\mathbf{u}). \tag{2.2}$$

The discretization of the previous equations and relations by the finite element method [11] leads to the matrix version of the equation of motion:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t), \tag{2.3}$$

where the terms $\mathbf{M}$,$\mathbf{C}$ and $\mathbf{K} \in \mathbb{R}^{N_d \times N_d}$ represent the inertial, damping and elastic forces respectively, and N_d is the number of considered degrees of freedom. $\mathbf{f}(t) \in \mathbb{R}^{N_d \times N_t}$ represents the external applied force, where N_t is the number of considered time steps. The external applied force will be conveniently written in a separated form as:

$$\mathbf{f}(t) = \mathbf{P}_s f(t), \tag{2.4}$$

where $\mathbf{P}_s$ collects the spatial distribution of the force and $f(t)$ modulates the force in time.

Several methods exist in the literature to solve the dynamic problem: the basis of actual techniques can be found in four methods: time integration methods, frequency based methods, modal method and impulse response method. The previous methods are detailed in this section given a light overview of the advantages and disadvantages.

2.1.2 Time integration methods

Time integration methods approximate the unknown function $\mathbf{u}$ in equation (2.3) in order to compute the next time step values. Among several approximations, one of the most used is Newmark β method, which considers an approximation of the unknown function $\mathbf{u}$ by a polynomial of degree *p*. By using a truncation of a Taylor series approximation at a degree $p = 2$ for the derivatives of the unknown function $\mathbf{u}$ [10], the next equations can be written:

$$\mathbf{u}_{n+1} = \dot{\mathbf{u}}_{n+1} + \frac{1}{2}\beta_2 \Delta t^2 \ddot{\mathbf{u}}_{n+1}, \tag{2.5}$$

$$\dot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_{n+1} + \beta_1 \Delta t \ddot{\mathbf{u}}_{n+1}, \tag{2.6}$$

where $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are displacements, velocity and acceleration respectively, and the subscript n indicates the actual time step.

Assuming that the values of $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are known for the previous time steps, $\ddot{\mathbf{u}}_{n+1}$ is obtained from:

$$\ddot{\mathbf{u}}_{n+1} = -\mathbf{A}^{-1}\{\mathbf{f}_{n+1} + \mathbf{C}\bar{\dot{\mathbf{u}}}_{n+1} + \mathbf{K}\bar{\mathbf{u}}_{n+1}\}, \tag{2.7}$$

where

$$\mathbf{A} = \mathbf{M} + \beta_1 \Delta t \mathbf{C} + \frac{1}{2}\beta_2 \Delta t^2 \mathbf{K}, \tag{2.8}$$

and

$$\bar{\mathbf{u}}_{n+1} = \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_n + \frac{1}{2}(1-\beta_2)\Delta t^2 \ddot{\mathbf{u}}_n,$$

$$\bar{\dot{\mathbf{u}}}_{n+1} = \dot{\mathbf{u}}_n + \Delta t \ddot{\mathbf{u}}_n + \frac{1}{2}(1-\beta_1)\Delta t \ddot{\mathbf{u}}_n. \tag{2.9}$$

Once $\ddot{\mathbf{u}}_{n+1}$ is obtained, $\mathbf{u}_{n+1}$ and $\dot{\mathbf{u}}_{n+1}$ can be obtained from (2.5) and (2.6), and this process is repeated for the next time steps.

These equations produce an explicit algorithm that allows to compute displacements, velocity and accelerations for any structure. The main problem of this kind of schemes is the maximum allowed time step to be considered in the calculus. The time step is related to the speed of the elastic wave propagation and the size of the mesh. The time step should not be greater than a critical time step following:

$$\Delta t \leq \frac{2}{\sqrt{3}} \frac{h}{C}, \tag{2.10}$$

where h is the smallest size of the mesh elements and C is the speed of the elastic wave, which is related to the Young's modulus E and the density of the material ρ:

$$C = \sqrt{\frac{E}{\rho}}. \tag{2.11}$$

Fine meshes will require small time steps, and consequently the simulation for a fixed time length will be slower.

There also exist the so called implicit methods, which allows greater values for the time step by paying the price of solving a more complicated system of equations where some iterations must be done each time step.

One of the most advantages of the method is the applicability to nonlinear problems. Structural matrices $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ can be actualized each time step considering the nonlinearities involved in the problem as nonlinear structural behaviour, contact, coupled problems, etc.

2.1.3 Modal method

Modal method is based on the mathematical treatment of the system matrices to reduce the resolution computational cost. By applying the properties of the eigenvectors decomposition the system equations can be uncoupled, it is, the matrix operators become diagonal after projecting in the eigenvectors conformed space [9]. Let us take the undamped version of equation (2.3) and consider a harmonic solution of the free vibration problem (no external force is applied) as

$$\mathbf{u} = \hat{\mathbf{u}}\, sin(\omega t + \theta). \tag{2.12}$$

The second derivative is:

$$\ddot{\mathbf{u}} = -\omega^2 \hat{\mathbf{u}}\, sin(\omega t + \theta) = -\omega^2 \hat{\mathbf{u}}. \tag{2.13}$$

Substituting in equation (2.3), the following equation is obtained:

$$[\mathbf{K} - \omega^2 \mathbf{M}]\hat{\mathbf{u}} = 0, \tag{2.14}$$

which is an eigenvalue problem. By using the orthogonal property of the eigenvectors, equations can be uncoupled pre-multiplying and post-multiplying by the eigenvectors:

$$\mathbf{M}_g = \Phi_n^T \mathbf{M} \Phi_n,$$

$$\mathbf{K}_g = \Phi_n^T \mathbf{K} \Phi_n, \tag{2.15}$$

which are known as normal-coordinate generalized mass and stiffness for mode *n*.

The resulting matrices $\mathbf{M}_g$ and $\mathbf{K}_g$ have a diagonal structure:

$$\mathbf{M}_g = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix}, \tag{2.16}$$

$$\mathbf{K}_g = \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix}. \tag{2.17}$$

The orthogonality properties of the transformation are defined by:

$$\Phi_m \mathbf{M} \Phi_n = 0 \qquad m \neq n, \tag{2.18}$$

$$\Phi_m \mathbf{K} \Phi_n = 0 \qquad m \neq n. \tag{2.19}$$

When the method is applied on forced damped problems, the transformation must be also applied on the damping matrix and the force term. If the damping matrix $\mathbf{C}$ is proportional to $\mathbf{M}$ and $\mathbf{K}$ it will be also diagonal under the transformation; if it is not proportional, a quadratic problem must be solved [14]. For the sake of simplicity, consider that $\mathbf{C}$ is proportional. Applying the transformation on the damping matrix and the force term:

$$\mathbf{C}_g = \Phi_n^T \mathbf{C} \Phi_n,$$

$$\mathbf{f}_g = \Phi_n^T \mathbf{f}. \tag{2.20}$$

The uncoupled system is written now as:

$$\mathbf{M}_g \ddot{\mathbf{u}}_g(t) + \mathbf{C}_g \dot{\mathbf{u}}_g(t) + \mathbf{K}_g \mathbf{u}_g(t) = \mathbf{g}_g(t), \tag{2.21}$$

where $\mathbf{u}_g$ are the normal-coordinate displacements. The uncoupled system can now be solved by applying any time step method, for example the Newmark method

explained in (2.1.2). Notice that as the matrix operators are now diagonal, the resolution is faster than in the original system (the inversion of the operator $\mathbf{A}$ in (2.7) is faster because of the diagonal structure of the matrix). On the other hand, linear behaviour must be assumed in order to keep coherence with the projection in the eigenvectors space.

Finally, displacements can be transformed back to their original coordinates by undoing the transformation:

$$\mathbf{u}(t) = \Phi_n^T \mathbf{u}_g(t). \tag{2.22}$$

2.1.4 Harmonic Analysis

Harmonic analysis or frequency analysis is one of the most used methods in structural dynamics. The knowledge about the spectral response of the structure is a key point in structural design, and it is built from the harmonic analysis. Based on Fourier's Theory, time domain equations can be transformed into frequency domain equations and vice versa. The relation between the two domains follows:

$$\hat{y}(\omega) = \int_{-\infty}^{\infty} y(t)\text{exp}(-\text{i}\omega t)\text{d}t,$$

$$y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{y}(\omega)\text{exp}(-\text{i}\omega t)\text{d}\omega. \tag{2.23}$$

Notice that, in this work, the imaginary unit is written by using non italic type i, and should be distinguished from the italic type i used as an index.

This relation permits a fast and easy transformation of the equations taking profit of the benefits in each domain. The computation of the frequency spectrum reveals information that can be hidden in time domain analysis, as the appearance of natural frequencies or nonlinear behaviours [15]. In most physical systems, as electronics, acoustics or structural dynamics, the concept of transfer function is widely used. The transfer function is the response of the system to a unit input. In frequency analysis, it gives the amplitude as a function of the frequency of the input. The version of the transfer function in the time domain is the impulse response. The impulse response is, then, the response of the system for a unit impulse. While transfer functions are very used in the phase design, and in some real time applications, impulse response framework is better adapted to real time computations when displacements are required in time domain.

Transfer function

The transfer function can be computed from the frequency analysis of the dynamic problem. Consider a solution of the forced damped dynamics equation (2.3) as:

$$\mathbf{u} = \mathbf{U}e^{-\mathrm{i}\omega t}, \tag{2.24}$$

where $\mathbf{U}$ collects the nodal coefficients and $e^{-\mathrm{i}\omega t}$ modulates the function over the time (this expression can be related to the sinus and cosine functions by the Euler identity $e^{-\mathrm{i}\omega t} = cos(\omega t) + i\, sin(\omega t)$. Consider also that the data is obtained at a frequency $f_s = 1$ Hz, in a time interval $I_t = [0, t_f]$. The corresponding sampling in the frequency domain is related to the sample in the time domain by the relations:

$$f_{max} = \frac{1}{\Delta t} f_s = \frac{2N_w - 1}{t_f}, \tag{2.25}$$

from which the number of harmonics N_ω can be obtained. The force term can be written now as a linear combination of N_ω harmonics as:

$$\mathbf{f}(t) = \mathbf{P}_s f(t), \tag{2.26}$$

where the vector $\mathbf{P}_s \in \mathcal{C}^{N \times N_\omega}$ recovers the nodal coordinates and the Fourier coefficients and the function $f(t) = exp^{i\omega t}$ modulates in time the amplitude of the force.

Differentiating twice the equation (2.24), is obtained:

$$\dot{\mathbf{u}}(t) = \mathrm{i}\omega \mathbf{U} e^{-\mathrm{i}\omega t},$$

$$\ddot{\mathbf{u}}(t) = -\omega^2 \mathbf{U} e^{-\mathrm{i}\omega t}. \tag{2.27}$$

Considering then, that the force can be expressed as a sum of sinusoidal forces, and introducing (2.1.4), (2.1.4) and (2.27) in (2.3), the dynamics equation in frequency domain is obtained:

$$\left(-\omega^2 \mathbf{M} + \mathrm{i}\omega \mathbf{C} + \mathbf{K}\right) \mathbf{h}(\omega) = \mathbf{P}_s f(\omega). \tag{2.28}$$

Applying a unit force on the structure, a $\delta(\omega)$ in frequency domain, the transfer function $\hat{\mathbf{h}}$ is obtained.

$$\hat{\mathbf{h}}(\omega) = \left(-\omega^2 \mathbf{M} + \mathrm{i}\omega \mathbf{C} + \mathbf{K}\right)^{-1} \mathbf{P}_s. \tag{2.29}$$

Transfer function allows to compute the displacements operating in the Fourier's space when an external force $f(t)$ is applied on the structure. By using the Fourier's Theory relations, the forces $f(t)$ can be transformed into the frequency domain. Once the force is transformed to the frequency domain, displacements spectrum can be computed by doing:

$$\hat{\mathbf{u}}(\omega) = \hat{\mathbf{h}}(\omega)\hat{f}(\omega), \tag{2.30}$$

where the structure of $\hat{\mathbf{u}}$ is:

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{u}_1(\omega) = \hat{f}(\omega)\hat{h}_1(\omega) \\ \vdots \\ \hat{u}_{Nd}(\omega) = \hat{f}(\omega)\hat{h}_{Nd}(\omega) \end{bmatrix}. \tag{2.31}$$

An important issue among the transfer function characteristics is the possibility of compute the displacements just in some nodes of interest, avoiding the computation in the whole spatial domain. In fact, the computation of the displacements in a node $j \in [1, \ldots, N_d]$ only implies the multiplication of the corresponding row in $\hat{\mathbf{u}}$. Another important characteristic of the frequency analysis come out from (2.29): the equations are uncoupled in frequency. It avoids the computation of the solution for all ω in $I_\omega = [1, N_\omega]$, but just in those frequencies contained in the force. This result is absolutely in concordance with the resolution of the PDE of the dynamics equation, which states that the forced solution of the equation vibrates at the frequency of the force. It is important to remark that the solution is only concerned with the stationary part of the solution.

If displacements are needed in time domain, Fourier equations (2.23) can be applied on $\hat{\mathbf{u}}(\omega)$ to directly obtain $\mathbf{u}(t)$. Notice that for real time applications, two Fourier transformations (direct and inverse) and the resolution of the equations are needed. This operation can be summarized in the following equation, where the symbol $\mathcal{F}$ denotes the Fourier Transform and the symbol $\mathcal{F}^{-1}$ denotes the Inverse Fourier Transform:

$$u(t) = \mathcal{F}^{-1}\big(\mathcal{F}(h) \cdot \mathcal{F}(f)\big). \tag{2.32}$$

Impulse response

There is also the possibility to operate directly in the time domain if one wants to avoid the transformation and anti-transformation operations. By applying Fourier's

Theory, the transfer function $\hat{\mathbf{h}}(\omega)$ is transformed into the impulse response $\mathbf{h}(t)$, and the computation of the displacements is done by applying the integral of convolution:

$$\mathbf{u}(t) = \int_0^t f(t-\tau)\mathbf{h}(\tau)\mathrm{d}\tau, \tag{2.33}$$

where $\mathbf{u}(t)$ is:

$$\mathbf{u} = \begin{bmatrix} u_1(t) = \int_0^t f(t-\tau)h_1(\tau)\mathrm{d}\tau \\ \vdots \\ u_{N_d}(t) = \int_0^t f(t-\tau)h_{N_d}(\tau)\mathrm{d}\tau \end{bmatrix}. \tag{2.34}$$

Some advantages of the use of the impulse response theory are the following ones:

- For real time applications, this is a useful application because the convolution operation can be done with a minimum cost, and the computation of displacements in the whole domain is avoided.
- If the data is obtained in time domain, as it usual in most laboratories, Fourier's transformation and anti-transformation computational cost is avoided.
- The time step to adopt in the impulse response is not limited by the used mesh size in the finite element discretization, as in time integration schemes. The election of the time step is usually related to the physic characteristics of the problem, as natural frequencies or the force frequency range, which are usually quite less limiting than for time integration schemes.

2.1.5 Example

A simple example of application for the aforementioned structural dynamic methods is shown here: displacements on the edge of the bar are computing applying the harmonic method, the impulse response theory, modal superposition and Newmark method.

Figure 2.1: 1D Free-fixed Bar

Young's modulus	[10 100] Pa
Density	1 N/m^3
Damping ratio	[10 50] %

Table 2.1: 1D Free-fixed Bar

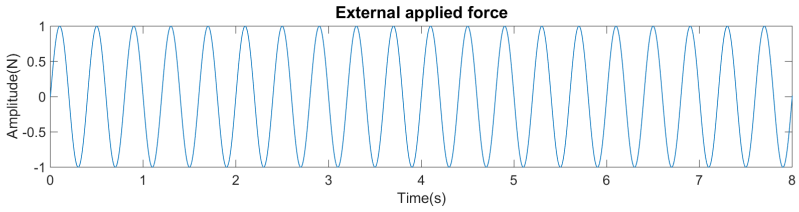

Figure 2.2: External applied force

Consider the free-fixed one dimensional bar shown in the figure 2.1, which is discretized in 11 elements then containing 12 degrees of freedom. The bar is fixed on the left side. The mass density has been considered as $1N/m^3$; two different values of the Young's modulus have been considered being 1 and 10 Pa. The Kelvin-Voigt time constant μ is set in order to obtain a damping factor ξ of 10% and 50% following the relations:

$$\xi = \frac{1}{2}\mu\omega_0 \quad \text{with} \quad \omega_0 = \sqrt{\sigma_0}, \tag{2.35}$$

where ω_0 is the smallest eigenvalue coming from the resolution of the eigenvalue problem defined in (2.14). The physical properties are summarized in the table 2.1. Both two Young's modulus and damping factors are considered in order to compare the results with the parametric method proposed in section 2.5.

An external force $f = sin(\omega_{ext}t)$ with $\omega_{ext} = 2.5$ Hz is applied on the free edge of the bar, where the frequency value has been selected far from the natural frequencies of the bar. Those are 0.31 Hz for the value of 10 Pa and 1 Hz for the value of 100 Pa. The force is shown in figure 2.2.

For the computation of the displacements by the harmonic method, a frequency range of $[0, 25]$Hz with a $\Delta\omega = 1$mHz has been considered. The corresponding time vector has a range of $[0, 1000]$s with a time step $\Delta t = 20$ms. The same discretization has been used to compute the impulse response. The transfer function and the impulse response corresponding to the computation of the impulse

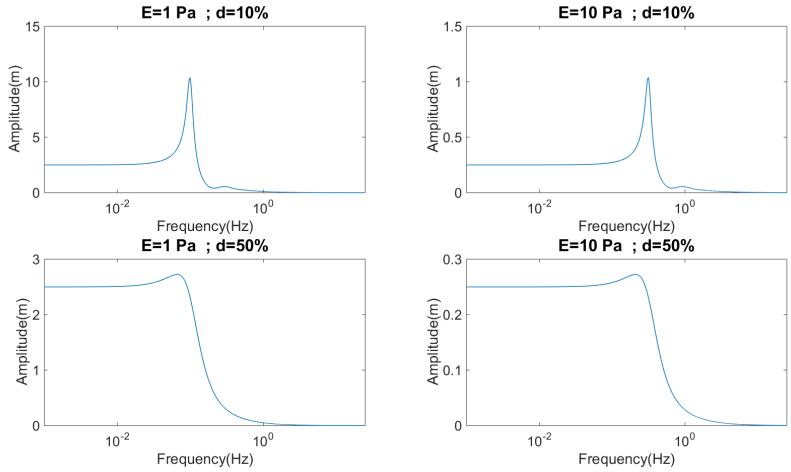

Figure 2.3: Transfer function at the edge of the bar. Logarithmic scale

response method are shown in figures 2.3 and 2.4 respectively.

For the application of the Newmark method, a time step $\Delta t = 1 \cdot 1e - 4$ has been used, and the parameter values were $\beta_1 = 0.25$ and $\beta_2 = 0.5$. Also the Newmark method has been used to compute the displacements in the uncoupled system coming from the modal superposition method. The parameters values were kept at $\beta_1 = 0.25$ and $\beta_2 = 0.5$ and the time step was increased to $\Delta t =$.

Finally, in figure 2.5 a comparison of the obtained displacements for the four different methods is shown (see figure 2.6). The transient phase of the displacements is well captured by modal superposition, impulse response and Newmark methods. On the other hand, the solution computed in the frequency domain only concerns the steady state solution as expected. Figure 2.7.

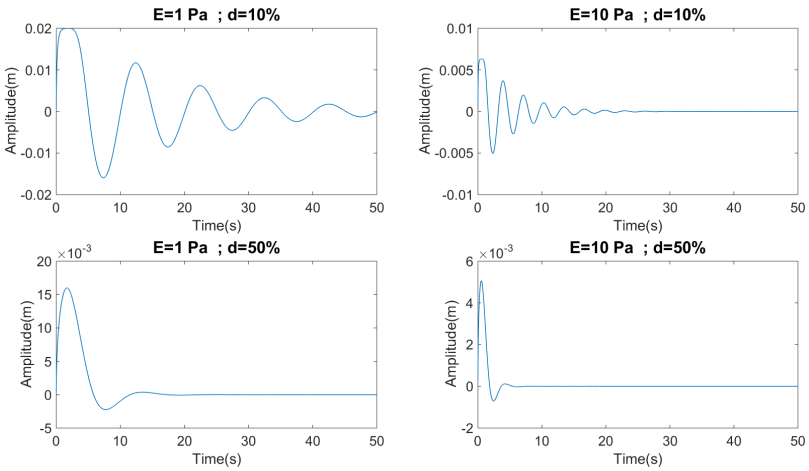

Figure 2.4: Impulse response at the edge of the bar

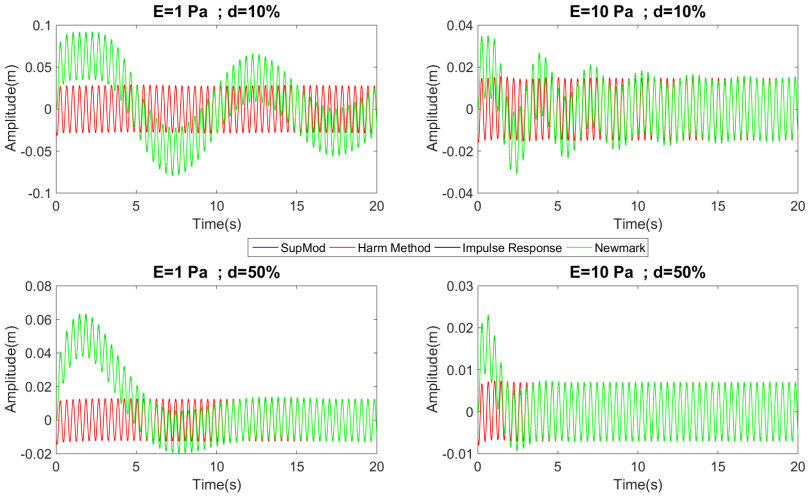

Figure 2.5: Displacements at the edge of the bar

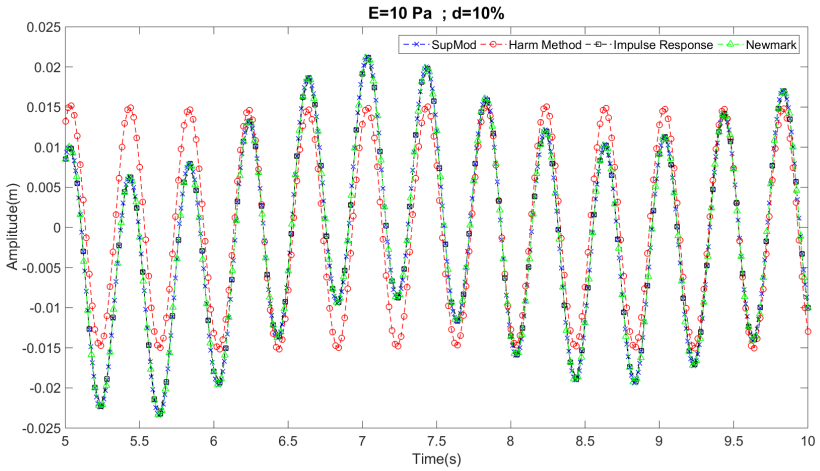

Figure 2.6: Detail of the displacements at the edge of the bar

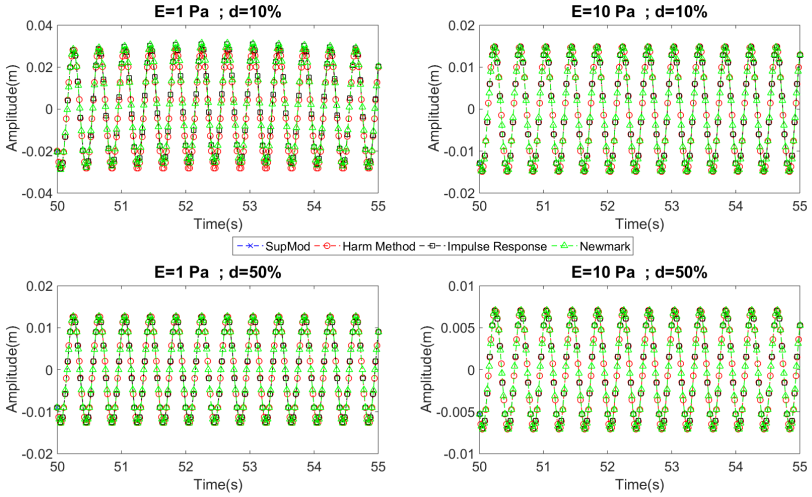

Figure 2.7: Stationary displacements at the edge of the bar

2.2 Model order reduction techniques

The computational cost of the dynamics equation solution is directly related to the number of degrees of freedom. For certain applications it can reach millions of degrees of freedom, what makes unaffordable any classic approach for the problem. In those cases there is the necessity of the reduction of the computational complexity of the model. A light difference can be made among MOR techniques by looking at the goal of the methods: while some methods are interested in reducing the computational cost of the resolution of one instance, other methods are interested in reducing the computational cost of the resolution for every instance. This difference makes that the first case looks for a reduced basis in a smaller dimension, while the second case looks for an approximation of the solution in the whole parametric domain.

Some of the most used methods in model order reduction are the *POD (Proper Orthogonal Decomposition)* and the *RBM (Reduced Basis Method)*. *POD* and *RBM* algorithms start generating a certain number of snapshots to explore the space of the solution in order to find a reduced basis of a smaller space. These methods are classified as *a posteriori* methods, as they need to explore the solution space in order to find a reduced one. In opposite to that, *PGD* is classified as an *a priori* method, where the reduced basis is built without sampling the original solution's space. The strategy followed by the *PGD* algorithm is to apply an iterative process to build the basis of a reduced space, enriching the solution until a prescribed error is achieved. Thanks to the suitable procedure, it is specially suited for multi-parameter problems, as it avoids the exploration of the multi-parameter nature of the space where the solution lives.

2.2.1 Proper orthogonal decomposition

The *POD* is one of the most used model order reduction techniques, widely used in different domains [85][84]. The strategy of the *POD* consists in the sampling of the full model, collecting the so called *snapshots*. From them, a reduced basis is extracted in order to build a subspace where the full model can be projected, expecting that:

$$S_{red} \ll S_{full} \tag{2.36}$$

where S_{red} is the size of the reduced space and S_{full} is the size of the full model. The size of the reduced space can be chosen by truncation, and the error of the

approximation will depend on this truncation.

POD method searches a solution in the form:

$$u(x,t) = \sum_{n=1}^{N} a_n(t)\Phi_n(x), \tag{2.37}$$

where a_n are time dependent coefficients and Φ are basis elements.

To build the POD functions, first data is collected in matrix U with size $N_n \times N_t$, where N_n is the space discretization size and N_t is the number of time steps:

$$U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & ... & u_t \\ | & | & & | \end{bmatrix}. \tag{2.38}$$

A correlation in the data is searched in order to find relations which allow to express the problem in a reduced space. The following eigenvalues problem must be solved:

$$\mathbf{CA} = \lambda \mathbf{A}, \tag{2.39}$$

where $\mathbf{C}$ is the correlation matrix and $\mathbf{A}$ collects the coefficients. The basis can be computed as follows:

$$\Phi_n(x) = \frac{1}{N_t \lambda_n} \sum_{k=1}^{N_t} a_n^k u^k(x). \tag{2.40}$$

Notice that the previous equation produces normalized functions. The *energy* corresponding to each function Φ_n is given by λ_n, so usually a truncation is done by regarding the sorted eigenvalues.

2.2.2 Reduced Basis Method

The *RBM* is a similar technique to the *POD*, but in an enrichment process idea [19]. The elements of the reduced space are obtained in an iterative algorithm. Each element is searched in order to maximize the best-approximation error of the reduced space. The reduced basis is computed in an *offline* phase. In the *online* phase, the extension of the basis is computed in the reduced space, where is expected to be computationally inexpensive.

A fundamental hypothesis is made by assuming that an approximation of the solution for the parameter *p*:

$$u_p \approx \sum_{i=1}^{n} \alpha_i \mathbf{e}^i \qquad \forall p \in (p^-, p^+) \tag{2.41}$$

is possible in the reduced space of size *m*:

$$u_p \approx \sum_{i=1}^{m} \alpha_i(p) \mathbf{e}^i, \tag{2.42}$$

where $m \ll n$.

To build the basis $B = (e^1, ..., e^m)$, first a learning phase is made. It is, several snapshots or resolutions of the problem for different values of the parameter *p* are computed. Then, by truncation of the eigenvectors and Galerkin projection a reduced basis is built. Based on the error of the approximation, the basis can be expanded. The computation of a new term is done for a concrete parameter value. This value is found where the error indicator is maximized.

Finally, in the *online* phase, a reduced system of size *m* must be solved.

Both *POD* and *RBM*, even if they are able to reduce the size of the problem, they cannot built an approximation of the solution in the parametric space, but the basis that represents the solution in the parametric domain.

2.2.3 Proper Generalized Decomposition

The basis of the *PGD* algorithm is the construction of the solution as a separated representation. Separated representations are quite common in many problems. One of the possible solutions for some ordinary or partial differential equations of second order as heat or wave equations is to follow the method of separation of variables, where the solution is sought as a product of time functions and space functions as $u(x,t) = X(x)T(t)$. Also POD gives a solution in a separated representation as shown in 2.2.1. The separated representation gives a simple form of solution, where particular solutions are just a parametrization of a general solution. The computation of this evaluation of the general solution is inexpensive, and presents an interesting tool for certain applications as optimization or real time computations.

PGD is an *a priori* method, it is, the approximation of the solution is built without exploring the whole parametric domain. By an enrichment process, the

approximation of the solution is built in a separated form, adding terms to the solution until a required error is reached. The terms of the approximation are computed by solving a *PDE*, but in a wise process that transforms the multi-domain problem into different 1D problems. The price to pay is the introduction of a nonlinearity into the equations, which is solved by an alternating direction scheme. The PGD solution takes the form:

$$u(p_1, \ldots, p_q) = \sum_{i=1}^{n} \Phi_1(p_1) \ldots \Phi_q(p_q), \tag{2.43}$$

where q parameters are considered and the functions Φ_q living on spaces $\mathcal{V}_q$ depend each one on a parameter p_q. The tensor product defines a multidimensional space built from: $\mathcal{V}_1 \otimes, \ldots, \otimes \mathcal{V}_q$, where the symbol $\otimes$ denotes the tensor product.

The algorithm of the PGD is presented in section 2.3 by means of an example in a structural dynamics problem, but the method has been applied in several other physical fields. The interested reader is referenced to [25] [30] [29] for more detailed information.

2.3 Generalized transfer function

The goal of this section is to compute the GIR which allows to compute in real time the displacements caused by an unknown external force, for any value in the considered range for some preselected parameters. In section 2.1.4, the computation of the transfer function and the impulse response is shown. By applying the PGD method, a parametric form of the transfer function and the impulse response can be computed.

Generalized solutions are suitable to be computed by the PGD method when more than a few parameters are considered in the problem. The best framework for the application of the PGD method in structural dynamics is the harmonic analysis, as it is shown in 2.1.4 the solution is already separated in frequency. Hence, by applying the PGD method to obtain a parametric solution of the harmonic equation the GTF can be obtained. Finally, to obtain the GIR, it rests to apply the inverse Fourier Transform.

As an illustrative example, the dynamic problem 2.1.5 is solved by computing the transfer function applying the PGD method considering three parameters: space, frequency and Young's modulus. This parametric function is what is called GTF. Frequency parameter, as show in 2.1.4 can be transformed to time domain by

using Fourier's Theory. Applying a unit force for each considered frequency in the range $[\omega_{min}, \ldots, \omega_{max}]$, the transfer function is computed in separated form, and, by applying (2.23), the impulse response is also obtained in separated form.

The solution in the PGD framework is sought assuming that an approximation can be computed in separated form as:

$$\mathbf{h}(\omega, E) = \sum_{i=1}^{n} \mathbf{X}_i \, W_i(\omega) \, \mathcal{E}_i(E), \tag{2.44}$$

where $\mathbf{X}$ is a vector that collects the nodal generalized space functions, W is a function which depends on the frequency $\omega \in I_\omega$ and $\mathcal{E}$ is a function which depends on the Young's modulus $E \in I_E$.

Recover (2.28) and remember that $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ come from space finite element discretization; now the space problem can be extended to the other two considered dimensions, frequency and Young's modulus. The frequency parameter appears explicitly in the formulation, and the dependence on the Young's parameter will be written as $\mathbf{K}(E)$. The weighted residual form of the equation with a Galerkin projection can be written as:

$$\int_{I_\omega} \int_{I_E} \mathbf{q}^H \mathbf{A}\mathbf{h} - \mathbf{q}^H \mathbf{f} \mathrm{d}\omega \mathrm{d}E = 0, \tag{2.45}$$

where $\mathbf{A}$ is:

$$\mathbf{A} = -\omega^2 \mathbf{M} + \mathrm{i}\omega \mathbf{C} + \mathbf{K}(E). \tag{2.46}$$

By integrating over the parametric domains one arrives to:

$$\mathbf{B}\mathbf{h} = \mathbf{f}, \tag{2.47}$$

where term $\mathbf{B}$ is now a tensor operator that reads:

$$\mathbf{B} = \mathbf{M}_s \otimes \mathbf{M}_\omega \otimes \mathbf{M}_E + \mathbf{C}_s \otimes \mathbf{C}_\omega \otimes \mathbf{C}_E + \mathbf{K}_s \otimes \mathbf{K}_\omega \otimes \mathbf{K}_E, \tag{2.48}$$

where the subscripts indicate the parametric dependence, and the force term is written as:

$$\mathbf{f} = \mathbf{P}_\mathbf{s} \otimes f_\omega \otimes f_E. \tag{2.49}$$

The PGD algorithm is applied to look for a solution $\hat{\mathbf{h}}$ in separated form where the first $n-1$ terms are known and the terms R, S y T belong to the iteration p and are unknown:

$$\hat{\mathbf{h}}(\omega,k)^{n,p}=\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k)+\mathbf{R}ST. \tag{2.50}$$

The test function is written as:

$$\mathbf{h}^*=\mathbf{R}^*ST+\mathbf{R}S^*T+\mathbf{R}ST^*. \tag{2.51}$$

Assuming known S and T, and substituting (2.51) and (2.50) in (2.45), the algorithm computes the function $\mathbf{R}$:

$$\int_{W\times E}(\mathbf{R}^*ST)(-\omega^2\mathbf{M}+\mathrm{i}\omega\mathbf{C}+\mathbf{K})(\mathbf{R}ST)=$$

$$-\int_{W\times E}(\mathbf{R}^*ST)(-\omega^2\mathbf{M}+\mathrm{i}\omega\mathbf{C}+\mathbf{K})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))+$$

$$\int_{W\times E}(\mathbf{R}^*ST)\mathbf{f}. \tag{2.52}$$

If the terms of the equation are expanded, the following equations can be written:

$$\int_{W\times E}(\mathbf{R}^*ST)(-\omega^2\mathbf{M})(\mathbf{R}ST)+\int_{W\times E}(\mathbf{R}^*ST)(\mathrm{i}\omega\mathbf{C})(\mathbf{R}ST)+$$

$$\int_{W\times E}(\mathbf{R}^*ST)(\mathbf{K})(\mathbf{R}ST)=$$

$$-\int_{W\times E}(\mathbf{R}^*ST)(-\omega^2\mathbf{M})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))$$

$$-\int_{W\times E}(\mathbf{R}^*ST)(\mathrm{i}\omega\mathbf{C})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))$$

$$-\int_{W\times E}(\mathbf{R}^*ST)(\mathbf{K})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))+\int_{W\times E}(\mathbf{R}^*ST)\mathbf{f}, \tag{2.53}$$

where the integrals over ω and k are known and can be computed (see Appendix 1), obtaining:

$$-\mathbf{R}^*\mathbf{M}\mathbf{R}l_{1\omega}l_{1k} + \mathbf{R}^*\mathrm{i}\mathbf{C}\mathbf{R}l_{2\omega}l_{1k} + \mathbf{R}^*\mathbf{K}\mathbf{R}l_{3\omega}l_{3k} =$$

$$+\mathbf{R}^*\mathbf{M}\sum_{i=1}^{n-1}\mathbf{X}_i r_{1i\omega}r_{1ik} - \mathbf{R}^*\mathrm{i}\mathbf{C}\sum_{i=1}^{n-1}\mathbf{X}_i r_{2i\omega}r_{1ik}$$

$$-\mathbf{R}^*\mathbf{K}\sum_{i=1}^{n-1}\mathbf{X}_i r_{3i\omega}r_{3ik} + \mathbf{R}^*F_x\eta_\omega\eta_k. \tag{2.54}$$

From (2.54), $\mathbf{R}$ can be obtained.

For the computation of S, the already computed $\mathbf{R}$ and keeping T as known are used:

$$\int_{W\times E}(\mathbf{R}S^*T)(-\omega^2\mathbf{M})(\mathbf{R}ST)$$

$$+\int_{W\times E}(\mathbf{R}S^*T)(\mathrm{i}\omega\mathbf{C})(\mathbf{R}ST)+$$

$$\int_{W\times E}(\mathbf{R}S^*T)(\mathbf{K})(\mathbf{R}ST) =$$

$$-\int_{W\times E}(\mathbf{R}S^*T)(-\omega^2\mathbf{M})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))$$

$$-\int_{W\times E}(\mathbf{R}S^*T)(\mathrm{i}\omega\mathbf{C})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))$$

$$-\int_{W\times E}(\mathbf{R}S^*T)(\mathbf{K})(\sum_{i=1}^{n-1}\mathbf{X}_i W_i(\omega)E_i(k))+$$

$$\int_{W\times E}(\mathbf{R}S^*T)\mathbf{f}, \tag{2.55}$$

where now the integrals over x and k can be computed (see Appendix 1):

$$\int_W S^*(-\omega^2)Sl_{1x}l_{1k} + \int_W S^*\,\mathrm{i}\omega\,Sl_{2x}l_{1k} + \int_W S^*Sl_{3x}l_{3k} =$$

$$\int_W S^*\omega^2\sum_{i=1}^{n-1}W_i(\omega)r_{1ix}r_{1ik}$$

$$-\int_W S^*\,\mathrm{i}\omega\sum_{i=1}^{n-1}W_i(\omega)r_{2ix}r_{1ik}$$

$$-\int_W S^* \sum_{i=1}^{n-1} W_i(\omega) r_{3ix} r_{3ik} + \int_W S^* F_\omega \eta_x \eta_k, \tag{2.56}$$

from which S can be computed.

For the computation of *T*, the already computed $\mathbf{R}$ and S are used:

$$\begin{aligned}
&\int_{W\times E} (\mathbf{R}ST^*)(-\omega^2\mathbf{M})(\mathbf{R}ST) + \\
&\int_{W\times E} (\mathbf{R}ST^*)(\mathrm{i}\omega\mathbf{C})(\mathbf{R}ST) + \\
&\int_{W\times E} (\mathbf{R}ST^*)(\mathbf{K})(\mathbf{R}ST) = \\
&-\int_{W\times E} (\mathbf{R}ST^*)(-\omega^2\mathbf{M})(\sum_{i=1}^{n-1} \mathbf{X}_i W_i(\omega) E_i(k)) \\
&-\int_{W\times E} (\mathbf{R}ST^*)(\mathrm{i}\omega\mathbf{C})(\sum_{i=1}^{n-1} \mathbf{X}_i W_i(\omega) E_i(k)) \\
&-\int_{W\times E} (\mathbf{R}ST^*)(\mathbf{K})(\sum_{i=1}^{n-1} \mathbf{X}_i W_i(\omega) E_i(k)) + \\
&\int_{W\times E} (\mathbf{R}ST^*)\mathbf{f}.
\end{aligned} \tag{2.57}$$

Computing the integrals over x and ω one obtains:

$$\begin{aligned}
&-\int_E T^*Tl_{1x}l_{1\omega} + \int_E T^*Tl_{2x}l_{2\omega} + \int_E T^*Tl_{3x}l_{3\omega} = \\
&\int_E T^* \sum_{i=1}^{n-1} E_i(k) r_{1ix} r_{1i\omega} \\
&-\int_E T^* \sum_{i=1}^{n-1} E_i(k) r_{2ix} r_{2i\omega} \\
&-\int_E T^* k \sum_{i=1}^{n-1} E_i(k) r_{3ix} r_{3i\omega} + \int_E T^* F_k \eta_x \eta_\omega,
\end{aligned} \tag{2.58}$$

where T can be easily calculated.

The algorithm is stopped when a prescribed error ϵ is achieved:

$$\epsilon(n) = \frac{\|\mathbf{X}_n(x) W_n(\omega) E_n(k)\|}{\|\mathbf{X}_1(x) W_1(\omega) E_1(k)\|} \tag{2.59}$$

2.4 Generalized impulse response

The GIR is an impulse response in parametric form. By means of the *PGD* algorithm the GTF has been obtained in the previous section 2.3. Here, the GIR, which is the version of the GTF in time domain, is obtained. The equivalence between the GTF and the GIR is:

$$\mathbf{h}(t,E) = \mathcal{F}^{-1}\left(\sum_{i=1}^{n} \mathbf{X}_i\, W_i(\omega)\, \mathcal{E}_i(E)\right) =$$

$$\sum_{i=1}^{n} \mathbf{X}_i\, \mathcal{F}^{-1}\left(W_i(\omega)\right)\, \mathcal{E}_i(E) = \sum_{i=1}^{n} \mathbf{X}_i\, W_i(t)\, \mathcal{E}_i(E), \tag{2.60}$$

where $\mathcal{F}^{-1}$ represents the inverse Fourier transform. Notice that the Fourier transform is applied only on those functions which depend on the time. The cost of transforming and anti-transforming is then independent of the number of considered parameters.

2.5 Generalized displacements

GTF and GIR can also be used to compute parametric displacements in frequency domain or in time domain. This possibility can be useful in optimization or monitoring problems: the exploration of the multi-dimensional displacements is a simple parametrization of the solution. Once the GTF and the GIR are computed, obtaining the displacements value for the considered parameters is an inexpensive post-processing. In this section it is shown how the parametrization of the general solution is done and some interesting aspects.

2.5.1 Generalized displacements in frequency space

Consider that the GTF is already computed for the parameters frequency and Young's modulus as in (2.44), and the displacements caused by an external force $\hat{f}(\omega)$ are required in a parametric form. The required displacements can be obtained by applying 2.1.4, which in parametric form is equivalent to:

$$\hat{\mathbf{u}}(\omega,E) = \sum_{i=1}^{n} \mathbf{X}_i\, W_i(\omega)\, \mathcal{E}_i(E) \cdot \hat{f}(\omega) = \sum_{i=1}^{n} \mathbf{X}_i\, \left(W_i(\omega) \cdot \hat{f}(\omega)\right)\, \mathcal{E}_i(E)$$

$$= \sum_{i=1}^{n} \mathbf{X}_i \Upsilon(\omega) \mathcal{E}_i(E). \tag{2.61}$$

Notice that the computation of the displacements involve only the multiplication of the force by the functions depending on ω as :$\Upsilon(\omega) = W_i(\omega) \cdot \hat{f}(\omega)$. The computation of the parametric displacements is then independent of the considered number of parameters.

Imagine now that one is interested on the amplitude of the displacements for some value of the Young's modulus parameter and for a certain degree of freedom. The corresponding index for the selected degree of freedom is denoted by j_x , and j_E the corresponding index for the selected value of the Young's modulus parameter, while the ω parameter is set free. Then, the displacements can be obtained from:

$$\hat{\mathbf{u}}^{j_x, j_E} = \sum_{i=1}^{n} \mathbf{X}_i^{j_x} \, \Upsilon_i(\omega) \, \mathcal{E}_i^{j_E}(E), \tag{2.62}$$

where $\hat{\mathbf{u}}$ stores the complex displacements associated to the preselected parameters. For any other parameter combination, the sum in (2.62) must be computed again. Obviously, this operation is inexpensive in comparison with the whole resolution of the dynamic equations.

2.5.2 Generalized displacements in time domain

In a similar way, parametric displacements in time domain can be obtained. Parametric displacements in time domain bring some advantages over the equivalence in time domain. Frequency analysis is only concerned with the stationary response of the structure, thus the temporary response is missing, while the time domain computation of displacements by using the impulse response theory the causality is assured.

Consider the already computed GIR (2.60) from section 2:

$$\mathbf{h}(t, E) = \sum_{i=1}^{n} \mathbf{X}_i \, W_i(t) \, \mathcal{E}_i(E). \tag{2.63}$$

Consider now an external force $f(t)$ applied on the structure. The displacements are computed by applying the convolution operation with the functions that depend on the time:

$$\mathbf{u}(t,E) = \sum_{i=1}^{n} \mathbf{X}_i\, W_i(t)\, \mathcal{E}_i(E) * f(t) = \sum_{i=1}^{n} \mathbf{X}_i\ \left(W_i(\omega) * f(t)\right)\ \mathcal{E}_i(E)$$

$$= \sum_{i=1}^{n} \mathbf{X}_i \Upsilon(t) \mathcal{E}_i(E), \tag{2.64}$$

where the symbol $*$ represents the convolution operation. The convolution operation can be considered computationally inexpensive, and it is shown that the parametrization of the general solution is also a trivial computation. The separated form of the solution for the displacements in time domain is the same as in the frequency domain, then the evaluation of the solution for concrete parameters is done by following the same procedure, and by the same computational cost. These characteristics make the present approach an interesting alternative in fast structural dynamics computation.

2.6 Results

Consider the free-fixed one dimensional bar shown in figure 2.1, used in section 2.1 with the physical properties in the table 2.1. The PGD was applied to obtain a parametric solution following the procedure developed in section 2.3, considering the parameters: frequency, Young's modulus and damping. The frequency space was discretized in a range of [0 25] Hz, with an interval of 1 mHz. The Young's modulus space was discretized in a range of [1 12] Pa with an interval of 1 MPA. Notice that both the two values of the Young's modulus considered in 2.1.5 are in the considered parameter range of the PGD solution. The discretization of the damping has been done considering the Kelvin-Voigt time constant μ in a range of [0.1, 1.6] with a step of 0.1, which covers a damping ratio between 3% and 159%.

In the figure 2.8 the real and imaginary parts of the 4 first frequency modes obtained by the PGD method are shown. The figure 2.9 shows the real and imaginary parts of the 4 first Young's modulus parameter modes. The figure 2.10 shows the real and imaginary parts of the 4 first damping factor parameter modes.

In the figure 2.11, the parametrization for different values of the Young's modulus parameter and the damping ratio is shown. Notice that this parametrization leads to the transfer function of the structure, where the amplitude of the response

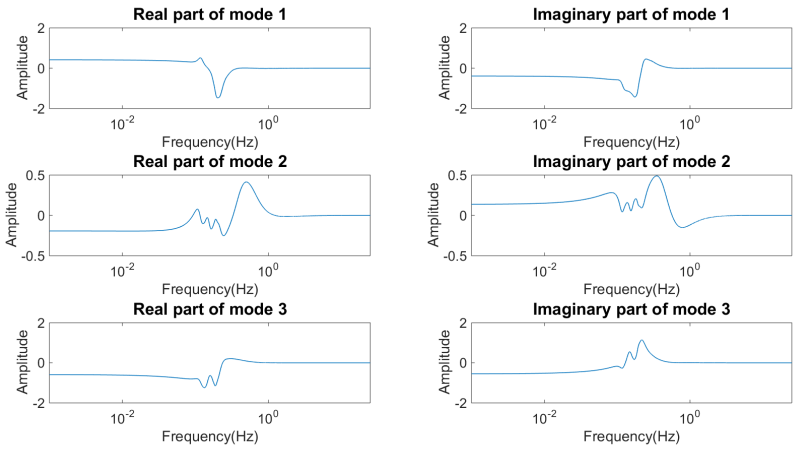

Figure 2.8: Frequency functions

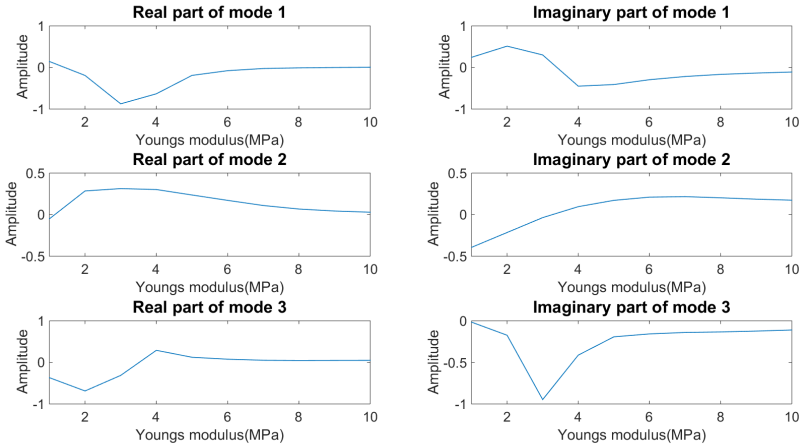

Figure 2.9: Young's modulus functions

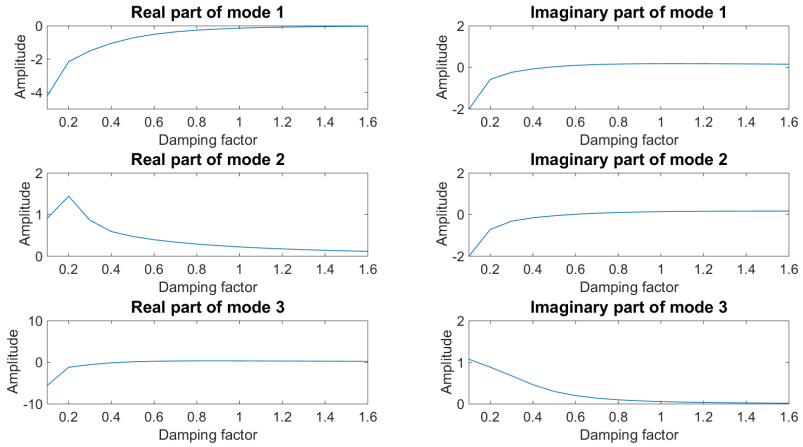

Figure 2.10: Damping functions

is related to the frequency. The results are shown for the node placed in the right edge of the bar. In the figure 2.12 the corresponding impulse responses to the previous transfer functions are shown. Remember that the impulse responses are obtained by applying the Inverse Fourier Transform.

Finally, the impulse responses are used to compute the displacements. The external applied force is the same as in 2.1, it is, $f_e = sin(\omega t)$ were $\omega = 2.5$Hz. Displacements are compared with the solution obtained by applying Newmark method in figure 2.13.

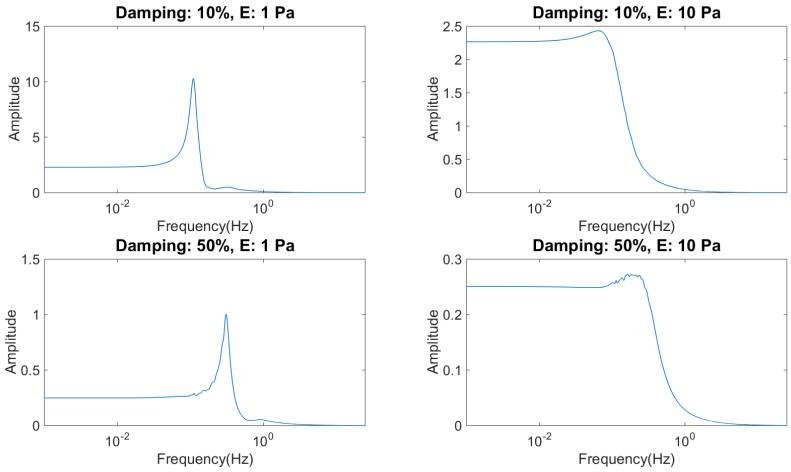

Figure 2.11: Transfer function for different values of Young's modulus and damping ratio

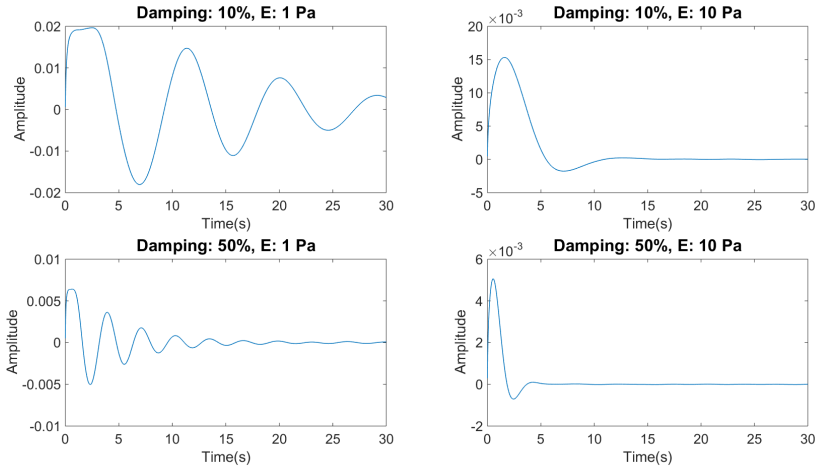

Figure 2.12: Transfer function for different values of Young's modulus and damping ratio

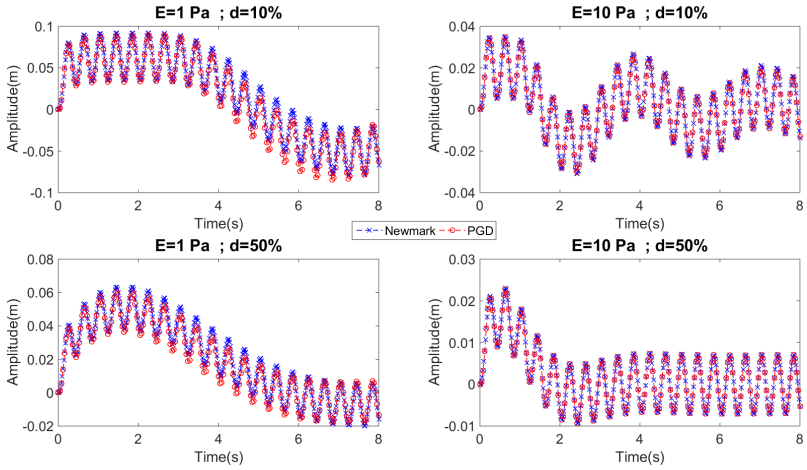

Figure 2.13: Displacements obtained by the PGD method compared with a Newmark method solution

2.7 Fractional damping

Fractional calculus is an ancient and not much studied part of mathematics. By the end of the 17th century, differential calculus was developed and fractional derivatives theory came with it. In comparison with differential calculus, fractional calculus has not been much studied, despite certain famous mathematicians have written about it. In the last decades, some natural processes have been simulated with fractional calculus, and them have been demonstrated more precise than classical differential calculus. In this section, fractional calculus is applied to provide a better approximation tool for the damping term. The application of fractional calculus provides an extra parameter to better represent the dissipation of the forces which are comprehended on the damping term.

Leibnitz and Marquis de l'Hôpital, introducing differential calculus notation, are widely considered as the first who studied fractional calculus, when they thought about a non-integer n^{th} derivative. Other mathematicians, as Euler, Laplace or Abel, developed theories about fractional calculus, but still nowadays there are discrepancies when defining fractional derivatives. In the "Traite du Calcul Differentiel et du Calcul Integral", Lacroix showed the deduction of:

$$\frac{d^{\frac{1}{2}}v}{dv^{\frac{1}{2}}} = \frac{2\sqrt{v}}{\sqrt{\pi}}. \tag{2.65}$$

First application of fractional calculus was provided by Abel when he was

working on the solution of the tautochrone curve. Abel found a solution using integrals and derivatives of non-integer orders. By the other hand, Liouville is considered to be the first who gave a formal definition of fractional derivative concept. Starting for a known formula:

$$\frac{d^m}{dx^m}e^{ax} = a^m e^{ax}, \tag{2.66}$$

he developed his calculus arriving to the *fractional derivative of a power function*:

$$\frac{d^v}{dx^v}f(x) = \sum_{n=0}^{\inf} c_n a^v e^{a_n x}. \tag{2.67}$$

Two of the most used fractional definitions are shown below in Eq. (2.68) and (2.70) . Main difference between them is the consideration of the fractional derivative of a constant (except when it is 0 value). Caputo's considers zero the fractional derivative of a constant, while Grünvald-Letnikov consider that it depends on some global space near the constant.

- Grünvald-Letnikov equation is:

$$ {}_aD_x^\alpha f(x) = \frac{1}{h^\alpha}\sum_{m=0}^{\frac{x-a}{h}} (-1)^m g_m^\alpha f(x-h), \tag{2.68}$$

  evaluated in the interval $x \in [a, b]$. h is the distance between two nodes of the finite difference discretization and the coefficient g_m^α is defined as follows:

$$g_m^\alpha = \frac{\Gamma(\alpha+1)}{m!\Gamma(\alpha+1-m)}, \tag{2.69}$$

  where Γ is the Euler's Gamma function.

- Caputo defined his equation as:

$$ {}_a^C D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)}\int_a^x (x-\tau)^{n-\alpha+1} f^{(n)}(\tau), \tag{2.70}$$

  where $n-1 < \alpha \leq n$ and $f^{(n)}$ is the integer order derivative of order n. Caputo's formula provided a useful tool, as the derivative of a constant is equal to zero in his formula, and it let solve several problems that previous formulae couldn't or had difficulties.

2.7.1 Fractional damping as a parameter

On structural dynamics, fractional calculus can be interesting as it provides a more complete tool than the concept of complex damping. This comes from the fact that complex damping is a particular case of fractional damping with α=1.

Consider the model problem in section 2.1, where a Kelvin-Voigt linear viscoelastic model was assumed. Let us now add the fractional derivative to the viscous behaviour as:

$$\boldsymbol{\sigma} = \mathbf{D} : (\epsilon + \tau \frac{\partial^\alpha \epsilon}{\partial t^\alpha}), \tag{2.71}$$

where instead an integer differentiation of the velocity a fractional differentiation is considered. The discretization of the equation leads to the equation (notice that the same procedure as in 2.1 is followed):

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\frac{\partial^\alpha u}{\partial t^\alpha} + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t). \tag{2.72}$$

If the force can be expressed as a harmonic term, or a sum of harmonic terms, the transformation of the damping term to the space of the frequency leads to a complex term:

$$\mathbf{C}(\mathrm{i}\omega)^\alpha, \tag{2.73}$$

so finally the discrete dynamics equation with fractional damping is written as:

$$\left(-\omega^2\mathbf{M} + (\mathrm{i}\omega)^\alpha\mathbf{C} + \mathbf{K}\right)\hat{\mathbf{h}}(\omega) = \mathbf{P}_s f(\omega), \tag{2.74}$$

where the complex number i can be written as:

$$\mathrm{i}^\alpha = cos\frac{\pi}{2}\alpha + \mathrm{i}\ sin\frac{\pi}{2}\alpha = \phi + \mathrm{i}\mu. \tag{2.75}$$

Despite of the complex definitions of the fractional derivative, the application in the frequency domain leads to a very simple term. To better understand the role played by the fractional parameter, a phasorial representation 2.14 gives us a good tool to illustrate it. Analysing it with phasorial calculus, inertial force (1) and stiffness (2) are in phase with the force (f), and complex damping is at an angle of 90 degrees (3). Fractional damping can give a different phase angle. Varying the modulus and the angle of the damping term can make an adjustment of the damping effect.

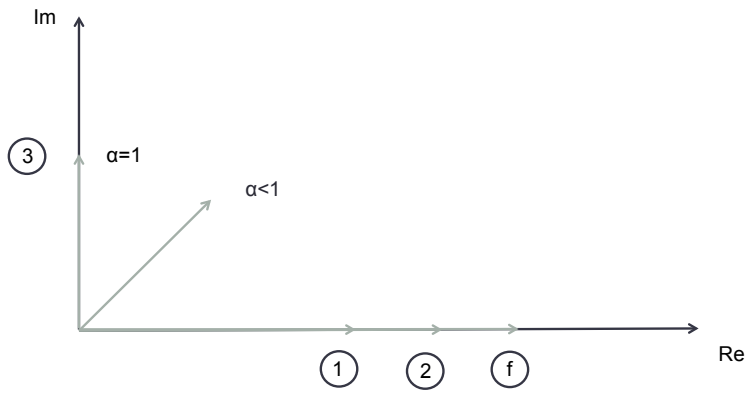

Figure 2.14: Phasorial comparison on dynamics equation

Complex stiffness and Rayleigh damping are commonly used as parameters to consider the damping effects in structures. Complex stiffness is not depending on the velocity nor the frequency, and obviously, could be not enough precise to simulate some dynamic behaviours. Rayleigh damping considers the damping as a linear combination of the stiffness and mass matrices. This consideration involves certain problems as it means that the structure is viscous and linear. Applying fractional calculus, is possible to correct in some way the assumption made on considering the damping as a viscous effect, as it evaluates not the whole frequency but some fractional part of it. Equations can then be solved by following the same procedure detailed in 2, where the added fractional parameter does not imply longer complications.

2.7.2 Results

Variation of alpha and damping coefficient produces a sort of curves that can represent the behaviour of the material more properly than simple complex term. For the evaluation, the same bar as in 2.1.5 has been chosen to evaluate and illustrate the results. The displacements showed in the figures belong to the edge of the bar.

In the graphic 2.15 different values for alpha and the damping coefficient are shown. Alpha values have been taken from 0.5 to 1, and, for each alpha value, the value of the damping coefficient has been varied from 16 to 26.

In the graphic 2.16, comparison is made choosing 6 damping coefficients, and varying alpha values from 0.5 to 1 on each graphic:

In figure 2.17, similar behaviour of combination of alpha and damping coef-

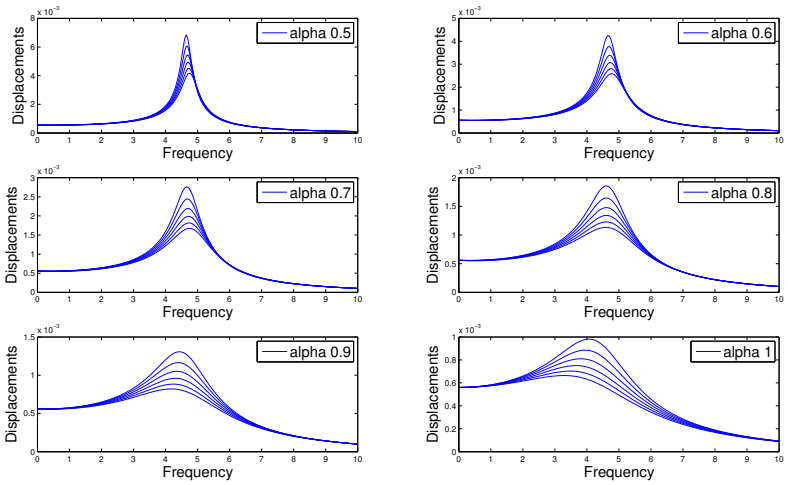

Figure 2.15: Comparison fixing alpha and varying the damping coefficient

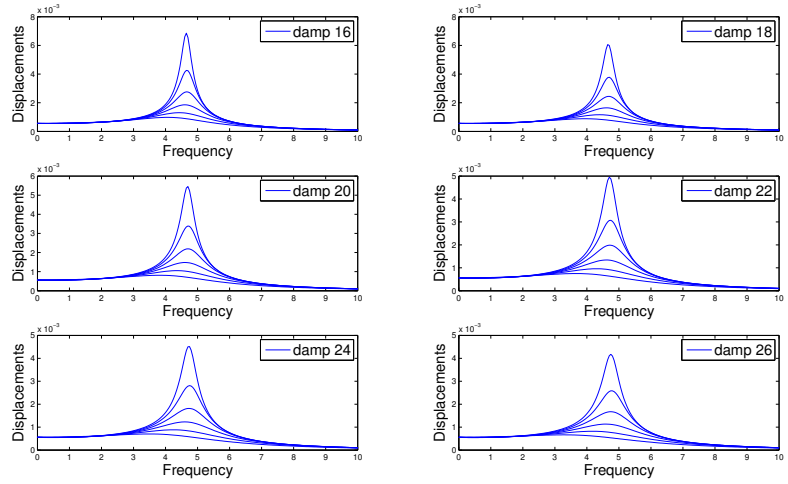

Figure 2.16: Comparison fixing the damping coefficient and varying alpha

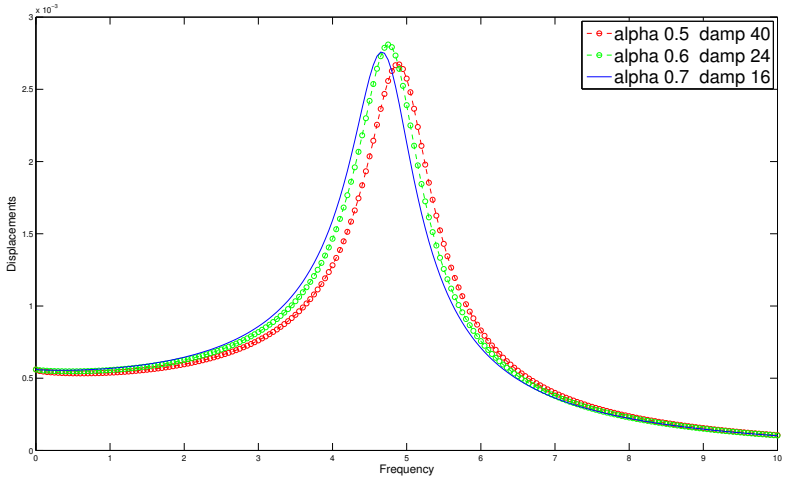

Figure 2.17: Different simulation of damping effect varying alpha and damping coefficient values

ficient term are shown. Next figure tries to present some different proposals to simulate the behaviour of damping effect of a structure. In the case of a more precise simulation is searched, a proper combination of damping and fractional coefficient can be proposed to define the behaviour of the model:

Chapter 3

Monitoring of forces

Monitoring and control of structures is a key point in many industrial applications. The increase of automation process in the industry demands fast and accurate algorithms capable of monitoring the dynamics of the robots or structures and positioning them in a certain configuration.

Soft robots and soft materials operated by robots are some of the issues nowadays in the simulation field. The nonlinearities arising from these problems or the required response ratio are a still a challenge and an object of several publications per year.

The impulse response theorem was used in Chapter 2 to solve the computational time issue. The mathematical operation which arises from the impulse response theory is the convolution. It is easy to implement and it's a fast operation in terms of computational cost, as it only involves sums and multiplications. The objective of this section is to apply the impulse theorem to compute in real time the required forces to achieve a certain configuration in displacements.

In addition, in Chapter 2 the impulse response was computed in parametric form. The parametric form brings enormous possibilities for real time applications, optimization, etc. To keep these advantages in the inverse problem, the GIIR is computed using the GIR and keeping the parametric structure. The formulation of the impulse response theory is used and the PGD, the model order reduction technique explained in section 2.2.3 is applied in order to obtain the GIIR.

The advantage of using the GIR is double: the pre-computed solution allows optimization of the forces, material selection, etc. for different displacements compositions, and allows to compute in real time the required force to control the displacements under desired values.

While the problem of computing the displacements from known forces can be

well established by the discretization of the dynamics equation, the inverse problem has theoretical and practical issues that complicate the resolution. The resolution in the frequency domain can compromise the real time requirements as it demands direct and inverse transformation of the data, and despite of the computational speed of the Fast Fourier Transform in some applications could not be fast enough. The resolution of the problem in the time domain leads to the resolution of the convolution problem, which is usually an ill-posed problem and numerically ill-conditioned.

Inverse problems are widely studied, as they arise in many physical problems: most of the physical processes are modelled by a function which links inputs (heat sources, forces, etc.) with outputs (temperature, displacements, etc.). The obtaining of the input when the output is known in the integral of convolution results an inverse problem called deconvolution, and it usually needs numerical treatment to be solved. There is also the possibility of obtaining the function when both the input and the output are known. This is the main interest in system identification field, which is also classified as an inverse problem.

In this section a method to solve the problem of real time deconvolution is presented. The inverse of the impulse response is computed in an *offline* phase, and the computation of the force is done in the *online* phase by applying the convolution operation.

While the computation of the GIIR is also an inverse problem, the special structure of the PGD allows to avoid some inconveniences of the numerical methods which arise in the resolution of the inverse problem. These methods involve some parameters that are unknown and difficult to find, and they can affect considerably the solution, as the regularization parameter in the Tikhonov method or the relaxation parameter in the Landweber method. Avoiding the use of these parameters can be related to the Truncated Singular Values Decomposition.

3.1 Formulation of the problem

Let us consider the impulse response obtained following 2.1.4, and take from it the function at a single degree of freedom, i.e. the centre of the bar used in section 2.1. Let us also recover the definition of the convolution integral (1.2) to compute the displacements in the centre of the bar when an external force is applied at the edge

of the bar, reading:

$$u_c(t) = \int_0^t f(t-\tau)h_c(\tau)\mathrm{d}\tau, \tag{3.1}$$

where the subscript $(.)_c$ denotes the node placed in the centre of the bar.

Obtaining f from (3.1) is a deconvolution problem which has been exhaustively explained in 1, and where specific numerical methods need to be applied. Obtaining f from (3.1) in real time means the resolution of an inverse problem per time step, which is unaffordable for most of the classic methods.

The approach used in this thesis is to avoid the real time deconvolution by transferring it to an *offline* phase, and kept a convolution operation for the *online* phase. For that, the formulation is slightly rewritten. Instead of computing f in real time, what is computed is the inverse of the impulse response in an *offline* phase, in a way that, once it is computed, the computation of the forces in the *online* phase is done by a convolution operator as:

$$f(t) = \int_0^t u_c(t-\tau)g_c(\tau)\mathrm{d}\tau, \tag{3.2}$$

where g is the inverse impulse response. It is evident to deduce from this equation the relation between h and g: one function is the inverse of the other in the sense of the convolution operation. The inverse property is defined in the convolution theorem by:

$$\delta(t) = \int_0^t h_k(t-\tau)g_k(\tau)\mathrm{d}\tau, \tag{3.3}$$

where $k = 1, \ldots, N_d$, and being N_d the number of degrees of freedom following the notation in 2.1. The position in the integral of the functions g and h is interchangeable by the properties of the integral of convolution.

The reader may imagine the possibility of obtaining the inverse impulse response in the frequency domain rather in time domain. Then, the equation (3.3) where g is defined is transformed to the frequency domain, obtaining:

$$\delta(\omega) = \hat{\mathbf{h}}(\omega)\hat{\mathbf{g}}(\omega). \tag{3.4}$$

Recovering the definition of the transfer function $\hat{\mathbf{h}}$, $\hat{\mathbf{g}}$ is defined then as:

$$\hat{\mathbf{g}}(\omega) = -\omega^2\mathbf{M} + \mathrm{i}\omega\mathbf{C} + \mathbf{K}, \tag{3.5}$$

which is a not bounded function, as $\hat{\mathbf{g}}(\omega) \to \infty$. Therefore, the inverse impulse response defined as $\mathcal{F}^{-1}(\hat{\mathbf{g}})$ does not exist, as Fourier inverse transform is not defined for not bounded functions.

By the other hand, this method allows to compute the force in real time by paying the cost of direct/inverse transformations per time step. Measured displacements in time domain must be transformed to the frequency domain, then the force must be computed and finally it must be transformed back to the time domain. This is possible as longer as the force vector is bounded, which is common in structural dynamics. This operation can be summarized in the following equation:

$$f(t)_p = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(u_p)}{\mathcal{F}(h)}\right), \tag{3.6}$$

where p denotes the current time step.

3.2 Generalized inverse impulse response

In Chapter 2, where the GIR was presented based in the impulse theory, the objective was to compute the displacements in real time, when a external forced is applied in the structure. It was computationally inexpensive by using the convolution operation. In this Chapter, the objective is the opposite one: to compute the force that is causing some known displacements, or, in another point of view, which force must be applied to obtain a certain configuration in displacements. The general idea of the approach is to keep the advantages of the impulse response theory while avoiding the deconvolution in real time. For that, the GIIR is pre-computed, and used in real time to obtain the required force by convolution. This two-steps method is detailed as:

- Compute first the GIIR from the equation:

$$\delta(t) = \int_0^t \mathbf{h}(t-\tau)\mathbf{g}(\tau)\mathrm{d}\tau, \tag{3.7}$$

- Compute then in real time the required force by applying

$$f(t) = \int_0^t \mathbf{u}(t-\tau)\mathbf{g}(\tau)\mathrm{d}\tau. \tag{3.8}$$

To compute the GIIR $\mathbf{g}$ the GIR $\mathbf{h}$ must be obviously known. The approach followed in this work is to compute the GIR $\mathbf{h}$, and then use it to compute the GIIR $\mathbf{g}$.

3.2.1 Dual problem. Flexibility method

In the monitoring problem the GIR was obtained. A physical model and its equations, discretized by the finite element method or the stiffness method, and the harmonic analysis where used to that end. To obtain the GIIR two approaches can be followed: formulate the problem in the dual version, called flexibility method, or compute it from the GIR solving the arising inverse problem.

In the direct problem, one can take profit of the impulse response theory to compute the displacements without cost. In the inverse problem, in this section the same idea is applied: a convolution operation to compute the force in real time. Therefore, if the convolution operation comes from the stiffness method, where displacements are computed from external forces, formulating the dual problem one will expect to obtain a convolution to compute the required forces.

The flexibility method is nearly reduced to academic lessons in static computation of structures. The laboriousness to automatize the formulation has reduced the use and research of this method. However, some works have been published on the development of the method, by extending the formulation to the structural dynamics field [37][39], but to the knowledge of the author, no transfer function or impulse response has been computed yet by applying those methods.

Following a standard formulation [40], the matrix form of the problem can be written as:

$$\mathbf{A}\mathbf{f} = \mathbf{u}, \tag{3.9}$$

where $\mathbf{A}$ is known as the flexibility matrix, with a similar role as the stiffness matrix in the stiffness method.

Consider the bar in section 2.1, which is discretized by using the classic beam theory: two elements are considered, with lumped mass in the edges. The flexibility matrix is then:

$$A = \begin{bmatrix} \frac{L^3}{3EI} & \frac{5L^3}{6EI} \\ \frac{5L^3}{6EI} & \frac{8L^3}{3EI} \end{bmatrix}. \tag{3.10}$$

If dynamic forces are considered, the force term can be written as:

$$\mathbf{f} = \begin{bmatrix} -m\ddot{u}_2 \\ -m\ddot{u}_3 + F^* \end{bmatrix}. \tag{3.11}$$

The previous arrangement considers that the acting forces on the nodes are the inertial ones and the external applied force in the edge. Rearranging the equations

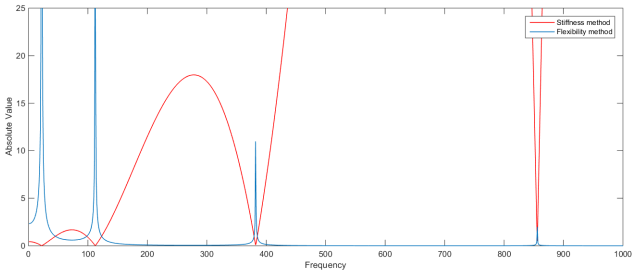

Figure 3.1: Comparison between Transfer Function and Inverse Transfer Function

in order to place the unknowns on the right hand side of the equations and the known variables on the left hand side of the equation one arrives to:

$$\begin{bmatrix} A_{11}m_2\omega^2 - 1 & A_{12} \\ A_{21}m_2\omega^2 & A_{22} \end{bmatrix} \begin{bmatrix} u_2 \\ F \end{bmatrix} = \begin{bmatrix} -A_{12}m_3\omega^2 \\ A_{22}m_3\omega^2 + 1 \end{bmatrix} u_3, \tag{3.12}$$

which in fact are the same equations than the direct stiffness method, isolating as unknown the displacements at point 3 instead of the forces. Solving for a unit displacement at node 3 for each frequency the inverse transfer function is obtained. This transfer function relates displacements on node 3 with the force applied in the edge.

The figure 3.1 reveals the relation between the inverse transfer function and the transfer function computed by the harmonic method for the same bar.

The obtained transfer function is the inverse of the transfer function as expected. One would want now to compute the corresponding impulse response, which is obtained by applying the inverse Fourier Transform. Here appears one of the reasons because a deconvolution problem must be solved: Fourier's inverse transform can not be applied because the obtained function is not bounded; on the contrary, it tends to infinity: $g \to \infty$ when $\omega \to \infty$. Rembeber that Fourier's theory is applied on integrable functions f satisfying:

$$\int_{-\infty}^{\infty} |f(x)| \mathrm{d}x < \infty. \tag{3.13}$$

3.2.2 Training

The GIIR is computed from the definition of the inverse of a function in the convolution theory. This definition was presented in (3.3). The approach for the compu-

tation of the GIIR is to minimize the least squares problem to find the best approximation of $\mathbf{g}$:

$$\Pi = \sum_{l=1}^{L} (h * g - \delta)^2. \tag{3.14}$$

A training set is proposed in order to improve the information under the least squares minimization. As the GIR is available, parametric displacements can be computed. A set of displacements are computed by following:

$$u_l = f_l * h, \tag{3.15}$$

where the forces are created synthetically as:

$$f_i = sin(w_i t) \qquad\qquad w_i = [\omega_{min}, \ldots, \omega_{max}]. \tag{3.16}$$

The function to minimize is then:

$$\Pi = \sum_{l=1}^{L} (u_l * g - f_l)^2 + \lambda S(g), \tag{3.17}$$

where " $*$ " denotes the convolution operator applied component to component. Considering a number m of training sets, each $\mathbf{u}_\ell$ is computed applying equation (1.2) for different synthetic forces p_ℓ for $\ell = 1, \ldots, m$. The matrix form of the solution to the Tikhonov's regularization reads:

$$\mathbf{g} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{p}, \tag{3.18}$$

where $\mathbf{U}$ is built from $\mathbf{u}_l(t)$, which contains the functions $\mathbf{X}_i$, $\mathcal{E}_i$ and the functions $\boldsymbol{\Upsilon}_i$ which are the Toeplitz matrix built from Υ_i, $\mathbf{D}$ comes from the regularization condition imposed over $\mathbf{g}(t)$ and $\mathbf{p}$ is the vector coming from $p_l(t)$.

3.2.3 Avoiding regularization with the separated representation

The separated construction of the solution implemented in the PGD algorithm allows to avoid the use of the regularization as proved in section 3.3. In fact, the bad conditioning of the Toeplitz matrix inversion can appear in the PGD algorithm, but the convergence can be achieved during the enrichment process. The approach can be related to the Truncated Singular Values Decomposition detailed in section 1.2.

This method expresses an approximated solution as a decomposition in singular values and singular right and left vectors as:

$$\tilde{\mathbf{f}} = \sum_{i=1}^{n} \frac{\boldsymbol{\chi}_i^T \mathbf{u}}{\sigma_i} \boldsymbol{\eta}_i. \tag{3.19}$$

High singular values are associated with high frequencies, which are the most numerically affected by noise. A precise truncation can eliminate the spurious information provided for singular vectors associated to high frequencies. Notice that no regularization parameter is used in the TSVD. The PGD enrichment process computes first the modes associated with more *energy*, so an appropriate stop criterion can avoid the introduction of spurious modes. Farther research must be done in order to relate PGD and TSVD solutions.

3.2.4 Computation of the generalized inverse impulse response

To obtain the GIIR from equation (3.17), a training set of forces and displacements must be known in advance. If m training sets are computed, the minimization problem in separated form reads:

$$\Pi(\mathbf{g}) = \sum_{\ell=1}^{m} \left(\left(\sum_{i=1}^{n} \mathbf{X}_i \ (W_i(t) * p_\ell(t)) \ \mathcal{E}_i(E) \right) * \mathbf{g}(t,E) - p_\ell(t) \right)^2. \tag{3.20}$$

The displacements have been introduced in the parametric form, which has been computed by doing:

$$\mathbf{u}_\ell(t,E) = p_\ell(t) * \mathbf{h}(t,E) = \sum_{i=1}^{n} \mathbf{X}_i(x) \ (W_i(t) * p_\ell(t)) \ \mathcal{E}_i(E). \tag{3.21}$$

If one is interested in the computation of the displacements in some nodes, for example, those points where the displacements will be measured, a subset of points can be extracted from $\mathbf{X}$ and collected in $\boldsymbol{\alpha}$. The corresponding subset of $\mathbf{g}$ is noted as $\bar{\mathbf{g}}$. The resulting minimization equation is written as:

$$\sum_{\ell=1}^{m} \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \Upsilon_{i,\ell} \mathcal{E}_i \right)^T \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \Upsilon_{i,\ell} \, \mathcal{E}_i \right) * \bar{\mathbf{g}} = \sum_{\ell=1}^{m} \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \Upsilon_{i,\ell} \mathcal{E}_i \right)^T p_\ell, \tag{3.22}$$

where:

$$\Upsilon_{i,\ell} = W_i(t) * p_\ell(t). \tag{3.23}$$

Introducing the notation:

$$\Lambda_{i,\ell}(t) = \Upsilon_{i,\ell}(t) p_\ell(t), \tag{3.24}$$

it results:

$$\sum_{\ell=1}^{m} \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \boldsymbol{\Upsilon}_{i,\ell} \mathcal{E}_i \right)^T \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \boldsymbol{\Upsilon}_{i,\ell} \, \mathcal{E}_i \right) \bar{\mathbf{g}} = \sum_{\ell=1}^{m} \sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \boldsymbol{\Lambda}_{i,\ell} \mathcal{E}_i, \tag{3.25}$$

where $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Lambda}$ are the Toeplitz matrices of Υ and Λ respectively. The final number of operators is equal to $n^2 \cdot m$.

Now the PGD algorithm is applied to find a separated solution of $\bar{\mathbf{g}}$ in the form:

$$\bar{\mathbf{g}}(t, E) = \sum_{k=1}^{r} \mathbf{a}_k(x) \Gamma_k(t) \beta_k(E), \tag{3.26}$$

where $\mathbf{a}$, $\Gamma(t)$ and $\beta(E)$ meaning is equivalent to $\boldsymbol{\alpha}$, W and $\mathcal{E}$ in $\mathbf{h}$. The computation of $\bar{\mathbf{g}}$ can be done by following the same procedure as in 3, but for the sake of clarity a detailed process is also presented in Appendix 2.

3.3 Results

Consider the structure in the figure 3.2, which consists in a square plate of size 1×1 with a circular hole of radius 0.5m. The structure has been discretized with a mesh containing 124 elements and 78 nodes. As boundary conditions, the structure is fixed in the bottom of the surface and the forces are applied on the left side. The mass density is $\rho = 1kg/m^3$, and the stiffness is considered as a parameter with possible values in the range $E \in I_E = [10, 200]$Pa with an interval of discretization of $\Delta E = 10$Pa. The structural parameters are detailed in table 3.3. The Kelvin-Voigt time constant μ is set in order to obtain a damping factor ξ of 10%.

Young's modulus	100 Pa
Density	$1N/m^3$
Damping ratio	10%

In the *offline* phase, both GIR and GIIR are computed. The computation of the GIR is done by considering the frequency and the stiffness as parameters in the

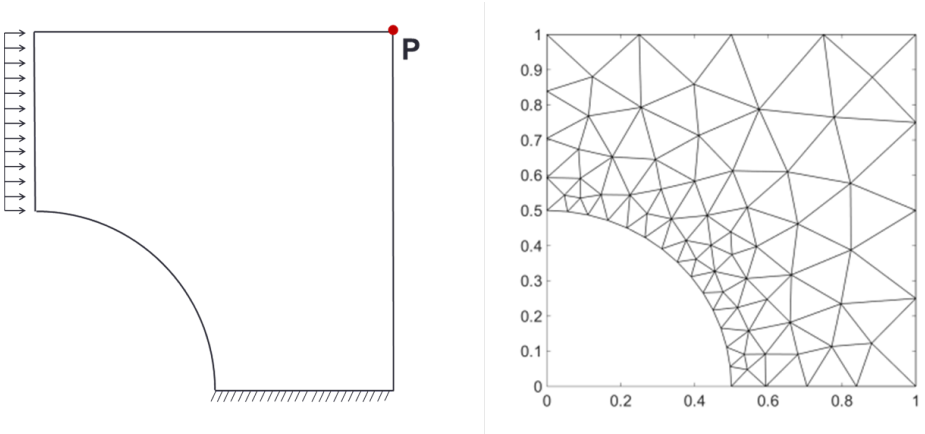

Figure 3.2: 2D plane stress structure

PGD algorithm. The frequency is discretized by considering a range $\omega \in I_\omega = [0, 500]$Hz, with an interval of 10mHz. This discretization leads to a time step $\Delta t = 1ms$ and a time length of 100s after the Inverse Fourier Transform. 15 PGD modes have been considered and a tolerance of $7.9 \cdot 10^{-3}$ has been achieved.

For the computation of the GIIR a training set of 10 forces $f = cos(\omega_{tr} t)$ has been considered. This set of forces covers a range of frequencies $\omega_{tr} \in I_\omega = [1, 50]$Hz, which pretends to represent the range of frequencies that could be applied on the structure. A time window of 350 milliseconds has been considered for the computation. The PGD solution of the GIIR results in a parametric solution containing 80 modes and an achieved tolerance of 0.22.

Figures 3.3, 3.4 and 3.5 show the space, frequency and Young's modulus modes of the DIR. Figures 3.6 and 3.7 show the frequency and Young's modulus modes of the GIIR.

In the *online* phase displacements and forces can be computed. Three forces classified as low, medium and high frequency (respect to the training frequency range) are randomly created as a combination of simple sinusoidal signals. The amplitude of the conforming signals are randomly selected from a range of $[0.1, 2$N, and the frequencies are randomly selected from a range of frequencies $[1, 5]$Hz for *Force 1*,$[3, 25]$Hz for *Force 2* and $[4, 50]$Hz for *Force 3*. The final selected spectrum

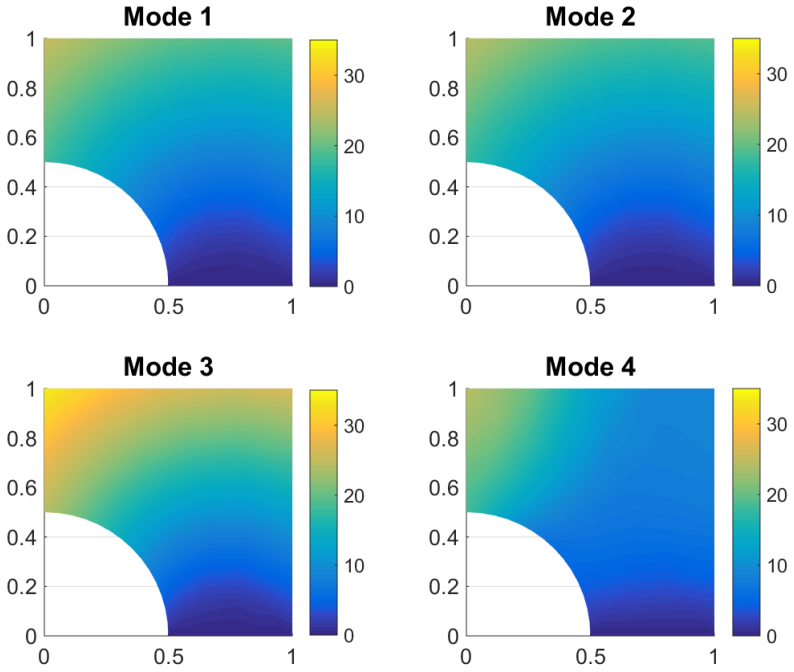

Figure 3.3: First 4 space modes of the generalized transfer function

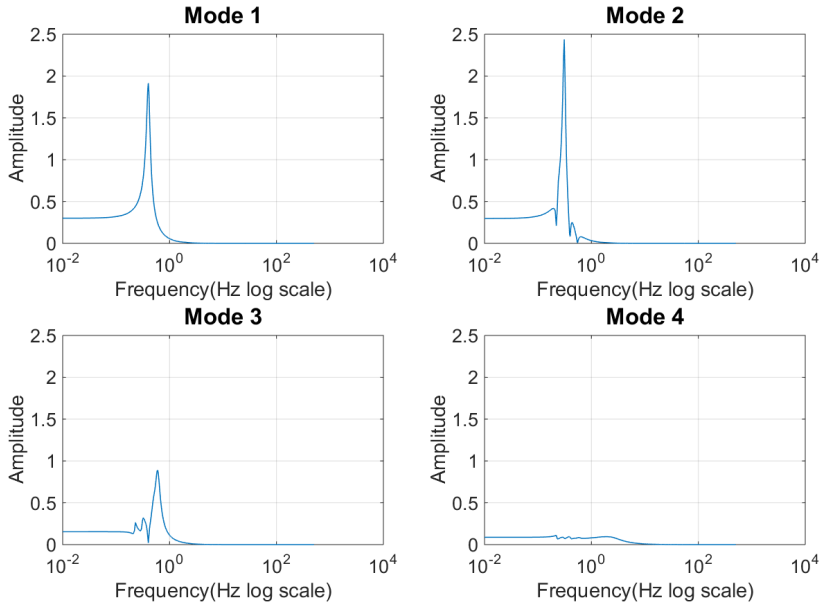

Figure 3.4: First 4 frequency modes of the generalized transfer function in logarithmic scale

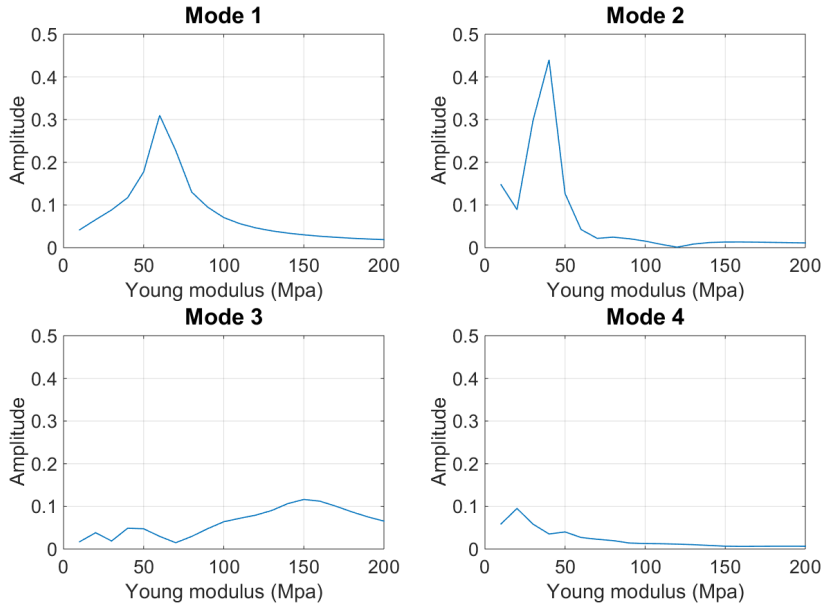

Figure 3.5: First 4 Young modulus modes of the generalized transfer function

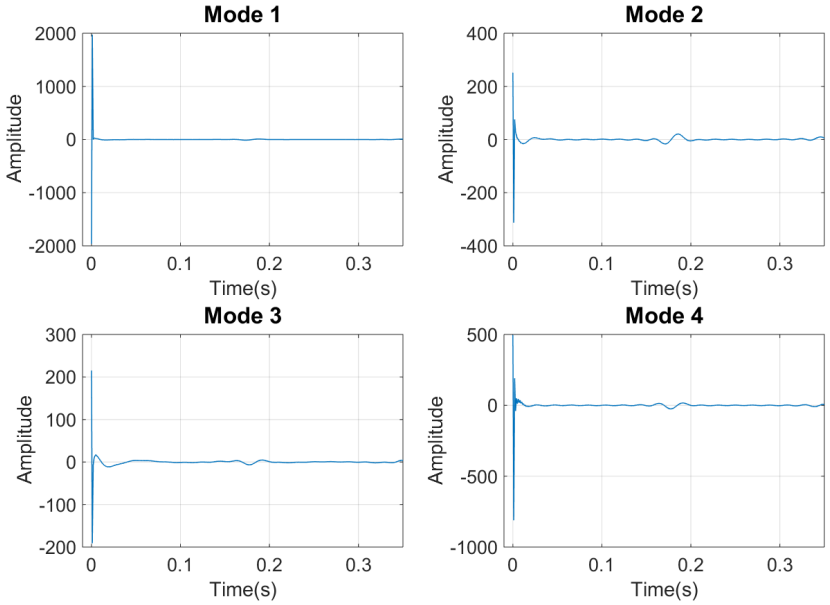

Figure 3.6: First 4 time modes of the GIIR

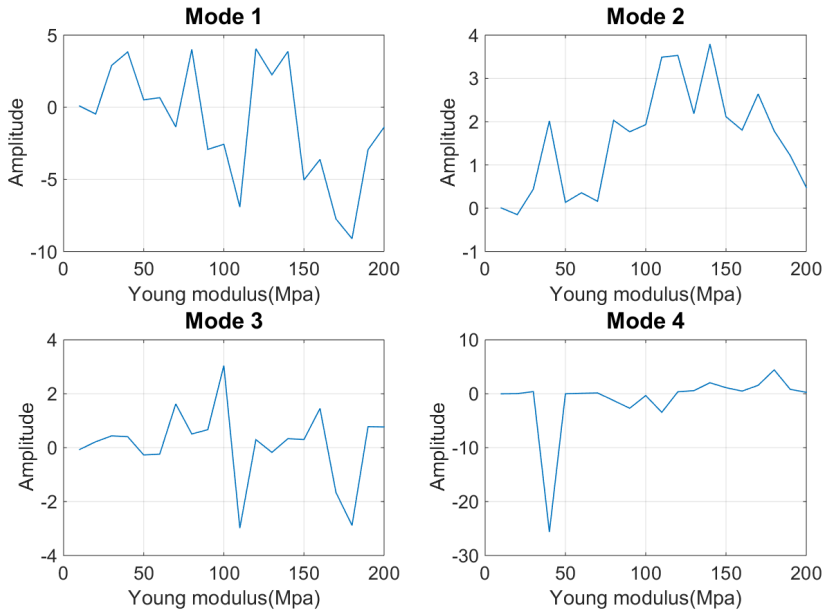

Figure 3.7: First 4 Young modulus modes of the GIIR

is detailed in table 3.1 and the three forces are shown in figure 3.8.

Previous to the computation of the displacements, a value of the stiffness parameter corresponding to 100Pa is chosen. Once both GIR and GIIR are parametrized, the corresponding impulse response and inverse impulse response at the point **P** are obtained. They are shown in figures 3.9 and 3.10.

The displacements are computed in real time at point **P** of the structure and shown on figure 3.11. At the same time, the external applied forces are recovered simply by applying:

$$u_{P_f}(t) = \int_0^t f(t-\tau)\, h_{P_f}(\tau)\, \mathrm{d}\tau, \quad ; \quad f(t) = \int_0^t u_{P_f}(t-\tau)\, g_{P_f}(\tau)\, \mathrm{d}\tau, \quad (3.27)$$

where h_p is the impulse response that allows to compute the displacements in the point p, and g_p is the impulse response that computes the force f by convolution with the displacements in the point p. Results recovering the applied forces by measuring the displacements at point **P** are shown in figures 3.12. The error com-

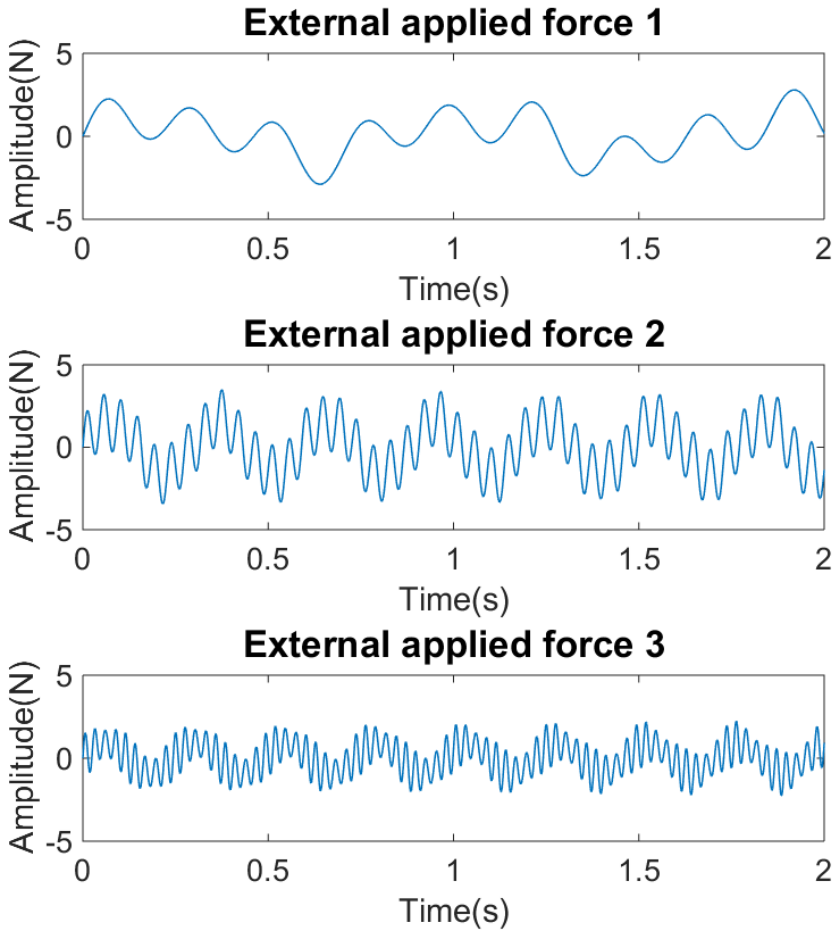

Figure 3.8: External applied forces on the left side of the structure

Forces		Frequency (Hz)	Amplitude (N)
Force 1	f_1	1.1	1.2
	f_2	0.5	2.7
	f_3	1.3	4.3
Force 2	f_1	0.2	8.7
	f_2	1.7	21.8
	f_3	1.6	3.4
Force 3	f_1	0.9	4.1
	f_2	0.3	44.9
	f_3	1.1	37.2

Table 3.1: External applied forces spectrum

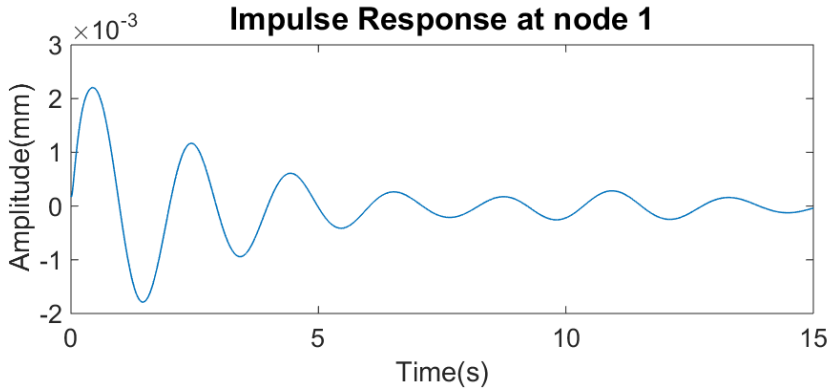

Figure 3.9: Impulse response at point *P* for a selected value of 100 Pa

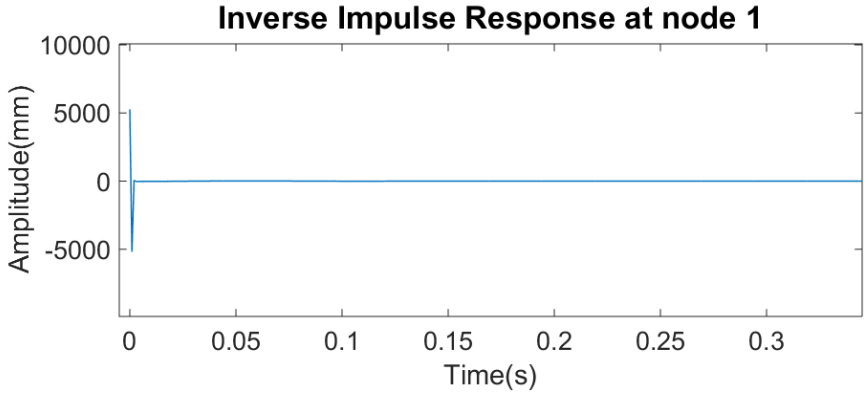

Figure 3.10: Inverse impulse response at point *P* for a selected value of 100 Pa

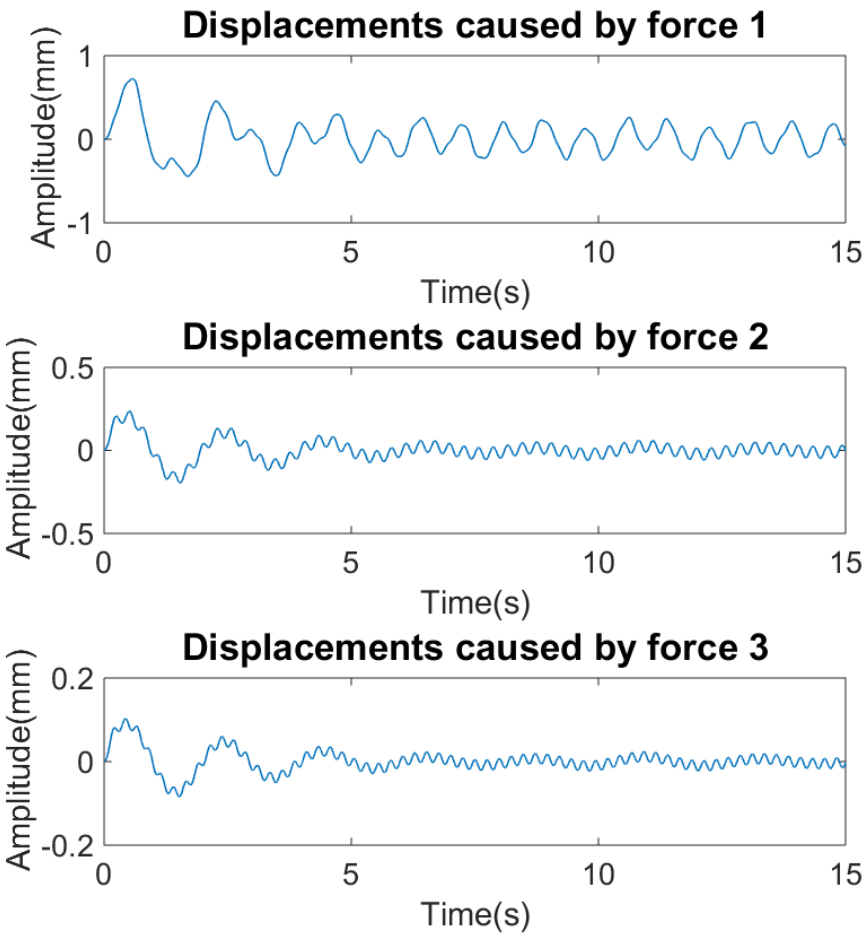

Figure 3.11: Displacements at point *P*

ϵ_1	3.510^{-6}
ϵ_2	6.310^{-7}
ϵ_3	8.110^{-7}

Table 3.2: Error in the recovery of the forces

puted as (3.28) is shown in table 3.2.

$$\epsilon_i = \frac{||F_{recovered} - F_{exact}||}{F_{exact}||} \tag{3.28}$$

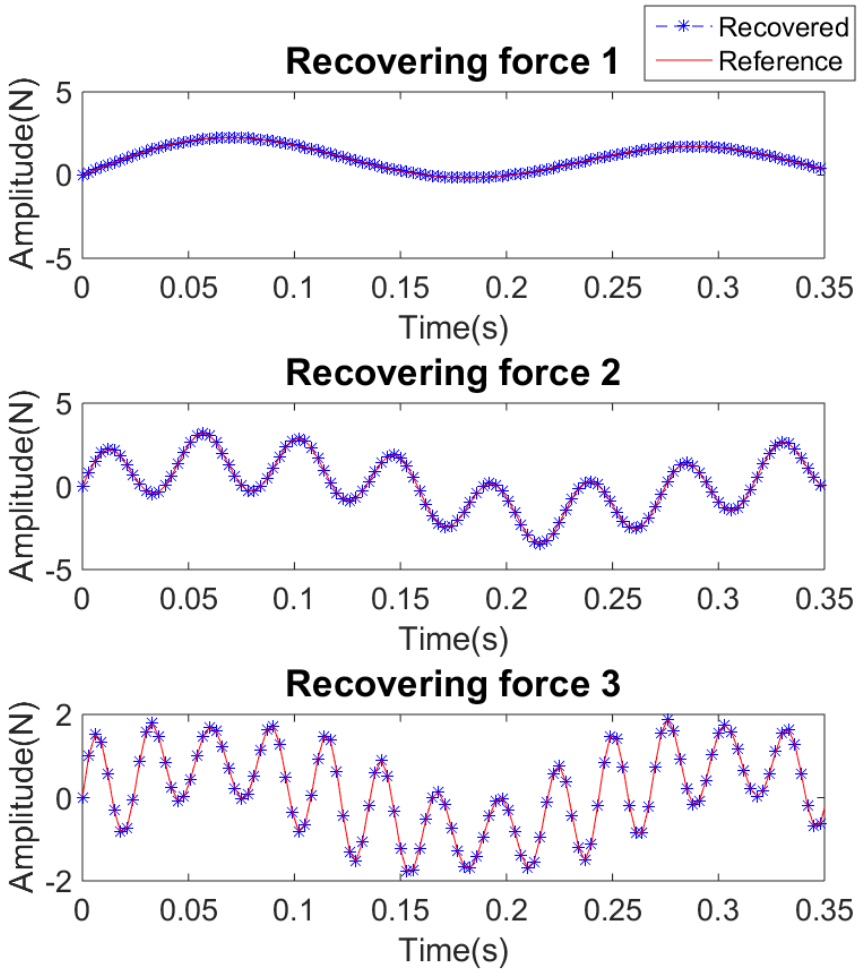

Figure 3.12: Recovered force by measuring at node 1

Chapter 4

Nonlinear applications of the Generalized Impulse Response

Soft robots and the manipulation of soft materials by robots are two incoming issues in the industry. New uses, applications or materials are evolving very fast in the last years. Soft robots characteristics as adaptability, flexibility or compliance bring new opportunities to the industry which can not be achieved by usual robots. Biomedical applications, where soft materials can avoid damages to the patience and work in cluttered environments, or worker's safety in factories, where the interaction with the hard parts of the robots can be a hazard, are two of a large list of examples where soft materials are becoming more and more a part of the industry.

Soft robots control and monitoring is being one the most growing fields in the scientific research. The development of the use of soft robots is followed by new methods of computation capable of understand and compute the physics behind it. Despite nonlinear behaviours in structural dynamics have been studied for many years, fast and accurate computations are still a challenge [91][89]. Different approaches have been carried to deal with the dynamics problem: quasi static methods [92], corotational method [90], or mass-spring methods [95] [94] are some of the methods applied in soft robotics.

Separated representations bring a useful tool in real time applications. Precomputing some part of the solution in an *offline* phase can save time computing in the *online* phase. The *Proper Generalized Decomposition* is an *a priori* model order reduction technique which allows the computation of parametric solutions, avoiding the curse of dimensionality [26] [27].

Impulse theory is a classic method [9] which allows to calculate the displace-

ments in structures under dynamic loads. Once the impulse response of the structure is known, the computation of displacements is fast. They can be obtained only in those parts where they are required, avoiding the computation in the whole structure, and consequently, reducing the computational cost.

In this Chapter, the PGD technique is used in order to compute the necessary GIR, which permits to evaluate the dynamics of a structure for different parameters at a low computational cost. Some different nonlinear behaviours are also considered, showing the possibilities of the method.

4.1 Generalized impulse response application in nonlinear problems

One of the most used methods are time integration schemes, which allow the actualization of the parameters at each time step. These methods are easy to compute, and can be applied in linear or nonlinear behaviour. In counterpart, these methods can be time consuming or unstable [88]. Impulse response theory is applied while the behaviour of the structure remains linear, but it can also be applied for some nonlinear behaviours by splitting the problem in linear and nonlinear behaviour.

4.2 Nonlinear external applied force

Recover the dynamics equation:

$$\mathbf{M}\ddot{u} + \mathbf{C}\dot{u} + \mathbf{K}u = \mathbf{F} + \mathbf{Q}(u), \tag{4.1}$$

where a nonlinear behaviour is considered by means of an external force $\mathbf{Q}$ applied on some part of the boundary.

The impulse response functions must be computed on those nodes where a force will be applied. These forces will be both linear external applied force $\mathbf{F}$ and nonlinear external applied forces $\mathbf{Q}(u)$. Once all impulse responses are computed and stored, displacement at point j can be calculated by computing the sum of displacements caused by: the external applied forces plus the nonlinear forces. Note that the impulse response allows to compute the displacements where they are needed, and one can avoid the computation of the displacements on that nodes out of interest, saving time and computational cost.

Then the displacements are computed following:

$$u^j = \int_0^t h_p^j(t-\tau) f_p(t) + \sum_{k=1}^{l} \int_0^t h_k^j(t)(t-\tau) q_k(u_k, t), \tag{4.2}$$

where the upper index indicates the point where the displacements are being computed and the lower index indicates the point where the force is applied.

As q_k depends on u_k, an iteration must be solved for each time step. If the external applied force concerns a few degrees of freedom, the iteration can be solved inexpensively, and the proposed method can be considered as an option for real time computations.

4.3 Nonlinear stiffness

A nonlinear stiffness can be considered in the form:

$$\mathbf{K}(u) = \mathbf{K}_l + \mathbf{K}_n(u), \tag{4.3}$$

where the nonlinear behaviour in the stiffness property can be split in two terms: a linear stiffness $\mathbf{K}_l$ and a nonlinear stiffness $\mathbf{K}_n(u)$. The transfer functions are computed from the linear part of the equation (4.1), and the nonlinear part is computed as a nonlinear term in the right side of the equation, and evaluated during the *online* computation.

4.4 Numerical examples

Consider the problem of the bridge in the figure 4.1, where plane stress has been considered. The discretization of the bridge contains 140 elements and 100 nodes. The displacements and rotations are fixed at the bottom of the left pillar. The mass is $7850kN/m^3$. The stiffness has been considered as a parameter and computed in the range $[100, 300]$ MPa, and the Kelvin-Voigt time constant μ is set in order to obtain a damping factor ξ of 10%. Frequency domain has been discretized in a range of $[0, \ldots, 50]$ Hz and an increment of 10 mHz. The GIR has been computed comprehending the 100 nodes (excluding those with restricted displacements and rotations), and used for the displacements computation.

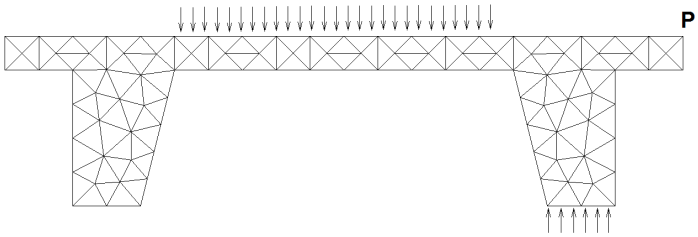

Figure 4.1: Discretization elements of the 2D plane stress bridge

Young's modulus	[100 300] MPa
Density	$7850kN/m^3$
Damping ratio	10%

4.4.1 Nonlinear external applied force

A nonlinear force acting on the down part of the right pillar is considered by the equation:

$$\mathbf{F}_{nl} = A_f sin(\mathbf{u} \cdot B_f), \tag{4.4}$$

where A_f and B_f are arbitrary constants.

An external force, as shown in figure 4.2, is applied on the top of the structure, and the displacements at the point *P* are computed in real time. In the figure 4.3, a comparison between the linear displacements (without considering the nonlinear term) and the nonlinear displacements (considering the nonlinear term) are shown for some values of the arbitrary constant A_f. Linear behaviour is shown in order to evaluate how significant are the considered nonlinearities. The linear displacements are named as *linear* in the figure, **PGD** stands for the nonlinear behaviour computed by applying the procedure explained in 4 and *Newmark* stands to the solution of the problem computed by the Newmark method. In the computation of the Newmark method, a time step Δt of $1.5 \cdot 10{-}11$ has been used, and the values of the Newmark parameters were $\beta_1 = 0.25$ and $\beta_2 = 0.5$. Figure show good results, even when the nonlinear behaviour completely differs from linear behaviour.

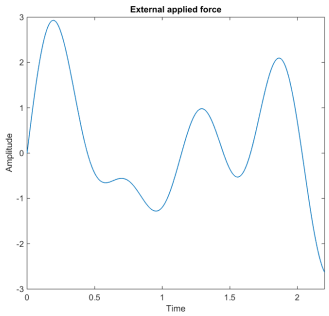

Figure 4.2: External applied force

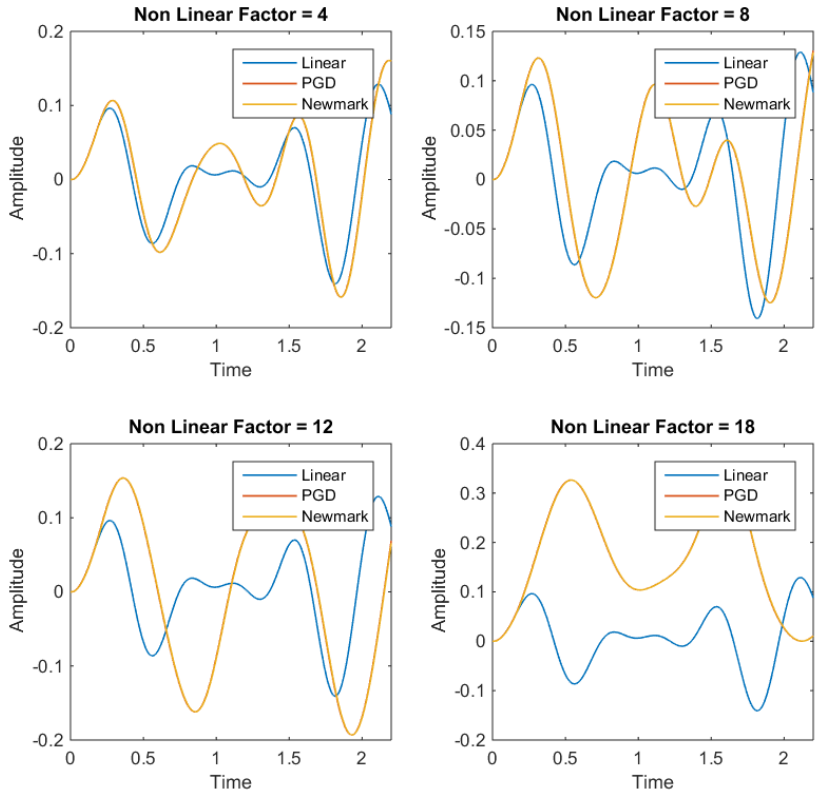

Figure 4.3: Displacements at point P for different values of A_f

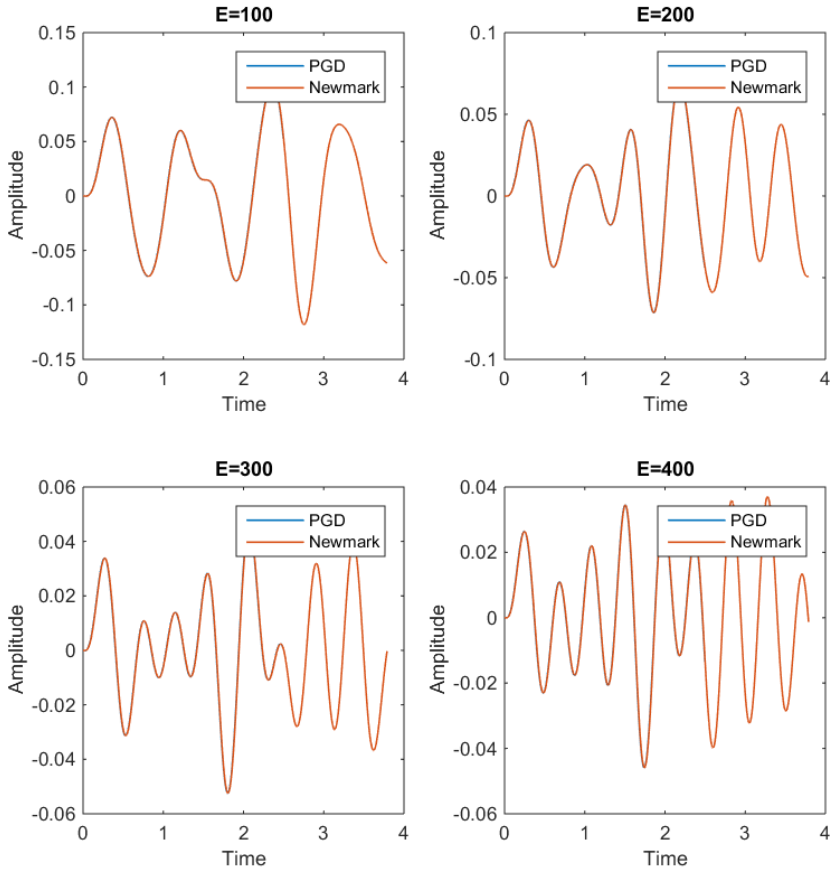

Figure 4.4: Displacements at point *P* for different values of the Young's modulus

4.4.2 Nonlinear stiffness

In this example, a nonlinear stiffness concerning the whole structure has been considered as:

$$\mathbf{K}_n(u) = 0.2\mathbf{K}\mathbf{u}, \tag{4.5}$$

where the term $\mathbf{K}_n(u)$ is updated at each time step.

An external applied force is applied in the top of the structure. The applied force is shown in figure 4.2. In the figure 4.4, the displacements for Young's modulus values $[100, 200, 300, 400]$MPa are shown. The impulse response method applied on this nonlinear case show good accuracy accordingly to the images.

Conclusions

Three challenges in structural dynamics problem are addressed at the same time:

- Real time computation.
- Monitoring of forces, which leads to the resolution of ill-posed inverse problems.
- Multi-dimensional problems.

In this thesis, a method to compute displacements and forces in real time for multidimensional problems is developed. It is also presented an application of the method for some nonlinear problems.

The adopted solution is a separated representation solution computed by means of an *a priori* model order reduction method (PGD) applying the harmonic analysis theory on the dynamics equation. This approach gives a solution the three aforementioned challenges at the same time:

- Harmonic analysis theory brings a separated form in frequency of the dynamics equation, which is suitable for the *a priori* model order reduction technique PGD. The application of this theory leads to the impulse theory in time domain, and therefore, a fast computation procedure is obtained.
- The properties of the PGD algorithm allow to compute the solution of the inverse problem avoiding the regularization parameters of the Tikhonov family methods or the relaxation parameters in the iterative methods, which commonly differs from one problem to another, and whose evaluation is difficult to automatize.
- The resolution of multi-dimensional problems is also solved by the PGD method, in which additional parameters do not compromise the computational cost.

Impulse response theory brings two advantages when dealing with linear invariant systems: the information of the system is in some way condensed in the transfer functions and uncoupled in the space domain, it is, the displacements can be computed in some nodes avoiding the computation in the whole space, reducing the number of computational operations. Displacements caused by an external force are then computed by multiplication between the transfer function and the force in the frequency domain, or by convolution between the impulse response and the force in the time domain: both operations are computationally inexpensive, and suitable to be applied on real time computations.

Precomputed multi-dimensional solutions are actually a promising option to address the curse of dimensionality, even if computational power is growing, specific algorithms are needed for multi-dimensional approaches rather than classic Monte-Carlo simulations. In this field, PGD algorithm is well defined for this kind of problems when the solution can be expressed as a separated representation. Fourier relations between time and frequency domains allow to take profit of each domain advantages. In this thesis, the frequency analysis is applied to obtain a separated form of the frequency parameter ω, which will be useful for the resolution of the multi-dimensional problem, and the time domain resolution to take profit of the advantages for real time computation.

The multi-parametric resolution of the harmonic equation leads to the computation of the Generalized Transfer Function and the Generalized Impulse Response. Both generalized solutions can be applied in optimization problems or real time applications in structural dynamics ,e.g., the monitoring of displacements. The computation of both functions is detailed in Chapter 2.

The monitoring of forces approach is developed in Chapter 3. The chosen approach is to use the knowledge about the direct problem to resolve the inverse problem. This inverse problem is defined as the computation of the required external force to achieve a certain configuration in displacements. The arising theoretical and numerical difficulties when dealing inverse problems are detailed in Chapter 1. A state of the art in inverse problems, and more specifically, in deconvolution problems, is presented. Deconvolution is the inverse of the convolution operation, which appears naturally when impulse response theory is applied to solve the dynamics equation. In this work the problem of deconvolution appears in the definition of the GIIR, as it is computed from a convolution equation. Once computed, the GIIR allows to compute the monitoring of the forces in real time. Some results are provided to illustrate the performance of the method.

Finally, some nonlinear problems in structural dynamics are approached by using pre-computed GIR. New industrial procedures or the evolving soft robots demands new methods in simulation and computation. In this thesis is presented the application of the GIR in computing displacements under the effect of some different kinds of nonlinearities.

While the application of the impulse theory is restricted to linear invariant systems, some kind of nonlinearities can be linearised. In this thesis, two examples where the nonlinear behaviour can be divided in a linear behaviour and a nonlinear behaviour are shown. Classic impulse response theory is then applied on the linear behaviour part of the structure and the nonlinear behaviour is solved in real time.

The results show a good performance of the method when the nonlinearities affect a few degrees of freedom, as in the example of the nonlinear external applied force. In this case, displacements are only needed in a part of the boundary, and only a few impulse responses are required. The computation of the displacements is then fast, requiring a small computational cost for the iteration phase. On the other hand, displacements in the whole structure can be also computed by paying the corresponding computational cost when the nonlinearity affects a huge number of degrees of freedom.

Bibliography

Bibliography

[1] S.W. Doebling, C.L. Farrar, M.B. Prime, D.W. Shevitz, *Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review*, Los Alamos National Laboratory Report, LA-13070-MS, 1996.

[2] Farrar, C.R.; Fasel, T. R. et al., *Experimental modal analysis and damage detection in a simulated three-story building*. October 1, 2001. New Mexico, University of North Texas Libraries, Digital Library.

[3] Kong, X. , Cai, C., Hu, J. *The State-of-the-Art on Framework of Vibration-Based Structural Damage Identification for Decision Making*. Applied Sciences. V7. 2017. N5.

[4] S. Giclais, P. Lubrina, C. Stephan. *Aircraft Ground Vibration Testing at ONERA*. AerospaceLab. Journal, ONERA, 2016, p. 1-18.

[5] Lin P. Y., Roschke P. N., Loh C. H., Cheng C. P. *Hybrid controlled base isolation system with semi-active magnetorheological damper and rolling pendulum system*. Taipei, National Center for Research on Earthquake Engineering, 1999.

[6] Abbiati, G., Bursi, O. S., Caperan, P., Di Sarno, L., Molina, F. J., Paolacci, F., and Pegon, P. (2015). Hybrid simulation of a multi-span RC viaduct with plain bars and sliding bearings. Earthquake Engng Struct. Dyn., 44: 2221–2240.

[7] Mei Z, Wu B, Bursi OS, Yang G, Wang Z. Hybrid simulation of structural systems with online updating of concrete constitutive law parameters by unscented Kalman filter. Struct Control Health Monit. 2018;25:e2069.

[8] Körner, T. (1988). Fourier Analysis. Cambridge: Cambridge University Press.

[9] Clough, R. Penzien, J. *Dynamics of structures*, McGraw-Hill, 1975.

[10] Zienkiewicz OC, Taylor RL. *The Finite Element Method*. Vol. 2 Solid Mechanics (5th edn). Butterworth Heinemann: Oxford, 2000.

[11] Fish, J., Belytschko, T. *A First Course in Finite Elements*. 2007. 0470035803. John Wiley and Sons, Inc. USA

[12] Mrabet E., Guedri M., Najib M., Ghanmi S., Soula M. *A new reliability based optimization of tuned mass damper parameters using energy approach*. Journal of Vibration and Control. Vol 24, Issue 1, pp. 153-170. First Published March 9, 2016.

[13] Cen L., Melkote S., *Effect of Robot Dynamics on the Machining Forces in Robotic Milling*, Procedia Manufacturing, Volume 10, 2017, p 486-496.

[14] Tisseur, F. and Meerbergen, K.. *The Quadratic Eigenvalue Problem*. SIAM Rev., 43(2):235-286. 2001.

[15] Wang, L., Zhang, J. Wang, C. *Time-Frequency Analysis of Nonlinear Systems: The Skeleton Linear Model and the Skeleton Curves*. 2003/04/01. N1, 10.1115/1.1545768. Journal of Vibration and Acoustics.

[16] Aguado J.V., Huerta A., Chinesta F., Cueto E. *Real time monitoring of thermal processes by reduced order modeling*. International Journal for Numerical Methods in Engineering, 102 (5), 991-1017, 2015.

[17] Bialecki, R., Kassab, A., and Fic, A. *Proper Orthogonal Decomposition and Modal Analysis for acceleration of transient FEM thermal analysis*. Int. J. Numer. Meth. Engng., 62(6):774-797. 2005.

[18] Maday, Y., Patera, A., and Turinici, G. *A priori convergence theory for reduced-basis approximations of single-parametric elliptic partial differential equations*. J. Sci. Comput., 17(1-4):437-446. 2002.

[19] Ohlberger, M., Rave, S. (2016). *Reduced Basis Methods: Success, Limitations and Future Challenges*. Proceedings Of The Conference Algoritmy, , 1-12.

[20] Chen T., Ohlsson H., Lunj L. *On the estimation of transfer functions, regularizations and Gaussian processes*. Automatica 48(8):1525-35. 2012.

[21] J. W. Cooley, P. A. W. Lewis and P. D. Welch, "The Fast Fourier Transform and Its Applications," in IEEE Transactions on Education, vol. 12, no. 1, pp. 27-34, March 1969. doi: 10.1109/TE.1969.4320436

[22] Folland, G. *Fourier Analysis and Its Applications*. Wadsworth & Brooks/Cole Advanced Books & Software, 1992.

[23] Z Jiang, SJ Kim, S Plude, R Christenson. *Real-time hybrid simulation of a complex bridge model with MR dampers using the convolution integral method*. Smart Materials and Structures. 22 105008. 2013.

[24] Proakis JG, Manolakis DG, *Digital Signal Processing*, 4th edition, Prentice Hall, ISBN, Sec 2. 2002

[25] Zlotnik, S, Díez, P, Modesto, D, and Huerta, A (2015), Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. Int. J. Numer. Meth. Engng, 103, 737–758. doi: 10.1002/nme.4909.

[26] Chinesta, F., Ladevèze, P., and Cueto, E. *A short review on model order reduction based on Proper Generalized Decomposition*. Arch. Comput. Methods Eng., 18(4):395-404. 2011.

[27] Chinesta,F., Ammar, A., Leygue, A., Keunings, R.. *An overview of the proper generalized decomposition with applications in computational rheology*. Journal of Non-Newtonian Fluid Mechanics, Elsevier, 2011, 166 (11), pp.578-592.

[28] Chinesta F, Leygue A, Bordeu F, Aguado J, Cueto E, Gonzalez D, Alfaro I, Ammar A, Huerta A. *PGD-based computational vademecum for efficient design, optimization and control*. Archives of Computational Methods in Engineering. 20(1):31-59. 2013.

[29] L. Tamellini, O. Le Maître, and A. Nouy. Model Reduction Based on Proper Generalized Decomposition for the Stochastic Steady Incompressible Navier-Stokes Equations. SIAM Journal on Scientific Computing 2014 36:3, A1089-A1117

[30] Neron, D, Boucard, P-A, and Relun, N (2015), Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context. Int. J. Numer. Meth. Engng, 103, 275-292. doi: 10.1002/nme.4893.

[31] Alberto Badías, Icíar Alfaro, David González, Francisco Chinesta, Elías Cueto. Reduced order modeling for physically-based augmented reality. Computer Methods in Applied Mechanics and Engineering, Elsevier, 2018, <10.1016/j.cma.2018.06.011>. <hal-01829292>

[32] Gómez-Aguilar, J.F., Yépez-Martínez, H., Calderón-Ramón C., Cruz-Orduña, I. Escobar-Jiménez R.F. Olivares-Peregrino V.H., Modeling of a Mass-Spring-Damper System by Fractional Derivatives with and without a Singular Kernel. Entropy. V. 17. 2015. N.9. P. 6289-6303.

[33] Li He, Gang Qin, Yunqing Zhang, Liping Chen. Non-stationary Random Vibration Analysis of Vehicle with Fractional Damping. IEEE. 22008.

[34] Lawson C.L. and Hanson R.J. (1974): Solving Least Squares Problems. —Englewood Cliffs, NJ: Prentice-Hall

[35] Santamarina J.C., Fratta D. (1998). Introduction to Discrete Signals and Inverse Problems in Civil Engineering, ASCE, 327 pgs.

[36] P. Jiménez-Rodríguez, S. Maghsoudi, G.A. Muñoz-Fernández, Convolution functions that are nowhere differentiable, Journal of Mathematical Analysis and Applications, Volume 413, Issue 2, 2014, Pages 609-615, ISSN 0022-247X, https://doi.org/10.1016/j.jmaa.2013.12.008.

[37] Patnaik, S. An integrated force method for discrete analysis. Int. J. Numer. Meth. Engng. V. 6. 0029-5981. 2005.

[38] C.A. Felippa, K.C. Park. A direct flexibility method, Computer Methods in Applied Mechanics and Engineering, V. 149. Issues 1–4. 1997. Pages 319-337. ISSN 0045-7825.

[39] Reynders E., Roeck G. A local flexibility method for vibration-based damage localization and quantification, Journal of Sound and Vibration. V. 329. Issue 12. 2010. p. 2367-2383. ISSN 0022-460X.

[40] Paris, F., ETSII. Cálculo matricial de estructuras. 1980. Escuela Técnica Superior de Ingenieros Industriales, Sección de Publicaciones.

[41] Hadamard, Jacques (1902). Sur les problèmes aux dérivées partielles et leur signification physique. Princeton University Bulletin. pp. 49-52.

[42] C.R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, vol. 23, Society for Industrial Mathematics, 2002.

[43] Bunch, J. *Stability of Methods for Solving Toeplitz Systems of Equations*. SIAM J. Sci. Statist. Comput. V6. 1985. pp. 349 364.

[44] Wing, M. *A Primer on Integral Equations of the First Kind: The Problem of Deconvolution and Unfolding*, 1991, Society for Industrial and Applied Mathematics.

[45] Richardson, L., *The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, 210, 307-357, 1911, The Royal Society.

[46] Engl HW, Hanke M, Neubauer A, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

[47] Landweber, L. *An iteration formula for Fredholm integral equations of the first kind*. Amer. J. Math. 73, 615–624, 1951.

[48] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Fluids Engineering, vol. 82, no. 1, pp. 35-45,1960

[49] Bertero, M., Bocacci, P., *introduction to inverse problems in imaging*, 1998, CRC Press.

[50] Tikhonov AN, Arsenin VY. *Solutions of ill-posed problems* Wiley. New York. 1997.

[51] Kochukhov O., Makaganiuk V., Piskunov N. *Least-Squares deconvolution of the stellar intensity and polarization spectra*. Astron. Astrophys.524, A.5.

[52] Heinz. E, Hanke M., Neubauer, A. *Regularization of Inverse Problems*. 375. 1996. Kluwer Academic Publishers.

[53] Strohmer, T. Vershynin, R. *A randomized Kaczmarz algorithm for linear systems with exponential convergence*. 2009. Journal of Fourier Analysis and Applications, 15 (2): 262–278.

[54] Bandžuch, P., Morháč M. Krištiak J. *Study of the Van Cittert and Gold iterative methods of deconvolution and their application in the deconvolution of experimental spectra of positron*. 1997, 506-515. Nuclear Instruments and Methods in Physics Research.

[55] Demoment G. *ADELE. A fast suboptimal estimator for real time deconvolution*. 1983. Vol 19. Electronic Letters.

[56] Zanella R., Zanghirati G., Cavicchioli R., Zanni L., Boccacci P., Bertero M., Vicidomini G. *Towards real-time image deconvolution: application to confocal and STED microscopy*. Scientific Reports 3 2523, 2013.

[57] Ljung, L. *State of the art in linear system identification: time and frequency domain methods*, Proceedings of the 2004 American Control Conference, 2004, p650-660 vol.1.

[58] Stavroulakis, G. *Inverse and Crack Identification Problems in Engineering Mechanics*. 2001. V.1. Springer.

[59] Silvia. M., Robinson, E. *Deconvolution of Geophysical Time Series in the Exploration for Oil And Natural Gas*, V. 10. 1st Edition. 1979. Elsevier.

[60] Yingying Song, David Brie, El-Hadi Djermoune, Simon Henrot. Regularization parameter estimation for non-negative hyperspectral image deconvolution. IEEE Transactions on Image Processing, Institute of Electrical and Electronics Engineers, 2016, 25 (11), pp.5316-5330.

[61] Levin, E. and Meltzer, A., *Estimation of the Regularization Parameter in Linear Discrete Ill-Posed Problems Using the Picard Parameter*, SIAM Journal on Scientific Computing, V39, N6, p. A2741-A2762. 2017.

[62] R.A. Renaut, K. Hnetynkova, J. Mead, *Regularization parameter estimation for large-scale Tikhonov regularization using a priori information*, Comput. Stat. Data An. 54 (12) (2010) 3430–3445.

[63] Hansen, P.C. Deconvolution and Regularization with Toeplitz Matrices. Numerical Algorithms (2002) 29:323.

[64] Christian Hansen, Per. (1990). The discrete Picard condition for discrete ill-posed problems. BIT. 30. 658-672. 10.1007/BF01933214.

[65] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion (SIAM, Philadelphia, PA, 1998).

[66] P.C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. SIAM Rev., 34 (1992), pp. 561-580.

[67] Krawczyk-Stańdo, D., Rudnicki, M. (2007). Regularization Parameter Selection in Discrete Ill-Posed Problems — The Use of the U-Curve, International Journal of Applied Mathematics and Computer Science, 17(2), 157-164.

[68] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards 49 (1952) 409–436.

[69] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes 3rd Edition: The art of Computing. 2007. Cambridge.

[70] Ronald. Musicus, Bruce. (1981). Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices. Internal Report, Lab. of Electronics, MIT.

[71] Burdick JT, Murray JI. Deconvolution of gene expression from cell populations across the C. elegans lineage. BMC Bioinformatics. 2013;14:204. doi:10.1186/1471-2105-14-204.

[72] Jérôme Idier. Bayesian Approach to Inverse Problems. ISTE Ltd and John Wiley and Sons Inc, pp.384, 2008. hal-00400668

[73] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero. Efficient deconvolution methods for astronomical imaging: algorithms and IDL-GPU codes. Astronomy and Astrophysics, 539 (2012) A133.

[74] Robinson, Enders. Tomographic Deconvolution of Echograms, Communications, Computation, Control, and Signal Processing: a tribute to Thomas Kailath. 1997. Springer US. Boston MA. p. 535-546.

[75] Einicke, G.A.; White, L.B. (September 1999). "Robust Extended Kalman Filtering" (PDF). IEEE Trans. Signal Processing. 47 (9): 2596–2599

[76] Mirza M. A Modified Kalman Filter for Non-gaussian Measurement Noise. Springer. 2011. p. 401-409.

[77] L. Eldén, An efficient algorithm for the regularization of ill-conditioned least squares problems with triangular Toeplitz matrices, SIAM J. Sci. Statist. Comput. 5 (1984) 229-236.

[78] A.W. Bojanczyk, R.P. Brent and F.R. de Hoog, QR factorization of Toeplitz matrices, Numer. Math. 49 (1986) 210-221.

[79] H. Park and L. Eldén, Stability analysis and fast algorithms for triangularization of Toeplitz matrices, Numer. Math. 76 (1997) 383-402.

[80] J.N. Franklin, Minimum principles for ill-posed problems, SIAM J. Math. Anal. 9 (1978) 638–650.

[81] Lassas M, Saksman E and Siltanen S 2009, Discretization invariant Bayesian inversion and Besov space priors. Inverse Problems and Imaging 3(1), pp. 87-122.

[82] Zhaohui Gao, Dejun Mu, Shesheng Gao, Yongmin Zhong, Chengfan Gu. Adaptive unscented Kalman filter based on maximum posterior and random weighting. Aerospace Science and Technology. V. 71. 2017. P 12-24. ISSN 1270-9638.

[83] T. Lacey, Tutorial;: The Kalman Filter. Computer Vision

[84] Corey Winton, Jackie Pettway, C.T. Kelley, Stacy Howington, Owen J. Eslinger. Application of Proper Orthogonal Decomposition (POD) to inverse problems in saturated groundwater flow. Advances in Water Resources. V. 34, I. 12. 2011. P 1519-1526, ISSN 0309-1708.

[85] Roberto Melli, Enrico Sciubba, Claudia Toro. An application of the Proper Orthogonal Decomposition method to the thermo-economic optimization of a dual pressure, combined cycle powerplant. Energy Conversion and Management. V. 85. 2014. P 638-645. ISSN 0196-8904

[86] Diana Ugryumova, Katrina Lau, Julio Braslavsky, Gjerrit Meinsma. An application of system identification techniques to impedance estimation in magnetotelluric surveying. IFAC Proceedings. 2009. N 10. Pag. 970-975. 15th IFAC Symposium on System Identification 2009.

[87] Marco P. Schoen and Ji-Chao Lee, "Application of System Identification for Modeling the Dynamic Behavior of Axial Flow Compressor Dynamics," International Journal of Rotating Machinery, vol. 2017, Article ID 7529716, 14 pages, 2017.

[88] Hughes, T.J.R., *Analysis of Transient Algorithms with Particular Reference to Stability Behavior*. Computational Methods for Transient Analysis, North-Holland, 1983, pp. 67–155.

[89] Duriez. Y, Coevoet, E., Largilliere, F., Bieze T., Zhang Z., et al.. *Framework for online simulation of soft robots with optimization-based inverse model*. SIMPAR: IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Dec 2016, San Francisco, United States. 2016, Proceedings of SIMPAR 2016 conference.

[90] Felippa C., *A systematic approach to the element-independent corotational dynamics of finite elements*, in Technical Report, Center for Aerospace Structures, 2000.

[91] D. Rus and M. T. Tolley, *Design, fabrication and control of soft robots* Nature, vol. 521, no. 7553, 2015

[92] J. Bosman, T. M. Bieze, O. Lakhal, M. Sanz, R. Merzouki and C. Duriez, *Domain decomposition approach for fem quasistatic modeling and control of continuum robots with rigid vertebras* in IEEE International Conference on Robotics and Automation (ICRA), 2015

[93] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, *Real-time simulation of contact and cutting of heterogeneous soft-tissues* Medical image analysis, vol. 18, no. 2, 2014

[94] J. Hiller and H. Lipson, *Dynamic Simulation of Soft Multimaterial 3D-Printed Objects* Soft robotics, vol.1, no. 1, 2014

[95] Kühnapfel, U., Cakmak, H., Maaß, H., 2000. *Endoscopic surgery training using virtual reality and deformable tissue simulation*. Comput. Graph. 24 (5), 671-682.

Appendix

Appendix 1: Computation of the integrals in the PGD algorithm

One of the advantages of the PGD method is to split the multi-dimensional problem into several 1-D problems. Here are detailed the 1-D integrals of the PGD method applied in 2:

- Integrals in frequency ω dimension:

$$
\begin{aligned}
l_{1\omega} &= \int_{\omega_o}^{\omega_f} S^2 \cdot \omega^2 \cdot \mathrm{d}\omega \\
l_{2\omega} &= \int_{\omega_o}^{\omega_f} S^2 \cdot \omega \cdot \mathrm{d}\omega \\
l_{3\omega} &= \int_{\omega_o}^{\omega_f} S^2 \cdot \mathrm{d}\omega \\
r_{1i\omega} &= \int_{\omega_o}^{\omega_f} S \cdot W_i \cdot \omega^2 \cdot \mathrm{d}\omega \\
r_{2i\omega} &= \int_{\omega_o}^{\omega_f} S \cdot W_i \cdot \omega \cdot \mathrm{d}\omega \\
r_{3i\omega} &= \int_{\omega_o}^{\omega_f} S \cdot W_i \cdot \mathrm{d}\omega \\
\eta_{\omega} &= \int_{\omega_o}^{\omega_f} S \cdot \mathrm{d}\omega
\end{aligned}
$$

.

- Integrals in Young's modulus k dimension:

$$
\begin{aligned}
l_{1k} &= \int_{k_o}^{k_f} T^2 \cdot \mathrm{d}k \\
l_{3k} &= \int_{k_o}^{k_f} T^2 \cdot k \cdot \mathrm{d}k \\
r_{1ik} &= \int_{k_o}^{k_f} T \cdot E_i \cdot \mathrm{d}k \\
r_{3ik} &= \int_{k_o}^{k_f} T \cdot E_i \cdot k \cdot \mathrm{d}k \\
\eta_k &= \int_{k_o}^{k_f} T \cdot \mathrm{d}k.
\end{aligned}
$$

- Integrals in space dimension:

$$
\begin{aligned}
l_{1x} &= \mathbf{R}^H \mathbf{M} \mathbf{R} \\
l_{2x} &= \mathbf{R}^H \mathbf{C} \mathbf{R} \\
l_{3x} &= \mathbf{R}^H \mathbf{K} \mathbf{R} \\
r_{1ix} &= \mathbf{R}^H \mathbf{M} \mathbf{X}_i \\
r_{2ix} &= \mathbf{R}^H \mathbf{C} \mathbf{X}_i \\
r_{3ix} &= \mathbf{R}^H \mathbf{K} \mathbf{X}_i \\
\eta_x &= \mathbf{R}^H F.
\end{aligned}
$$

Appendix 2: PGD algorithm for the GIIR computation

The algorithm is started under the assumption that the first $r-1$ terms of the separated representation are computed, and the term r is to be computed:

$$
\mathbf{g} = \mathbf{g}(t, E) = \sum_{k=1}^{r-1} \mathbf{a}(x)\Gamma(t)\beta(E) + RST, \tag{4.9}
$$

where R,S, and T are unknown. If equation 4.9 is introduced in 3.25:

$$
\sum_{\ell=1}^{m} \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \boldsymbol{\Upsilon}_{i,\ell} \mathcal{E}_i \right)^T \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \, \boldsymbol{\Upsilon}_{i,\ell} \, \mathcal{E}_i \right) \left(\sum_{k=1}^{r-1} \mathbf{a}_k \Gamma_k \beta_k + R^* S T \right) =
$$

$$\sum_{\ell=1}^{m}\sum_{i=1}^{n}\boldsymbol{\alpha}_i\,\boldsymbol{\Lambda}_{\ell,i}\mathcal{E}_i. \tag{4.10}$$

Introducing the following notation for the product of modes:

$$\boldsymbol{\Psi}_{i,j} = \boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j, \tag{4.11}$$

$$\boldsymbol{\Phi}_{\ell,i,j} = \mathbf{V}_{i,\ell}^T\mathbf{V}_{j,\ell}, \tag{4.12}$$

$$\Upsilon_{i,j} = \mathcal{E}_i^T\mathcal{E}_j, \tag{4.13}$$

where $i = 1,\ldots,n$ and $j = 1,\ldots,n$, and substituting in (4.10), it results:

$$\sum_{\ell=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\Psi}_{i,j}\,\boldsymbol{\Phi}_{\ell,i,j}\Upsilon_{i,j})\left(\sum_{k=1}^{r-1}\mathbf{a}_k\Gamma_k\beta_k + RST\right) = \sum_{\ell=1}^{m}\sum_{i=1}^{n}\boldsymbol{\alpha}_i\,\boldsymbol{\Lambda}_{\ell,i}\mathcal{E}_i. \tag{4.14}$$

A nonlinearity in form of the product has been introduced, and an alternating direction scheme will be followed to solve it. Starting by setting an initial guess for S and T, R^* is computed:

$$\sum_{\ell=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\Psi}_{i,j}\,\boldsymbol{\Phi}_{\ell,i,j}\Upsilon_{i,j})\,RST =$$

$$-\sum_{\ell=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\Psi}_{i,j}\,\boldsymbol{\Phi}_{\ell,i,j}\Upsilon_{i,j})\sum_{k=1}^{r-1}\mathbf{a}_k\Gamma_k\beta_k + \sum_{\ell=1}^{m}\sum_{i=1}^{n}\boldsymbol{\alpha}_i\,\boldsymbol{\Lambda}_{\ell,i}\mathcal{E}_i. \tag{4.15}$$

Applying Galerkin projection:

$$(RST)^T\sum_{\ell=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\Psi}_{i,j}\,\boldsymbol{\Phi}_{\ell,i,j}\Upsilon_{i,j})\,RST =$$

$$-(RST)^T\sum_{\ell=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\Psi}_{i,j}\,\boldsymbol{\Phi}_{\ell,i,j}\Upsilon_{i,j})\sum_{k=1}^{r-1}\mathbf{a}_k\Gamma_k\beta_k + (RST)^T\sum_{\ell=1}^{m}\sum_{i=1}^{n}\boldsymbol{\alpha}_i\,\boldsymbol{\Lambda}_{\ell,i}\mathcal{E}_i. \tag{4.16}$$

Now the following products can be computed:

$$\mu_{\ell,i,j} = S^T\boldsymbol{\Phi}_{\ell,i,j}S, \tag{4.17}$$

$$\xi_{i,j} = T^T \Upsilon_{i,j} T, \tag{4.18}$$

$$\sigma_{i,\ell,k} = S^T \mathbf{\Phi}_{\ell,i,j} \Gamma_k, \tag{4.19}$$

$$\theta_{i,\ell,k} = T^T \Upsilon_{i,j} \beta_k, \tag{4.20}$$

$$\rho_{\ell,i} = S^T \Lambda_{\ell,i}, \tag{4.21}$$

$$\zeta_i = T^T \mathcal{E}_i, \tag{4.22}$$

Function R can be computed from:

$$R^T \sum_{\ell=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{\Psi}_{i,j} R \mu_{\ell,i,j} \xi_{i,j} =$$

$$-R^T \sum_{\ell=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{\Psi}_{i,j} \sigma_{i,\ell,k} \theta_{i,\ell,k} \sum_{k=1}^{r-1} \mathbf{a}_k(x) + R^T \sum_{\ell=1}^{m} \sum_{i=1}^{n} \boldsymbol{\alpha}_i \rho_{\ell,i} \zeta_i. \tag{4.23}$$

The computation of functions S and T is realized following the same procedure as in the computation of R.