Titre: Title:	Parameterized Model Order Reduction and Data Assimilation towards a Digital Twin of a Vertical Axis Rotating Machine
Auteur: Author:	Sima Rishmawi
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Rishmawi, S. (2025). Parameterized Model Order Reduction and Data Assimilation towards a Digital Twin of a Vertical Axis Rotating Machine [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/63327/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/63327/
Directeurs de recherche: Advisors:	Frederick Gosselin
Programme: Program:	Génie mécanique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Parameterized Model Order Reduction and Data Assimilation towards a Digital Twin of a Vertical Axis Rotating Machine

SIMA RISHMAWI
Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie mécanique

Février 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Parameterized Model Order Reduction and Data Assimilation towards a Digital Twin of a Vertical Axis Rotating Machine

présentée par **Sima RISHMAWI**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Njuki MUREITHI, président
Frédérick GOSSELIN, membre et directeur de recherche
Alain BATAILLY, membre
Amélie FAU, membre externe

DEDICATION

In the tapestry of this dissertation, woven with countless threads of learning and perseverance, there are profound influences that have shaped both the scholar and the person behind these pages. To those who have been my anchors, my guides, and my pillars of support, I dedicate this work with immense gratitude.

To my beloved mother, Ghada, and father, Walid, whose infinite love, efforts, and unwavering support laid the foundation of my journey. Your faith in my dreams empowered me, and each step I made was driven by the principles you cultivated within me.

To my sisters, Sandra and Samar, for their continuous laughter and companionship, which have brought me immense joy. Their support and understanding throughout the ups and downs of this endeavor have made the journey less solitary. To my dear nieces, Lamar and Elena, and my nephew, Ramzi, while you might be too young to fully understand it now, the optimism you exude has been a guiding light in my life each day. You are the hope of the future.

To all my dear close friends, particularly Mirella Sahih, who have been there for me through all the challenges of life. Your constant support, late-night talks, and shared moments of joy have provided comfort and motivation.

To my respected educators who sparked my curiosity. Dima Mogannam Kaibni, Vera Mustaklem, Lutfi Al-Shareef, Zaer Abou Hammour, and William Singhose, your mentorship goes beyond the confines of the classroom, influencing not just my academic comprehension but also my outlook on life.

Lastly, I would like to express my gratitude to my committed supervisor, Frédérick Gosselin, whose intelligence, guidance, and unwavering confidence in my abilities have been the guiding force that has helped me navigate the complex world of research. Your mentorship has played a crucial role in shaping this dissertation, and I am deeply grateful for the opportunities you have given me. You pushed me to the edge, allowing my work to stand out and shine.

By expressing my gratitude to all of you, I acknowledge the various factors that have shaped this academic pursuit. May the content within these pages depict the accomplishments, experiences, and development that have enriched my Ph.D. experience.

ACKNOWLEDGEMENTS

I would like to begin by expressing my sincere appreciation to my supervisor, Frédérick Gosselin. I am extremely grateful for his invaluable guidance, expertise, and encouragement throughout this research endeavor. His insightful feedback and unwavering support have greatly influenced both the development of this thesis and my personal growth as a researcher.

I express my gratitude to our industrial partners, Hydro-Québec and Maya HTT, for their continuous support. Their generous financial contributions and invaluable technical knowledge have played a crucial role in the success of this project. The dedication of all the experts involved in this collaboration, particularly Quentin Dollon, Christopher Blake, and Elsa Piollet, has greatly improved the study and its practical implications. I am sincerely thankful for their partnership and the opportunity to collaborate in advancing knowledge and innovation in our field.

I also express my gratitude to the Natural Sciences and Engineering Research Council of Canada (NSERC), L'Innovation en Énergie Électrique (InovÉÉ), IVADO, and Mitacs for their generous financial support. Their funding has not only enabled the completion of this research project, but has also provided me with the opportunity to explore new areas of investigation and innovation. This support has played a crucial role in the cover of various expenses, such as research materials, equipment, travel, and personal funding, allowing me to delve deeper into the subject matter and generate meaningful findings. I truly appreciate their dedication to promoting academic endeavors and advancing knowledge.

The completion of this project would not have been possible without the assistance of Professor Francisco Chinesta and his team at the Procédés et Ingénierie en Mécanique et Matériaux (PIMM) Laboratory, which is situated at the École Nationale des Arts et Métiers in Paris, France. I express my gratitude for the specific contributions made by Sebastian Rodriguez, a postdoctoral fellow who played an essential part in the development of the techniques that were instrumental in achieving the goals of this thesis.

Finally, my heartfelt thanks go to my colleagues and peers at the Laboratory of Multiscale Mechanics (LM2) at Polytéchnique Montréal for their camaraderie, stimulating discussions, and shared experiences. Their technical and moral support has made this academic journey both intellectually stimulating and rewarding. I would like to specifically mention Hongyan Miao, Souheil Serroud, Ludivine Moyne, Mohammed Abda, Maryam Boukor, Lucas Berthet, and Esmaeil Ghorbani.

RÉSUMÉ

Les jumeaux numériques sont des répliques numériques sophistiquées de systèmes, objets ou processus physiques, répliquant leur comportement actuel en temps réel. En utilisant des technologies comme l'Internet des objets (IoT), l'intelligence artificielle (IA), l'apprentissage automatique et les techniques d'optimisation, les jumeaux numériques offrent une surveillance, une analyse et une optimisation précises de leurs équivalents réels. Leur utilisation s'étend à divers secteurs tels que la fabrication, la santé et la planification urbaine, où ils augmentent l'efficacité, anticipent les besoins de maintenance et stimulent l'innovation. En connectant les mondes physique et numérique, les jumeaux numériques favorisent une prise de décision plus éclairée, optimisent les performances opérationnelles et stimulent le progrès dans divers domaines.

La création d'un jumeau numérique pour un système physique complexe présente des défis notables. La complexité de cette tâche est fortement influencée par l'application particulière et devient plus difficile à mesure que le système augmente en taille et que davantage de paramètres contribuent à son comportement, en particulier dans le cas des systèmes non linéaires. Les unités hydroélectriques en sont un exemple.

Afin de faciliter le développement d'un jumeau numérique pour une unité hydroélectrique à axe vertical, telle que les turbines Francis, Kaplan ou à hélice, l'objectif de cette thèse est de présenter une méthodologie approfondie centrée sur la construction d'un jumeau numérique d'une machine rotative à axe vertical (VARM) soumise à des forces de palier non linéaires à ses extrémités et subissant un déséquilibre de rotation. La VARM a été conçue pour présenter des propriétés dynamiques similaires à celles de la ligne d'arbre de l'alternateur de turbine hydroélectrique. Les deux systèmes présentent des dynamiques vibratoires comparables et peuvent être évalués et surveillés dans des conditions de charge et des paramètres similaires.

Dans cette thèse, le jumeau numérique adopte une approche de modélisation en boîte grise, intégrant un modèle mathématique de la VARM élaboré à partir des principes physiques et représenté par une équation aux dérivées partielles discrétisée, complété par des données mesurées à partir d'une installation expérimentale de la VARM conçue à Polytechnique Montréal pour quantifier les valeurs des paramètres inconnus. Le modèle mathématique décrit précédemment est intrinsèquement non linéaire, reposant sur divers paramètres tels que l'espace, le temps, les propriétés géométriques, les conditions initiales et les conditions aux limites. Cela le rend hautement dimensionnel et coûteux à résoudre avec des techniques de haute fidélité. Et c'est là que les techniques de réduction d'ordre de modèle (MOR) entrent en jeu.

La thèse atteint son objectif principal en deux phases. Initialement, un solveur global non linéaire espace-fréquence est présenté. Ce solveur, appelé méthode Harmonic-Modal Hybrid, utilise l'analyse de base modale pour générer une représentation à faible rang de la solution. Par la suite, ce solveur facilite des calculs rapides de la réponse d'un système en fonction d'un ensemble de paramètres, ce qui construit un modèle de substitution pour les prédictions en temps réel de la réponse paramétrée en utilisant l'algorithme de sparse Proper Generalized Decomposition (sPGD). Pour démontrer la faisabilité, cette méthode est appliquée à un système non linéaire reconnu : une poutre encastrée avec un ressort non linéaire fixé à son extrémité.

Au cours de la deuxième phase de l'étude, l'approche est élargie pour développer un modèle de substitution pour la VARM, en l'intégrant avec des données du déplacement planaire d'un seul nœud situé sur la ligne d'arbre en utilisant la méthode d'assimilation de données Levenberg-Marquardt (LM), afin de trouver les valeurs optimisées de plusieurs paramètres inconnus dans le modèle de la VARM.

Les résultats montrent que la méthode Harmonic-Modal Hybrid a considérablement réduit le temps de calcul pour un scénario particulier par rapport aux méthodes de haute fidélité, accélérant ainsi le développement de l'ensemble de données nécessaire à la construction du modèle de substitution utilisant la sPGD. Cela permet des calculs en ligne rapides pour tout ensemble de paramètres dans les plages spécifiées. De plus, l'intégration de cette méthode avec la technique LM permet d'optimiser efficacement les paramètres inconnus de la VARM, aboutissant à un modèle complet capable de fournir la réponse du système presque instantanément pour tout ensemble de valeurs de paramètres, également connu sous le nom de jumeau numérique.

En fin de compte, la contribution principale de cette thèse est de proposer une méthodologie détaillée pour le développement d'un jumeau numérique pour des systèmes partageant des cadres mathématiques, des caractéristiques dynamiques et des paramètres similaires. Le modèle proposé peut être ajusté et intégré à d'autres sous-modèles, aboutissant à la construction d'une unité de production hydroélectrique pour former un jumeau numérique exhaustif de ces systèmes.

ABSTRACT

Digital twins are advanced digital replicas of physical systems, objects, or processes that mimic their actual behavior in real-time. By leveraging technologies such as the Internet of Things (IoT), artificial intelligence (AI), machine learning, and optimization techniques, digital twins allow accurate monitoring, analysis, and optimization of their real-world counterparts. They find applications across diverse industries, including manufacturing, healthcare, and urban planning, boosting efficiency, forecasting maintenance requirements, and fostering innovation. By connecting the physical and digital realms, digital twins support better decision-making, enhance operational performance, and propel progress across various sectors.

However, creating a digital twin for a complex physical system presents notable challenges. The complexity of this task is heavily influenced by the particular application, and it becomes more challenging as the system grows in size and as more parameters contribute to its behavior, especially in the case of non-linear systems. Hydroelectric generating units serve as an example of such systems.

Toward facilitating the development of a digital twin for a vertical axis hydroelectric unit, such as Francis, Kaplan, or propeller turbine, the objective of this thesis is to introduce a thorough methodology centered on constructing a digital twin of a vertical axis rotating machine (VARM) subjected to non-linear bearing forces at its ends and undergoing rotational unbalance. The VARM was engineered to exhibit dynamic properties akin to those of the hydroelectric turbine-generator shaft line. Both systems exhibit comparable vibrational dynamics and can be evaluated and monitored under similar loading conditions and parameters.

In this thesis, the digital twin adopts a grey-box modeling approach, integrating a mathematical model of the VARM crafted from physics principles and represented by a discretized partial differential equation (PDE), supplemented by data measured from an experimental setup of the VARM designed at Polytechnique Montreal to quantify values of unknown parameters. The previously described mathematical model is inherently non-linear, relying on various parameters such as space, time, geometric properties, initial conditions, and boundary conditions. This renders it high-dimensional and costly to solve with high-fidelity techniques. And this is where model order reduction (MOR) techniques come into play.

The thesis achieves its main goal through two phases. Initially, a global non-linear space-frequency solver is presented. This solver, referred to as the Harmonic-Modal Hybrid Method, uses modal basis analysis to generate a low-rank representation of the solution. Subsequently, this solver facilitates fast calculations of a system's response as a function of a sparse parameter set, which constructs a surrogate model for real-time predictions of the parameterized response using the sparse Proper Generalized Decomposition (sPGD) algorithm. To demonstrate feasibility, this method is applied to a recognized non-linear system: a cantilevered beam with a non-linear spring affixed to its end.

During the second phase of the study, the approach was broadened to develop a surrogate model for the VARM, integrating it with data that recorded the planar displacement of a single node along the shaft line by using the Levenberg-Marquardt (LM) data assimilation method, in order to find the optimized values of several unknown parameters in the VARM's model.

Results show that the Harmonic-Modal Hybrid Method significantly reduced computation time for a particular scenario compared to high-fidelity methods, thereby expediting the development of the sparse data set necessary for constructing the surrogate model using sPGD. This, in turn, enabled swift online calculations for any parameter set within the specified ranges. Additionally, integrating this method with the LM technique efficiently optimized the unknown parameters of the VARM, resulting in a comprehensive model capable of delivering an almost instantaneous system response for any set of parameter values, also known as the digital twin.

Ultimately, the primary contribution of this thesis is offering a detailed methodology for developing a digital twin for systems that share related mathematical frameworks, dynamic characteristics, and parameters. The proposed model can be adjusted and integrated with other sub-models, culminating in the construction of a hydroelectric unit to form an exhaustive digital twin of these systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AFT	Alternating Frequency/Time Method
AI	Artificial Intelligence
DDDAS	Dynamic Data-Driven Application System
DEIM	Discrete Empirical Interpolation Method
DOF	Degree of Freedom
EIM	Empirical Interpolation Method
$\mathcal{FFT}$	Fast Fourier Transform
HMH	Harmonic Modal Hybrid Approach
$\mathcal{IFFT}$	Inverse Fast Fourier Transform
LATIN	LArge Time INcrement
LM	Levenberg-Marquardt
MOR	Model Order Reduction
MQR	Multi-Quadric Radial Basis Function
PDE	Partial Differential Equation
PGD	Proper Generalized Decomposition
PLA	Polylactic Acid
POD	Proper Orthogonal Decomposition
ROM	Reduced-Order Model
RTM	Resin Transfer Molding
SDDAS	Static Data-Driven Application System
sPGD	Sparse Proper Generalized Decomposition
SVD	Singular Value Decomposition
VARM	Vertical Axis Rotating Machine

LIST OF SYMBOLS

GENERAL SYMBOLS

d	Number of variables/dimensions in an arbitrary model
F_j^i	$1D$ Function of variable j in enrichment step i
$\mathcal{L}_0$	Loss function of the LM algorithm
m	Number of parameters in an arbitrary model
$\mathcal{M}$	Number of nodes (DOFs) of an arbitrary model
n_M	Number of data points
N	Number of summation terms (enrichment steps)
$\boldsymbol{p}$	Set of parameters to be optimized using LM
p_i	Parameter i
t	Time
$u_{PGD}(\boldsymbol{x}, t, p_1, p_2, \ldots, p_m)$	PGD representation of the solution of a general PDE
$u_M(\boldsymbol{s}_j)$	Measured data from sensor j
$\boldsymbol{x}$	$3D$ space coordinates

ARTICLE 1

α	Non-linearity coefficient	
γ_1, γ_2	Relaxation hyper-parameters	
$\delta(\square)$	Dirac-delta function	
ζ_i	Damping coefficient of mode i	
τ	Minimum distance between two parameter values in the sparse domain	
ω	Frequency domain	
ω_j	j^{th} natural frequency	$\mathrm{rad\,s^{-1}}$
Ω	Excitation frequency	$\mathrm{rad\,s^{-1}}$
d	Damping coefficient	$\mathrm{kg\,m^{-1}\,s^{-1}}$
E	Young's Modulus	GPa
E_{stg}	Stagnation error percentage	
F_0	Amplitude of the harmonic force	N
h	sPGD hyper-parameter	
I	Cross-sectional area moment of inertia	$\mathrm{m^4}$
k_{NL}	Non-linear force proportionality constant	$\mathrm{N\,m^{-3}}$

L	Length	m
m	Mass per unit length	$\mathrm{kg\,m^{-1}}$
M	Bending moment	N m
n_s	Number of samples	
$N_j^k(\eta)$	Multiquadric radial basis function of the parameter η	
t	Time	s
$u(x,t)$	Transverse displacement of the beam as a function of space and time	m
U_b	Amplitude of the base excitation	
V	Shear force	N
x	Space coordinate along the x-axis	
$\{\overrightarrow{\bar{\mathbf{\Phi}}}\}_j$	Eigenvector of mode j	
$\{\overrightarrow{\mathbf{a_j}}\}$	Vector of constant nodal values calculated by sPGD	
$\{\overrightarrow{\mathbf{c}}\}$	DEIM constant vector of unknown amplitudes	
$\{\overrightarrow{\mathbf{f_0}}\}$	Non-dimensional nodal force	
$\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}$	$\mathcal{FFT}$ of $\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}\cos\left(\bar{\Omega}\bar{t}\right)$	
$\{\overrightarrow{\mathbf{h}}\}$	DEIM constant vector	
$\{\overrightarrow{\bar{\mathbf{u}}}\}$	Non-dimensional nodal displacement	
$\{\overrightarrow{\hat{\mathbf{U}}}\}$	$\mathcal{FFT}$ of $\{\overrightarrow{\bar{\mathbf{u}}}\}$	
$\{\overrightarrow{\bar{\mathbf{y}}}\}$	Non-dimensional modal displacement	
$\{\overrightarrow{\hat{\mathbf{Y}}}\}$	$\mathcal{FFT}$ of $\{\overrightarrow{\bar{\mathbf{y}}}\}$	
$\mathbf{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}$	Basis matrix of of the non-linear force in the frequency domain	
$\mathbf{\Xi}_{\bar{f}_{\mathbf{NLp}}}$	Basis matrix of of the non-dimensional non-linear force in the time domain	
$\mathbf{\Xi}_{\hat{\mathcal{U}}p}$	Basis matrix of the beam tip deflection in the frequency domain	
$\mathbf{\Xi}_{\bar{\mathbb{U}}p}$	Basis matrix of the non-dimensional beam tip deflection in the time domain	
$\mathbf{\Phi}$	Matrix containing eigenvectors	
$\mathbf{\Omega}$	Matrix of eigenvalues	
$\mathbf{d}$	Diagonal modal damping matrix	
$\mathbf{D}$	Non-dimensional damping matrix	
$\bar{f}_{\mathbf{NL}}(1,\bar{t})$	Snapshot matrix of the non-dimensional non-linear force in the time domain	
$\hat{\mathcal{F}}_{\mathbf{NL}}(1,\omega)$	Snapshot matrix of the non-linear force in the frequency domain	
$\mathbf{k}$	Diagonal modal stiffness matrix	

$\mathbf{K}$	Non-dimensional stiffness matrix
$\mathbf{K_{nl}}$	Non-dimensional non-linear stiffness matrix
$\mathbf{m}$	Diagonal modal mass matrix
$\mathbf{M}$	Non-dimensional mass matrix
$\mathbf{P}_p$	Point selection operator
$\mathcal{S}$	Sample matrix
$\hat{\mathcal{U}}(1, \omega)$	Snapshot matrix of the beam tip deflection in the frequency domain
$\bar{\mathbb{U}}(1, \bar{t})$	Snapshot matrix of the non-dimensional beam tip deflection in the time domain
$\partial/\partial t$	Partial derivative with respect to time
$\partial/\partial x$	Partial derivative with respect to space
$\bar{\square}$	Non-dimensional value
$\{\dot{\square}\}$	Time derivative
$\{\ddot{\square}\}$	Second order time derivative

ARTICLE 2

α_d	Coefficient of proportionality to the mass matrix in the Raleigh damping term	
$\delta(\square)$	Dirac-delta function	
$\zeta_{b_{lo}}$	Lower bearing damping ratio	
$\zeta_{b_{up}}$	Upper bearing damping ratio	
ζ_s	Structural damping coefficient	
θ_i	Angle of rotation	°
λ	Hyper-parameter to weight the significance of sparsity	
τ	Number of trials executed within one optimization procedure	
ϕ	1 DOF version of the first eigenvector of the VARM	
ω_1	First modal frequency	$\mathrm{rad\,s^{-1}}$
Ω	Rotational speed of the VARM	RPM
c	Damping coefficient	$\mathrm{kg\,m^{-1}\,s^{-1}}$
$c_{b_{lo}}$	Lower bearing damping coefficient	$\mathrm{kg\,m^{-1}\,s^{-1}}$
$c_{b_{up}}$	Upper bearing damping coefficient	$\mathrm{kg\,m^{-1}\,s^{-1}}$
dt	Time step	s
E	Young's Modulus	GPa
$f_{b_{lo}}$	Lower bearing force	N

$f_{b_{up}}$	Upper bearing force	N
$f_x(z,t)$	External forces due to rotational unbalance along the $x-$axis	N
$f_y(z,t)$	External forces due to rotational unbalance along the $y-$axis	N
F_s	Single spring force	N
F_T	Net spring force	N
I	Cross-sectional area moment of inertia	m^4
k	Lower bearing stiffness	$\mathrm{N\,m^{-1}}$
$\hat{k}$	Stiffness of 1 DOF of the shaft	$\mathrm{N\,m^{-1}}$
k_1 - k_7	Upper bearing stiffness parameters	
k_b	Upper bearing spring stiffness	$\mathrm{N\,m^{-1}}$
k_w, k_z	Upper bearing stiffness parameter	$\mathrm{N\,m^{-1}}$
l_f	Spring's deformed length	m
l_i	Spring's unstretched length	m
L	Length of the VARM shaft	m
$\mathcal{L}_1$	Sparsity-promoting term in the loss function	
$\mathcal{L}_2$	Least squares error term in the loss function	
m	Mass per unit length	$\mathrm{kg\,m^{-1}}$
$m_{b_{lo}}$	Lower bearing mass	kg
$m_{b_{up}}$	Upper bearing mass	kg
m_d	Number of reduced-order modes	
m_e	Eccentric mass	kg
m_{el}	Mass of one element	kg
M	Number of enrichment steps in the sPGD formulation	
n	Number of finite elements	
n_{DOF}	Number of spatial degrees of freedom	
n_{MC}	Number of times the optimization procedure is executed within the Monte Carlo framework	
n_p	Number of parameters	
n_s	Number of scenarios	
n_t	Number of time steps	
$N_j^k(\eta)$	Interpolation function of parameter η	
p_i^j	The i^{th} parameter within the j^{th} scenario	
q_0	Initial shaft bending coefficient	
q_x	Distributed eccentric force coefficient along the $x-$axis	
q_y	Distributed eccentric force coefficient along the $y-$axis	
r	Radius of VARM shaft	m

r_{m_e}	Lever arm of the eccentric mass	m
t	Time	s
T_{2cyc}	Time period covering 2 cycles	s
$u(z,t)$	Transverse displacement of the VARM shaft as a function of space and time along the $x-$axis	m
$u_0(z)$, $v_0(z)$	Minor deformations in the shaft when it is stationary	m
$v(z,t)$	Transverse displacement of the VARM shaft as a function of space and time along the $y-$axis	m
x	Spatial coordinate along the $x-$axis	
y	Spacial coordinate along the $y-$axis	
z	Spacial coordinate along the $z-$axis	
z_{m_e}	Spatial position of the eccentric mass	m
$\{\overrightarrow{\gamma}\}_{z_{m_e}}$	A vector of zeros with 1 at the node linked to z_{m_e}	
$\{\overrightarrow{\mathbf{\Phi_1}}\}$	Primary eigenvector of the system	
$\{\overrightarrow{\mathbf{a_j}}\}$	Vector of constant nodal values calculated by sPGD	
$\{\overrightarrow{\mathbf{f}}\}$	Forcing term amplitude	N
$\{\overrightarrow{\hat{\mathbf{p}}}\}$	Vector of optimized parameters	
$\{\overrightarrow{\mathbf{R}\{\overrightarrow{\mathbf{u}}\}}\}$	Nonlinear contribution that depends on $\{\overrightarrow{\mathbf{u}}\}$	
$\{\overrightarrow{\mathbf{u}}\}$	Nodal displacement along the $x-$axis	m
$\{\overrightarrow{\mathbf{u}^-}\}$	Time dependent solution from the previous iteration	
$\{\overrightarrow{\mathbf{v}}\}$	Nodal displacement along the $y-$axis	m
$\mathbf{C}$	Damping matrix	$\mathrm{kg\,m^{-1}\,s^{-1}}$
$\mathbf{C_s}$	Structural damping matrix	$\mathrm{kg\,m^{-1}\,s^{-1}}$
$\mathbf{K}$	Linear stiffness matrix	$\mathrm{N\,m^{-1}}$
$\mathbf{K_{nl1}}$, $\mathbf{K_{nl2}}$	Nonlinear stiffness matrices	$\mathrm{N\,m^{-3}}$
$\mathbf{M}$	Mass matrix	kg
$\mathbf{P}_{LHC}$	Matrix of parameter scenarios generated by Latin hypercube sampling	
$\mathbf{R}_i$	Rotation matrix	
$\mathcal{R}_t$	Temporal reduced-order basis	
$\mathbf{U}(z,t)$	Solution matrix along the $x-$axis	m
U	Solution snapshot matrix	
$\mathbf{U}_i$	Solution matrix corresponding to scenario i	
$\mathcal{U}_i$	Reduced-order portrayal of $\mathbf{U}_i$	
$\mathbf{V}(z,t)$	Solution matrix along the $y-$axis	m
$\partial/\partial t$	Partial derivative with respect to time	

$\partial/\partial x$	Partial derivative with respect to space
$\{\dot{\square}\}$	Time derivative
$\{\ddot{\square}\}$	Second order time derivative

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Hydro power is an environmentally friendly, cost-effective, and renewable energy source that transforms the kinetic energy of running water into electrical power by using turbines and generators (EERE, 2021). Hydro-Québec is acknowledged as a prominent provider of clean energy. It generates, transmits, and distributes electricity throughout the province of Québec in Canada (Hydro-Québec, 2021a).

Almost all the electricity distributed by Hydro-Québec to its customers is generated from renewable resources, with hydropower representing 98.254% of this total. Hydro-Québec has a fleet of 63 hydro power generation units and 28 reservoirs (Hydro-Québec, 2021b), the Francis turbine being the most widely used turbine (Hydro-Québec, 2021c). Given the need to reduce greenhouse gas emissions, the significance of this renewable energy source for the Canadian economy cannot be emphasized enough.

The Francis turbine is classified as a mixed flow turbine since it incorporates elements from both impulse and reaction turbines. This type of turbine operates by allowing water to enter the impeller blades in a radial direction and exit in an axial direction. The rotation of the blades is achieved through a combination of the reaction and impulse forces generated by the flowing water, resulting in a more efficient electricity production process. It is worth mentioning that the Francis turbine is commonly employed in hydroelectric plants of large or medium scale, and its rotation axis is oriented vertically, as described by Fasi Ur Rahman (2021).

Traditionally, hydroelectric generating units (or simply hydroelectric units) were typically over designed, well maintained by competent personnel, and often operated continuously under rated load conditions for many years. Therefore, there were few maintenance problems, outages could be scheduled at regular intervals, and there was a little need for condition monitoring of these units (ISO, 2020).

Today, there are stricter demands for operating regimes, availability, and reliability. Power producers aim to maximize profits by minimizing service interruptions for consumers. Different types of hydropower units, such as those that use synchronous compensation, load following, and peak demand, often require frequent starting and stopping, sometimes in unstable areas, and operate at partial loads. Furthermore, new units are specifically designed for particular applications and are less robust, while older units are typically refurbished to prolong their lifespan or enhance their power output (ISO, 2020).

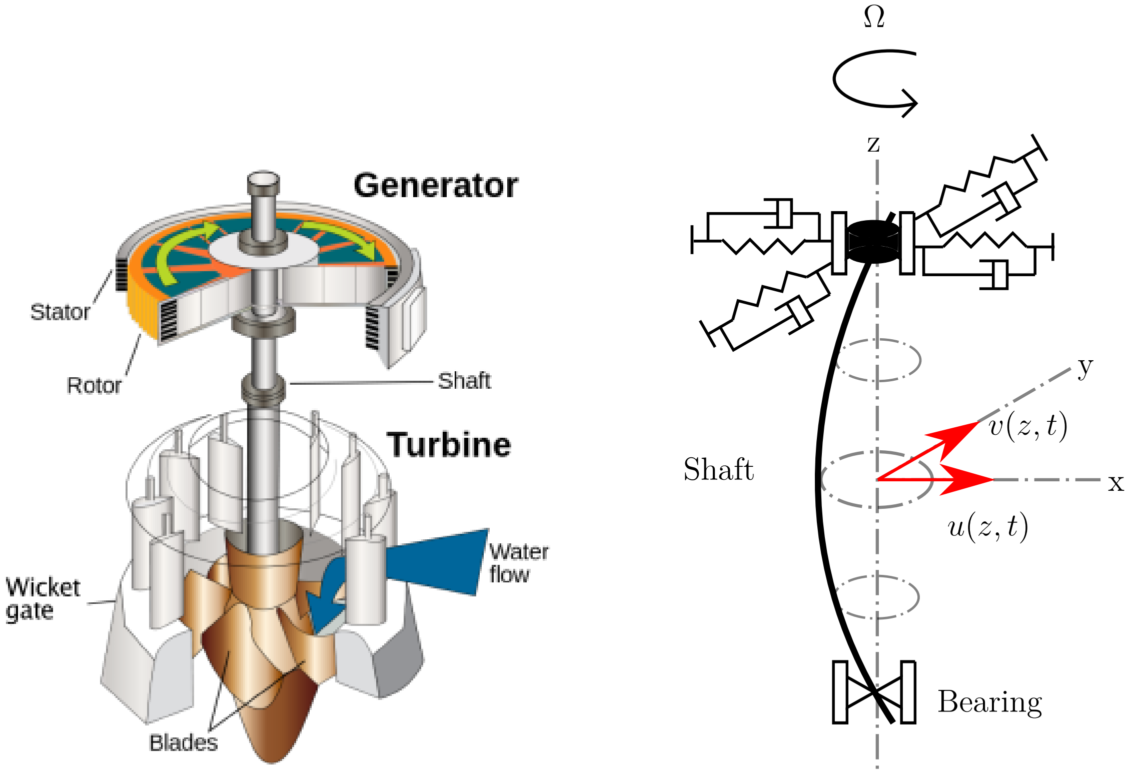
This implies that machines experience increased levels of stress, which can result in early or unforeseen failure of components and even the emergence of new failure modes. Simultaneously, there is a growing trend of reducing maintenance personnel and specialists responsible for machine monitoring. Therefore, there is a greater requirement for an efficient strategy for monitoring the condition of the machines, rather than solely relying on a protection system. Moreover, it is desirable for the condition monitoring solution for these machines to encompass more than just vibration monitoring (ISO, 2020).

To address this challenge, hydroelectric power producers have developed expertise in numerical simulation of the dynamics of the different components of their equipment. They are also collecting an enormous amount of data from their operating equipment with a variety of sensors. Historically, this data on real equipment is only compared to simulations in rare and punctual validations. The recent advance of artificial intelligence is a technological push to implement better integration of real equipment data and numerical simulations. This is where the motivation for the development of digital twins grew.

Glaessgen and Stargel (2012) first defined a Digital Twin *"as an integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin."*

It is not advisable, however, to create a digital twin for a full-scale Francis turbine since the concept of digital twins is still fairly new, and the Francis turbine itself is a complex multi-physics system that encompasses multiple physical phenomena and several undetermined parameters. Consequently, we selected one subsystem of the turbine, specifically the internal vertical shaft linked to bearings at its ends, as depicted in Figure 1.1a. The difficulty with this subsystem lies in determining the bearing forces affecting the shaft, as these forces are typically non-linear and difficult to quantify.

To further simplify and idealize this subsystem, it can be represented by a vertical flexible shaft linked to two bearings at its ends, which can either be considered fixed or flexible. The unknown forces exerted by those bearings are described as functions of displacements, velocities, and other parameters, involving some unknown coefficients that need to be determined. Figure 1.1b illustrates a schematic of this idealized model. Constructing this model is feasible using accessible and cost-effective equipment in an academic lab, allowing it to be tested as a proof of concept instead of the genuine hydroelectric turbine-alternator shaft line. We call this experimental setup the Vertical Axis Rotating Machine (VARM).



(a) Internal shaft of a Francis Turbine (Water Turbine, 2025)

(b) Vertical Axis Rotating Machine with unknown nonlinear bearing forces acting at its extremities.

Figure 1.1 Idealization of the Francis turbine internal shaft connected to bearings at its extremities.

The objective of this thesis is to create a digital twin for a VARM that exhibits dynamical properties comparable to a hydroelectric turbine-alternator shaft line, as shown by Adams (2010). The two machines exhibit similar dynamic traits, including lateral vibrations resulting from both whirling and rotational unbalance. Each machine contains a vertical flexible rotating shaft that is supported by bearings at the ends, which exert non-linear forces on the shaft during motion. In addition, they can be assessed and monitored using similar parameters and similar loading conditions. Consequently, the development of a digital twin for a VARM serves as an academic demonstrator in the process of creating a digital twin for the internal shaft of an industrial hydroelectric unit, thereby setting the foundation for subsequent research that could culminate in a comprehensive digital twin of the entire turbine.

1.2 Project Impact

A digital twin of a hydroelectric turbine has many technical and economic benefits. Below are a few examples summarized by Evangeline and Anandhakumar (2020):

1. It allows early problem or fault detection, which is essential to avoid long unnecessary down-times of the turbines. The longer the machines remain faulty or defective, the more costs the company accumulates.
2. It helps predict highly accurate results for a variety of operations. It saves both time and money needed to perform physical experiments. Furthermore, it enables the design and manufacturing of better products as well as the improvement of efficiency and the extension of product life cycles.
3. As the new operation schemes for hydroelectric turbines involve many starts and stops and partial load operations, a digital twin will be helpful in optimizing this procedure to minimize wear and provide a quantitative measure of performance efficiency.

Although digital twins are relatively new, they will be associated with many engineering products within a few years, such as electric cars, airplanes, jet engines, and manufacturing machines. The availability of cheap sensors and the omnipresence of Artificial Intelligence (AI) will make that very tempting and attainable.

1.2.1 Dissertation Structure

This dissertation contains seven chapters. Chapter 1 introduces the background of the project, its impact, and the general structure of the dissertation. Chapter 2 provides a review of the literature focusing on the key theoretical principles that are the foundation for the topics discussed in this study. Chapter 3 articulates the project's research goals and identifies the knowledge gaps addressed. The primary findings of the research are detailed in two articles, which have been published or submitted to peer-reviewed scientific journals, and can be found in Chapters 4 and 5. The first article introduces the Harmonic-Modal Hybrid Frequency method, integrated with the Discrete Empirical Interpolation Method (DEIM) and sPGD, to develop a parameterized solution to a non-linear dynamic problem. In the second article, this method is applied to model a VARM with unknown parameters. The Levenberg-Marquardt (LM) optimization technique is then used to estimate and validate these parameters using data collected from an experimental rig of the VARM, creating its digital twin. Finally, Chapter 6 provides a comprehensive discussion of the findings and highlights the limitations of the proposed methods, and Chapter 7 consolidates the conclusions of the project and offers recommendations for future research.

CHAPTER 2 LITERATURE REVIEW

The literature review delves into the foundational concepts and theories essential for comprehending the methodology and findings of this thesis. Evaluating this theoretical groundwork and examining prior studies aided in establishing the thesis objectives and identifying the optimal approach to accomplish them. Initially, digital twins are introduced alongside the challenges encountered in their design. Subsequently, the significance of Model Order Reduction (MOR) techniques in constructing digital twins is highlighted. This thesis primarily concentrates on projection-based MOR methods, which are discussed in greater detail, especially Proper Generalized Decomposition (PGD). Following that, a discussion on fundamental principles of rotor dynamics is provided, specifically targeting vertical axis rotating machinery and the forces in their bearings. Concluding the section, data assimilation methods for parameter identification are presented.

2.1 Digital Twins

2.1.1 The Concept of Digital Twins

The digital twin idea was introduced informally in 2002 by Grieves (2015), during a presentation about product life-cycle management. NASA was interested in developing the context of digital twins for its space assets. Therefore, exactly ten years after Grieves' presentation, NASA researchers presented seminal work about digital twins where Glaessgen and Stargel (2012) defined the concept as *"an integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin"*.

There are various definitions of a digital twin in the literature. In relation to this project, our focus lies on the definition suggested by Chinesta et al. (2019) where they introduce the term "Virtual Twin", which denotes a "static" mathematical model representing the behavior of an engineering system. Here, "static" indicates the creation of a model grounded in the system's physics through its parametric Partial Differential Equations (PDEs). While these models are rich and reliable, they are limited by the fact that they cannot account for changes in response caused by real-time data.

On the other hand, Chinesta et al. (2019) use the term "Digital Twin" to refer to a "dynamic" model that treats the system as a black box in the sense that it relates inputs to outputs, thus it can perform real-time predictions and calculations as data is fed into the system in real-time. An important drawback is the learning stage that requires significant computational effort and time in real time.

A hybrid twin, as Chinesta et al. (2019) call it, is a solution that combines both of the aforementioned concepts. However, the difficulty would be solving physics-based models in real-time. It is trivial to overcome this problem by using more powerful computers, but this is not always possible due to technical and financial constraints. In this case, a new modeling and simulation procedure can be employed, namely MOR in which the virtual twin operates offline during the design phase, while the digital twin performs online functions using real-time collected data.

In this project, we propose a digital twin akin to Chinesta's Hybrid Twin, which involves a physics-based model solved offline using MOR techniques, in addition to a data assimilation model predicting the model's unknown parameters. In this dissertation, we will employ the term "Digital Twin" to refer to Chinesta's "Hybrid Twin" combining both physics insights from the PDEs and data.

2.1.2 Challenges

Developing a digital twin is a use-case-specific application. As a result, reading the literature rarely provides a systematic or repeatable design procedure, as explained by Bickford et al. (2020) and Fuller et al. (2020).

Barricelli et al. (2019) describe digital twins as entities that can be conceptually understood by considering two distinct life cycles, spanning from initial design to eventual disposal. First, there is the case where the physical system being twinned does not yet exist; therefore, the physical and digital twins should be designed at the same time. In contrast, within the second type of life cycle, the physical counterpart is present during the operation and maintenance phase, yet it is not accompanied by its digital counterpart. Here, the design process emphasizes linking the physical system with its digital twin. However, according to Bickford et al. (2020) and Fuller et al. (2020), the majority of digital twins covered in the literature adhere to the second type of life cycle. As a result, the design process is often adapted to the existing physical system, resulting in a lack of a standard process for defining the digital twin requirements.

Due to the previous challenges, the unclear path of development, as well as the steep learning curve in the early stages of implementation, stakeholders are less interested in investing in the development of digital twins. There is no guarantee that a proposed approach will give reliable results, which makes development risky and expensive. Our project's importance becomes apparent at this point.

By developing a digital twin for the VARM and using a simple experimental rig to test it, we will be able to provide clear guidelines that will make the process of developing a digital twin for an industrial hydroelectric unit cheaper and more reliable, and that is the main contribution of this thesis.

2.2 Model Order Reduction Techniques

2.2.1 Motivation for MOR Techniques

Complex systems require more realistic simulations. Schilders et al. (2008) explain that a realistic simulation is one whose errors are as small as possible. For that, finer meshes are needed, which complicates numerical manipulations. Therefore, in the case of a dynamic system whose behavior is driven by many equations, affected by a large number of parameters, and characterized by a fine mesh with a large number of degrees of freedom, reducing the model will help us perform numerical simulations without consuming too much computation power and memory. Added to the complexity is the fact that the calculations need to be performed in real-time.

High-dimensional models suffer from the curse of dimensionality defined by Bellman et al. (1957). This is common in numerical analysis. As dimensionality increases, the available data becomes sparse; therefore, the number of data points needed to generate reliable results grows exponentially with dimensionality, making the calculation more complex.

If we think of dimensions in engineering applications as only spatial dimensions, then the curse of dimensionality will not make sense. Our later explanations will show how certain MOR methods, treat system parameters as additional dimensions, and the solution is found for a range of these parameters, creating what Chinesta et al. (2010a) define as a parametrized model. A parametrized model is, therefore, a mathematical framework governed by various parameters influencing a system's behavior, attributes, or states. By modifying these parameters, one can evaluate how the model reacts under different configurations or situations. The strength of parametrized models lies in their flexibility and adaptability, enabling the simulation of diverse conditions. These models can prove valuable for optimizing performance, forecasting future behavior, or gaining insight into intricate systems by breaking them down into more comprehensible parts. Therefore, when we speak of system dimensions in this thesis, we refer to the spatial dimensions as well as other parameters such as geometric parameters, material parameters, initial conditions, and boundary conditions.

As an example, the computing complexity of a model with d dimensions and $\mathcal{M}$ nodes can be quantified by $\mathcal{M}^d$. Assuming a domain of 30 parameters and dimensions (quite a simple system), discretized with a mesh of 1000 nodes (quite a coarse mesh), then the computational complexity is 10^{90} which is simply beyond the capabilities of the most powerful computers available today as illustrated by Chinesta et al. (2011, 2014) and Ladevèze and Chamoin (2011). This is where the advantage of MOR Techniques comes into play. Aguilera (2017) proved that using these techniques is effective in challenging scenarios, such as multiple-parameter and multidimensional systems, where a single simulation may take days. This applies to the concept behind this thesis.

2.2.2 State of the Art

A simple definition of MOR techniques is given by Aguilera (2017): *"any numerical approach that aims to replace a high-fidelity simulation by another one with a much lower computational cost"*. High-fidelity methods are traditional methods that have been used repeatedly, providing accurate and reliable results, such as Finite Elements, Finite Volumes, or Spectral Methods. Even though these methods are reliable, they are computationally expensive as systems become more complex.

To accurately capture the essential characteristics of a system, Schilders et al. (2008) and Chinesta et al. (2019) show that an MOR needs to capture the most fundamental properties of the original model in smaller approximations. Then, once the reduction process is stopped, all the necessary properties are captured with good precision. The entire process needs to be automatic. In other words, MOR techniques do not simplify the model; they approximate the solution in a way that simplifies the calculation procedure without compromising accuracy, taking into account real-time constraints.

2.2.3 Projection-Based MOR Techniques

MOR techniques are widely available in the literature, yet this thesis focuses on a particular class of techniques, those of projection-based techniques, where we look for the solution to the model's equation in a lower-dimensional subspace that is specific to the problem at hand, as shown by Lauzeral et al. (2019) who divide projection-based techniques into two categories: *A posteriori*, such as Proper Orthogonal Decomposition (POD), and *A priori* such as PGD.

A popular method of MOR is the POD method, since it is simple to implement and can be applied to non-linear systems without much difficulty. The mathematical implementation of POD and more details about it can be found in the work of Lauzeral et al. (2019), Chinesta et al. (2019), Schilders et al. (2008), and Yan et al. (2020). This concept involves approximating a process based on its most energetic modes, expressed in what we call the snapshot matrix. Therefore, a number of samples need to be taken from the parametric space of the problem whose output is already known experimentally or calculated in advance. The output of these samples is then used to extract the most important modes of the model offline, resulting in a snapshot matrix that is considered a reduced basis. Once this reduced basis is determined, the system's equations are projected onto it and solved. After this, the reduced solution can be projected back onto the complete basis.

POD, however, has limitations because the snapshot matrix is essential in the solution, and it may not always be available or too expensive to compute. Also, it is not a recommended method for systems involving a large number of parameters, as the computations will be heavy due to the curse of dimensionality. An alternative MOR method, PGD, overcomes these drawbacks since it is a completely *a priori* method that does not require prior knowledge of the solution, *i.e.* the snapshot matrix. This is a more generalized version of the POD method. PGD performs an *a priori* computationally intensive offline calculation stage, thus enabling fast online calculations later as explained by Lauzeral et al. (2019).

Chinesta et al. (2019) showed that during the offline stage, a parametric solution with all possible scenarios is calculated using all the computational power and time necessary. Then, this solution would be made available online, where users could extract the solution quickly and easily for any scenario. As a result of this technique, efficient simulation, optimization, inverse analysis, uncertainty propagation, and simulation-based control can also be performed under real-time constraints.

In this context, it is necessary to clarify why this offline-online approach is preferable to traditional simulations that are done in a single step. On the one hand, the offline computation is done effectively using a reduced model which provides accurate results at a much lower computational cost. On the other hand, this allows the online stage to be performed on mobile or light computer platforms, as well as on smartphones and tablets. This is a big advantage for some applications. The offline stage of PGD entails a higher workload than the online stage, but the offline stage is done once and for all to produce a handbook of the solution, also known as Computational *Vademecum* as defined by Aguilera (2017) and Chinesta et al. (2013).

The following section will elaborate on PGD since it is the focus of this project.

2.3 Proper Generalized Decomposition

2.3.1 Origin of PGD

Chinesta et al. (2013) define PGD as "*an efficient multidimensional solver that allows introducing model parameters (boundary conditions, initial conditions, geometrical parameters, material and process parameters ...) as extra-coordinates. Then by solving only once and off-line the resulting multidimensional model we have access to the parametric solution that can be viewed as a sort of meta-model or virtual chart that can be then used online*".

The basic idea is to seek a variable-separated representation of an unknown field as explained by Ladevèze and Chamoin (2011) and Cueto et al. (2016). The principle of variable-separation itself is not novel, it is based on the Fourier method for solving PDEs; however, the novelty lies in the fact that PGD is able to construct a reduced basis without any *a priori* knowledge of the solution, as well as eliminating the need to use the snapshot matrix. Also, it does not require any orthogonality properties, which is an advantage over POD as shown by Dumon et al. (2010) and Ladevèze and Chamoin (2011).

2.3.2 PGD state of the art

Chinesta et al. (2010b) and Cueto et al. (2016) explain that PGD seeks a separated representation of an unknown field known as the finite sum decomposition. Assuming that the solution of a general Partial Differential Equation (PDE) is a field that is dependent on space, time, and a number of m parameters, then it can be represented as:

$$u_{PGD}(\boldsymbol{x}, t, p_1, p_2, \ldots, p_m) = \sum_{i=1}^{N} F^i_{\boldsymbol{x}}(\boldsymbol{x}) \cdot F^i_t(t) \cdot F^i_1(p_1) \cdot F^i_2(p_2) \ldots F^i_m(p_m), \tag{2.1}$$

where, $\boldsymbol{x}$ represents the $3D$ space, t represents the time, and p_i can be any parameter such as material parameters, geometrical parameters, boundary conditions, or initial conditions.

In equation (2.1), the field u_{PGD} is a function of d variables, where, in the most general case, d includes 3 spatial dimension variables, one variable corresponding to time, and m variables related to additional parameters. Thus, if $\mathcal{M}$ nodes are used to discretize each variable, then the total number of unknowns will be $N \cdot d \cdot \mathcal{M}$ instead of $\mathcal{M}^d$ using traditional discretization methods. N is the number of summation terms forming the solution u_{PGD}. Each summation term is called an enrichment term as it enriches the solution and gives better accuracy as demonstrated by Dumon et al. (2010), Cueto et al. (2016), and Chinesta et al. (2013).

Ideally, an infinite number of enrichment terms gives an exact result; however, Ammar (2009) showed that a generally small number of summation terms N (a few dozen) is enough to give reliable results, and convergence is usually guaranteed using a greedy algorithm, as we will explain next.

It is necessary to solve a strictly nonlinear problem regardless of the nature of the original problem in order to determine these functions F_j^i. Fortunately, Ammar et al. (2006, 2007), Cueto et al. (2016), and Chinesta et al. (2013) proved that simple techniques are effective in addressing this issue.

Ammar et al. (2006, 2007) developed a set of numerical techniques for solving multidimensional partial differential equations in a space with dimensions $d \gg 1$. These solvers are based on the PGD separated representation, where the solution is constructed as the sum of the product of d functions, each of which is defined in one dimension.

An efficient greedy algorithm is proposed, in which one sum is computed per iteration, and simple linearization strategies are used for calculations at each iteration, such as the fixed-point alternating direction algorithm.

This is a greedy algorithm in the sense that the summation terms that contribute the most to the solution are calculated first, and then as more summation terms are calculated in each term, their contribution to the solution decreases with each iteration. Since the solution obtained in all the previous iterations up to iteration i is used in calculating the summation term at iteration $i+1$, this causes the residual of the PDE to be reduced with each addition of a term. Therefore, fewer terms need to be added.

As per the works of Ammar et al. (2006, 2007), Chinesta et al. (2014), and Cueto et al. (2016), the greedy algorithm strategy consists of three steps:

1. *Discretization:* where the PDE is discretized with respect to all parameters and the Galerkin variational formulation is found. In this step, the solution to the PDE presented in equation (2.1) is inserted into the PDE during the derivation of the Galerkin formulation, setting up the equations for resolution via the greedy algorithm, which calculates the enrichment steps discussed next. In the next step, the PDE is integrated by parts to find the problem *vademecum*.
2. *Enrichment:* after calculating i summation terms, a new summation term is calculated using the fixed-point alternating direction strategy and added to the solution to improve its accuracy. This step takes up most of the calculating time of the solution, but it is noted that these algorithms converge relatively quickly, even with high-dimensional problems.
3. *Convergence:* A decision is made about whether to end the enrichment process or not. If

the newly calculated enrichment term does not add any new information to the already computed solution, the enrichment process ends. There are several stopping criteria where a calculated value must fall below a certain tolerance. The relative weight is an example of this. It calculates the ratio of the newly calculated term to the complete solution calculated before it. Alternatively, we could calculate the ratio of the newly calculated term to the first term. Lastly, we can use a traditional approach, which involves plugging the PGD solution into the PDE, and then calculating the residual, which should be as close to zero as possible.

The advantage of this method is that it does not require knowing the solution in advance, collecting data, or creating snapshots. Even though it is an iterative procedure, convergence usually requires only a few iterations. Additionally, for each iteration, a single-dimensional problem is solved as shown by Cueto et al. (2016).

2.3.3 Previous Work on Implementing PGD to Solve PDEs

PGD has been used to solve a variety of different systems. The method is considered relatively new (less than 20 years old); therefore, there is still a lot of room for exploration.

In the works of Ghnatios et al. (2020), Zou et al. (2018), Zlotnik et al. (2014), Vidal et al. (2012, 2013), Leygue and Verron (2010), and Ammar et al. (2006) examples of static linear systems are investigated. The PGD formulation used included space and one or more parameters. On the other hand, the research of Ammar et al. (2007); Ammar (2009), Dumon et al. (2010), Pruliere et al. (2010), Chinesta et al. (2011), Ladevèze et al. (2010), González et al. (2014), and Boucinha et al. (2013) considered linear dynamic systems, included time in their PGD formulations, and solved initial value problems. Linear systems have also been solved by Montagud et al. (2020), Meyrand et al. (2019), and Haris et al. (2018) in the frequency domain using frequency as an extra parameter in the PGD formulation. Nonlinear static systems were investigated by Qin et al. (2016), Halabi et al. (2016), Henneron et al. (2012), and Niroomandi et al. (2013).

Despite the fact that those aforementioned articles only considered linear static and dynamic systems, in addition to nonlinear static systems, they actually inspired us to extend the principle of those methods to solve a nonlinear dynamic system in the frequency domain.

In terms of dynamic non-linear systems, Néron et al. (2015) and Cremonesi et al. (2013) employed a combination of PGD and the LArge Time INcrement (LATIN)[1] method to study

[1] The LATIN method was introduced by Ladevèze (1999) for solving nonlinear problems in structural mechanics. Using this method, global linear variables are separated from nonlinear local variables, and a two-step iterative scheme is then used to build the solution.

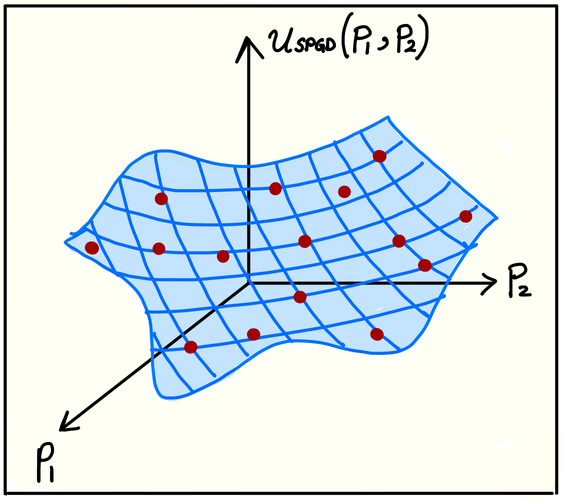


Figure 2.1 sPGD graphical representation of a function of two parameters $u_{sPGD}(p_1, p_2)$. The red circles represent the sparse data set. The example demonstrates the case with 2 parameters for better visibility, but it can be expanded to include multiple parameters.

a quasi-static isothermal evolution of a structure and the visco-elastic behavior of a 2D structure, respectively. Also, Aguado et al. (2013) solved a transient heat equation with a quadratic non-linearity using PGD. However, they interpolated the nonlinear term using a reduced basis. Grolet and Thouverez (2012) on the other hand, solved a beam problem featuring geometric nonlinearities using PGD. And finally, Germoso et al. (2020) studied the non-linear behavior of soil during a seismic event using PGD, while Quaranta et al. (2019) investigated the non-linear behavior of a 1D cantilevered beam subjected to a horizontal force acting at its tip. Both authors used a frequency approach.

2.3.4 Sparse Proper Generalized Decomposition (sPGD)

sPGD was introduced by Ibáñez et al. (2018) as a sparse reduced-order regression technique based on the PGD framework. In the way it efficiently separates variables, this method is similar to PGD, offering quick convergence for offline computation, which subsequently produces a surrogate Reduced-Order Model (ROM) for rapid online system response calculations. However, unlike PDE-dependent PGD, sPGD is employed in an *a posteriori*, non-intrusive approach, which involves carrying out an offline calculation and compressing the results to create a model that can interpolate over a sparse data set that does not need to be large, typical of machine learning methods. Another notable difference from traditional PGD is the use of interpolation functions rather than finite element shape functions. Nevertheless, the greedy algorithm is employed comparably to approximate the $1D$ separate variable functions.

Figure 2.1 illustrates a visual depiction of an sPGD surrogate model for a particular quantity u that depends on two parameters p_1 and p_2. The red circles signify the sparse, observed values of u based on various random combinations of p_1 and p_2. The capability of sPGD is demonstrated through the calculation of the blue surface, which can estimate the value of u for any combination of the parameters p_1 and p_2. The data indicated by the red dots may come from high-fidelity calculations, reduced-order models, or empirical data.

Further implementation details of this method are provided in Chapters 4 and 5, and in Appendix E.

2.4 Rotating Machines

2.4.1 Basics of Rotor Dynamics

Whenever we study the dynamics of a rotating machine, it is crucial to study the rotor at its center. Essentially, Tiwarti (2006) defines a rotor as a body suspended by a set of cylindrical hinges or bearings that allow it to rotate freely about an axis that is fixed in space. The rotor can be modeled as a single continuous beam or a series of beam elements and rotating disks as explained by Friswell et al. (2010). Stators are the fixed parts of a rotating machine. Their interactions with the rotors must be studied to obtain detailed information about a rotating machine's behavior.

Figure 2.2 shows two major types of dynamics, explained by Ishida and Yamamoto (2012), that we are interested in studying in this thesis: a) shaft line deformation and b) rigid body motion with flexible supports. In the former case, the shaft deforms with respect to its central axis during rotation, and in the latter, the shaft experiences radial motion within the bearing. These two types of motions can happen at the same time in some cases.

According to Tiwarti (2006) and Friswell et al. (2010), flexible rotors experience axial, tor-

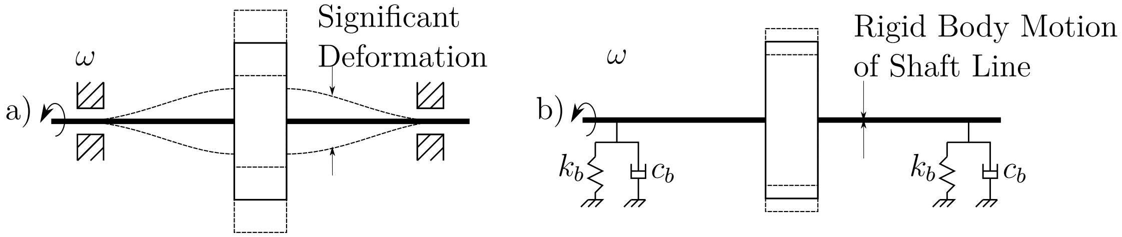


Figure 2.2 Major types of dynamics of rotating machines. (a) Shaft Line Deformation, (b) Rigid Body Motion of the Shaft with Flexible Supports

sional, and bending (lateral) vibrations. These vibrations cause the rotor to orbit around its axis of rotation and can occur independently or be coupled in complex systems. Lateral vibrations, being the largest in high-speed machinery, are crucial to control to prevent bearing wear and failure. Excessive lateral vibrations can lead to contact between rotating and stationary parts, causing potential damage.

One cause of lateral vibrations is the whirling phenomenon. At certain speeds, rotating shafts tend to bow out and whirl in a complex manner. Marghitu et al. (2001) define whirling as the rotation of the plane made by the bent shaft and the line passing through the center of the bearings. This induces a centrifugal force causing the shaft to vibrate. To maintain the shaft's stability, the elastic restoring force needs to overcome the centrifugal force effect; otherwise, any small perturbation to the shaft can cause the radial deflection to be large and the system to become unstable. Another cause of lateral vibrations is rotating unbalance. According to Ishida and Yamamoto (2012), the term "unbalance" refers to the nonuniform distribution of the mass of a rotor about its axis of rotation. This causes the principal axis of inertia to deviate from the axis of rotation.

Consequently, the unbalanced mass generates a centrifugal force akin to what was previously mentioned, which depends on both the rotation speed and the unbalance's eccentricity. Notably, the shaft's eccentricity is a constant factor, independent of whether the shaft is disturbed. The system's stability is contingent on the rotation speed. If this speed matches the system's natural frequency, resonance occurs, potentially leading to the system's failure due to excessive vibration amplitudes. This specific speed is referred to as the "critical speed". For further information, readers are encouraged to consult the works of Friswell et al. (2010), Ishida and Yamamoto (2012), and Adams (2010).

2.4.2 Horizontal vs. Vertical Axis Rotating Machines

Figure 2.3 shows two types of rotors distinguished by Ishida and Yamamoto (2012): vertical shaft systems and horizontal shaft systems. Rotating machinery with a horizontal rotor axis is more addressed in the literature. The reason behind this is that when the shaft is horizontal, the dead weight of the rotor maintains its equilibrium position where the bottom surface of the rotor is in contact with the bearing while the clearance is above the top surface of the shaft, as shown in Figure 2.4a. The bearing reaction forces can then be calculated based on the static radial loads caused by the dead weight of the rotors, as shown in the works of Nasselqvist et al. (2014) and Chen and Gunter (2007).

There are many commercial software programs that use numerical algorithms such as Navier-Stokes or Reynolds equations to calculate these reactions for the horizontal shaft at a certain

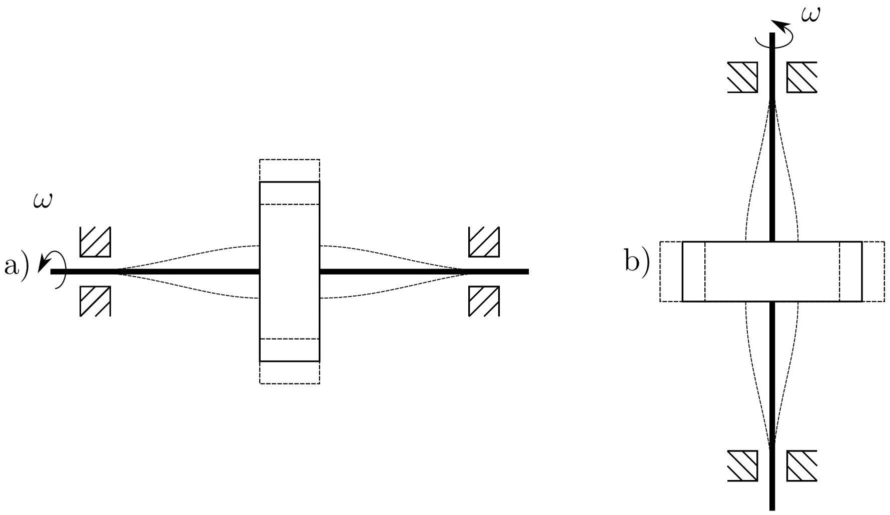


Figure 2.3 (a) Horizontal Axis Rotating Machine, b) Vertical Axis Rotating Machine [2]

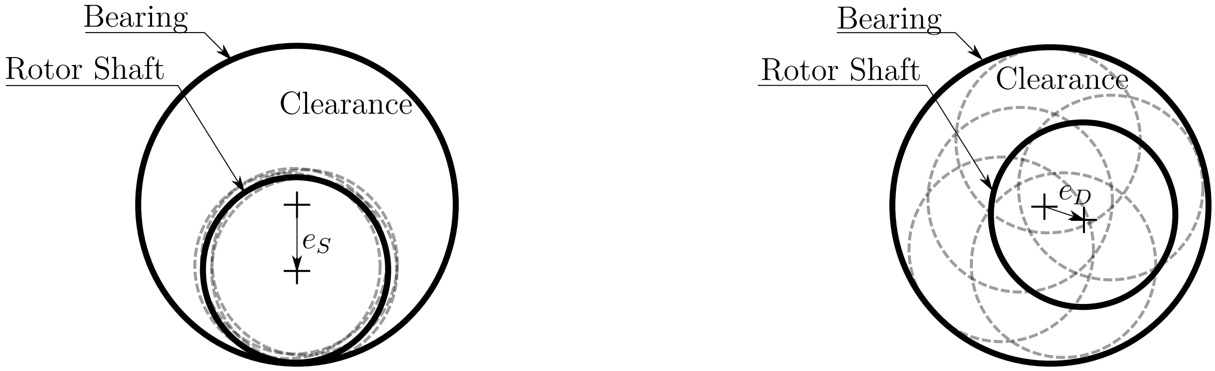
static eccentricity as shown by Nasselqvist et al. (2014), but they are usually not derived to perform calculations for vertical shafts. Moreover, in horizontal shaft systems, when the rotational speed is relatively low, the dynamic behavior of the bearing can be approximated as linear, as indicated by Shi et al. (2012).

In the case of a vertical rotor axis, the shaft is free to move within the clearance, hitting the inner surface of the bearing randomly as shown in Figure 2.4b. Haslam et al. (2020) showed that the clearance between the shaft and the bearing is the main cause of instability in the system, due to sudden changes in the bearing stiffness when the clearance gap is closed.

There are a number of methods described in the literature for calculating bearing reaction forces in vertical machines for various types of bearings. These methods will be discussed in the following sections.

Considering these categories of rotating machinery, it is essential to clarify that this project focuses on the lateral vibrations of a flexible, vertically-oriented rotating machine, where the bearings themselves are also flexible. In our VARM, we significantly reduced the clearance between the shaft and the bearing, representing the bearing's non-linearity using an alternative method detailed in Chapter 5. The bearings will be modeled as non-linear springs and dampers within the plane perpendicular to the vertical axis. We will examine two vibration phenomena: whirling and rotating unbalance, although in both cases, the rotational speed is below the critical speed.

[2] The schematics show rigid supports, however both configurations can be implemented with flexible supports



(a) Horizontal Shaft. e_S is the Static Eccentricity

(b) Vertical Shaft. e_D is the Dynamic Eccentricity

Figure 2.4 Section View of a Rotor Shaft inside a Bearing showing Clearance and Eccentricity - Clearance is exaggerated for demonstration purposes.

2.4.3 Modeling Bearing Reaction Forces

The rotor is connected to the supporting structure by means of bearings, which come in several types. In small machines with a light load, the bearing may be a simple bush in which the rotor rotates. When bearing load increases, rotors are mounted on bearings with rolling elements or ball bearings. Although these increase both load capacity and rigidity, they also play a crucial role in the machine's dynamics owing to their internal moving components, as discussed by Friswell et al. (2010).

Friswell et al. (2010) also discuss journal (hydrodynamic) bearings which are commonly used in large, heavy rotating machinery. There is a thin layer of oil in a small gap between the rotating and stationary parts. A hydrodynamic pressure distribution within the oil film is caused by the rotor, and this assists in supporting the unbalanced forces of the rotor. Other types of bearings can be found in the literature, but they are out of the scope of this project. The majority of hydropower plants use tilted pad journal bearings that are lubricated with oil, as discussed by Nasselqvist (2009).

Significant research efforts have focused on formulating theories about the influence of interconnections, like bearings, generators, or dampers, on rotor dynamics. Bearings are sometimes considered as localized rotor-stator interactions characterized by their unique damping and stiffness properties, which produce reaction forces. Nonetheless, these properties can be influenced by various factors, making them challenging to accurately quantify. The coefficients of bearings are crucial for modeling the forces that bearings exert on the shafts, which significantly affect the dynamic behavior of the rotating machinery.

Overall, Chen and Gunter (2007) indicate that the forces and moments associated with or interacting between bearings and shafts in rotating machinery can be represented as functions of the degrees of freedom. These include the displacements and bending of the shaft within the plane perpendicular to its axis of rotation, along with their time derivatives, specifically the velocities. Additionally, these forces and moments depend on operating parameters that vary with conditions such as speed, load, and temperature, and geometric parameters like clearance, length, and preload, which are constant once the components are manufactured. Notably, most of these forces are inherently non-linear, necessitating a non-linear analysis to accurately predict their dynamic behavior.

To efficiently calculate the stiffness and damping coefficients of bearings, complex numerical non-linear calculations of bearing coefficients are needed, because they are:

1. Strictly non-linear at high eccentricities and high rotating speeds (Nasselqvist, 2009; Chen and Gunter, 2007).
2. A function of preload at dynamic loading (Wygant et al., 2004).
3. A function of the load angle.
4. Cross-coupled, which is a fact that is usually ignored by researchers to simplify bearing models.

Earlier studies concerning the modeling of bearing reactions are reviewed in the introduction of the article presented in Chapter 5.

To sum up, the most complicated part of the dynamic model of a vertical axis rotating machine is the bearing model, where bearing stiffness and damping are difficult to quantify. There are several approaches used in the literature, including model linearization methods, numerical methods to solve non-linear models, experimental methods, and commercial software.

2.5 Data-Driven Systems towards a Digital Twin

As mentioned previously, the purpose of this project is to produce a Digital Twin where a physics-based model is solved offline using PGD as an MOR technique, and a data-driven model helps estimate unknown parameters of the model.

2.5.1 Static vs. Dynamic Data-Driven Application Systems

We can categorize data-driven systems into two types: static and dynamic. Here, the terms "static" and "dynamic" do not describe the system’s mechanics. Instead, Static Data-Driven

Application System (SDDAS) represent models that rely on the parameters and boundary conditions of a given system. When data is used in this case it would be collected *a priori*, *i.e.* data is not constantly fed into the system so that it can be updated in real-time as explained by Chinesta et al. (2019) and Knight (2003). Dynamic Data-Driven Application System (DDDAS) on the other hand, was introduced by Blasch et al. (2018) and Kapteyn and Willcox (2020) in 1998 as a paradigm where computing and instrumentation aspects of a system are dynamically integrated in a feedback control loop. In DDDAS, data is continuously fed to the system to update its digital twin in real time and improve its performance as shown by Knight (2003). This allows the digital twin to perform various tasks such as inference, assimilation, prediction, control, and planning. However DDDAS is out of the scope of this work.

In this project, data will be used for offline parameter identification to estimate unknown bearing parameters. This will be explained in the next sections.

2.5.2 Offline Parameter Identification

Data science and machine learning have made it possible to extract patterns from vast multimodal data. These advancements have caused a renaissance in the analysis and understanding of complex data. However, deriving a physical model from data is still a very challenging problem, especially when the system is non-linear, as shown by Brunton et al. (2016).

When a mechanical system has a physical understanding that is incomplete, *i.e.* we know the differential equations that govern the behavior of the system, but some parameters in these equations are difficult to identify. Using the measured data, these unknown parts of the model can be estimated. The premise of grey-box modeling is described by Sohlberg and Jacobsen (2008) as follows: *"don't estimate what you already know but test your knowledge against experimental data"*. The introduction in Chapter 5 provides additional information on grey-box modeling.

This technique can be implemented for developing a digital twin by first constructing a parameterized model of the system, in terms of its unknown parameters. An optimization method is then employed to solve a problem aimed at determining these unknown parameters that minimize the discrepancy between simulation results and real-world data gathered from the system's physical counterpart.

A common optimization technique is the LM algorithm, which was developed in the early 1960s to solve non-linear least squares problems. It combines the Gradient-Descent and the Gauss-Newton as explained by Gavin (2020). Detailed derivations of those two techniques in addition to their combination in the Levenberg-Marquardt algorithm can be found in Appendix F.

Fletcher (1971) explained that the Gradient-Descent method is known for its convergent nature, but it is slow, while the Gauss-Newton method is less reliable in terms of convergent behavior, but in general more rapid. The LM algorithm takes the Gradient-Descent approach when the parameter value is far from the optimum value, while the Gauss-Newton approach is used when the parameter value is close to the optimum value, thus combining the strengths of both approaches.

CHAPTER 3 SYNTHESIS OF THE LITERATURE AND RESEARCH OBJECTIVES

3.1 Analysis of the opportunities and gaps in the literature

In this section, we discuss how the literature review presented in Chapter 2 was used to establish the three main objectives of this thesis. It should be noted that this thesis is primarily an applied engineering project in which we do not aim to develop new algorithms or methodologies. Instead, we aim to combine and use previously developed techniques to create our digital twin, which is our main contribution.

3.1.1 Digital Twins

This project aims to create a digital twin of a VARM, which is an existing physical system. In the literature, we were unable to find a detailed recipe for building a digital twin from scratch. Instead, we demonstrated that designing a digital twin is a use-case-specific application. Thus, creating the digital twin for a VARM is a scientific and engineering contribution. Its importance lies in the fact that it lays the groundwork for designing a digital twin for a Francis turbine. This provides inspiration for systems with similar characteristics, thus eliminating the need to start the digital twin design process from scratch each time.

3.1.2 MOR Techniques and PGD

PGD is an interesting technique for MOR as it allows creating parametrized ROMs. To our knowledge, non-linear dynamic problems have not been sufficiently investigated when using PGD to solve PDEs. There is a notable gap in the literature concerning the application of PGD for addressing nonlinear dynamic problems within the frequency domain. Previous work has been limited to basic academic problems, lacking expansion into practical applications or integration within a digital twin framework. In the literature, the modal basis solution, the DEIM, and sPGD have been introduced independently. However, no previous attempt has been made to combine them in order to develop a rapid solver capable of generating data for the construction of surrogate models. These surrogate models play a crucial role in the development of digital twins.

3.1.3 Rotating Machines

Based on Chapter 2, to obtain a model of the VARM inspired by a Francis turbine, the VARM has to be a flexible vertical axis rotating machine, where the bearings are also flexible. We focus on lateral vibrations and flexible motion in the supports for the purposes of this project. This VARM model was chosen for an additional reason, that is, the coefficients of the nonlinear forces exerted by the bearings on the rotor shaft are difficult to calculate. Thus, we can use this fact to demonstrate the power of PGD and data-driven systems in handling such cases.

Most of the previous work on modeling reaction bearing forces has focused on horizontal axis machines. In the case of vertical axis machines, there is still room for exploration. It has been shown that these forces contain a nonlinear component, are hard to estimate analytically and experimentally, and in most cases, cross-coupling between these forces is ignored.

As a solution to these gaps, the bearing forces can be expressed as a function of displacements, with unknown bearing stiffness and damping parameters. It is possible to try different nonlinear combinations of these variables, then estimate the stiffness and damping parameters by minimizing the error between the PGD solution and the experimental data using optimization techniques.

3.1.4 Data-Driven Systems for Parameter Identification

In existing studies, it has been demonstrated that the LM optimization method is the most effective choice for integration with sPGD in parameter optimization, particularly in the frequency domain. This assertion is supported by the following arguments: Firstly, the sPGD surrogate model depicts the quantity of interest as a function of all unknown parameters within their designated ranges. Additionally, the sPGD algorithm independently calculates and maintains the Jacobian matrix, facilitating its integration into the LM algorithm, with this step being the most computationally demanding. As a result, numerical errors do not build up during the iterative process if the Jacobian is recalculated at each step. Furthermore, the sPGD library offers almost immediate responses for any parameter scenario, thus accelerating the convergence of the LM algorithm. Lastly, the LM method does not require time integration, an aspect removed from the approach when solving in the frequency domain.

3.2 Research Objectives

After reviewing possible opportunities in the literature that fit our area of interest and the scope of this thesis, we have outlined three main objectives for this research.

3.2.1 Objective 1: Development of a PGD framework to solve non-linear dynamic problems

To develop a ROM for a nonlinear dynamic system using PGD. Offline calculations will create the complete PGD *Vademecum*, which will allow online solution extraction for various scenarios efficiently. A proof of concept will be demonstrated using a simple academic problem in which a damped cantilevered beam is subjected to a harmonic force with nonlinear boundary conditions.

3.2.2 Objective 2: ROM of the VARM using PGD

To create a mathematical model of a VARM with two bearings at its extremities. Bearings exert a non-linear force on the shaft. Then a PGD *vademecum* will be used to reduce the order of the VARM model and to create a library of solutions for all possible combinations of unknown parameters. Unknown parameters related to bearing forces will be identified and prepared for estimation.

3.2.3 Objective 3: Data-Driven Parameter Estimation towards a complete Digital Twin

To use the data collected from the VARM experimental setup to create a digital twin using an SDDAS approach. Our main focus is on integrating these data through an optimization method designed to determine the values of unknown parameters. This approach facilitates the creation of a digital twin, which can then be employed to predict the system's response under various untested scenarios in an effective and reasonably precise way. This is useful for predicting various dynamic phenomena.

The main results of this research are presented in the form of accepted or submitted peer-reviewed journal articles in Chapter 4 which completes research objective 1 and Chapter 5 which completes research objectives 2 and 3.

CHAPTER 4 ARTICLE 1: HARMONIC-MODAL HYBRID FREQUENCY APPROACH FOR PARAMETERIZED NON-LINEAR DYNAMICS

Here, we present the article, *"Harmonic-modal hybrid frequency approach for parameterized non-linear dynamics"*, published in Computers and Structures (DOI: `https://doi.org/10.1016/j.compstruc.2024.107461`). In this paper, we achieve the first objective by introducing a novel approach to develop a surrogate model for a nonlinear system, which provides real-time predictions of a parameterized dynamic response through the use of the sPGD method. To address the sparse dataset employed in forming the sPGD model, we introduce the Harmonic Modal Hybrid Approach (HMH) as a global non-linear solver in the space-frequency domain. This method produces a low-rank solution representation by employing Modal Basis analysis, and DEIM is applied to accelerate convergence by constructing a reduced basis for the non-linear function(s) using either empirical data or computed results.

Our HMH solver is principally based on the solver developed at the Procédés et Ingénierie en Mécanique et Matériaux (PIMM) Laboratory, located at the École Nationale des Arts et Métiers in Paris, France by the team of Professor Francisco Chinesta. It was used to find the forced response simulation of soils exhibiting nonlinear behavior. The idea of combining it with DEIM and sPGD to create a parametrized surrogate model was suggested by Professor Francisco Chinesta, and implemented by the author of the thesis during an internship at ENSAM in October-December 2022. The author of the thesis derived the necessary equations and used the developed methodology to solve for a renowned non-linear system: a cantilevered beam with a non-linear spring fixed at its end, which is subject to two kinds of excitation, namely harmonic and base. Sebastian Rodriguez provided technical insight on the coding procedure and contributed the sPGD training algorithm featured in one of his earlier publications. The thesis author drafted the entire initial version of the paper, which was then reviewed by the co-authors.

The citation style was altered to fit the thesis format. A comprehensive bibliography is provided at the end of this document.

Date of Submission: 21 February 2024
Date of Publication: 1 September 2024

Authors

Sima Rishmawi[1,*], Sebastian Rodriguez[2], Francisco Chinesta[2], Frédérick P. Gosselin[1]

[1] Multi-scale Mechanics Laboratory (LM2), École Polytechnique de Montréal, 2500 Chemin de Polytechnique, Montréal, H3T 1J4, QC, Canada

[2] PIMM Laboratory, Arts et Métiers Institute of Technology, CNRS, Cnma, HESAM Université, 151 Boulevard de l'Hôpital, 75013, Paris France

CRediT authorship contribution statement

Sima Rishmawi: Conceptualization of the study, Formal Analysis, Methodology, Software (All codes), Validation, Model construction and analysis, Writing - original draft. **Sebastian Rodriguez**: Formal Analysis, Methodology, Software (sPGD algorithm), Model construction and analysis, Writing - review and editing. **Francisco Chinesta**: Conceptualization of the study, Writing - review and editing. **Frédérick P. Gosselin**: Conceptualization of the study, Writing - review and editing, Funding acquisition, Project administration.

Abstract

Structural dynamics systems are represented by discretized partial differential equations, whose solutions depend on various parameters. Developing high-fidelity numerical models for multi-dimensional systems or those with multiple parameters can be computationally expensive, particularly if the systems are non-linear. Consequently, the concept of a precalculated library of the system's response to a wide range of parameters is appealing. Thus, a global non-linear space-frequency solver is proposed that produces a low-rank representation of the solution using Modal Basis analysis known as the Harmonic Modal Hybrid Method. The DEIM is also used to accelerate its convergence by creating a reduced basis of the non-linear function(s) based on either calculated or experimental values. The optimized solver is then employed for rapid offline computations to construct surrogate models that can give real-time predictions of the parametrized dynamic response using the sPGD technique. These models serve as building blocks for virtual twins that necessitate near-instantaneous calculations when the system parameters and/or conditions are changed. A proof of concept is illustrated by using this technique to analyze a well-known non-linear system, the cantilevered beam with a non-linear cubic spring attached to its end. This method can be easily extended to solve other dynamical systems quickly and effectively.

4.1 Introduction

A "Virtual Twin" is a numerical model based on the physics of the system in terms of its parametric partial differential equations (PDEs), while a "Digital Twin" treats the system as a black box in the sense that it relates inputs to outputs, and thus it can perform real-time predictions and calculations as data are fed into the system in real-time. However, a "Hybrid Twin" is a solution that combines both concepts mentioned above, that is, enriching a physics-based model with measured data creating a virtual version of a real or potential physical system, as defined by Chinesta et al. (2019). To gain a more comprehensive understanding of hybrid twins, the interested reader is referred to the following sources: (Ghnatios et al., 2019; Ghnatios, 2021; Sancarlos et al., 2021b,a; Torregrosa et al., 2022; Ghnatios et al., 2023b,a; Rodriguez et al., 2023b), as well as the references therein. The use of hybrid twins is driven by the desire to take advantage of digital technology to improve comprehension, decision-making, and optimization of physical systems, resulting in increased productivity, decreased expenses, and innovation in a variety of industries. However, the difficulty would be to solve physics-based models in a fast and optimized manner. This problem may be overcome by using more powerful computers, but this is not always possible due to technical and financial constraints.

In many cases, the physics-based models used to construct the virtual twin are based on discretized PDEs. These equations are dependent on a range of parameters, such as space, time, geometric parameters, boundary conditions, initial conditions, etc. making them high-dimensional and expensive to solve. High-dimensional models suffer from the curse of dimensionality (Bellman et al., 1957), which is a common problem in numerical analysis. Without any simplifying assumptions, the number of discrete points needed to estimate a function with multiple variables at a specified accuracy in a given domain increases exponentially with the number of dimensions (Lee and Verleysen, 2007). Model Order Reduction (MOR) techniques have been developed to address this issue as they take advantage of the fact that the solution to a problem typically contains redundant information (Aguilera, 2017).

If our view of dimensions is limited to just spatial ones, the concept of the curse of dimensionality is not as critical. Ladevèze and Chamoin (2011) included time as an additional parameter, while Chinesta et al. (2011, 2014) demonstrated that system parameters can be considered as additional dimensions, and a solution can be obtained for a variety of these parameters. Thus, when we refer to system dimensions, we are referring to both spatio-temporal dimensions and other parameters such as geometric parameters, material parameters, initial conditions, boundary conditions, etc.

A low-rank approximation can thus be sought in place of the high-fidelity solution. MOR techniques do not make the model simpler, but rather approximate the solution in a way that reduces the complexity of the calculation process without sacrificing precision (Chinesta et al., 2019).

Projection-based methods are a popular type of MOR techniques. These methods look for a solution to the model equation in a lower-dimensional space that is tailored to the particular problem. Two types of projection-based techniques exist (Lauzeral et al., 2019): *a priori* and *a posteriori*. The former builds a low-rank approximation of the solution on-the-fly using the PDEs without requiring a dataset, while the latter uses a pre-existing dataset of the problem solution to generate a low-rank approximation. These techniques are also known as intrusive (corresponding to *a priori*) and non-intrusive (corresponding to *a posteriori*).

Proper Generalized Decomposition (PGD) is a common *a priori* technique initially proposed as a key element of the LATIN (LArge Time Increment) method (Ladevèze, 1985). This method was designed to solve non-linear mechanical problems through a sequence of alternating linear and non-linear steps. At each linear step, PGD is used to provide a cost-effective, low-rank approximation of the solution to a global space-time problem. In this context as well, Vella et al. introduced Galerkin-based PGD formulations through two different methods, namely the Lagrangian formalism and the Hamiltonian formalism, with the goal of enhancing the stability and energy conservation properties of the method, as discussed by Vella and Prudhomme (2022). Subsequently, the PGD approach, which relies on the Hamiltonian formalism, was further improved by incorporating a solver that integrates Modal Decomposition with the conventional PGD framework. This enhancement proved beneficial in expediting the convergence of the fixed point algorithm, as demonstrated by Vella et al. (2024). All this work limited the use of PGD to finding solutions of mechanical problems as functions of only space and time. It was Chinesta et al. (2010a); Chinesta and Cueto (2014) and Ammar et al. (2006) who extended the concept to include system parameters in addition to time and space, thus creating the concept of the parametric solution and realizing that it is in fact a generalization of the Proper Orthogonal Decomposition (POD) method. As PGD requires the alteration of the PDE that governs the system in order to calculate its solution in a separate variable format, it is considered an "intrusive" approach. This makes the implementation of this technique complicated when working with non-linear PDEs despite the fact that it does not necessitate prior knowledge of the solution.

On the other hand, POD is an *a posteriori* technique that was initially proposed by Karhunen (1946) and Loeve (1948). The approach involves generating a reduced basis for the reference problem using solutions obtained from a previous offline calculation stage. This is why it is referred to as *a posteriori*. It is computed by applying the Singular Value Decomposition (SVD) technique to a snapshot matrix, which is formed by concatenating the solutions at various time instants and various system parameters, where each column of this snapshot matrix represents the solution at a single time step, while each line represents the solution as a function of a random set of system parameters. Lauzeral et al. (2019); Chinesta et al. (2019); Schilders et al. (2008), and Yan et al. (2020) discussed this method in more details. POD has certain restrictions, as the snapshot matrix is necessary for the solution and may not always be accessible or too costly to calculate. Additionally, it is not a suggested approach for systems with a large number of parameters, as the computations will be burdensome due to the curse of dimensionality.

PGD can also be implemented in an *a posteriori*, non-intrusive approach, which involves carrying out an offline calculation using a high-fidelity model and compressing the results using the sparse Proper Generalized Decomposition (sPGD) method as shown by Rodriguez et al. (2023b). The sPGD method was introduced by Ibáñez et al. (2018) and Sancarlos et al. (2021c). It has the advantages of being able to work with a non-structured dataset, being robust in high dimensions, and not requiring a large dataset to converge. Another previous work done in this context is that of Sancarlos et al. (2021d) which shows that by keeping the offline calculations relatively inexpensive, the PGD decomposition can be calculated using a sparse dataset over a parametric space.

In this paper, we present an *a priori*, intrusive approach involving a non-linear global space-frequency solver that iteratively constructs a low-rank representation of the solution based on the use of the Harmonic-Modal Hybrid (HMH) method initially presented by Germoso et al. (2020), combined with the Discrete Empirical Interpolation Method (DEIM). The novelty here is that the HMH approach is used to carry out the offline calculations in the frequency domain, making the solver much faster than those involving a time integration scheme as it does not require the inversion of large matrices. The significance of incorporating DEIM lies in its ability to build an approximating reduced basis of the non-linear function(s) using calculated (if available) or experimental values, which helps to accelerate computations. The speed and accuracy of this solver make it an ideal tool for creating a parametrized solution to the problem. By running the solver multiple times with different sets of parameters, a sparse dataset is generated. This dataset is then used as input for an sPGD algorithm, which produces a library of solutions for the parametrized problem. This library can then be used to quickly and efficiently obtain the solution for any given set of parameters to give interpolated predictions in real-time. This facilitates system monitoring for control purposes or maintenance planning. The value of a parametrized solution is that it eliminates the need to recalculate the solution whenever a problem parameter is altered, making it a fundamental building block of hybrid twins. Overall, the approach presented is versatile and can be applied to any problem as long as the differential equation is clearly defined. Nevertheless, the model developed in this study is particularly tailored to address a specific one, the beam problem detailed in the subsequent section in our case.

This paper is structured as follows: Section 4.2, presents the mathematical model and equation of motion of a cantilevered beam under harmonic excitation with a non-linear spring attached to its end, used as a proof of concept to demonstrate the capabilities of the developed method. Then it introduces the HMH-DEIM non-linear global space-frequency solver and explains how it is used to generate the sparse dataset that will be used as input for the sPGD algorithm to construct the library of solutions. The outcomes of this study are outlined in Section 4.3, where the proposed method is applied to analyze the beam under two scenarios: the first scenario involves zero boundary conditions with harmonic excitation at the tip, while the second scenario involves the same beam with a base excitation applied at the clamped end as a non-zero Dirichlet boundary condition. Finally, the final thoughts and potential directions for further research are provided in Section 4.4.

4.2 Methodology

We consider the cantilevered beam shown in Figure 4.1 with a length L, a constant Young's modulus E, a constant cross-sectional area moment of inertia I, and a constant mass per unit length m, then the PDE that governs the transverse vibrations of the beam is as follows:

$$m\frac{\partial^2 u(x,t)}{\partial t^2} + d\frac{\partial u(x,t)}{\partial t} + EI\frac{\partial^4 u(x,t)}{\partial x^4} + k_{NL}u^3(L,t)\delta(x-L) = F_0\cos(\Omega t)\delta(x-L), \quad (4.1)$$

where x is the space coordinate along the beam length, t represents time, and $u(x,t)$ is the transverse deflection of the beam. A harmonic force is applied to the tip of the beam, where F_0 is the amplitude of the force and Ω is the excitation frequency. The details of the derivation of this equation can be found in the book by Meirovitch (2001). The beam is exposed to modal damping, which is represented by the damping coefficient d. Further details about this will be discussed later.

We define the boundary conditions of the system. At the clamped end, we define a Dirichlet boundary condition, and we set the slope of the deflection to zero:

$$u(0,t) = u_b(0,t), \quad \frac{\partial u(0,t)}{\partial x} = 0, \quad (4.2)$$

for the sake of simplicity, in our derivations we assume zero Dirichlet boundary conditions

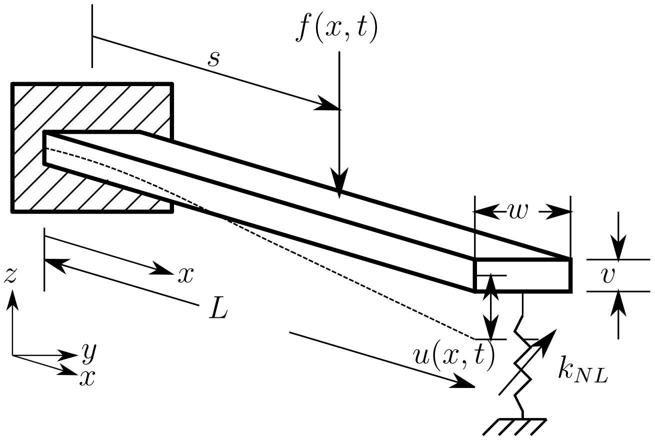


Figure 4.1 Schematic of a Cantilevered Beam with a Non-linear Spring Attached to its Tip.

initially, and subsequently in Section 4.3, we address a scenario where a harmonic base excitation is applied as a Dirichlet boundary condition at the clamped end.

At the free end, both the bending moment $M(L,t)$ and shear force $V(L,t)$ are zero:

$$\begin{aligned} M(L,t) &= EI\frac{\partial^2 u(L,t)}{\partial x^2} = 0, \\ V(L,t) &= -EI\frac{\partial^3 u(L,t)}{\partial x^3} = 0. \end{aligned} \tag{4.3}$$

The non-linearity comes in the form of an external force applied to the beam. It is the result of attaching a cubic spring to a certain node along the beam. This force is considered proportional to the cube of the beam deflection, and it has a value only at the node to which the spring is attached, where k_{NL} is the proportionality constant. This form of non-linearity yields a multidegree-of-freedom version of the Duffing oscillator equation, which is a widely studied non-linear system as demonstrated by Korsch et al. (2008). Salas et al. (2021) have shown that despite its simplicity, the Duffing oscillator exhibits a very rich dynamical behavior and is used to model physical systems in various domains. Furthermore, the single and multiple degree-of-freedom forms of the Duffing oscillator equation have been solved using a variety of analytical and numerical methods, as noted by Johannessen (2015), making it an ideal system for assessing the accuracy and efficiency of the proposed strategy.

It is recommended to express equation (4.1) in a non-dimensional format to simplify the choice of different parameters to study their effect on the response. From linear beam theory we know that the maximum deflection at the tip is given by:

$$u_{static} = \frac{F_0}{6EI}(x^3 - 3Lx^2)\bigg|_{x=L} = \frac{F_0 L^3}{3EI}. \tag{4.4}$$

Then, we define our non-dimensional parameters as follows (the $(\bar{\ })$ notation defines a non-dimensional value):

$$\begin{aligned} &\bar{u}(\bar{x},\bar{t}) = \frac{3EI}{F_0 L^3}u(x,t), \quad \bar{x} = \frac{x}{L}, \quad \bar{t} = \omega_1 t, \quad \omega_1 = \beta_1^2\sqrt{\frac{EI}{mL^4}}, \\ &\beta_1 = 1.8752, \quad \bar{\Omega} = \frac{\Omega}{\omega_1}, \quad \bar{d} = \frac{d}{m\omega_1}, \quad \alpha = \frac{k_{NL}F_0^2 L^9}{9E^3 I^3}, \end{aligned} \tag{4.5}$$

where ω_1 is the first natural frequency of the linear beam equation, and α is the non-linearity coefficient. Substituting the values from equation (4.5) into equation (4.1) we get the non-dimensional equation of motion as follows:

$$\frac{\partial^2\bar{u}(\bar{x},\bar{t})}{\partial\bar{t}^2} + \bar{d}\frac{\partial\bar{u}(\bar{x},\bar{t})}{\partial\bar{t}} + \frac{1}{\beta_1^4}\frac{\partial^4\bar{u}(\bar{x},\bar{t})}{\partial\bar{x}^4} + \frac{\alpha}{\beta_1^4}\bar{u}^3(\bar{x},\bar{t})\delta(\bar{x}-1) = \frac{3}{\beta_1^4}\cos\Big(\bar{\Omega}\bar{t}\Big)\delta(\bar{x}-1). \tag{4.6}$$

Solving the problem using dimensionless variables guarantees that the system dynamics are parametrized with a minimal number of parameters. After acquiring this dimensionless solution, it can be used to span a broad parameter space for beams of varying rigidity, cross-sectional area, length, etc.

Equation (4.6) describes the continuous analytical model of our beam, yet for a numerical solution, discretization is required. By employing a finite element formulation using beam shape functions, equation (4.6) can be written in matrix and vector form as follows for simpler manipulation via computer-based numerical methods:

$$\mathbf{M}\{\overrightarrow{\ddot{\bar{\mathbf{u}}}}\} + \mathbf{D}\{\overrightarrow{\dot{\bar{\mathbf{u}}}}\} + \mathbf{K}\{\overrightarrow{\bar{\mathbf{u}}}\} + \mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}}\}^3 = \{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}\cos\Big(\bar{\Omega}\bar{t}\Big), \tag{4.7}$$

where $\{\ddot{\ }\}$ and $\{\dot{\ }\}$ are the second and first dimensionless time derivative, $\mathbf{M}$ and $\mathbf{K}$ are the non-dimensional mass and stiffness matrices, $\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}$ is the non-dimensional amplitude of the forcing term made up of zeros except the element representing the node where the force is applied which contains a value of $3/\beta_1^4$, and the matrix $\mathbf{K_{nl}}$ is composed of zeros, with the exception of the node where the non-linear spring is connected, which has a value of α/β_1^4. This matrix form is derived using finite element analysis to create the weak formulation of the equation by applying the Galerkin method. For details, the interested reader is referred to the work Burnett (1988).

We define the modal damping coefficient ζ_j, corresponding to mode j, where $j = 1, \cdots, n_{modes}$ and ω_j is the j^{th} natural frequency of the system. Using the orthogonality property of the mode shapes with respect to the mass matrix, and in order to guarantee a diagonal damping matrix in the modal basis, the modal damping matrix $\mathbf{D}$ is defined such as:

$$\mathbf{d} = \boldsymbol{\Phi}^T \mathbf{D} \boldsymbol{\Phi}, \tag{4.8}$$

where $\boldsymbol{\Phi}$ is the matrix containing the eigenvectors of the system and $d_{ij} = 2\zeta_i \omega_j$ when $i = j$ and 0 otherwise. We can identify two parameters that affect the response $\{\overrightarrow{\bar{\mathbf{u}}}\}$, namely the damping coefficient ζ_i and the non-linearity coefficient α. The aim of the technique presented in this paper is to construct a repository of solutions for this equation in advance, so that we can rapidly and effectively compute the response for any combination of the two parameters mentioned above when needed.

Our technique comprises three stages. Initially, we introduce the Harmonic-Modal Hybrid (HMH) non-linear global space-frequency solver, which is used to solve for one set of parameters. Second, we incorporate DEIM into the aforementioned solver to make it more efficient. Finally, we demonstrate how this HMH-DEIM solver is used as a block to generate sparse data that will be fed into the sPGD algorithm to create the parametric library of solutions.

4.2.1 Harmonic-Modal Hybrid (HMH) Solver

Lets take another look at equation (4.7), where the vectors $\{\overrightarrow{\ddot{\bar{\mathbf{u}}}}\}$, $\{\overrightarrow{\dot{\bar{\mathbf{u}}}}\}$, and $\{\overrightarrow{\bar{\mathbf{u}}}\}$ are the non-dimensional transverse acceleration, velocity, and displacement at time $\bar{t}$, respectively; and $\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}$ is the magnitude of the harmonic force applied to the beam. It is important to note that the non-dimensional excitation frequency $\bar{\Omega}$ can be varied; however, our method is used to find the solution at one excitation frequency at a time.

The initial step of the HMH solver is to solve the eigenvalue problem to obtain the eigenvalues and eigenvectors of the system, $\boldsymbol{\Omega^2}$ and $\boldsymbol{\Phi}$ respectively. The beam transverse deflection, velocity, and acceleration can then be expressed as:

$$\{\overrightarrow{\bar{\mathbf{u}}}\} = \boldsymbol{\Phi}\{\overrightarrow{\bar{\mathbf{y}}}\},\ \{\overrightarrow{\dot{\bar{\mathbf{u}}}}\} = \boldsymbol{\Phi}\{\overrightarrow{\dot{\bar{\mathbf{y}}}}\},\ \{\overrightarrow{\ddot{\bar{\mathbf{u}}}}\} = \boldsymbol{\Phi}\{\overrightarrow{\ddot{\bar{\mathbf{y}}}}\}, \tag{4.9}$$

where $\{\overrightarrow{\bar{\mathbf{y}}}\}$, $\{\overrightarrow{\dot{\bar{\mathbf{y}}}}\}$, and $\{\overrightarrow{\ddot{\bar{\mathbf{y}}}}\}$ are defined as the modal deflection, velocity, and acceleration respectively. Substituting equation (4.9) into equation (4.7), and pre-multiplying by $\boldsymbol{\Phi}^T$ gives:

$$\boldsymbol{\Phi}^T \mathbf{M} \boldsymbol{\Phi}\{\overrightarrow{\ddot{\bar{\mathbf{y}}}}\} + \boldsymbol{\Phi}^T \mathbf{D} \boldsymbol{\Phi}\{\overrightarrow{\dot{\bar{\mathbf{y}}}}\} + \boldsymbol{\Phi}^T \mathbf{K} \boldsymbol{\Phi}\{\overrightarrow{\bar{\mathbf{y}}}\} + \boldsymbol{\Phi}^T \Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}}\}^3\Big) = \boldsymbol{\Phi}^T \Big(\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\} \cos\big(\bar{\Omega}\bar{t}\big)\Big), \tag{4.10}$$

which can be simplified to:

$$\mathbf{m}\{\overrightarrow{\ddot{\bar{\mathbf{y}}}}\} + \mathbf{d}\{\overrightarrow{\dot{\bar{\mathbf{y}}}}\} + \mathbf{k}\{\overrightarrow{\bar{\mathbf{y}}}\} + \mathbf{\Phi}^T\Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}}\}^3\Big) = \mathbf{\Phi}^T\Big(\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}\cos\big(\bar{\Omega}\bar{t}\big)\Big), \tag{4.11}$$

where **m**, **d**, and **k** are the diagonal modal mass, damping, and stiffness matrices. It should be noted here that **m** is effectively an identity matrix.

The next step is to apply a Fast Fourier Transform ($\mathcal{FFT}$) to equation (4.11) to obtain:

$$-\omega^2\ \mathbf{m}\{\overrightarrow{\hat{\mathbf{Y}}}\} + i\omega\ \mathbf{d}\{\overrightarrow{\hat{\mathbf{Y}}}\} + \mathbf{k}\{\overrightarrow{\hat{\mathbf{Y}}}\} + \mathcal{FFT}\Big(\mathbf{\Phi}^T\Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}}\}^3\Big)\Big) = \mathbf{\Phi}^T\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}\delta(\omega-\bar{\Omega}), \tag{4.12}$$

where $\{\overrightarrow{\hat{\mathbf{Y}}}\}$ is the $\mathcal{FFT}$ of $\{\overrightarrow{\bar{\mathbf{y}}}\}$, $\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}$ is the $\mathcal{FFT}$ of $\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}\cos\big(\bar{\Omega}\bar{t}\big)$, and ω represents the frequency domain. The modal displacement $\{\overrightarrow{\hat{\mathbf{Y}}}\}$ allows to calculate the spatial displacement $\{\overrightarrow{\hat{\mathbf{U}}}\}$ in the frequency domain by multiplying $\{\overrightarrow{\hat{\mathbf{Y}}}\}$ by the eigenvector matrix. The inverse Fast Fourier Transformation ($\mathcal{IFFT}$) of $\{\overrightarrow{\hat{\mathbf{U}}}\}$ returns the system response $\{\overrightarrow{\bar{\mathbf{u}}}\}$.

Examining equation (4.12), it is not possible to apply the $\mathcal{FFT}$ to non-linear functions, making the term $\mathcal{FFT}(\mathbf{\Phi}^T(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}}\}^3))$ inapplicable. Consequently, we employ the iterative fixed-point Alternating Frequency/Time Method (AFT) proposed by Cameron and Griffin (1989). This method involves taking the $\mathcal{FFT}$ of the response obtained in the first iteration and returning it to the time domain to calculate the non-linearity there, which is then transformed back to the frequency domain and used in the subsequent iteration to update the value of the system response until convergence is achieved.

Firstly, the solution of the linear version of equation (4.12) is expressed as follows:

$$\{\overrightarrow{\hat{\mathbf{Y}}^{[\mathbf{1}]}}\} = \frac{\{\overrightarrow{\mathbf{\Phi}}\}_j^T\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}\delta(\omega-\bar{\Omega})}{-\omega^2\ m_{jj} + i\omega\ d_{jj} + k_{jj}},\ j = 1\cdots n_{modes}, \tag{4.13}$$

where the superscript in brackets represents the iteration number.

In light of the fixed-point AFT method explained above, we include the non-linear force in our approach by computing the non-linear force component in each iteration using the $\{\overrightarrow{\bar{\mathbf{u}}}\}$ determined in the preceding iteration. The non-linear force term is then expressed as a numerical vector to which we apply an $\mathcal{FFT}$ and then use it to calculate the response in the next iteration as follows:

$$\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{nl}^{[\mathbf{i-1}]}}\} = \mathcal{FFT}\Big(\mathbf{\Phi}^T\Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}^{[\mathbf{i-1}]}}\}^3\Big)\Big), \tag{4.14}$$

where [**i**] represents the current iteration. Equation (4.13) can be modified to take into account the non-linear force by making the following adjustment:

$$\{\overrightarrow{\hat{\mathbf{Y}}^{[\mathbf{i}]}}\} = \frac{\{\overrightarrow{\boldsymbol{\Phi}}\}_j^T \left(\{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{0}}}\}\delta(\omega - \bar{\Omega}) - \{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{nl}}^{[\mathbf{i-1}]}}\}\right)}{-\omega^2 \; m_{jj} + i\omega \; d_{jj} + k_{jj}}, j = 1 \cdots n_{modes}. \tag{4.15}$$

When the damping is low and the non-linearities are high, the magnitude of the system's response can become too high, potentially leading to the divergence of the iterative resolution. This is because a high deflection amplitude makes the non-linear force huge, thus the "linearization" fixed-point approach we employ using the AFT method becomes unable to correctly approximate the solution. In order to address this limitation, a virtual high damping is added on the left-hand side of the equation, which is then counteracted by adding an equivalent damping force to the right-hand side. This force is calculated at each iteration using the solution from the previous iteration, so that the total damping force is equivalent to the desired low damping. The objective is to make minor adjustments to the iterative solver to prevent it from diverging. This damping is expressed as follows:

$$\mathbf{D} = \mathbf{D}_{high} - \mathbf{D}_{aux}, \tag{4.16}$$

where $\mathbf{D}$ is the required damping matrix, $\mathbf{D}_{high}$ is the virtual high damping matrix, and $\mathbf{D}_{aux}$ is the counteracting damping matrix. Then equation (4.13) can be modified as follows:

$$\{\overrightarrow{\hat{\mathbf{Y}}^{[\mathbf{i}]}}\} = \frac{\{\overrightarrow{\boldsymbol{\Phi}}\}_j^T \left(\{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{0}}}\}\delta(\omega - \bar{\Omega}) - \{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{nl}}^{[\mathbf{i-1}]}}\} + i\omega \; \mathbf{D}_{aux}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\right)}{-\omega^2 \; m_{jj} + i\omega \; d_{jj,high} + k_{jj}}, j = 1 \cdots n_{modes}. \tag{4.17}$$

This framework is suitable for the particular problem we are addressing, since we are applying an external force to the end of the beam. In cases of base excitation with Dirichlet boundary conditions, this approach needs to be modified. Modal Basis functions are zero at the boundary nodes, as they are calculated based on the free degrees of freedom, making it difficult to accurately represent the imposed deflection of the boundary nodes using the modal basis alone. To address this, Clough and Penzien (2003) and Rodriguez et al. (2023a) showed that a rigorous mathematical model that can solve for any Dirichlet boundary conditions expresses the total deflection of a structure as the sum of the deflection caused by a quasi-static force, and the deflection caused by the dynamic inertial and viscous forces, as follows:

$$\{\overrightarrow{\hat{\mathbf{Y}}}\} = \{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{hmh}}}\} + \{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{s}}}\}, \tag{4.18}$$

where $\{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{hmh}}}\}$ is the harmonic-modal hybrid solution and $\{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{s}}}\}$ is the quasi-static solution calculated using the equations below:

$$\mathbf{k}\{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{s}}}\} = \mathbf{\Phi}^T\Big(\{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{0}}}\}\delta(\omega-\bar{\Omega}) - \{\overrightarrow{\hat{\mathbf{F}}^{[\mathbf{i-1}]}_{\mathbf{nl}}}\} + i\omega\ \mathbf{D}_{aux}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\Big). \tag{4.19}$$

$$\{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{hmh}}}\} = \frac{\Big(\omega^2\ m_{jj} - i\omega\ d_{jj,high}\Big)}{\Big(-\omega^2\ m_{jj} + i\omega\ d_{jj,high} + k_{jj}\Big)}\{\overrightarrow{\hat{\mathbf{Y}}_{\mathbf{s}}}\}, j = 1\cdots n_{modes}. \tag{4.20}$$

The final step we take to enhance the convergence of our HMH solver is a relaxation technique that is designed to accelerate the convergence and prevent it from getting stuck around a local minimum without reaching it. This technique is presented by Salinas et al. (2017), and is applied as follows:

$$\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} = \gamma_1\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} + (1-\gamma_1)\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\} + \gamma_1(1-\gamma_1)^{\gamma_2+1}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-2}]}}\}, \tag{4.21}$$

where the values of γ_1 and γ_2 are selected by trial and error balancing convergence and stability.

Convergence is attained when the difference between two successive values of $\{\overrightarrow{\hat{\mathbf{U}}}\}$ falls below a certain tolerance, *i.e.*:

$$E_{stg} = \frac{\|\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} - \{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\|}{\|\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\|} < tol. \tag{4.22}$$

The HMH-solver algorithm can be summarized in Algorithm 1.

4.2.2 HMH Solver accelerated through DEIM (HMH-DEIM)

The method described above has a drawback in that it requires one $\mathcal{FFT}$ and one $\mathcal{IFFT}$ operations to be performed in each iteration i to update the value of the non-linear force. These operations are computationally intensive, slowing the method. To address this problem, we propose incorporating DEIM into the method. This method begins with a one-time, offline process. The outcomes of this process are stored to be used in the online procedure to quickly and efficiently calculate the solutions.

The goal is finding a solution to the problem for a variety of values of two parameters, as previously mentioned; the damping coefficient ζ and the non-linearity coefficient α. Therefore, we select a range of values for each of these parameters and use the latin hypercube sampling to randomly select a number of combinations of the two parameters:

$$\mathcal{S} = [(\zeta_1,\alpha_1),(\zeta_2,\alpha_2),\cdots,(\zeta_{n_s},\alpha_{n_s})], \tag{4.23}$$

where n_s is the number of samples.

The HMH-solver in Algorithm 1 is employed to find the solution for each of the scenarios in the sample matrix $\mathcal{S}$. The objective is to compute the values of the beam deflection and the non-linear force corresponding to each scenario in both the time and frequency domains. Therefore, for each scenario, the deflection of the tip of the beam is stacked in the time domain and in the frequency domain in two distinct snapshot matrices. Similarly, snapshots of the non-linear force that is exerted at the tip in each situation are also generated, as follows:

$$\begin{aligned}
\hat{\mathcal{U}}(1,\omega) &= [\{\overrightarrow{\hat{\mathbf{U}}}\}_1, \{\overrightarrow{\hat{\mathbf{U}}}\}_2, \cdots, \{\overrightarrow{\hat{\mathbf{U}}}\}_{n_s}], \\
\bar{\mathbb{U}}(1,\bar{t}) &= [\{\overrightarrow{\bar{\mathbf{u}}}\}_1, \{\overrightarrow{\bar{\mathbf{u}}}\}_2, \cdots, \{\overrightarrow{\bar{\mathbf{u}}}\}_{n_s}], \\
\hat{\mathcal{F}}_{\mathbf{NL}}(1,\omega) &= [\{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{NL}}}\}_1, \{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{NL}}}\}_2, \cdots, \{\overrightarrow{\hat{\mathbf{F}}_{\mathbf{NL}}}\}_{n_s}], \\
\bar{\mathscr{f}}_{\mathbf{NL}}(1,\bar{t}) &= [\{\overrightarrow{\bar{\mathbf{f}}_{\mathbf{NL}}}\}_1, \{\overrightarrow{\bar{\mathbf{f}}_{\mathbf{NL}}}\}_2, \cdots, \{\overrightarrow{\bar{\mathbf{f}}_{\mathbf{NL}}}\}_{n_s}].
\end{aligned} \tag{4.24}$$

Keeping in mind that $\hat{\mathcal{U}}(1,\omega)$ and $\hat{\mathcal{F}}_{\mathbf{NL}}(1,\omega)$ are in the frequency domain, and $\bar{\mathbb{U}}(1,\bar{t})$ and $\bar{\mathscr{f}}_{\mathbf{NL}}(1,\bar{t})$ are in the time domain, we use the symbols $\hat{\mathcal{U}}$, $\bar{\mathbb{U}}$, $\hat{\mathcal{F}}_{\mathbf{NL}}$, and $\bar{\mathscr{f}}_{\mathbf{NL}}$ to refer to the four snapshot matrices for the sake of simplicity.

We now proceed with the steps detailed in Appendix D to create an optimal basis for each of the four snapshot matrices using Singular Value Decomposition (SVD). The number of modes (columns) chosen are those corresponding to normalized singular values higher than 10^{-4}. This results in four basis matrices: $\boldsymbol{\Xi}_{\hat{\mathcal{U}}p}$, $\boldsymbol{\Xi}_{\bar{\mathbb{U}}p}$, $\boldsymbol{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}$, and $\boldsymbol{\Xi}_{\bar{\mathscr{f}}_{\mathbf{NLp}}}$. Then a number of rows corresponding to the interpolation points are chosen depending on the point selection operator $\mathbf{P}_p$ that is determined using the DEIM magic point strategy as explained by Brunton and Kutz (2019). It should be noted, however, that the indices representing the peaks of the beam deflection in the frequency domain are added to the automatically selected points as they provide valuable information about the system. An example of those sampling points is illustrated in Figure 4.2. Those points are chosen automatically by the DEIM algorithm to ensure that the reduced bases created can approximate the deflection and the non-linear force in both the time and frequency domains correctly. The offline procedure is concluded here, and the four reduced-order basis matrices are stored to be used in the online computations. The details of this process are outlined in Algorithm 2. The online process is the same as the HMH-solver algorithm, yet in every iteration i, as the beam deflection is determined in the frequency domain $\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\}$, rather than employing $\mathcal{IFFT}$, a constant vector $\{\overrightarrow{\mathbf{c}}\}$ is computed using the inverse of the basis matrix $\boldsymbol{\Xi}_{\hat{\mathcal{U}}p}$ in the frequency domain as follows:

$$\{\overrightarrow{\mathbf{c}}\} = \boldsymbol{\Xi}_{\hat{\mathcal{U}}p}^{-1}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\}. \tag{4.25}$$

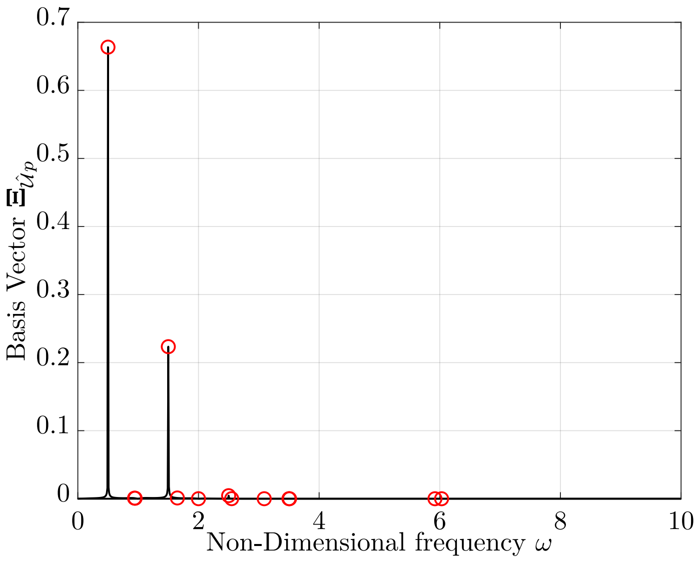


Figure 4.2 Representative example of the sampling points calculated by the DEIM to create the reduced order approximation basis $\boldsymbol{\Xi}_{\hat{\mathcal{U}}p}$ of the displacement in the frequency domain where $\bar{\Omega} = 0.5$. The red circles (○) represent the selected DEIM points.

Next, we use the constant vector $\{\overrightarrow{\mathbf{c}}\}$ to compute the beam deflection in the time domain. This is achieved by multiplying it with the basis $\boldsymbol{\Xi}_{\bar{\mathbb{U}}p}$ in the time domain as shown below:

$$\{\overrightarrow{\bar{\mathbf{u}}^{[\mathbf{i}]}_{\bar{\mathbf{t}}}}\} = \boldsymbol{\Xi}_{\bar{\mathbb{U}}p}\{\overrightarrow{\mathbf{c}}\}, \tag{4.26}$$

thereby converting the beam deflection from the frequency domain to the time domain through two straightforward matrix multiplication steps, which are considerably quicker than the $\mathcal{IFFT}$.

Subsequently, we compute the non-linear force within the time domain from the beam deflection using the same approach seen in the HMH solver, as illustrated below:

$$\{\overrightarrow{\bar{\mathbf{f}}^{[\mathbf{i}]}_{\mathbf{NL}}}\} = \boldsymbol{\Phi}^T\Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}^{[\mathbf{i}]}_{\bar{\mathbf{t}}}}\}^3\Big). \tag{4.27}$$

Finally, the non-linear force is reconverted into the frequency domain via the DEIM bases, employing the same method previously applied to the beam deflection transformation, specifically by computing a constant vector $\{\overrightarrow{\mathbf{h}}\}$ as follows:

$$\{\overrightarrow{\mathbf{h}}\} = \boldsymbol{\Xi}^{-1}_{\bar{f}_{\mathbf{NL_p}}}\{\overrightarrow{\bar{\mathbf{f}}^{[\mathbf{i}]}_{\mathbf{NL}}}\}, \tag{4.28}$$

$$\{\overrightarrow{\mathbf{\hat{F}^{[i]}_{NL}}}\} = \mathbf{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}\{\overrightarrow{\mathbf{h}}\}. \tag{4.29}$$

Those steps are presented in Algorithm 3.

4.2.3 sPGD Solver

The final step in producing the parametrized answer to our problem is to create some sparse data and use it to find the sPGD representation of the maximum amplitude of the tip of the beam $\bar{u}_{max}$. Our goal is to represent the solution as:

$$\bar{u}_{max}(\zeta, \alpha) = \sum_{k=1}^{M} F_1^k(\zeta) F_2^k(\alpha), \tag{4.30}$$

this formulation enables us to obtain the solution for any combination of the two variables, the damping coefficient ζ and and the non-linearity coefficient α. The functions $F_j^k(x)$ are each represented as:

$$F_j^k(\eta) = \{N_j^k(\eta)\}^T \{\overrightarrow{\mathbf{a_j}}\}, \tag{4.31}$$

where $N_j^k(\eta)$ is a multiquadric radial basis function which is selected as the interpolation function. It is expressed in the equation below:

$$N_j^k(\eta) = \sqrt{1 + (\epsilon\eta)^2}, \tag{4.32}$$

where j represents the node index, k is the index representing the number of interpolation function needed in the sum shown in equation (4.30), $\epsilon = h/(0.815\tau)$, h is a hyper-parameter controlling the shape of the function, τ is the minimum distance between two parameter values in the sparse domain, and η corresponds to the desired parameters; ζ and α in our case. The benefits of using this type of function are elucidated by Fasshauer (2007). $\{\overrightarrow{\mathbf{a_j}}\}$ is a vector of constant nodal values calculated by the sPGD algorithm. More details about the sPGD algorithm are found in Appendix E.

A collection of sparse data is generated by running the HMH-DEIM solver multiple times, each with a random combination of parameters ζ and α, and the solution is stored as a time series for a certain duration. For each time series, the maximum value of the steady-state deflection of the tip of the beam is extracted and saved. The sparse dataset is used as an input for the sPGD algorithm, which provides a parametrized solution in an efficient manner. This is done separately for each case of the excitation frequency $\bar{\Omega}$, which means that the excitation frequency is not parameterized; i.e. the solution can only be found for discrete values of $\bar{\Omega}$.

Algorithm 1: HMH solver algorithm to calculate the solution of a non-linear beam equation using a harmonic-modal hybrid approach in the frequency domain.

```
**Input:** Parameters of the problem to be solved: $\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{K_{nl}}, \{\overrightarrow{\mathbf{f_0}}\}, \bar{\Omega}$.
**Output:** Transverse beam deflection $\{\overrightarrow{\bar{\mathbf{u}}}\}$
$\mathbf{D_{aux}} = \mathbf{D_{high}} - \mathbf{D}$

**procedure** Transform Spatial Basis to Modal Basis()
  $\mathbf{\Phi} \leftarrow eigs(\mathbf{K}, \mathbf{M})$
  $\mathbf{m} \leftarrow \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi}$
  $\mathbf{d_{high}} \leftarrow \mathbf{\Phi}^T \mathbf{D_{high}} \mathbf{\Phi}$
  $\mathbf{k} \leftarrow \mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi}$

**procedure** Fast Fourier Transform()
  $\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}\delta(\omega - \bar{\Omega}) \leftarrow \mathcal{FFT}\Big(\{\overrightarrow{\bar{\mathbf{f}}_\mathbf{0}}\}\cos\Big(\bar{\Omega}\bar{t}\Big)\Big)$

**procedure** Solver's Main Loop()
  $i \leftarrow 1$
  $\{\overrightarrow{\hat{\mathbf{F}}^{[\mathbf{0}]}_{\mathbf{nl}}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}, \{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{0}]}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}, \{\overrightarrow{\hat{\mathbf{U}}^{[-\mathbf{1}]}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}$
  **while** $E_{stg} > tol$
    **do**
      $\{\overrightarrow{\hat{\mathbf{Y}}^{[\mathbf{i}]}}\} \leftarrow \dfrac{\{\overrightarrow{\mathbf{\Phi}}\}_j^T\Big(\{\overrightarrow{\hat{\mathbf{F}}_\mathbf{0}}\}\delta(\omega - \bar{\Omega}) - \{\overrightarrow{\hat{\mathbf{F}}^{[\mathbf{i-1}]}_{\mathbf{nl}}}\} + i\omega\ \mathbf{D}_{aux}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\Big)}{-\omega^2\ m_{jj} + i\omega\ d_{jj,high} + k_{jj}},\ j = 1 \cdots n_{modes}$
      $\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} \leftarrow \mathbf{\Phi}\{\overrightarrow{\hat{\mathbf{Y}}^{[\mathbf{i}]}}\}$
      $\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} \leftarrow \gamma_1\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} + (1 - \gamma_1)\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\} + \gamma_1(1 - \gamma_1)^{\gamma_2+1}\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-2}]}}\}$
      $\{\overrightarrow{\bar{\mathbf{u}}_{\bar{\mathbf{t}}^{[\mathbf{i}]}}}\} \leftarrow \mathcal{IFFT}\Big(\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\}\Big)$
      $\{\overrightarrow{\hat{\mathbf{F}}^{[\mathbf{i}]}_{\mathbf{nl}}}\} \leftarrow \mathcal{FFT}\Big(\mathbf{\Phi}^T\Big(\mathbf{K_{nl}}\{\overrightarrow{\bar{\mathbf{u}}^{[\mathbf{i}]}_{\bar{\mathbf{t}}}}\}^3\Big)\Big)$
      $E_{stg} \leftarrow \dfrac{\|\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i}]}}\} - \{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\|}{\|\{\overrightarrow{\hat{\mathbf{U}}^{[\mathbf{i-1}]}}\}\|}$
      $i \leftarrow i + 1$

**return** $(\{\overrightarrow{\bar{\mathbf{u}}^{[\mathbf{i-1}]}_{\bar{\mathbf{t}}}}\})$
```

Algorithm 2: HMH-DEIM offline algorithm to calculate the approximating basis of the beam deflection and the non-linear force acting on it in the time and frequency domains.

```
Output: The projection matrix $\mathbf{P}_p$, and the approximating basis matrices $\boldsymbol{\Xi}_{\hat{\mathcal{U}}p}$, $\boldsymbol{\Xi}_{\bar{\mathbb{U}}_{\mathbf{p}}}$, $\boldsymbol{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}$, $\boldsymbol{\Xi}_{\bar{f}_{\mathbf{NLp}}}$.

procedure Generating Scenarios()
  comment: Latin hypercube to generate random pairs
  $\mathcal{S} = [(\zeta_1, \alpha_1), (\zeta_2, \alpha_2), \cdots, (\zeta_{n_s}, \alpha_{n_s})]$
  comment: Run HMH Solver for each scenario
  return $(\hat{\mathcal{U}}(1,\omega), \bar{\mathbb{U}}(1,\bar{t}), \hat{\mathcal{F}}_{\mathbf{NL}}(1,\omega), \bar{f}_{NL}(1,\bar{t}))$

procedure Creating Approximating Basis($\hat{\mathcal{U}}(1,\omega)$, $\bar{\mathbb{U}}(1,\bar{t})$, $\hat{\mathcal{F}}_{\mathbf{NL}}(1,\omega)$, $\bar{f}_{NL}(1,\bar{t})$)
  $\boldsymbol{\Xi}_{\hat{\mathcal{U}}p} \leftarrow \mathrm{SVD}(\hat{\mathcal{U}})$
  $\boldsymbol{\Xi}_{\bar{\mathbb{U}}p} \leftarrow \mathrm{SVD}(\bar{\mathbb{U}})$
  $\boldsymbol{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}} \leftarrow \hat{\mathcal{F}}_{\mathbf{NL}}$
  $\boldsymbol{\Xi}_{\bar{f}_{\mathbf{NLp}}} \leftarrow \bar{f}_{\mathbf{NL}}$
  return $(\boldsymbol{\Xi}_{\hat{\mathcal{U}}p}, \boldsymbol{\Xi}_{\bar{\mathbb{U}}_{\mathbf{p}}}, \boldsymbol{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}, \boldsymbol{\Xi}_{\bar{f}_{\mathbf{NLp}}})$

procedure Interpolation Point Selection()
  comment: Run the Interpolation Indices Procedure from Algorithm 4 in D.

return $(\mathbf{P}_p)$
```

Algorithm 3: HMH-DEIM online solver algorithm to calculate the solution of a nonlinear beam equation in the frequency domain.

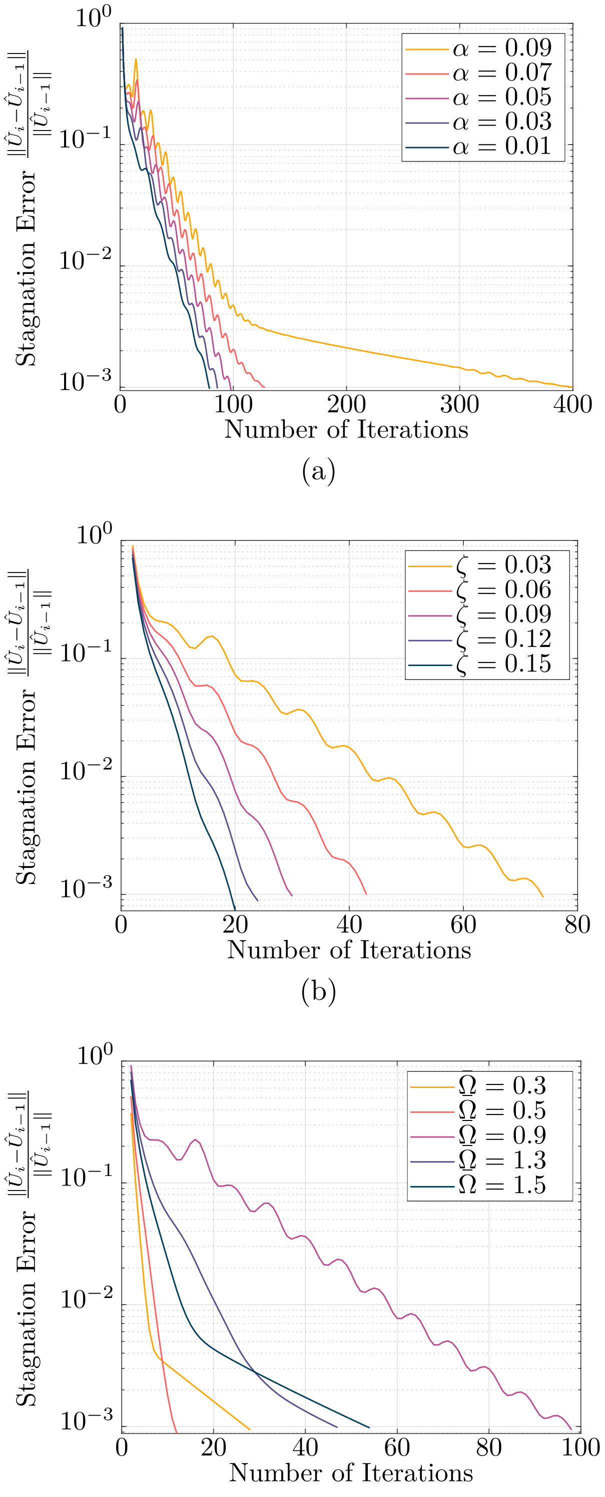
```
Input: Parameters of the problem to be solved: $\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{K_{nl}}, \{\overrightarrow{\mathbf{f_0}}\}, \bar{\Omega}$ .
Output: Transverse beam deflection $\{\overrightarrow{\mathbf{u}}\}$

⋮
procedure Solver's Main Loop()
  $i \leftarrow 1$
  $\{\overrightarrow{\mathbf{\hat{F}_{nl}^{[0]}}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}, \{\overrightarrow{\mathbf{\hat{U}}^{[0]}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}, \{\overrightarrow{\mathbf{\hat{U}}^{[-1]}}\} \leftarrow \{\overrightarrow{\mathbf{0}}\}$
  while $E_{stg} < tol$
    do
      $\{\overrightarrow{\mathbf{\hat{Y}}^{[i]}}\} \leftarrow \frac{\{\overrightarrow{\mathbf{\Phi}}\}_j^T \left(\{\overrightarrow{\mathbf{\hat{F}_0}}\}\delta(\omega - \bar{\Omega}) - \{\overrightarrow{\mathbf{\hat{F}_{nl}^{[i-1]}}}\} + i\omega \ \mathbf{D}_{aux}\{\overrightarrow{\mathbf{\hat{U}}^{[i-1]}}\}\right)}{-\omega^2 \ m_{jj} + i\omega \ d_{jj,high} + k_{jj}}, \ j = 1 \cdots n_{modes}$
      $\{\overrightarrow{\mathbf{\hat{U}}^{[i]}}\} \leftarrow \mathbf{\Phi}\{\overrightarrow{\mathbf{\hat{Y}}^{[i]}}\}$
      $\{\overrightarrow{\mathbf{\hat{U}}^{[i]}}\} \leftarrow \gamma_1\{\overrightarrow{\mathbf{\hat{U}}^{[i]}}\} + (1 - \gamma_1)\{\overrightarrow{\mathbf{\hat{U}}^{[i-1]}}\} + \gamma_1(1 - \gamma_1)^{\gamma_2+1}\{\overrightarrow{\mathbf{\hat{U}}^{[i-2]}}\}$
      $\{\overrightarrow{\mathbf{c}}\} \leftarrow \mathbf{\Xi}_{\hat{\mathcal{U}}p}^{-1}\{\overrightarrow{\mathbf{\hat{U}}^{[i]}}\}$
      $\{\overrightarrow{\mathbf{\bar{u}_{\bar{t}}^{[i]}}}\} \leftarrow \mathbf{\Xi}_{\bar{\mathbb{U}}p}\{\overrightarrow{\mathbf{c}}\}$
      $\{\overrightarrow{\mathbf{\bar{f}_{NL}^{[i]}}}\} \leftarrow \mathbf{\Phi}^T\left(\mathbf{K_{nl}}\{\overrightarrow{\mathbf{\bar{u}_{\bar{t}}^{[i]}}}\}^3\right)$
      $\{\overrightarrow{\mathbf{h}}\} \leftarrow \mathbf{\Xi}_{\bar{f}_{\mathbf{NLp}}}^{-1}\{\overrightarrow{\mathbf{\bar{f}_{NL}^{[i]}}}\}$
      $\{\overrightarrow{\mathbf{\hat{F}_{NL}^{[i]}}}\} \leftarrow \mathbf{\Xi}_{\hat{\mathcal{F}}_{\mathbf{NLp}}}\{\overrightarrow{\mathbf{h}}\}$
      $E_{stg} \leftarrow \frac{\|\{\overrightarrow{\mathbf{\hat{U}}^{[i]}}\} - \{\overrightarrow{\mathbf{\hat{U}}^{[i-1]}}\}\|}{\|\{\overrightarrow{\mathbf{\hat{U}}^{[i-1]}}\}\|}$
      $i \leftarrow i + 1$

  return $(\{\overrightarrow{\mathbf{\bar{u}_{\bar{t}}^{[i-1]}}}\})$
```

4.3 Results and Discussions

In this section, we test our method by comparing the solutions it generates to those produced by the traditional Newton-Raphson approach. We also evaluate its dependability, precision and restrictions.



(a)

(b)

(c)

Figure 4.3 Stagnation error with respect to the number of iterations as a function of: (a) α where $\bar{\Omega} = 0.9$ and $\zeta = 0.02$; (b) ζ where $\bar{\Omega} = 0.9$ and $\alpha = 0.05$; and (c) $\bar{\Omega}$ where $\alpha = 0.05$ and $\zeta = 0.02$.

4.3.1 Case Study: Harmonic Excitation with Null Dirichlet Boundary Conditions

Initially, we will present outcomes associated with zero boundary conditions at the fixed end and a harmonic excitation at the beam's free end.

HMH-Solver

The HMH-Solver takes the damping coefficient ζ, the non-linearity coefficient α, and the excitation frequency $\bar{\Omega}$ as inputs and produces the solution $\bar{u}(\bar{x}, \bar{t})$ corresponding to those parameters. If any of the three parameters is altered, the algorithm needs to be re-run and a new solution is generated.

The HMH-Solver always produces a convergent solution, however, examining Figure 4.3, it is evident that varying the parameters has an effect on the convergence rate and the number of iterations required. Figure 4.3a reveals that increasing the non-linearity coefficient α makes it more difficult to converge; with a higher α, more iterations are necessary and there is a noticeable oscillation in the stagnation error, as well as a slower convergence rate. The damping ratio ζ has an inverse effect on the convergence; with a higher ζ, fewer iterations are necessary and the convergence rate is faster. This is seen in Figure 4.3b. The excitation frequency $\bar{\Omega}$ has a distinct effect, as demonstrated in Figure 4.3c. The closer the excitation frequency is to the resonant frequency, the slower the convergence is, with more oscillations in the stagnation error and a greater number of iterations needed for convergence.

This leads us to the conclusion that the higher the non-linearity, the lower the damping, and the closer the excitation frequency is to the resonant frequency, the more difficult it is for the HMH-Solver to converge. This is because in this case, the vibration amplitudes are relatively high, resulting in higher non-linearities, thus the approximations of the fixed-point AFT method used suffer to accurately approximate the actual solution of the highly non-linear system. Additionally, the solution of equation (4.17) tends to give high values that become even higher with each iteration when the damping is low and frequencies are close to resonance. Due to that, it should be noted that when the normalized excitation frequencies are between $\bar{\Omega} = [0.95 - 1.20]$, a higher virtual damping (as explained in equation (4.16)) is needed to guarantee convergence, which increases the number of iterations. It is important to note that this was taken into account when generating all results. Finally, the sudden change of convergence rate noticed in some cases of Figure 4.3 is due to the relaxation technique used in the algorithm.

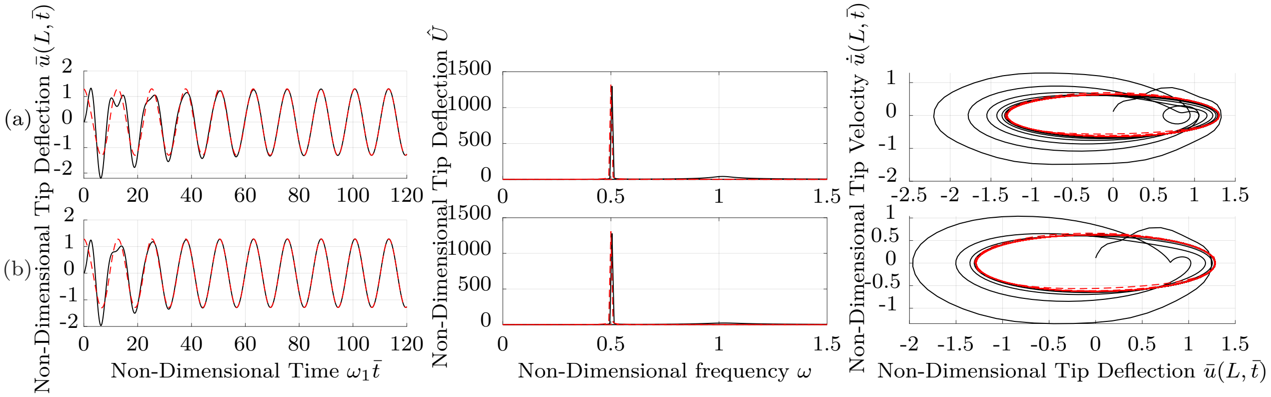
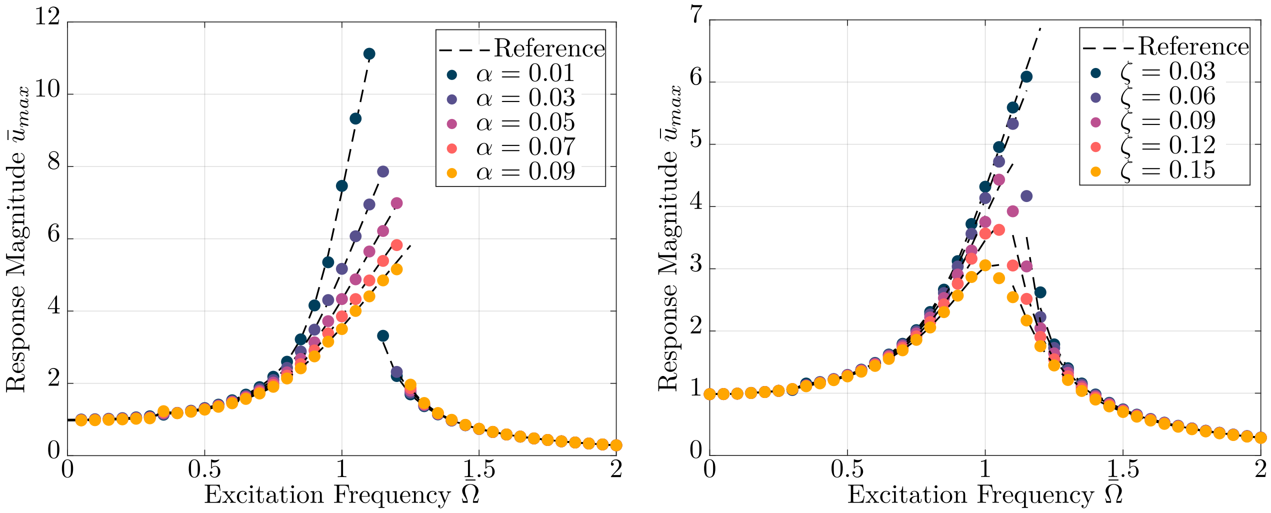


Figure 4.4 Comparison of the dynamics as simulated by the HMH-DEIM Solver and a conventional time-integration scheme with a Newton-Raphson Approach. Graphs on the left represent the time series of the non-dimensional displacement of the tip of the beam, graphs in the middle represent the $\mathcal{FFT}$ of the time series of the non-dimensional displacement of the tip of the beam, and graphs on the right represent the phase plot showing the relationship between the non-dimensional velocity and the non-dimensional displacement of the tip of the beam when $\bar{\Omega} = 0.5$, and (a) is a scenario used to create the DEIM approximating bases with $\zeta = 0.0504$ and $\alpha = 0.0305$, and (b) is a scenario used to test the DEIM approximating bases with $\zeta = 0.09$ and $\alpha = 0.05$. The red dashed line - - - - - is the solution of the HMH-DEIM Solver, and the black solid line ——— is the Newton-Raphson Solution. Notice how the HMH-DEIM Solver successfully captures the limit cycle but fails to reproduce the transients.

HMH-DEIM Solver

The HMH-Solver has a clear advantage in terms of computation speed when compared to traditional methods such as Newton-Raphson, which is demonstrated further in this section. Introducing DEIM made it even faster as described in Section 4.2.2. This eliminated the need to perform one $\mathcal{FFT}$ and one $\mathcal{IFFT}$ at each iteration of the solver.

The use of DEIM does not compromise the accuracy of the solution. As seen in the left-hand side of Figure 4.4, graph (a) is a scenario used to create the DEIM approximation bases, while the graph (b) is a scenario that was interpolated using the DEIM approximating bases. The black line is the solution from a traditional Newton-Raphson solver, which is in agreement with the red dashed line in the steady-state limit-cycle regime in terms of amplitude and frequency. However, the HMH-DEIM solver fails to capture the transient regime, which is expected since an $\mathcal{FFT}$ approach is used, resulting in the loss of transient information. The $\mathcal{FFT}$ of the time series is shown in the middle of Figure 4.4. The HMH-DEIM successfully captures the amplitude and the frequency of the response in the steady-state regime. The right-hand side graphs in Figure 4.4 are phase plots demonstrating the relationship between the non-dimensional velocity and the non-dimensional displacement of the tip of the beam for the same two scenarios.



(a)

(b)

Figure 4.5 Steady-state amplitude of the non-dimensional displacement of the tip of the beam. (a) Solution for multiple values of α when $\zeta = 0.02$. (b) Solution for multiple values of ζ when $\alpha = 0.05$.

Figures 4.5a and 4.5b are the results of running the HMH-DEIM solver with various damping and non-linearity coefficient values and excitation frequencies, and then extracting the steady-state amplitude of the non-dimensional displacement of the tip of the beam in each case. The black dashed line in the figures illustrates the solution obtained with the Newton-Raphson solver. The HMH-DEIM solver produces reliable results that are in agreement with the reference solution. It is noted that the accuracy decreases slightly for frequencies close to resonance, for the same previously discussed reasons. Near resonant frequencies, an observed average error escalates to 6.3% in the HMH-DEIM results compared to the Newton-Raphson results, in contrast to a typical error of 0.52% for other frequencies. It is also noticed that, at the point of discontinuity, if multiple solutions exist, the solver tends to converge to the lower branch. This is due to the fact that simulations for any excitation frequency start with the same initial value of zero beam deflection.

sPGD Solver

To generate a surrogate model that can provide real-time predictions of the parametrized dynamic response using sPGD, each discrete value of the excitation frequency is considered as a separate problem. For each of these problems, a latin hypercube is used to generate a sparse dataset of parameters, with α ranging from 0.01 to 0.09 and ζ from 0.01 to 0.15. The HMH-DEIM solver is used to solve for the sparse dataset at each value of the excitation frequency.

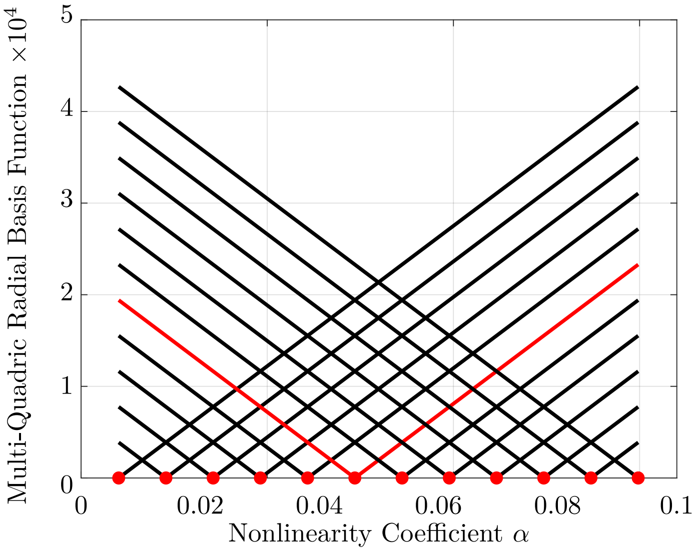


Figure 4.6 Multiquadric radial basis functions used to interpolate the response of the system in terms of the parameter α. The red circles (●) represent control points.

This data is then used as an input to the sPGD algorithm to create a parametric solution of the problem in terms of α and ζ. The sPGD surrogate model can interpolate the peak deflection at the beam's tip for any given values of ζ and α within the established ranges, but it cannot extrapolate beyond these specified intervals. For frequencies away from resonance, i.e. below 0.95 and above 1.25, the sPGD hyper-parameters selected for the interpolation functions of equation (4.32) were $h = 7$ and 12 control points for each parameter. Figure 4.6 illustrates the Multi-Quadric Radial (MQR) Basis functions aligned with the specified hyper-parameters for the parameter α, serving as the interpolation functions within the sPGD algorithm. These functions are depicted as described in equation (4.32). A single function is defined for each control point k across the full range of the parameter. To demonstrate, we have displayed one such function, corresponding to the sixth control point, highlighted in red in Figure 4.6. The sharpness of these functions is governed by the parameter h. A similar figure can be created for these functions with respect to the parameter ζ. In theory, the construction of the surrogate model requires 2^p data points, where p represents the number of parameters, translating to 4 data points for our scenario. Nonetheless, we opted for a denser grid to enhance the accuracy of the interpolation, benefiting from the rapid performance of our HMH-DEIM solver. We needed 300 sparse data points to obtain good approximations for the full parametric space, because it is noticed in Figures 4.5a and 4.5b that for those frequencies, changing the parameters ζ and α barely affects the value of the maximum deflection of the tip of the beam. On the other hand, for frequencies close to resonance; i.e. between 0.95 and 1.25, a dataset of 500 points, $h = 0.5$ and 25 control points are needed, as the function to be interpolated becomes more intricate due to the fact that a slight alteration in α and/or ζ has a greater influence on the response at those frequencies than frequencies further away from resonance.

The figures 4.7a and 4.7b illustrate the accuracy of sPGD in approximating the solution for different values of α and ζ respectively. It is evident that sPGD is able to predict the solution for a variety of parameter sets, even if they were not used in the training process. However, the accuracy of the prediction decreases at frequencies close to resonance, particularly at points of discontinuity. This is because the relationship between the parameters is more complex in these cases, as a small change in one parameter can cause the solution to jump from the upper branch to the lower branch, or vice versa. To address this issue, one can increase the number of sparse data points used by the sPGD algorithm, adjust the hyper-parameters, or use different interpolation functions. Alternative methods for data generation are discussed by Guilhaumon et al. (2024) and Torregrosa et al. (2024), which describe intelligent sampling strategies focused on selecting data points strategically to enhance the accuracy of the surrogate model at critical points of interest. For instance, Guilhaumon et al. (2024) use the Fisher information matrix, which quantifies the information that an observable random variable provides with respect to an unspecified parameter. In formal terms, it represents the score's variance, or the expected value of the observed information. On the other hand, Torregrosa et al. (2024) propose a physics-based Active Learning method that progressively targets data points that meet specific goals during the sampling of training data. Consequently, this results in a surrogate model that is precise in regions with potentially significant data points, and less detailed in areas where the data points are irrelevant to the study.

To further explain the concept of interpolating the parametrized solution using sPGD, we consider a single excitation frequency. Figure 4.8a displays the interpolated values of the maximum deflection of the beam's tip for a mesh of 300×300 scenarios. The circles represent the sparse data points used to train the sPGD algorithm. This figure illustrates how solving for those sparse data points using our solver and using them as inputs to the sPGD algorithm enables us to obtain the parametrized solution for an infinite number of scenarios which can be found quickly and efficiently. Figure 4.8b shows the discrepancy between the values calculated using our HMH-DEIM solver and the values predicted by sPGD for the same dense mesh. We observe that in the areas where there is a noticeable gap between the sparse data points the error increases. Nevertheless, the error values are acceptable. Additionally, when this model is combined with experimental data from a physical entity these error values can be further reduced using correction models. This will be investigated in future research.

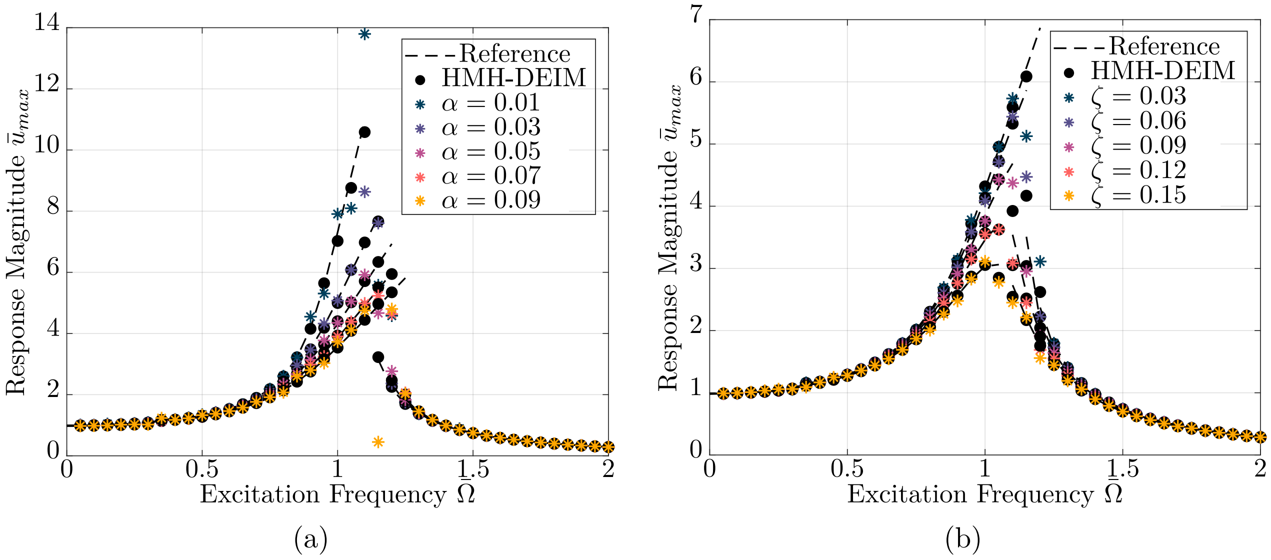


Figure 4.7 Comparison of the steady-state amplitude of the non-dimensional displacement of the tip of the beam approximated by sPGD to that calculated using the HMH-DEIM solver for various scenarios. The black dashed line - - - - - is the Newton-Raphson Solution, the black circles (•) represent the solution generated by the HMH-DEIM Solver and the colored stars ($*$) represent the solution predicted by sPGD. (a) Solution for multiple values of α when $\zeta = 0.02$. (b) Solution for multiple values of ζ when $\alpha = 0.05$.

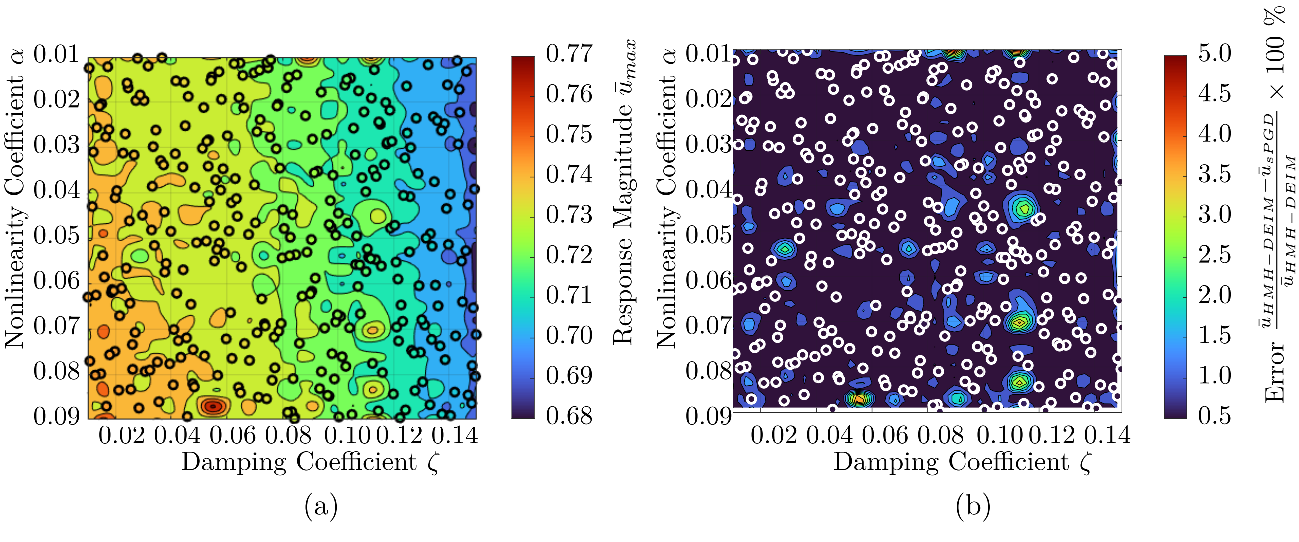


Figure 4.8 (a) Steady-state amplitude of the non-dimensional displacement of the tip of the beam for the full domain of parameters when $\bar{\Omega} = 1.5$. The circles represent the sparse dataset used to create the sPGD interpolation functions. (b) Error in the sPGD prediction for the full domain of parameters when $\bar{\Omega} = 1.5$. The circles represent the sparse dataset used to create the sPGD interpolation functions.

Table 4.1 Average simulation time of one scenario using the different discussed solvers.

Solver	**Simulation Time (s)**							
	Offline	**Online**	**Offline**	**Online**	**Offline**	**Online**	**Offline**	**Online**
Mesh Size	**5 Elements**		**10 Elements**		**20 Elements**		**40 Elements**	
Newton-Raphson	0.31		0.46		0.78		2.75	
HMH Solver	0.17		0.22		0.63		1.01	
HMH-DEIM Solver	5.88	0.07	7.68	0.12	17.00	0.34	32.07	0.67
Data Generation	12.02		23.47		53.62		87.93	
s-PGD	0.84	1.13×10^{-5}	1.01	1.02×10^{-5}	0.83	0.90×10^{-5}	0.98	0.87×10^{-5}
Mesh Size	**60 Elements**		**80 Elements**		**100 Elements**		**500 Elements**	
Newton-Raphson	3.92		8.30		12.09		286.49	
HMH Solver	1.61		2.26		3.06		28.4	
HMH-DEIM Solver	46.98	1.03	63.08	1.44	85.65	1.93	1144.93	20.72
Data Generation	137.25		189.78		203.70		2350.22	
s-PGD	1.02	0.99×10^{-5}	0.95	0.99×10^{-5}	0.96	1.22×10^{-5}	1.03	0.99×10^{-5}

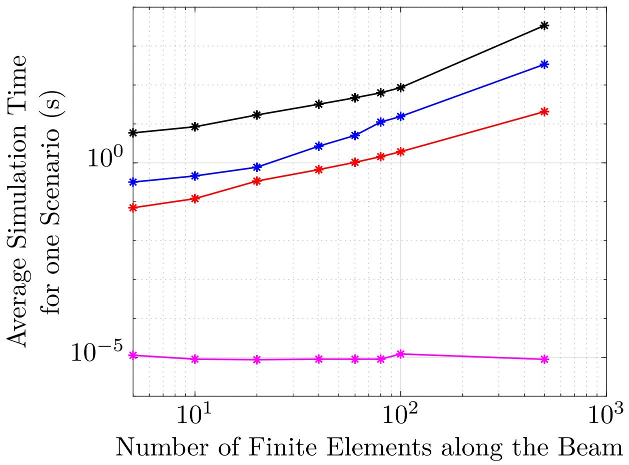


Figure 4.9 Average Simulation Time of one Scenario as a function of the Number of Finite Elements along the Beam. The blue stars (∗) represent the Newton-Raphson Solution, the black stars (∗) represent the Offline Stage of the HMH-DEIM Solver, the red stars (∗) represent the Online Stage of the HMH-DEIM Solver, and the magenta stars (∗) represent the Online Stage of the sPGD.

Table 4.1 shows that the average simulation time for one scenario is significantly reduced when the parametrized solution library is computed and stored for each value of the excitation frequency. For example, for a mesh of 10 elements, the time required to obtain any solution is approximately 1.02×10^{-5} s, compared to 1.2×10^{-1} s when using the HMH-DEIM solver alone, and 4.6×10^{-1} s when using Newton Raphson. This implies that by investing in the offline calculation, which is done once and for all, we can save more than 99% of the calculation time for one scenario. The total offline calculation time is the sum of the offline time of the HMH-DEIM Solver, the time needed for data generation, and the offline time of the sPGD algorithm. It should be noted that all calculations were conducted on a laptop computer with an Intel® Core™ i9-10885H CPU, 32 GB of RAM, and a 64-bit Windows Operating System.

As the order of the system increases, the importance of using a reduced-order model such as sPGD becomes more evident. For instance, when examining the effect of increasing the number of finite elements used to discretize a beam on the simulation time using Newton-Raphson and the HMH-DEIM Solver, Figure 4.9 shows that the simulation time increases with a larger number of elements. However, with sPGD, the online time needed to calculate the response for one scenario remains the same regardless of the number of elements after the offline calculations are completed, making this method more advantageous for higher-order systems and promoting its application for monitoring or control actions of dynamic systems, whether in 1D, 2D, or 3D.

4.3.2 Case Study: Base Excitation

To further illustrate the effectiveness of the proposed method, we applied it to analyze the deflection of the same cantilevered beam, but this time we eliminated the harmonic excitation and introduced a base excitation as an imposed Dirichlet boundary condition:

$$\bar{u}(0, \bar{t}) = U_b \cos(\bar{\Omega}\bar{t}), \tag{4.33}$$

where U_b represents the amplitude of the base excitation, expressed as a percentage of the total length of the beam. U_b will be treated as a third parameter in the system. This boundary condition imposes a Dirichlet force used in the algorithms to determine the tip deflection of the beam under these conditions.

Here, the distinction lies in the scenario of base excitation, where the solution is expressed as the sum of a quasi-static solution and a harmonic-modal hybrid solution, as detailed in Section 4.2.

To demonstrate the effectiveness of the method, a wider range of excitation frequencies was used in order to see the results when the excitation triggers wave propagation at elevated frequencies. We employed the HMH-DEIM solver to analyze a single scenario to determine if the technique can effectively detect wave propagation. In this case, the parameters set are $\alpha = 0.03$, $\zeta = 0.03$, $U_b = 0.1$, and $\bar{\Omega} = 5$. Figure 4.10 illustrates the deflection of the beam as a function of its spatial coordinate $\bar{x}$ over several time steps within a specific duration. The illustration confirms that the beam undergoes wave propagation, effectively captured by the proposed technique. This is evident from the absence of a zero-displacement node at $\bar{x} = 7$, and the intersection of the lines at $\bar{x} = 9$ and $\bar{x} = 10$, two signs of wave propagation.

Then, to develop the surrogate model, a latin hypercube is employed to produce a sparse dataset of 3 parameters at value of the excitation frequency, where α varies from 0.01 to 0.09, ζ from 0.02 to 0.15, and U_b from 0.05 to 0.2. The sparse dataset is solved using the HMH-DEIM solver described in the first case study. Subsequently, this dataset serves as input for the sPGD algorithm, which constructs the parametric solution of the problem based on α, ζ, and U_b.

For frequencies away from resonance, i.e. below 0.95 and above 1.25, and below 5.5 and above 7.0, the chosen sPGD hyper-parameters for the interpolation functions included $h = 15$ and 15 control points per parameter (see Section 4.2.3). To accurately approximate the entire parametric space, 300 data points were produced for each value of excitation frequency. On

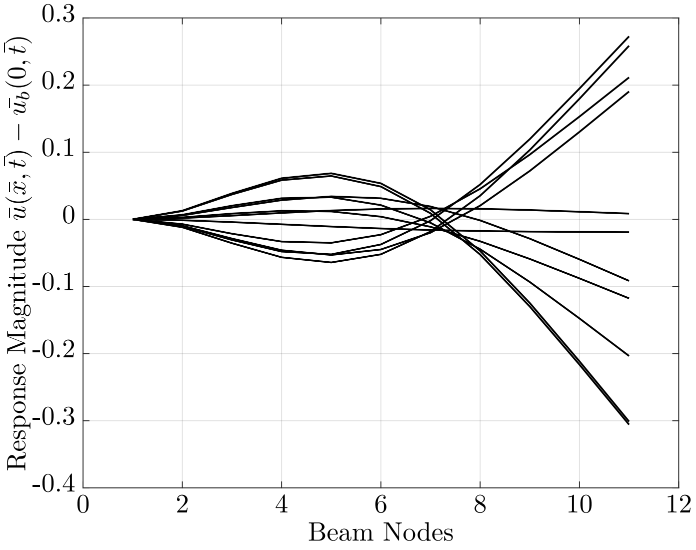


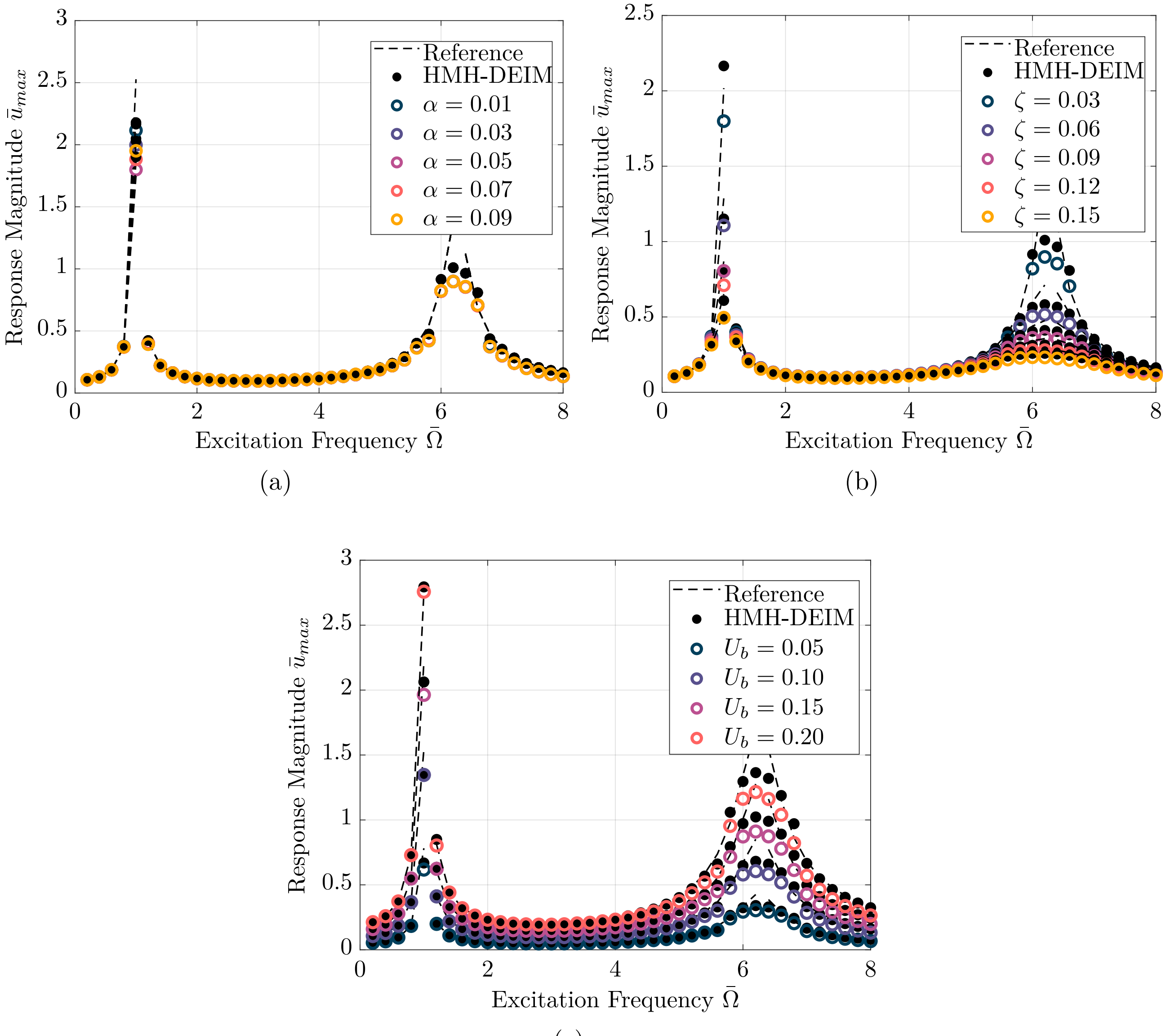
Figure 4.10 HMH-DEIM solution of the deflection of the beam across its spatial coordinates at multiple time steps within one period of motion, where each line represents the position of the beam nodes at a given time step. This scenario corresponds to $\alpha = 0.03$, $\zeta = 0.03$, $U_b = 0.1$, and $\bar{\Omega} = 5$.

the other hand, for frequencies close to resonance, i.e. between 0.95 and 1.25 and between 5.5 and 7.0, a dataset of 1500 points, $h = 7$, and 30 control points are needed. The results are shown in Figure 4.11.

As demonstrated in Figure 4.11, the approach continues to resolve the problem effectively, even under the imposition of a Dirichlet boundary condition. Furthermore, it remains operational when the response depends on more than two parameters.

4.4 Conclusion and Future Work

This paper presents a novel *a priori*, intrusive PGD approach that combines the Harmonic-Modal Hybrid solution in the frequency domain with the DEIM to create a fast global solver that gives a reliable low-rank approximation of the system's solution. This solver is used to generate a sparse dataset offline, which is then compressed to create a surrogate model that can provide real-time predictions of the parametrized dynamic response using sPGD. To demonstrate its effectiveness, the method was applied to a multiple degrees-of-freedom variation of the Duffing oscillator, i.e., a cantilevered beam with a non-linear cubic spring attached to its end. The library of solutions was created for the steady-state amplitude of the tip of the beam as a function of the non-linearity and damping coefficients of the system. The method was found to be reliable and to greatly reduce the computational time



(a)

(b)

(c)

Figure 4.11 Comparison of the steady-state amplitude of the non-dimensional displacement of the tip of the beam approximated by sPGD to that calculated using the HMH-DEIM solver for various scenarios. The black dashed line - - - - - is the Newton-Raphson Solution, the black circles (●) represent the solution generated by the HMH-DEIM Solver and the colored rings (o) represent the solution predicted by sPGD. (a) Solution for multiple values of α when $\zeta = 0.03$ and $U_b = 0.1$. (b) Solution for multiple values of ζ when $\alpha = 0.05$ and $U_b = 0.1$. (c) Solution for multiple values of U_b when $\alpha = 0.03$ and $\zeta = 0.05$.

once the offline calculation is completed. In our first example, we have taken into account two parameters, but it is important to recognize that this methodology can be extended to estimate values dependent on multiple parameters as demonstrated in the case study of base excitation. However, to construct the surrogate model via sPGD in that case, an increased number of data points is required. Furthermore, it is important to acknowledge that the entire HMH-DEIM approach relies on discrete excitation frequency values, a constraint imposed by employing DEIM approximating bases in the frequency domain. Nonetheless, parametrization of the excitation frequency remains feasible if the data required for building the sPGD surrogate model is solely produced using the HMH solver, that is, excluding the use of DEIM.

It is clear why the offline-online approach is preferable to traditional simulations that are done in a single step. The offline computation is done efficiently with a reduced model, which yields accurate results with a much lower computational cost. This allows the online stage to be performed on mobile or light computer platforms, such as smartphones and tablets, which is a great advantage for some applications. The offline stage of our method requires more work than the online stage, but the offline stage is done only once to create a solution handbook, also known as a Computational *Vademecum*, as defined by Chinesta et al. (2013) and Aguilera (2017). This reflects the approach used in Machine Learning Techniques where the model undergoes an extensive one-time training (which might take hours or days), but upon completion, it enables the rapid generation of solutions, similar to our online phase. Therefore, the duration of offline computations is not considered when assessing the efficiency of these methods compared to other traditional high-fidelity techniques, since the model, once developed, is usable indefinitely.

This method can be extended to other engineering systems with similar equations of motion, such as rotating machines with non-linear bearing forces at their ends. Thus, the approach and process can be used to solve for various systems, though the derived Reduced-Order Model will be tailored to the specific problem at hand. A parametrized library of solutions is seen as a key component of virtual twins, which, when supplemented with real-time experimental data, can create a powerful hybrid twin. This will be explored in future research. Possible enhancements may involve the addition of more parameters such as the location where linear and non-linear forces are applied, geometric and material parameters, time, etc. to generate richer models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI in scientific writing

During the preparation of this work the authors used *"Writefull"* version 2024.3.0 in order to rephrase certain parts and check grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

The authors would like to acknowledge the technical and financial support of Hydro-Québec and Maya HTT, and the financial support of NSERC, InovÉÉ, IVADO, and Mitacs.

CHAPTER 5 ARTICLE 2: PARAMETER IDENTIFICATION OF A NONLINEAR VERTICAL AXIS ROTATING MACHINE THROUGH REDUCED ORDER MODELING AND DATA ASSIMILATION

This chapter presents the article, *"Parameter identification of a nonlinear vertical axis rotating machine through reduced order modeling and data assimilation"*, published in Nonlinear Dynamics (DOI: `https://doi.org/10.1007/s11071-025-11032-3`). In this article, we accomplish objectives 2 and 3. The initial section focuses on developing a mathematical representation of a VARM, which includes a slender shaft linked to two bearings at its ends: a fixed lower bearing and an upper bearing attached to four perpendicular springs, enabling movement in a plane perpendicular to the rotation axis. Implementing the method from the article presented in Chapter 4, we constructed a parametrized surrogate model of the VARM based on its unknown parameters. Subsequently, experimental data from the VARM setup was gathered to estimate these unknown parameters, as well as the linear and non-linear terms to approximate the unknown bearing forces, thereby creating a digital twin of the VARM.

The thesis author applied the approach detailed in Chapter 4 to develop a parameterized model of the VARM, then integrated the developed model with the LM optimization method to determine the unknown parameters of the VARM. Sebastian Rodriguez contributed the general LM optimization algorithm that was adapted by the author of the thesis to achieve the objectives of this paper. Ludivine Moyne and Souheil Serroud designed and prepared the experimental setup of the VARM for data acquisition, adjusting it as needed to maintain high data quality. Experimental tests were done by the author of the thesis who also drafted the entire initial version of the paper, which was then reviewed by the co-authors.

The citation style was altered to fit the thesis format. A comprehensive bibliography is provided at the end of this document.

Date of Submission: 13 December 2024

Authors

Sima Rishmawi[1,*], Ludivine Moyne[1], Souheil Serroud[1], Sebastian Rodriguez[3], Francisco Chinesta[3], Oguzhan Tuysuz[2], Frédérick P. Gosselin[1]

[1] Multi-scale Mechanics Laboratory (LM2), École Polytechnique de Montréal, 2500 Chemin

de Polytechnique, Montréal, H3T 1J4, QC, Canada

[2] Virtual Manufacturing Research Laboratory, École Polytechnique de Montréal, 2500 Chemin de Polytechnique, Montréal, H3T 1J4, QC, Canada

[3] PIMM Laboratory, Arts et Métiers Institute of Technology, CNRS, Cnma, HESAM Université, 151 Boulevard de l'Hôpital, 75013, Paris France

CRediT authorship contribution statement

Sima Rishmawi: Writing - original draft, Validation, Software, Resources, Methodology, Formal analysis, Conceptualization, Data curation, Investigation. **Ludivine Moyne**: Building the experimental rig, Carrying out experiments, Data collection. **Souheil Serroud**: Building the experimental rig. **Sebastian Rodriguez**: Writing - review and editing, Software, Methodology. **Francisco Chinesta**: Writing - review and editing, Conceptualization of the study. **Oguzhan Tuysuz**: Writing - review and editing, Supervision. **Frédérick P. Gosselin**: Writing - review and editing, Project Administration, Conceptualization of the study, Funding acquisition.

Abstract

One challenge in modeling nonlinear dynamic systems involves the uncertainty associated with certain parameters that cannot be directly measured or estimated, along with the complexity of incorporating all relevant physical phenomena into a mathematical model without increasing computational cost. A hybrid twin represents an advanced modeling approach that combines the system's physics-based mathematical model with the empirical data collected from the real-world system, using data assimilation. This strategy enhances the accuracy and reliability of both estimating the system's unknown parameters and predicting its overall behavior. Further improvements are achieved by using a reduced-order model, which significantly lowers the computational burden of the entire procedure. In this study, we construct a surrogate model for a Vertical Axis Rotating Machine (VARM) by deploying sparse Proper Generalized Decomposition (sPGD). The model is parametrized in terms of the machine's unidentified parameters, and we apply the Harmonic-Modal Hybrid (HMH) Frequency Approach to solve for sparse scenarios creating the library of solutions. This is then combined with the Levenberg-Marquardt optimization technique to identify the unknown parameters using the measured shaft displacements of an experimental rig of the machine. The results demonstrate that this method is effective for parameter estimation in complex nonlinear systems and allows for fast computations, whether the unknown force function is specified

explicitly or presumed. Precisely estimating a system's parameters can serve as a crucial indicator for scheduling maintenance or predicting failures.

5.1 Introduction

Grey-box modeling combines elements of white-box models, which are formulated with detailed insights into system physics, and black-box models, which rely entirely on data. The grey-box method strikes a balance between precision and flexibility (Wagg et al., 2020; Brunton et al., 2016). It is especially effective when the differential equations that govern the behavior of the system are known, but some parameters in these equations are difficult to identify; then measured data can be used to estimate these unknown features. According to this concept, a grey-box model aligns with the Hybrid Twins framework presented by Chinesta et al. (2019). This framework enhances a physics-based model by incorporating empirical data to create a virtual equivalent of a real or anticipated physical system.

Rotating machines are an example of systems with incomplete knowledge. Typically, these machines experience vibrations caused by both the shaft's and supports' flexibility (Ishida and Yamamoto, 2012; Friswell et al., 2010; Tiwarti, 2006). Shaft flexibility is usually captured using models for bending and torsion of beams, while modeling the supports proves more challenging, due to the diverse physical phenomena involved, including bearing physics with hydrodynamic lubrication, rolling components, thermal influences, and mounting compliance. The nonlinearly coupled multi-physical characteristics of the supports necessitate simulations that depend on parameters that are challenging to quantify, thereby making simulations difficult to achieve. This is particularly true for machines with a vertical axis of rotation. Conversely, in horizontal axis machines, the calculation of bearing reaction forces is based on the static radial loads due to the rotor's dead weight (Nasselqvist et al., 2014; Chen and Gunter, 2007).

Extensive research has explored how bearing forces affect rotor dynamics. Bearings, often seen as localized rotor-stator interactions with specific damping and stiffness, generate reaction forces. These properties are crucial for modeling the shaft forces, but are influenced by various factors and are typically nonlinear in vertical machines, making them difficult to quantify (Friswell et al., 2010; Chen and Gunter, 2007; Turaga et al., 1999).

Researchers have developed three primary categories of approaches to evaluate bearing forces and coefficients in rotating machinery. The first category involves the use of sensor data or experimental measurements to determine unknown bearing parameters in horizontal and vertical machines (Gustavsson et al., 2005; Diamond et al., 2009, 2011). The second approach is based on solving the Navier-Stokes equations. It is used to calculate the coefficients of journal bearings. This is generally accomplished using computer software programs, such as the one provided in Rotordynamic-Seal Research (2015). The third approach, which is commonly used, models the bearing forces as functions of the lateral displacements and velocities of the shaft in the plane perpendicular to its axis of rotation involving parameters that require identification (Shi et al., 2012).

When addressing a differential equation with one or more unknown parameters, three strategies are available: (1) A cautious approach involves determining the parameter's value before solving the equation, for instance, through sensor measurements if feasible. (2) Solving the equation repeatedly with varying parameter values, and choosing the one providing the closest approximation of the response. (3) Solving the equation comprehensively for an entire range of the unknown parameter(s), with the range being refined according to the problem's specifications, resulting in a parametrized solution (Ghnatios et al., 2012).

Proper Generalized Decomposition (PGD) is an effective method for crafting a parametrized solution to differential equations. Initially, PGD was introduced as an *a priori* intrusive Model Order Reduction (MOR) projection-based technique that yields an economical low-rank approximation of the solution to a global space-time problem. The notion of a parametrized solution emerged when Chinesta et al. expanded PGD to incorporate system parameters alongside space and time (Chinesta et al., 2010a; Chinesta and Cueto, 2014; Ammar et al., 2006). PGD aims at achieving a distinct representation of an unknown field referred to as a finite sum decomposition. In the context of problems involving n_p parameters, each summation term (enrichment step) within the PGD representation consists of a set of n_p one-dimensional functions, which are simpler to resolve compared to dealing with a single function of n_p dimensions.

Another way to apply PGD is through an *a posteriori*, non-intrusive method. This involves performing an offline calculation with a high-fidelity model and then using the sparse Proper Generalized Decomposition (sPGD) technique to compress the results into a parametrized reduced order model (ROM) (Rodriguez et al., 2023b). The sPGD technique was initially introduced by Ibáñez et al. (2018) and Sancarlos et al. (2021c). It offers benefits such as handling unstructured datasets, maintaining robustness in high-dimensional spaces, and not requiring an extensive dataset for convergence.

Rishmawi et al. (2024) introduced an alternative technique for conducting the offline sPGD calculations. This approach leverages a nonlinear global space-frequency solver to iteratively construct a low-rank solution for a nonlinear differential equation, employing the Harmonic-Modal Hybrid (HMH) method as detailed by Germoso et al. (2020). Implementing this method during the offline calculation stage of the sPGD ensures precise outcomes while significantly reducing the computational cost.

In the realm of data assimilation methods, Static Data-Driven Application Systems (SDDAS) are characterized by models that gather data *a priori* through sensors from a physical source; hence, data is not continuously fed into the system for real-time updates (Chinesta et al., 2019; Knight, 2003). This collected data is subsequently employed to solve an optimization problem, where it serves to estimate unknown parameter values within a differential equation while minimizing the disparity between the observed data and the simulated response. The Levenberg-Marquardt (LM) algorithm serves as an optimization method to solve such problems. Originating from early 1960s, LM was designed to address nonlinear least square problems. It merges the advantageous features of the Gradient-Descent and Gauss-Newton optimization methods, specifically convergence and rapidity. For comprehensive derivations of these techniques and their integration within the LM algorithm, the reader is advised to refer to the work of Gavin (2020), where a suggested numerical implementation with codes can be found. It is advisable to integrate LM with sPGD because the LM algorithm's costliest step is the computation of the Jacobian matrix, a task that is performed automatically by the sPGD algorithm, storing the Jacobians within the sPGD library, thereby accelerating the optimization procedure. The most relevant example in the literature involves the integration of intrusive PGD with the LM optimization method to determine the optimal parameters for a 2D thermal problem presented by Ghnatios et al. (2012). This method was broadened to incorporate sPGD in the work of Rodriguez et al. (2023b), where it was used to create a hybrid twin of a resin transfer molding Resin Transfer Molding (RTM) process to ensure correct filling.

Here we introduce a parametrized ROM for a nonlinear Vertical Axis Rotating Machine (VARM) using a surrogate sPGD model in conjunction with the LM optimization method for inverse analysis. This approach is used to identify unknown parameters essential to characterizing the nonlinear bearing forces on the VARM shaft within a grey-box hybrid twin framework. To accelerate the offline computations needed for constructing the sPGD solution library, we employ the HMH global nonlinear space-frequency solver. While all techniques applied in this method have been previously developed and validated individually, they have never been combined in this manner to create a dependable hybrid twin for such a system.

This article is organized as follows: Section 5.2 outlines the overall framework of the developed methodology by describing the steps where each numerical technique is employed, and explaining how they are linked to create a hybrid twin. Section 5.3 presents the newly designed and tested vertical axis rotating machine. The findings of this research are presented in Section 5.4. Lastly, the concluding remarks and suggestions for future research directions are provided in Section 5.5.

5.2 General Methodology

Figure 5.1 illustrates the three primary stages of the proposed method to develop a hybrid twin of a vibrating system using a grey-box framework. The first step involves determining the unknown parameters and estimating a plausible range for each of them. Then a set of sparse parameter scenarios is generated using a Latin hypercube. Following this, the HMH method is employed to calculate the system's response as a function of the unknown parameters for each one of those scenarios. By executing the solver multiple times with various sets of unknown parameters within their expected ranges, we generate a sparse dataset. This dataset is subsequently used as input for an sPGD algorithm, resulting in a library of solutions for the parametrized problem. The second step entails recording data from the physical entity of the system. Subsequently, the final step consists of leveraging the collected data to derive the optimal values for the unknown parameters that reduce the discrepancy between the observed and simulated shaft displacements. This is achieved by employing the LM optimization technique to develop a comprehensive model of the system at hand. The sPGD solution library accelerates the LM optimization technique considerably, as it can promptly provide the solution for any set of parameters which is required in these iterative processes, thereby obviating the need for time integration and other computationally intensive methods. The thoroughly developed mathematical model can be used for future system monitoring and control, as well as maintenance planning and fault detection. Naturally, the approach should be customized to meet the requirements of the specific models under consideration.

5.2.1 White-Box Model

This part of the methodology describes the development of a surrogate model based on pre-computed scenarios aimed at rapidly predicting the parameterized dynamic response of a system. Initially, a Latin hypercube sampling generates n_s random scenarios, each consisting of a specific combination of n_p unknown parameter values within their defined ranges as follows:

$$\mathbf{P}_{LHC} = \begin{bmatrix} p_1^1 & p_2^1 & \cdots & p_{n_p}^1 \\ p_1^2 & p_2^2 & \cdots & p_{n_p}^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^{n_s} & p_2^{n_s} & \cdots & p_{n_p}^{n_s} \end{bmatrix}. \tag{5.1}$$

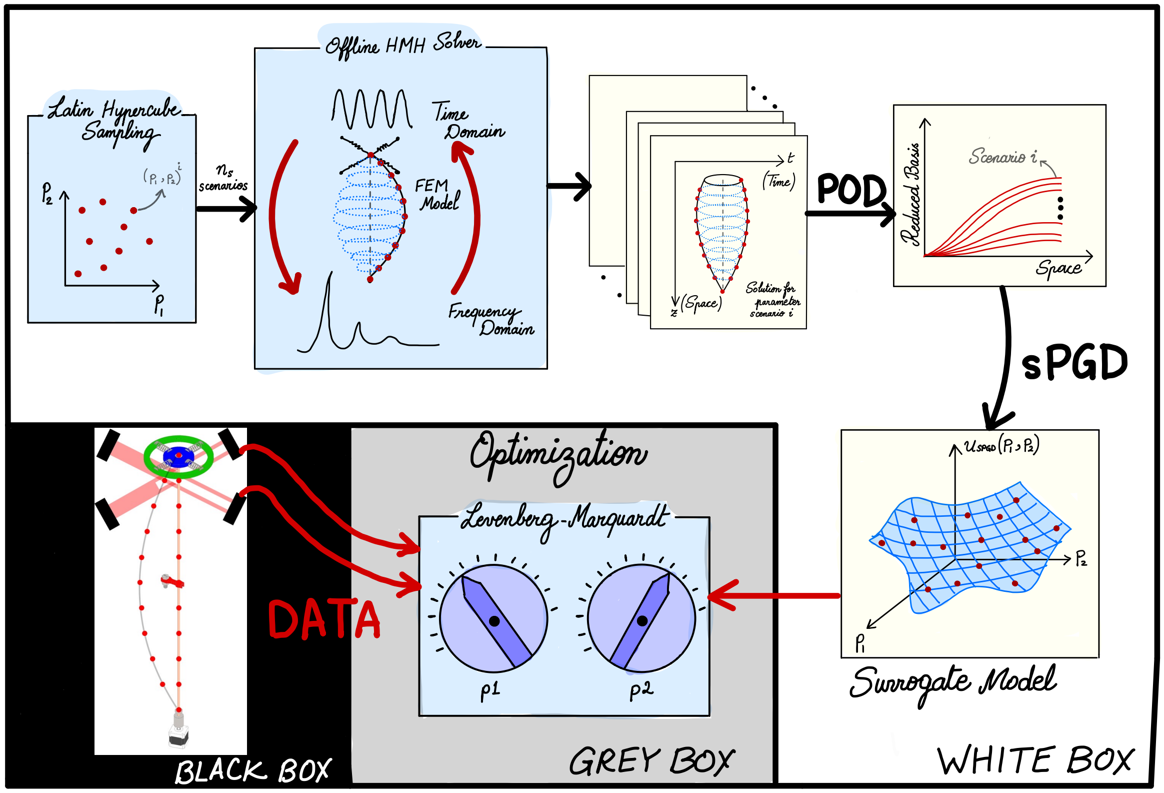


Figure 5.1 A diagram reflecting the overall outlined methodology, detailing each step involved in the white, black, and grey-box models, along with their respective inputs and outputs, illustrating their interconnection in forming the comprehensive model. To enhance clarity and visibility, the example focuses on optimizing two unknown parameters, though this approach can be expanded to optimize several parameters.

Here, p_i^j denotes the i^{th} parameter within the j^{th} scenario. Each scenario in the matrix $\mathbf{P}_{LHC}$, generates a distinct system response that must be calculated explicitly for every scenario. To achieve reliable and rapid computations, we employ the HMH solver Rishmawi et al. (2024) to solve for those scenarios.

The iterative HMH solver relies on modal basis analysis in the frequency domain to address a nonlinear Partial Differential Equation (PDE) without requiring a time integration scheme. Its core concept involves employing the fixed-point Alternating Frequency Time method (AFT) (Cameron and Griffin, 1989), in which the Fast Fourier Transform ($\mathcal{FFT}$) is used on the linear segment of the equation of motion to convert it into the frequency domain, where the initial response is computed in the first iteration. This response is then converted back to the time domain to determine the nonlinear force. In subsequent iterations, the nonlinear force component is calculated using the previously obtained response, represented as a numerical vector. This vector undergoes $\mathcal{FFT}$ again, facilitating the response calculation for the next iteration. It is essential to note that in the frequency domain, the solution is explicit. This implies that when employing modal basis analysis, the equations are entirely decoupled in the frequency domain.

Considering an arbitrary nonlinear differential equation of the form:

$$\mathbf{M}\{\overrightarrow{\ddot{\mathbf{u}}}\} + \mathbf{C}\{\overrightarrow{\dot{\mathbf{u}}}\} + \mathbf{K}\{\overrightarrow{\mathbf{u}}\} - \{\overrightarrow{\mathbf{R}(\{\overrightarrow{\mathbf{u}}\})}\} = \{\overrightarrow{\mathbf{f}}\}, \tag{5.2}$$

where the notation $\{\ddot{}\}$ and $\{\dot{}\}$ represent the second and first derivatives with respect to time, respectively, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are general mass, damping, and stiffness matrices of an arbitrary vibration system, respectively, and $\{\overrightarrow{\mathbf{R}(\{\overrightarrow{\mathbf{u}}\})}\}$ is a nonlinear contribution that depends on $\{\overrightarrow{\mathbf{u}}\}$, then the aforementioned approach is based on calculating the nonlinear term from the solution at the previous iteration and considering it as an extra loading as follows:

$$\mathbf{M}\{\overrightarrow{\ddot{\mathbf{u}}}\} + \mathbf{C}\{\overrightarrow{\dot{\mathbf{u}}}\} + \mathbf{K}\{\overrightarrow{\mathbf{u}}\} = \{\overrightarrow{\mathbf{R}(\{\overrightarrow{\mathbf{u}^-}\})}\} + \{\overrightarrow{\mathbf{f}}\}, \tag{5.3}$$

where $\{\overrightarrow{\mathbf{u}^-}\}$ corresponds to the time dependent solution calculated at the previous iteration. Now, $\mathcal{FFT}$ can be directly applied to the right side as well as the left side, to find the solution in the frequency domain. This is further detailed in the works of Rishmawi et al. (2024), Germoso et al. (2020), and Quaranta et al. (2019).

Typically solving dynamic equations in terms of space and time, the HMH solver results in n_s solution matrices with rows corresponding to the degrees of freedom in space and columns corresponding to the time steps. Due to the complexity of creating a surrogate model based on those matrices using sPGD, as interpolation becomes too intricate, we reduce the rank of these matrices by projecting them onto a reduced basis derived from Proper Orthogonal Decomposition (POD) (Karhunen, 1946; Loeve, 1948). Thus, Singular Value Decomposition (SVD) is applied to a snapshot matrix formed by concatenating the n_s matrices obtained during the initial offline computation phase with the HMH solver. The snapshot matrix can be defined as:

$$\mathtt{U} = [\mathbf{U}_1^T, \cdots, \mathbf{U}_{n_s}^T], \tag{5.4}$$

where $\mathbf{U}_i$ is the solution matrix corresponding to scenario i having a dimension of $n_{DOF} \times n_t$, making the total dimension of $\mathtt{U}$ $n_t \times n_s n_{DOF}$. By applying SVD to the snapshot matrix $\mathtt{U}$, the first few temporal modes can be extracted to form a temporal reduced basis $\mathcal{R}_t$ with the dimension of $n_t \times m_d$, where n_t is the number of time steps and m_d denotes the number of selected modes. The temporal reduced basis $\mathcal{R}_t$ is used to project each complete solution matrix onto the reduced basis, as shown below:

$$\mathcal{U}_i = \mathbf{U}_i \mathcal{R}_t, \tag{5.5}$$

where $\mathcal{U}_i$ represents the reduced-order portrayal of $\mathbf{U}_i$ and has dimensions of $n_{DOF} \times m_d$ instead of $n_{DOF} \times n_t$, making it significantly easier to manage. The number of selected modes, m_d, is influenced by the level of correlation within the data set. A greater correlation results in fewer modes that need to be extracted.

The concluding phase of this stage involves developing a surrogate model with sPGD which offers a parameterized solution for the problem following the framework presented by Ibáñez et al. (2018). The matrix $\mathcal{U}$ can be represented as a function of the n_p unknown parameters according to the sPGD formulation below:

$$\mathcal{U}(p_1, p_2, \cdots, p_{n_p}) = \sum_{k=1}^{M} F_1^k(p_1) F_2^k(p_2) \cdots F_{n_p}^k(p_{n_p}), \tag{5.6}$$

where M is the number of enrichment steps needed to obtain an acceptable approximation of $\mathcal{U}$ using sPGD. So by defining the functions $F_j^k(\eta)$ we can obtain the solution $\mathcal{U}$ for any combination of the n_p parameters instantaneously. The functions $F_j^k(\eta)$ are expressed as:

$$F_j^k(\eta) = \{N_j^k(\eta)\}^T \{\overrightarrow{\mathbf{a_j}}\}, \tag{5.7}$$

In this context, $N_j^k(\eta)$ represents a function chosen for interpolation purposes, with j defined as the number of control points (Fasshauer, 2007). The vector $\{\overrightarrow{\mathbf{a_j}}\}$ contains constant nodal values computed via the sPGD algorithm.

At this stage, a library of solutions is compiled for any parameter combination within the previously defined ranges, encapsulating current knowledge from physics. It is important to highlight that once the response $\mathcal{U}_i$ is computed for a specified set of parameters using the sPGD surrogate model, it must be projected back onto the full basis using the following equation to obtain the complete system solution:

$$\mathbf{U}_i = \mathcal{U}_i \mathcal{R}_t^T. \tag{5.8}$$

5.2.2 Black-box Model

A physical entity of the system being studied is necessary at this step. Sensors can be used to gather data from the setup across different operational scenarios, which are chosen according to the problem's characteristics. While kinematic data is commonly collected, other types of data might also be relevant. Should the measured quantity not precisely represent the system's response, an extra step is needed to compute the response from the data to facilitate a comparison with the simulated response.

5.2.3 Grey-box Model (Hybrid Twin)

This stage solves an optimization problem that estimates the values of the unknown parameters of the system based on the collected data. The optimization problem can be defined as:

$$\{\overrightarrow{\hat{\mathbf{p}}}\} = \underset{p_1, p_2, \cdots, p_{n_p}}{\operatorname{argmin}} \; \mathcal{L}_2 + \mathcal{L}_1, \tag{5.9}$$

where

$$\mathcal{L}_2 = \left\| \tilde{\mathcal{U}} - \mathcal{U}(p_1, p_2, \cdots, p_{n_p}) \right\|_2, \tag{5.10}$$

$$\mathcal{L}_1 = \lambda \left\| p_1, p_2, \cdots, p_{n_p} \right\|_1. \tag{5.11}$$

Equation (5.9) determines the vector of optimized parameters $\{\overrightarrow{\hat{\mathbf{p}}}\}$ by minimizing a loss function, which includes the $\mathcal{L}_2$ norm reflecting the least squares error between the simulated response $\mathcal{U}$ and the measured data $\tilde{\mathcal{U}}$. In addition to the $\mathcal{L}_1$ norm that features a term promoting sparsity. The parameter $\lambda \geq 0$ is a hyper-parameter used to weight the significance of sparsity. The sparsity term is crucial in identifying the parameters that significantly impact the solution, enabling discarding insignificant terms and thus simplifying the problem.

Numerous optimization techniques that solve such problems are available in the literature; however, we chose the Levenberg-Marquardt (LM) algorithm because it does not necessitate time-integration, converges rapidly, and addresses the limitations of other methods by integrating both the Gradient-Descent and Gauss-Newton methods.

To prevent being trapped in a local minimum, the optimization procedure is executed in τ independent trials, each initiated with a different random set of unknown parameters. The set of parameters that results in the lowest value of the loss function is then chosen as the solution. We further enhanced the technique by applying a Monte Carlo simulation which executes the aforementioned optimization procedure n_{MC} times, averaging the results for each parameter. This method ensures that the results closely reflect reality and avoids local minima. Such comprehensive calculations can be performed without compromising on computational resources because the entire process uses the sPGD library, which pre-computes the Jacobian matrix and extracts solutions for various parameter sets in milliseconds. According to Rishmawi et al. (2024), the duration required to obtain the system's response for

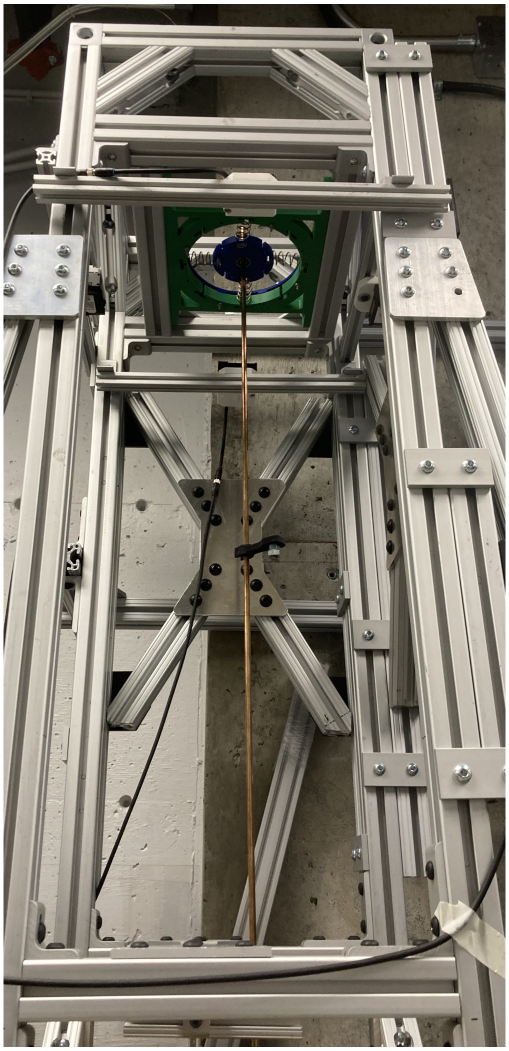


Figure 5.2 A photograph of the VARM built and tested at Polytechnique Montréal. For reference, the shaft length from the upper bearing to the lower bearing is 850 mm.

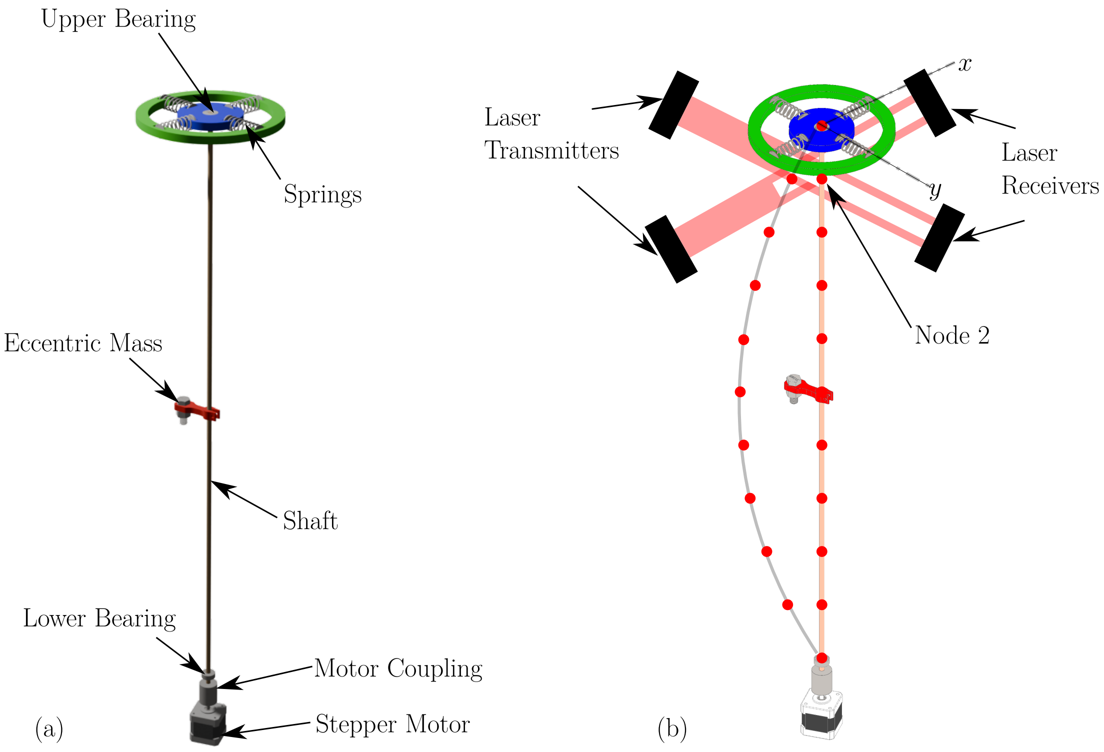


Figure 5.3 Schematic of the VARM built at Polytechnique Montréal. (a) Components of the VARM. (b) Laser Sensor mounting on the VARM. The red circles (●) represent nodes that divide the shaft into 10 finite elements.

a specific parameter set from the sPGD library is 99% less compared to solving the system for the same parameter set using high-fidelity methods like Newton-Raphson at the expense of performing large and expensive offline calculations which are done once and for all. Depending on computational capabilities, the online computation time can range from $1e-5$ to $1e-3$ s. Thus, the LM algorithm is able to achieve convergence through an iterative parameter identification process in real-time.

It is important to note that it is highly recommended to normalize the parameters to ranges between -1 and 1, or 0 and 1, based on their physical significance, to ensure more accurate approximations during the LM optimization process.

5.3 Case Study: Vertical Axis Rotating Machine

In this section, we present the VARM as a case study to validate the developed methodology. Figure 5.2 shows a photograph of the machine built and tested at Polytechnique Montréal.

5.3.1 VARM Experimental Setup

Figure 5.3a presents a 3D illustration of a VARM designed and built to test hybrid twins. This device features a vertical shaft and experiences nonlinear forces arising from shaft displacements at the bearings, which are typically challenging to quantify. To address this, a long and thin bronze shaft measuring 850 mm in length with a Young's modulus of 110.3 GPa was mounted between two distinct bearings: a fixed flanged miniature pillow block mounted MFL4 ball bearing at the lower end of the shaft, and an Uxcell R4A-2RS deep groove ball bearing housed in a 3D printed structure composed of Polylactic Acid (PLA) plastic at the upper end. This upper bearing was made flexible by mounting it on four perpendicular tension-compression springs. The springs are identical, constructed from a 1.702 mm gauge stainless steel wire, featuring an outer diameter of 18.3 mm, and an unstretched length of 50.8 mm and consisting of 8 coils. To simulate unbalances and imperfections, a mass was asymmetrically attached to the beam using a 3D printed PLA attachment to create a rotating unbalance. The whole setup is mounted on an NI ISM-7402 stepper motor.

Two Keyence IG-028 Laser micrometers were positioned perpendicular to each other in each direction of motion, to measure the displacement of a point located at one-tenth of the shaft's full length from the top bearing. This point, as will become evident in subsequent steps, aligns with the second node on the shaft, which will be divided into 10 elements using 11 nodes, as illustrated in Figure 5.3b. These sensors have a gate size of 28 mm, with precision ranging from 0.03 mm to 0.17 mm based on the spacing between the transmitter and receiver, which can reach up to 1500 mm. In this machine, the actual separation distance of the transmitter and receiver placed along the x-axis is 343 mm, while for those along the y-axis, it is 317 mm. Using a sampling rate of 2 kHz, these instruments are able to capture every significant displacement response of the shaft. The design of this machine and the selection of its various parts are explained by Serroud (2023).

5.3.2 VARM Mathematical Model

Figure 5.4 illustrates a schematic of the VARM, which includes a uniform shaft of length L, Young's modulus E, radius r, cross-sectional area moment of inertia I, and mass per unit length m. At rest, the shaft lies on the $z-$axis and its motion in time t in the x

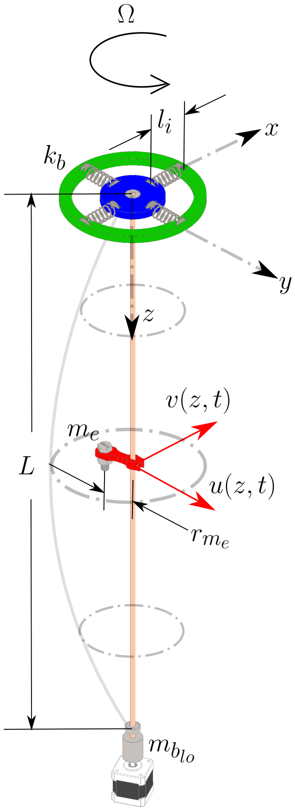


Figure 5.4 Schematic representation of the VARM showing different parameters.

and y directions is described by $u(z,t)$ and $v(z,t)$, respectively. Excluding the upper and lower bearings, the partial differential equations that describe the rotating motion of the shaft under a rotational speed of Ω and induce transverse vibrations in two perpendicular directions are as follows:

$$m\frac{\partial^2 u(z,t)}{\partial t^2} + c\frac{\partial u(z,t)}{\partial t} + EI\frac{\partial^4 u(z,t)}{\partial z^4} - m\Omega^2 u(z,t) = f_x(z,t), \tag{5.12}$$

$$m\frac{\partial^2 v(z,t)}{\partial t^2} + c\frac{\partial v(z,t)}{\partial t} + EI\frac{\partial^4 v(z,t)}{\partial z^4} - m\Omega^2 v(z,t) = f_y(z,t), \tag{5.13}$$

where the external forces $f_x(z,t)$ and $f_y(z,t)$ act on the shaft due to rotational unbalance, and the terms $m\Omega^2 u(z,t)$ and $m\Omega^2 v(z,t)$ correspond to the centrifugal force that stems from the unbalance caused by the shaft's deflection relative to its rotational center. The derivation of equations (5.12) and (5.13) was conducted using the principles outlined by Adams (2010) and leveraging the Euler Beam Element theory. However, it is important to mention that the rotational inertia term was omitted due to the shaft's radius being much smaller than its length, specifically $r^2/L^2 \ll 1$. Additionally, the gyroscopic term was disregarded because of the factor $mr^2\Omega/L^2c \ll 1$.

The force terms $f_x(z,t)$ and $f_y(z,t)$ are a combination of two components, the first of which arises from the rotating unbalance due to the eccentric mass m_e with a lever arm r_{m_e} positioned at $z = z_{m_e}$, represented by:

$$f_{x1}(z,t) = m_e r_{m_e} \Omega^2 \cos(\Omega t)\delta(z - z_{m_e}), \tag{5.14}$$

$$f_{y1}(z,t) = m_e r_{m_e} \Omega^2 \sin(\Omega t)\delta(z - z_{m_e}). \tag{5.15}$$

The second force component consists of a distributed unbalance force arising from the deflection of each element along the shaft, represented by:

$$f_{x2}(z,t) = m(u(z) + u_0(z))\Omega^2 \cos(\Omega t), \tag{5.16}$$

$$f_{y2}(z,t) = m(v(z) + v_0(z))\Omega^2 \sin(\Omega t), \tag{5.17}$$

where $u_0(z)$ and $v_0(z)$ represent minor deformations in the shaft when it is stationary, caused by material flaws or mounting imperfections. The shaft experiences damping, characterized by the damping coefficient c. Additional details on this will be provided later.

The bearings at the top and bottom are considered mixed boundary conditions. Each bearing is modeled as a spring-mass-damper system located at each extremity of the shaft. The lower bearing restricts the shaft's movement in both directions, thus it can be represented as a straightforward linear mass-spring-damper system with a very high stiffness coefficient:

$$f_{b_{lo_x}} = -m_{b_{lo}} \frac{\partial^2 u(L,t)}{\partial t^2} - c_{b_{lo}} \frac{\partial u(L,t)}{\partial t} - ku(L,t), \tag{5.18}$$

$$f_{b_{lo_y}} = -m_{b_{lo}} \frac{\partial^2 v(L,t)}{\partial t^2} - c_{b_{lo}} \frac{\partial v(L,t)}{\partial t} - kv(L,t), \tag{5.19}$$

where $m_{b_{lo}}$, $c_{b_{lo}}$, and k are the lower bearing mass, damping coefficient, and stiffness coefficient, respectively.

Similarly, the upper bearing can be modeled as a mass-spring-damper system, the mass being that of the bearing and its housing and the damping force being that due to friction and air resistance. Since this force is difficult to measure precisely, we will model it as a viscous damping force with an unknown coefficient affecting both directional movements. This coefficient will have to be determined from data.

We present two models for the upper bearing stiffness force: A Geometric Model, and an Expansion Model which uses a Taylor series expansion. Details are discussed next.

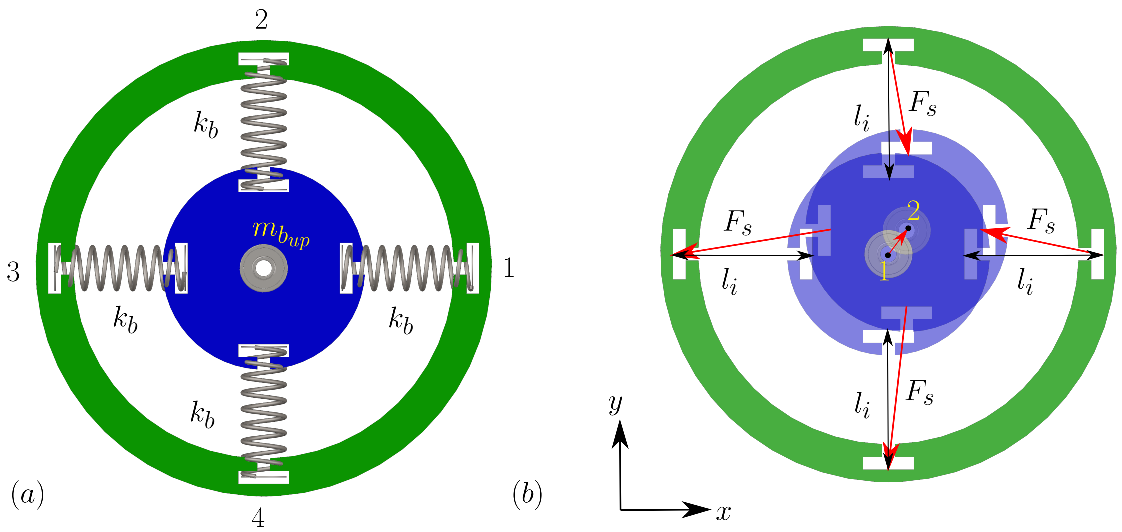


Figure 5.5 Compliant bearing attached to the upper end of the shaft. (a) Bearing parameters. (b) Schematic showing spring forces.

Geometric Model - Geometric Stiffness Force

Figure 5.5a presents a diagram of the upper bearing. The mass of the bearing housing is $m_{b_{up}}$, and it is attached to four springs arranged perpendicularly, each having an unspecified stiffness denoted by k_b. When the bearing and its housing move from position 1 to position 2, the springs are either stretched or compressed, depending on the movement direction as illustrated in Figure 5.5b. Each spring generates a force, F_s, following Hooke's law along its axis. This force is described by:

$$F_s = -k_b(l_f - l_i), \tag{5.20}$$

where l_i is the spring's unstretched length and l_f represents its deformed length.

To determine the overall force produced by the four springs in both the x and y directions, we use a rotation matrix to multiply the force in each spring F_s:

$$\mathbf{R}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}, \; i = 1, 2, 3, 4, \tag{5.21}$$

with θ_i being assigned one of the four values (0^o, 90^o, 180^o, and 270^o) that represent each spring's orientation. Based on that, the net spring forces along the x and y axes are represented as follows:

$$\begin{bmatrix} F_{T_x} \\ F_{T_y} \end{bmatrix} = \sum_{i=1}^{4} \mathbf{R}_i \begin{bmatrix} F_{is_x} \\ F_{is_y} \end{bmatrix} = \begin{bmatrix} -\frac{k_b}{l_i^2} u^3(0,t) + \frac{4k_b}{l_i^2} u(0,t)v^2(0,t) - 2k_b u(0,t) \\ -\frac{k_b}{l_i^2} v^3(0,t) + \frac{4k_b}{l_i^2} v(0,t)u^2(0,t) - 2k_b v(0,t) \end{bmatrix}, \tag{5.22}$$

where F_{is_x} and F_{is_y} represent the force components produced by spring i along the x and y axes, respectively.

Expansion Model - Taylor Expansion of the Stiffness Force

The model developed in the previous subsection is based on having prior knowledge of both the quantity and arrangement of the springs, which is not always available. Therefore, we have broadened our approach to address scenarios in which the stiffness force function of the upper bearing remains unknown. Acknowledging the inherently nonlinear nature of the stiffness function, we propose the following equations to characterize the stiffness force in the upper bearing:

$$\begin{aligned} F_{T_x} = &-\frac{k_1}{l_i^2} u^3(0,t) - \frac{k_2}{l_i} u^2(0,t) - k_3 u(0,t) - k_4 \frac{\partial u(0,t)}{\partial z} \\ &-\frac{k_5}{l_i} u(0,t)v(0,t) - k_6 (\frac{\partial u(0,t)}{\partial z})^3 - \frac{k_7}{l_i^2} u(0,t)v^2(0,t), \end{aligned} \tag{5.23}$$

$$\begin{aligned} F_{T_y} = &-\frac{k_1}{l_i^2} v^3(0,t) - \frac{k_2}{l_i} v^2(0,t) - k_3 v(0,t) - k_4 \frac{\partial v(0,t)}{\partial z} \\ &-\frac{k_5}{l_i} v(0,t)u(0,t) - k_6 (\frac{\partial v(0,t)}{\partial z})^3 - \frac{k_7}{l_i^2} v(0,t)u^2(0,t). \end{aligned} \tag{5.24}$$

It is noted that we retained the terms from the explicit model and incorporated additional linear and nonlinear terms that could impact the response. The parameters k_1 through k_7 are variables that must be determined later.

In both models, the upper bearing mass-spring -damper system is represented by:

$$f_{b_{up_x}} = -m_{b_{up}} \frac{\partial^2 u(0,t)}{\partial t^2} - c_{b_{up}} \frac{\partial u(0,t)}{\partial t} - F_{T_x}, \tag{5.25}$$

$$f_{b_{up_y}} = -m_{b_{up}} \frac{\partial^2 v(0,t)}{\partial t^2} - c_{b_{up}} \frac{\partial v(0,t)}{\partial t} - F_{T_y}, \tag{5.26}$$

where $c_{b_{up}}$ is the damping coefficient of the upper bearing.

Finally, combining equations (5.12), (5.14), (5.16), (5.18), (5.25) in the x direction and equations (5.13), (5.15), (5.17), (5.19), (5.26) in the y direction, we derive the complete mathematical model of the VARM as represented by the following two equations:

$$\begin{aligned} m\frac{\partial^2 u(z,t)}{\partial t^2} + c\frac{\partial u(z,t)}{\partial t} + EI\frac{\partial^4 u(z,t)}{\partial z^4} - f_{b_{lo_x}}\delta(z-L) - f_{b_{up_x}}\delta(z) - m\Omega^2 u(z,t) \\ = m_e r_{m_e}\Omega^2\cos(\Omega t)\delta(z - z_{m_e}) + m(u(z) + u_0(z))\Omega^2\cos(\Omega t), \end{aligned} \tag{5.27}$$

$$\begin{aligned} m\frac{\partial^2 v(z,t)}{\partial t^2} + d\frac{\partial v(z,t)}{\partial t} + EI\frac{\partial^4 v(z,t)}{\partial z^4} - f_{b_{lo_y}}\delta(z-L) - f_{b_{up_y}}\delta(z) - m\Omega^2 v(z,t) \\ = m_e r_{m_e}\Omega^2\sin(\Omega t)\delta(z - z_{m_e}) + m(v(z) + v_0(z))\Omega^2\sin(\Omega t). \end{aligned} \tag{5.28}$$

Equations (5.27) and (5.28) serve as the continuous analytical framework of the VARM for both movement directions. To derive a numerical solution, these equations must be discretized, facilitating manipulation through computer-based numerical methods. The Galerkin method is used in a finite element analysis to formulate the weak form of the equations. This finite element model relies on $1D$ beam shape functions, which specify two degrees of freedom at each node located at the ends of each beam element: deflections and the slopes of deflection curves. For further information on this step, the reader is advised to consult the work of Burnett (1988). The discretized version of the equations is presented by:

$$\begin{aligned} \begin{bmatrix} \mathbf{M_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{M_y} \end{bmatrix} \begin{Bmatrix} \overrightarrow{\ddot{\mathbf{u}}} \\ \overrightarrow{\ddot{\mathbf{v}}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_y} \end{bmatrix} \begin{Bmatrix} \overrightarrow{\dot{\mathbf{u}}} \\ \overrightarrow{\dot{\mathbf{v}}} \end{Bmatrix} + \left[\begin{bmatrix} \mathbf{K_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_y} \end{bmatrix} - \Omega^2 \begin{bmatrix} \mathbf{M_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{M_y} \end{bmatrix} \right] \begin{Bmatrix} \overrightarrow{\mathbf{u}} \\ \overrightarrow{\mathbf{v}} \end{Bmatrix} \\ + \begin{bmatrix} \mathbf{K_{nl1_x}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{nl1_y}} \end{bmatrix} \begin{Bmatrix} \overrightarrow{\mathbf{u}}^3 \\ \overrightarrow{\mathbf{v}}^3 \end{Bmatrix} - \begin{bmatrix} \mathbf{K_{nl2_x}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{nl2_y}} \end{bmatrix} \begin{Bmatrix} \overrightarrow{\mathbf{v}}^2 \\ \overrightarrow{\mathbf{u}}^2 \end{Bmatrix} \begin{Bmatrix} \overrightarrow{\mathbf{u}} \\ \overrightarrow{\mathbf{v}} \end{Bmatrix} = \begin{Bmatrix} \overrightarrow{\mathbf{f_x}}\cos(\Omega t) \\ \overrightarrow{\mathbf{f_y}}\sin(\Omega t) \end{Bmatrix}, \end{aligned} \tag{5.29}$$

It is important to mention that this equation pertains to the Geometric Model for bearing stiffness, with a comparable method applied to the Expansion Model. In equation (5.29), the square and cube operations, as well as the vector multiplication in the fourth and fifth terms, are implemented element-wise. The subscripts x and y denote the directions of motion. Consequently, we will define the matrices without indicating the direction, without losing generality. The mass matrix $\mathbf{M}$ represents the mass associated with the shaft, combined with the masses of the bearings and the rotating unbalance. $\mathbf{K}$ represents the linear stiffness matrix including shaft stiffness and linear bearing stiffness coefficients. The matrices $\mathbf{K_{nl1}}$ and $\mathbf{K_{nl2}}$ are primarily zero matrices except the element corresponding to the node where the upper bearing is connected, where the nonlinear coefficient value is included. The vectors $\{\overrightarrow{\mathbf{f_x}}\}$ and $\{\overrightarrow{\mathbf{f_y}}\}$ specify the forcing term amplitudes due to the eccentric mass and the distributed unbalance force arising from the shaft's initial deflection, respectively. They can be expressed as follows:

$$\begin{Bmatrix} \overrightarrow{\mathbf{f_x}} \\ \overrightarrow{\mathbf{f_y}} \end{Bmatrix} = \begin{Bmatrix} m_e r_{m_e} \Omega^2 \{\overrightarrow{\gamma}\}_{z_{m_e}} + m_{el} q_x \{\overrightarrow{\mathbf{\Phi_1}}\} \Omega^2 \\ m_e r_{m_e} \Omega^2 \{\overrightarrow{\gamma}\}_{z_{m_e}} + m_{el} q_y \{\overrightarrow{\mathbf{\Phi_1}}\} \Omega^2 \end{Bmatrix}. \tag{5.30}$$

In this context, $\{\overrightarrow{\gamma}\}_{z_{m_e}}$ is a vector that consists predominantly of zeros, except it has a value of 1 at the node linked to the distance z_{m_e}. The second term can be perceived as a distributed eccentric force, where each element is an eccentric mass of m_{el}. The eccentricity can be determined by considering the beam's initial bending and the deflection caused by rotational movement, which can be expressed as $q_x\{\overrightarrow{\mathbf{\Phi_1}}\}$. The beam's dynamics are mainly influenced by its first mode; hence, we chose to express the beam's deflection using its primary eigenvector, $\{\overrightarrow{\mathbf{\Phi_1}}\}$, which is derived from solving the system's eigenvalue problem through its mass and stiffness matrices. This eigenvector is scaled with an unidentified factor q_x, which needs to be estimated from experimental data. A similar approach is applied for representation in the y-direction.

To conclude, the matrix $\mathbf{C}$ serves as the damping matrix, incorporating both structural damping and damping derived from bearing models. The structural damping component $\mathbf{C_s}$ is expressed using Rayleigh damping, specifically by considering only the mass-proportional term as shown below:

$$\mathbf{C_s} = \alpha_d \mathbf{M}. \tag{5.31}$$

Here, the structural damping coefficient ζ_s can be defined as:

$$\zeta_s = \frac{1}{2}\frac{\alpha_d}{\omega_1}, \tag{5.32}$$

with ω_1 representing the system's first modal frequency.

Table 5.1 presents all the known parameters of the general VARM model, which includes both measured and computed values. An arbitrarily large value was selected for the lower bearing stiffness, along with a comparatively elevated damping ratio, aligned with the damping properties of rubber (JPE - High Tech Engineering, 2025), since a rubber bushing was placed between the shaft and the lower bearing. This ensures that the node linked to the lower bearing consistently has a displacement of zero in our calculations due to the high stiffness and/or high damping ratio.

5.4 Results

This section verifies the effectiveness of the developed method by applying it to the Geometric and Expansion Models.

Table 5.1 List of known parameters of the VARM model.

Parameter	Symbol	Value
Mass per unit length [$\mathrm{kg\,m^{-1}}$]	m	0.35
Young's Modulus of bronze [GPa]	E	110.3
Shaft radius [m]	r	0.003175
Area moment of inertia [$\mathrm{m^4}$]	I	79.81e-11
Upper bearing mass [kg]	$m_{b_{up}}$	0.056
Unstretched spring length [m]	l_i	0.0508
Lower bearing mass [kg]	$m_{b_{lo}}$	0.176
Lower bearing damping ratio	$\zeta_{b_{lo}}$	0.05
Lower bearing stiffness [$\mathrm{N\,m^{-1}}$]	k	10^4
Eccentric mass [kg]	m_e	0.000, 0.0148, 0.0259
Eccentricity [m]	r_{m_e}	0.040
Eccentric mass position [m]	z_{m_e}	0.425
Shaft length [m]	L	0.850
Number of elements	n	10
Mass of one element [kg]	m_{el}	0.0281

5.4.1 Geometric Model

White-Box Model

In the Geometric Model, we identify $n_p = 5$ difficult-to-measure parameters and enumerate them in Table 5.2.

For a given rotational speed, the Latin hypercube generated $n_s = 1000$ scenarios. The HMH solver was used to solve these n_s scenarios to obtain their steady state limit cycles. The time series of two steady state cycles of all DOFs of the system were stored in the $\mathbf{U}(z,t)$ and $\mathbf{V}(z,t)$ matrices for the x- and y-direction displacements respectively. With 10 elements along the shaft, there are 11 nodes, each possessing 4 DOFs, thus resulting in $n_{DOF} = 44$. Figure 5.3b illustrates these nodes depicted as red circles positioned along the shaft. Each rotational speed corresponds to a unique time duration that covers 2 full cycles, represented

Table 5.2 List of unknown parameters of the VARM corresponding to the Geometric Model.

Parameter	Symbol	Expected range
Spring stiffness [$\mathrm{N\,m^{-1}}$]	k_b	2000 - 3500
Structural damping ratio	ζ_s	0.01 - 0.05
Upper bearing damping ratio	$\zeta_{b_{up}}$	0.00 - 0.05
Force coefficient (x-axis)	q_x	0.001 - 0.01
Force coefficient (y-axis)	q_y	0.001 - 0.01

by $T_{2cyc} = 120/\Omega$ s, where Ω is the VARM's rotational speed in RPM. The number of time steps, n_t, for 2 cycles is influenced by the rotational speed, typically calculated as $n_t = T_{2cyc}/dt$, with $dt = 5e-4$ s being the duration of one time step.

To choose a sufficient number of reduced-order modes to fully represent the VARM displacement, the first few normalized singular values where extracted. We observed that the ratio of the second singular value to the first is 0.0012, this means that the data can be effectively represented by the first reduced-order mode alone, with additional modes having an insignificant contribution, i.e. $m_d = 1$. This is due to the high correlation within the dataset, as variations in parameter values across scenarios only influence the response amplitude.

Applying equation (5.6) presented in Section 5.2.1 to the Geometric Model case with the parameters defined in Table 5.2, the matrix $\mathcal{U}$ can be represented as a function of the five specific parameters according to the sPGD formulation as follows:

$$\mathcal{U}(k_b, \zeta_s, \zeta_{b_{up}}, q_x, q_y) = \sum_{k=1}^{M} F_1^k(k_b) F_2^k(\zeta_s) F_3^k(\zeta_{b_{up}}) F_4^k(q_x) F_5^k(q_y). \tag{5.33}$$

where the interpolation functions chosen are the Kriging functions shown in Figure 5.6. The advantages of using such functions are explained by Fasshauer (2007). In general, using between 8 and 12 control points provided a satisfactory approximation while preventing over-fitting.

To evaluate the outcomes of the proposed method, it is crucial to initially confirm that the sPGD library yields accurate approximations of the system's response for a random set of parameters. Figure 5.7 illustrates the orbital motion of the VARM's second node, computed for a random set of parameters via the HMH solver and the sPGD library. The figure demonstrates a strong agreement between the two approaches, indicating that the sPGD is a dependable method for use in the optimization process.

Black-Box Model: The VARM system's sensors collect data, which is then processed using a LabView program and analyzed with MATLAB scripts. During the experiments, the motor maintained a constant rotational speed, and the resulting steady-state displacement of the second node was recorded as time series data in both the x and y directions with a time step of $5e-4$ s (a sampling frequency of 2 kHz). Figure 5.8 displays an example of the collected data. To ensure that the unbalance force acting on the shaft exceeds the preload requirements of the lower bearing without causing machine damage, an appropriate eccentric mass was selected for each speed range through experimental observations, as detailed in Table 5.3. Data were collected for rotational speeds with 50 RPM increments.

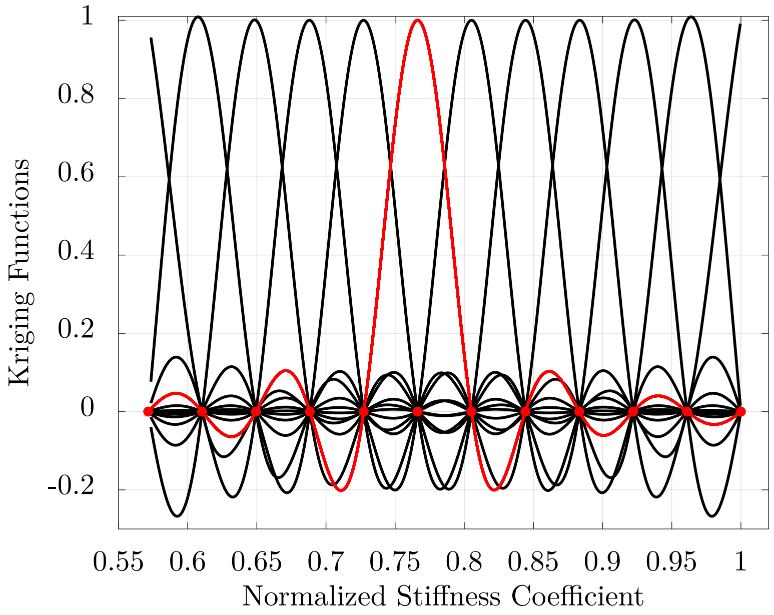


Figure 5.6 Globally defined Kriging basis functions used to interpolate the response of the system in terms of the normalized stiffness coefficient. The limits of the non-dimensional domain were derived by normalizing the range in Table 5.2 with its maximum value. Similar functions are used to interpolate the response in terms of the other parameters. The red circles (●) represent control points. The function in red represents the form of a single Kriging function defined globally on the parameter domain.

Grey-box Model: At this point, we address the earlier specified optimization problem using the LM optimization method. Given that the bearing stiffness is determined by geometric factors, we omit the sparsity-promoting term, hence $\lambda = 0$, and only the $\mathcal{L}_2$ norm is minimized. In each optimization process, conducting $\tau = 50$ trials proved sufficient to achieve satisfactory outcomes, and the procedure was executed 1000 times within the Monte Carlo framework, with the final result reflecting the average from these 1000 procedures.

Figure 5.9 illustrates the parameter values as a function of rotational speed and the eccentric mass attached to the shaft. It is observed that the bearing stiffness k_b, structural damping

Table 5.3 The values of the eccentric mass used for each range of rotational speed during the experiments.

Rotational Speed Range [(**RPM**]	**Eccentric Mass** [g]
350 - 650	25.9
550 - 850	14.8
800 - 1000	0.0

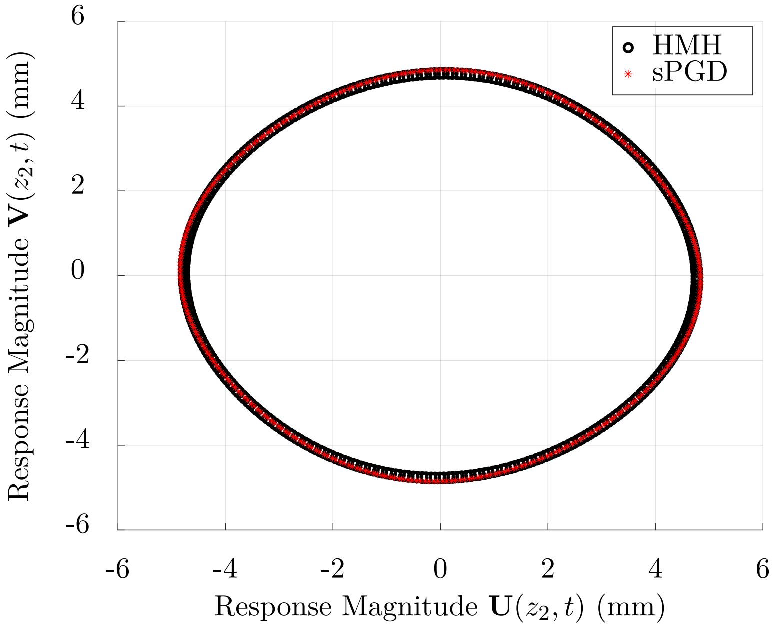


Figure 5.7 Orbital motion of the second node of the VARM calculated using the HMH solver and sPGD for the parameters: $k_b = 2750$ N m^{-1}, $q_x = 0.01$, $q_y = 0.01$, $\zeta_s = 0.02$, $\zeta_{b_{up}} = 0.02$.

ratio ζ_s, and the bearing damping ratio $\zeta_{b_{up}}$ maintain consistent values, irrespective of changes in the rotational speed or eccentric mass. This consistency emerges because these values signify physical properties expected to remain steady during operation. In contrast, q_x and q_y vary with the eccentric mass and increase as the rotational speed rises. These parameters are correction factors intended to account for shaft bending not otherwise physically modeled. Their variations are characteristic of a centrifugal pull; examining a 1 DOF static form of the equation of motion in the x direction yields:

$$\hat{k}\phi(q_x - q_0) = \Omega^2(m_{el} q_x \phi + m_e r_{m_e}), \tag{5.34}$$

where q_0 is the initial bending in the shaft, $\hat{k}$ is the stiffness of 1 DOF of the shaft, and ϕ is a 1 DOF version of the first eigenvector of the VARM. This equation rearranges to:

$$q_x = \frac{\hat{k}\phi q_0 + \Omega^2 m_e r_{m_e}}{\hat{k}\phi - \Omega^2 m_{el}\phi}, \tag{5.35}$$

displaying the relationship between q_x and Ω. Two scenarios arise: with an eccentric mass, $m_e >> m_{el}$, the equation simplifies to:

$$q_x = \frac{\hat{k}\phi q_0 + \Omega^2 m_e r_{m_e}}{\hat{k}\phi}, \tag{5.36}$$

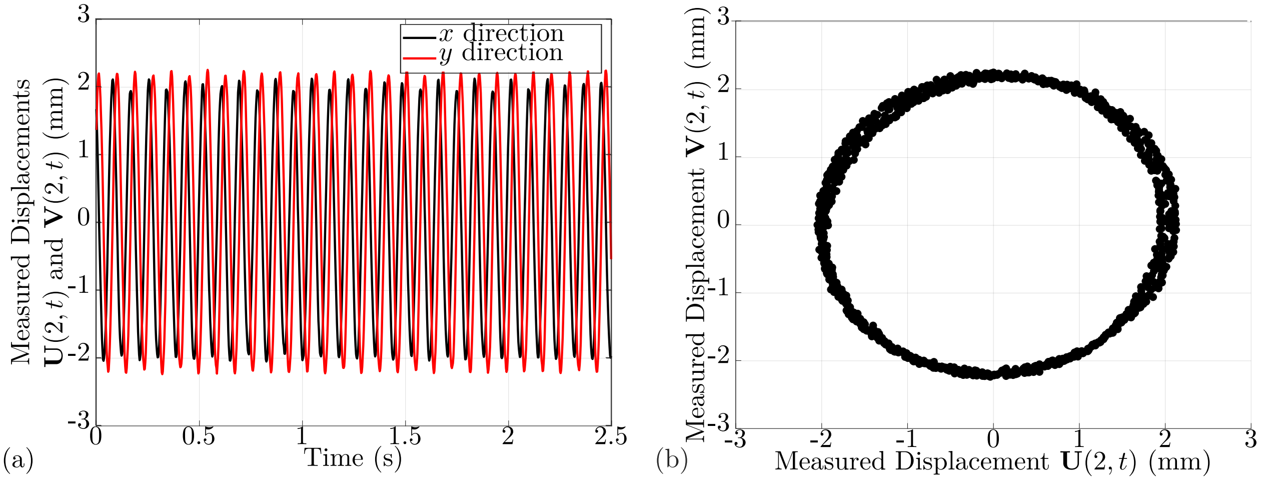


Figure 5.8 Example of the measured displacement of node 2 on the shaft in the x and y directions. Case corresponds to an eccentric mass of 14.8 g and a rotational speed of 700 RPM. (a) Time series. (b) Orbital motion.

without an eccentric mass, where $m_e = 0$, the equation becomes:

$$q_x = \frac{\hat{k}\phi q_0}{\hat{k}\phi - \Omega^2 m_{el}\phi}. \tag{5.37}$$

Equations (5.36) and (5.37) are explicitly used to define the optimal fitting line for q_x and q_y graphs relative to Ω. Figure 5.9 shows that these theoretical relations align well with the empirical results, thus confirming their reliability.

Figure 5.10 depicts the orbital displacement of the second node of the shaft as it relates to the rotational speed and eccentric mass. The black dots indicate data gathered from the VARM, while the blue line illustrates the calculated response derived from the optimized parameters calculated using the LM algorithm. The figure demonstrates that the calculated response agrees well with the measured data across all values of rotational speed and eccentric masses, confirming that the optimal parameters accurately represent the system's behavior.

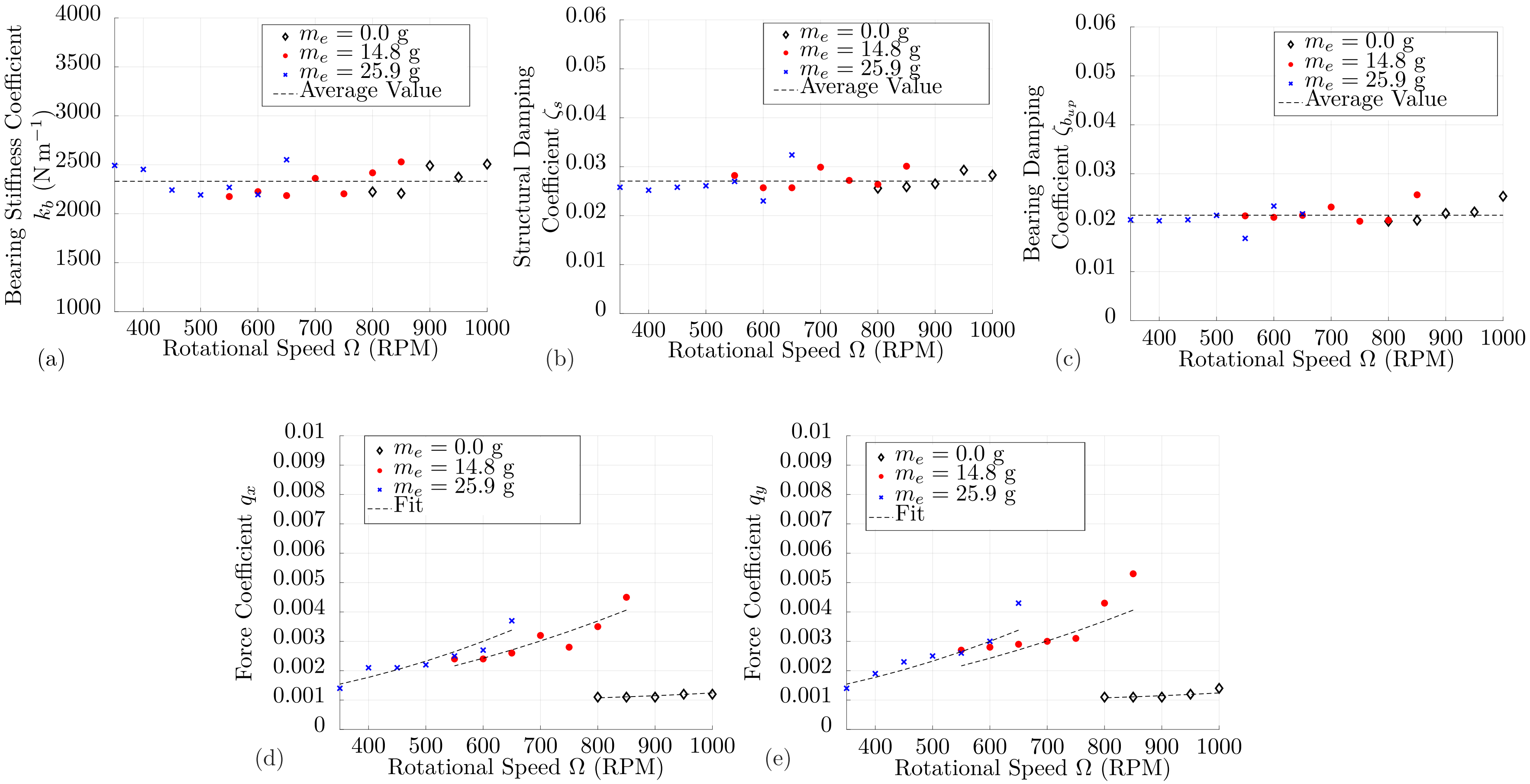


Figure 5.9 Optimal values of VARM parameters. (a) Bearing Stiffness Coefficient $k_b = 2331.5$ (N m^{-1}). (b) Structural Damping Coefficient $\zeta_s = 0.0271$. (c) Bearing Damping Coefficient $\zeta_{b_{up}} = 0.0215$. (d) Force Coefficient q_x. (e) Force Coefficient q_y.

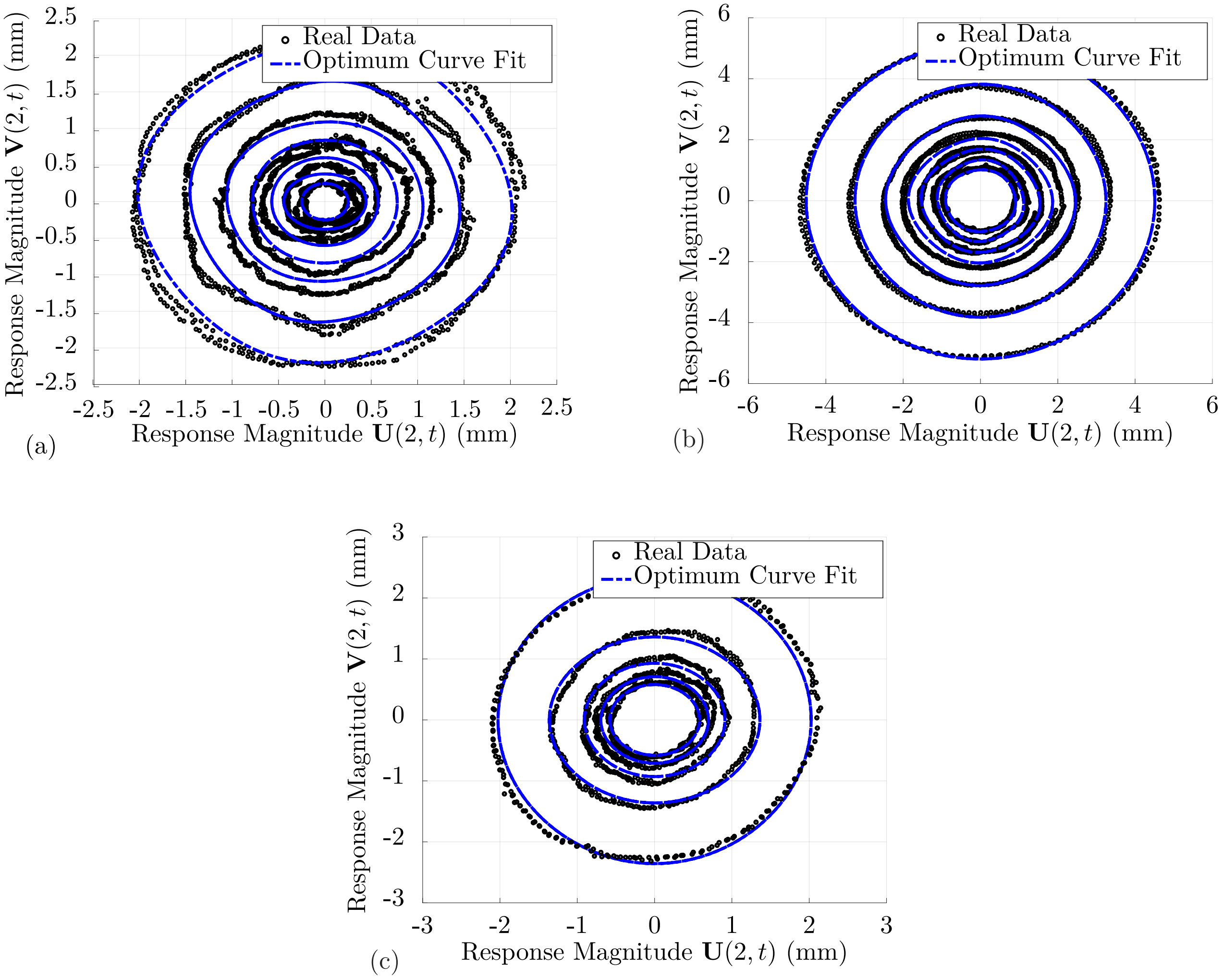


Figure 5.10 Optimal orbital displacement of the second node of the shaft vs. measured data for various values of eccentric mass and rotational speeds. (a) $m_e = 25.9$ g, $\Omega = [350, 650]$ RPM. (b) $m_e = 14.8$ g, $\Omega = [550, 850]$ RPM. (c) $m_e = 0$ g, $\Omega = [800, 1000]$ RPM.

Exploring the behavior of the LM optimization technique, an optimization process involves τ randomly initialized trials, each starting with a different random parameter set. The algorithm then selects the trial that minimizes the loss function to determine the solution for that optimization process. Figure 5.11 illustrates the progression of the loss function's value across 50 trials, highlighting in red the trial that minimizes the loss function. This corresponds to the case where $m_e = 14.8$ g and $\Omega = 700$ RPM.

Figure 5.12 illustrates the evolution of the 5 parameters toward their chosen optimal value in the case highlighted above. It is important to note that the parameter values depicted in the figure are normalized. Observing the data, we find that the parameters q_x and q_y have less impact on the system's response compared to the other parameters.

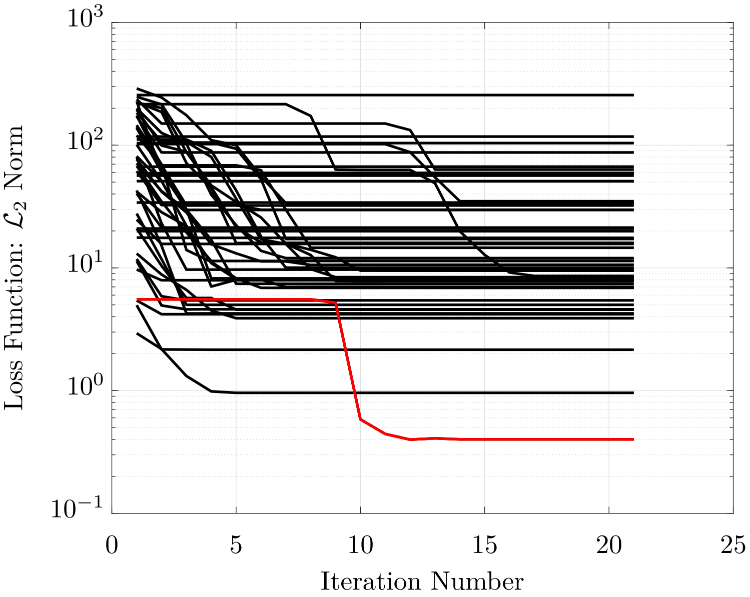


Figure 5.11 Loss function evolution for 50 trials of one optimization process when $m_e = 14.8$ g and $\Omega = 700$ RPM. The red line represents the selected trial that minimizes the loss function.

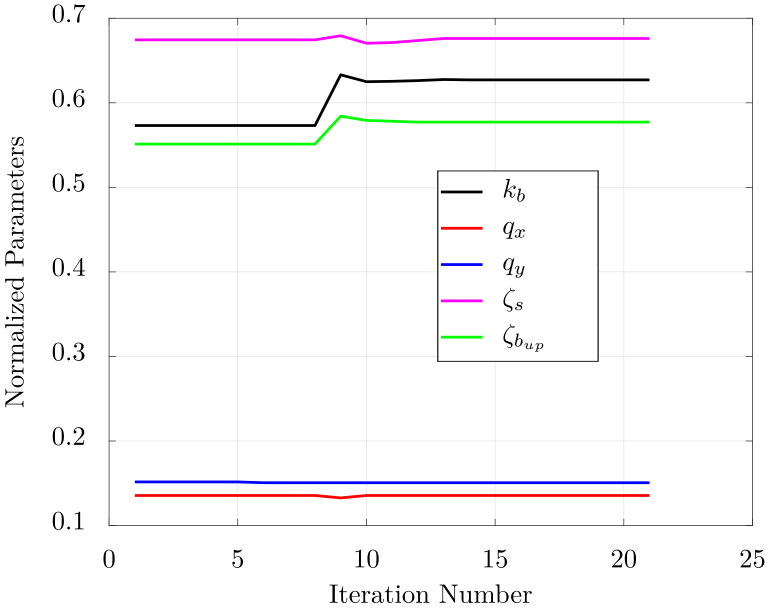


Figure 5.12 Parameter convergence for the selected trial that minimizes the loss function of one optimization process when $m_e = 14.8$ g and $\Omega = 700$ RPM.

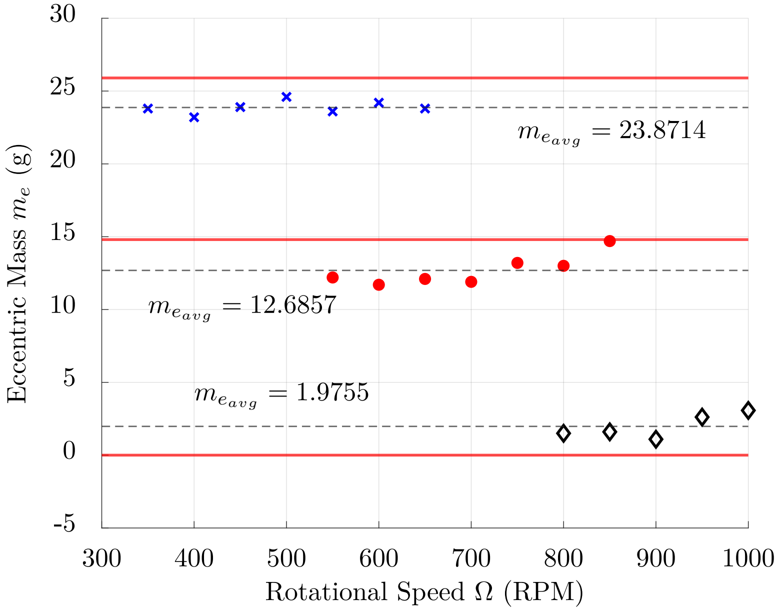


Figure 5.13 Optimized Value of the Eccentric Mass. The red line represents the true value, and the dashed line represents the average of the predicted values.

Eccentric Mass as an Additional Parameter

In our model, the eccentric mass was easily identified using a digital scale. However, in this section, we treat the eccentric mass as unknown, incorporating it as a sixth parameter in our system for estimation. The corresponding results are presented in Figure 5.13. We observe that the eccentric mass exhibits a consistent value irrespective of the rotational speed, although it is slightly underestimated in the initial two cases. In the first case, the actual value is 25.9 g, while our method predicts 23.9 g, resulting in a 7.7% error. In the second case, the true value is 14.8 g with our estimation at 12.7 g, leading to a 14.2% error. Notably, in the third case, there was no eccentric mass, yet the predicted value was 1.9 g. This can be attributed to the domain on which the sPGD was trained, which spans $[0, 10]$ g. Since 0 is an edge value, the sPGD interpolation struggles to precisely reach it. Nonetheless, these discrepancies arise because when the LM algorithm attempts to determine values for six parameters concurrently to minimize a single loss function, some compromise is inevitable to achieve an overall response that most closely aligns with the data. To provide a numerical explanation, we observed that in the Geometric Model, when the eccentric mass values are known, the predicted bearing stiffness came to 2331.5 $\mathrm{N\,m^{-1}}$. Conversely, in the model with predicted masses, the stiffness predicted was 2443.6 $\mathrm{N\,m^{-1}}$. This indicates that underestimating the eccentric mass led to an overestimation of the bearing stiffness as a compensatory measure to minimize the loss function.

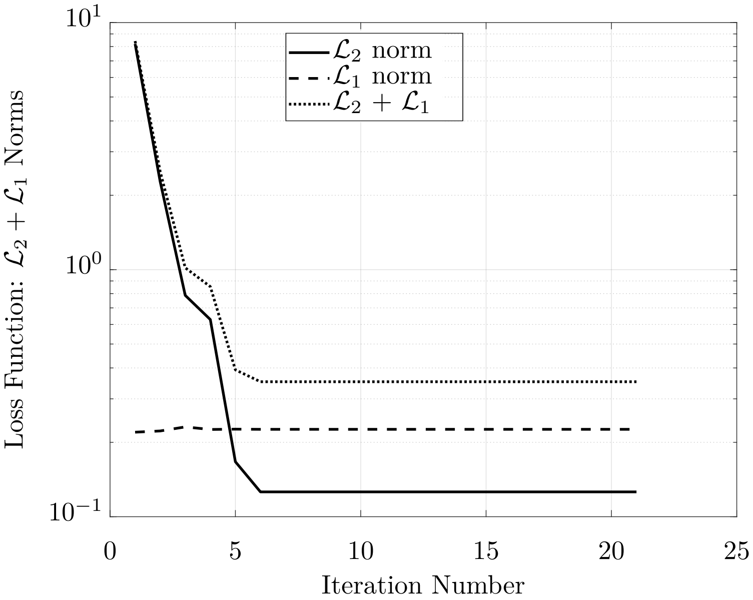


Figure 5.14 Loss function evolution for the selected trial of one optimization process when $m_e = 14.8$ g and $\Omega = 700$ RPM.

Table 5.4 Optimized Values of the Bearing Stiffness Parameters in the Expansion Model

Parameter	**Normalized Value**	**Actual Value**
k_1	0.732	2562.7 $[\mathrm{N\,m^{-1}}]$
k_2	-0.071	-70.9 $[\mathrm{N\,m^{-1}}]$
k_3	0.775	5428.5 $[\mathrm{N\,m^{-1}}]$
k_4	0.341	341.1 [N]
k_5	-0.061	60.9 $[\mathrm{N\,m^{-1}}]$
k_6	0.293	293.0 [N]
k_7	-0.045	-627.7 $[\mathrm{N\,m^{-1}}]$

5.4.2 Expansion Model

The Expansion Model involves determining $n_p = 9$ parameters, specifically k_1 through k_7 along with q_x and q_y. The values for ζ_s and $\zeta_{b_{up}}$ are sourced from predictions of the Geometric Model. Given the model's complexity, we employed Latin hypercube sampling to create $n_s = 2700$ scenarios. To accommodate the increased parameter count, the sPGD algorithm required additional control points, with 15 to 20 being adequate.

We employed the same optimization strategy to estimate these parameters but with a minor adjustment that involved incorporating a sparsity promoting term $\lambda = 0.1$ into the loss function as detailed in equations (5.9, 5.10, 5.11) in the methodology. λ is a hyper-parameter that was chosen to be sufficiently large to ensure that $\mathcal{L}_1$ contributed a meaningful sparsity component, aiding in assessing parameter significance, while remaining small enough to preserve the significance of $\mathcal{L}_2$. Figure 5.14 illustrates the original least squares error loss function, the added sparsity loss function, and their combined effect. Although the inclusion of the sparsity term raises the loss function's final value, it ensures that the total sum of predicted parameters is minimized, aiding in identifying which parameters significantly influence the system's response.

Additionally, the $\mathcal{L}_1$ norm tends to stay relatively stable throughout the optimization process. This is due to the fact that the algorithm selects the trial with the least error in each optimization procedure. In these successful trials, the initial parameter values were generally close to their eventual converged values, leading to only slight fluctuations during optimization, as depicted in Figure 5.12. By definition, the $\mathcal{L}_1$ norm mirrors the behavior of the parameter values, as it calculates their average multiplied by the hyper-parameter λ, thus maintaining a nearly steady value. On the other hand, the $\mathcal{L}_2$ norm depends on the system's response to the parameter set derived from the sPGD surrogate model, making it more prone to slight changes in parameter values.

The remaining hyper-parameters that were not specifically mentioned here were the same as those used for the Geometric Model. Same data sets were used as well.

Table 5.4 displays the optimum normalized values for the bearing stiffness parameters $k_1 - k_7$. These values represent the average of values obtained for the same ranges of eccentric mass and rotation speeds used in the Geometric Model. The parameters k_2, k_5, and k_7 are distinctly smaller but not negligible when assessed purely on a mathematical basis using the gathered data. However, incorporating physical considerations, a stronger case can be made for their exclusion. This further emphasizes the benefits of the grey-box framework. The terms associated with k_2 and k_5 have no physical relevance in the model and can thus be omitted.

In contrast, k_7 depicts the coupling between the shaft's motion along the x and y axes. Yet, as demonstrated in the upcoming section, omitting k_7 has a negligible effect on the overall results, allowing us to treat the shaft's motion as uncoupled. Nevertheless, k_7 converged to a negative value leading to a positive coupling term in equations (5.23) and (5.24), and this conforms with the Geometric model in equation (5.22). Another observation is that the two terms related to the displacement's slope (k_4 and k_6), which were overlooked in the Geometric Model, appear to have a significant effect. Lastly, as anticipated from the geometry, the actual value of k_1 is approximately half the value of k_3, a relationship that was explicitly depicted in the Geometric Model.

5.4.3 Combining the Geometric and Expansion Models towards the Hybrid Twin

Analyzing the parameter values using the Expansion Model provided more clarity on the overall depiction of the bearing stiffness force. We found that the coupled term in the Geometric Model is not substantial. However, two additional crucial terms concerning shaft bending have more influence on the accuracy of the entire function. Consequently, we chose to express the stiffness force in the upper bearing as follows:

$$F_{T_x} = -\frac{k_b}{l_i^2}u^3(0,t) - 2k_b u(0,t) - k_w\frac{\partial u(0,t)}{\partial z} - k_z(\frac{\partial u(0,t)}{\partial z})^3, \tag{5.38}$$

$$F_{T_y} = -\frac{k_b}{l_i^2}v^3(0,t) - 2k_b v(0,t) - k_w\frac{\partial v(0,t)}{\partial z} - k_z(\frac{\partial v(0,t)}{\partial z})^3, \tag{5.39}$$

These equations include three unknown parameters: k_b, k_w, and k_z, along with the two damping parameters $\zeta_{b_{up}}$ and ζ_s, and the two force coefficients q_x and q_y. We recalculated the values of these seven parameters in our final model. The values of q_x and q_y exhibit a pattern similar to that depicted in Figure 5.9, while the remaining values are documented in Table 5.5. It is important to mention that in this instance, the parameters were determined without incorporating the sparsity loss term, meaning $\lambda = 0$, because our findings suggest that all parameters hold significance.

Figure 5.15 illustrates that despite eliminating the extra terms, the data continues to align closely with the simulated response. This implies that the final model indeed acts as a hybrid twin of the physical machine, capable of predicting its behavior across different scenarios. The model's validity is further confirmed by testing it with a separate dataset not involved in the parameter optimization. In this instance, an eccentric mass of three different values was used. The findings are presented in Figure 5.16 where the empirical data aligns with the simulated response for eccentric mass values not used during the construction of the hybrid

Table 5.5 Optimized Values of the Bearing Final Model Parameters

Parameter	**Optimized Value**
k_b [N m^{-1}]	2411.3
k_w [N]	324.2
k_z [N]	301.1
ζ_s	0.024
$\zeta_{b_{up}}$	0.016

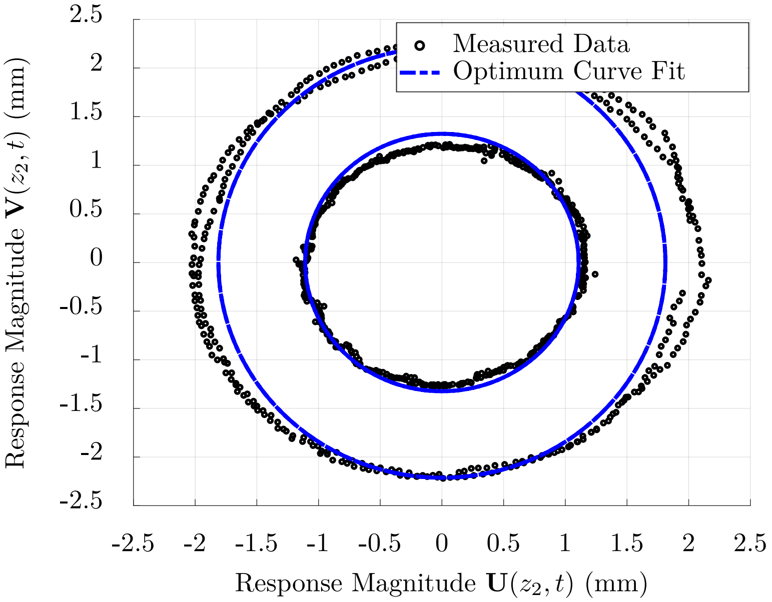


Figure 5.15 Orbital displacement of the second node of the shaft. The large circle represents the case where $m_e = 14.8$ g and $\Omega = 700$ RPM, and the small circle represents the case where $m_e = 25.9$ g and $\Omega = 550$ RPM.

twin. It is crucial to note that the parameters q_x and q_y were adjusted based on the mass values, as they depend on the mass values as seen in the results.

The method proved to be efficient in producing the anticipated results, but it still has limitations. First, it should be noted that when generating the parameter data set used to build the surrogate model via sPGD, the number of scenarios n_s is selectable, though the Latin Hypercube algorithm randomly distributes the parameter values of these scenarios throughout the parameter domain to ensure comprehensive coverage. Also, as sPGD is a form of interpolation, it cannot handle the situation where the true parameter values are outside the expected range. Therefore, initial estimates of the parameter ranges were established through an understanding of the system components and subsequently adjusted by evaluating the identified parameters. It was noted that if any parameters converged to values near the extremities of the predetermined range, it would suggest a need for revising these limits. Another limitation in the system is that although numerous complexities were incorporated during development, the system ultimately displayed a monomodal response. Further tests should be performed in the future with excitations giving rise to a richer modal response.

5.5 Conclusion and Future Work

In conclusion, this paper introduced an innovative method that integrates established numerical techniques to form a hybrid twin of a Vertical Axis Rotating Machine (VARM) subjected to non-linear bearing forces, rotational unbalance, and varying loading amplitudes and rotational speeds. This hybrid twin relies on the physical principles governing the VARM, which are modeled by partial differential equations comprising some unknown parameters determined through data assimilation from an experimental VARM setup. The findings indicate that when the bearing model is clearly defined, the method reliably predicts various parameters in accordance with physical principles and intuition. In cases where the bearing model is not clearly established, a provisional model was proposed, allowing the method to determine which terms meaningfully affect the machine's behavior by optimizing the parameter values.

In future work, the approach may be expanded to tackle increasingly intricate systems. Vertical-axis hydraulic turbines, like Francis, Kaplan, and propeller turbines, exhibit numerous resemblances to our VARM. These similarities include a flexible rotating shaft supported by bearings at both ends, which apply unknown nonlinear forces, the lateral vibrations induced by whirling and rotational unbalance, as well as the nonlinear dynamic behavior as a whole. However, the hydraulic turbine model not only incorporates dynamic vibrations but also complex fluid-structure interactions due to water flow. Nevertheless, a block method can be employed where each subsystem is independently modeled and solved with its own PDE. The output from one block serves as the input for the next, aiding in the analysis via the proposed methods. Unbalanced mass effects may originate from swirling cavitation, generator imbalance, or irregular water flow caused by blockages in the guide veins. Additionally, typical hydraulic turbine configurations use tilted-pad journal bearings. Literature indicates that the forces exerted by these bearings are described by equations with undetermined coefficients, which can be analyzed using a method akin to the one outlined in this paper.

The method itself can be further enhanced to study the transient regime, as our analysis was limited to the steady-state response. This is important in the application of hydraulic turbines as their operation involves starts and stops that need to be analyzed. There remains potential

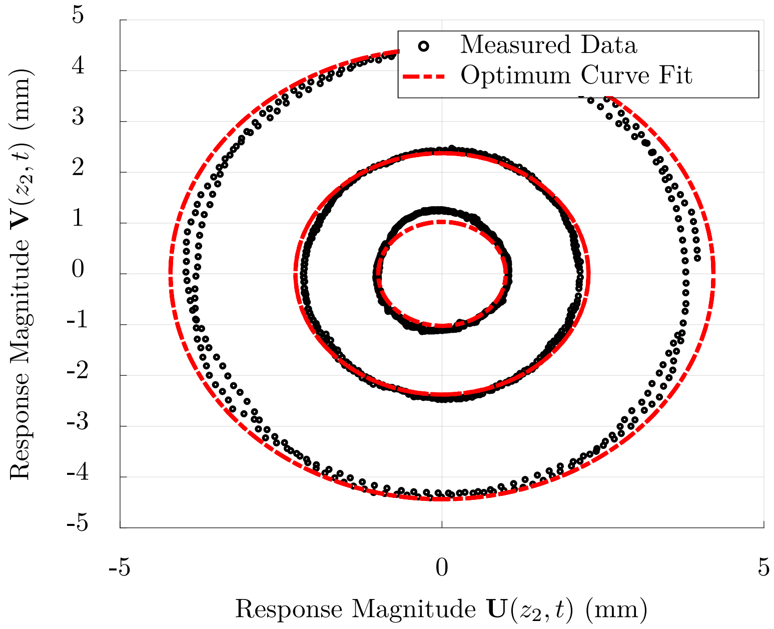


Figure 5.16 Orbital displacement of the second node of the shaft. The large circle represents the case where $m_e = 7.6$ g and $\Omega = 900$ RPM, the medium circle represents the case where $m_e = 19.5$ g and $\Omega = 700$ RPM, and the small circle is where $m_e = 36.0$ g and $\Omega = 450$ RPM.

to explore alternative optimization techniques that could anticipate parameter variations during operation, thereby identifying potential faults. Additionally, various scenarios could be explored by introducing varied loads or using different types of bearings.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments and Funding

The authors would like to acknowledge the technical and financial support of Hydro-Québec and Maya HTT, and the financial support of NSERC Alliance [Grant ALLRP 556353 - 20], InovÉÉ, IVADO [PhD Excellence Scholarship], and Mitacs [Globalink Research Award].

Data Availability

The datasets generated or analyzed during this study are available from the corresponding author upon reasonable request.

Ethical Approval

The authors declare that they comply with ethical standards.

CHAPTER 6 GENERAL DISCUSSIONS

6.1 Discussion

This project presents a blueprint for a digital twin of a VARM. By integrating reduced-order modeling with data assimilation, a robust model was formulated to effectively optimize unknown parameters. Various computational techniques, such as the HMH method, DEIM, sPGD, and the LM optimization approach, were employed to accomplish this.

The HMH method coupled with DEIM has proven highly effective in resolving PDEs with notable accuracy and speed. This facilitates the generation of the sparse data set necessary to construct the parametrized solution with sPGD. Although incorporating more parameters demands a more extensive data set for precise interpolation via sPGD, employing HMH guarantees that computational costs are kept fairly low during the offline phase of the sPGD process.

sPGD has shown to be a highly effective method to create a surrogate reduced-order model of the VARM. It efficiently expresses the shaft displacement as a function based on unknown parameters, allowing fast retrieval of the associated value for any given parameter scenario over defined ranges. Moreover, it dynamically calculates the Jacobian matrix and automatically stores it, making it ready for use in the next optimization phase.

The LM optimization technique pairs effectively with sPGD. Firstly, it eliminates the need for time integration, which was omitted from the overall approach. Secondly, its most computationally intensive step, that is, the calculation of the Jacobian matrix, is already performed by sPGD. Lastly, since LM is an iterative process, extracting solutions from the sPGD library in fractions of a second significantly speeds up convergence.

In summary, the selected techniques work seamlessly to determine the unknown parameters of the VARM and construct the digital twin, facilitating precise and rapid computation of the system response.

The principal advantage of the devised method lies in its applicability to more complex systems. Initially, this was achieved by adapting the method used for the cantilevered beam problem to address a realistic model, specifically the VARM. In this section, we explore how this method can be further extended to solve an even more complex system like the Francis Turbine in this context. The suggested VARM shares multiple similarities with the Francis Turbine, which include the following:

1. Both systems contain a vertically flexible rotating shaft supported by bearings at both ends.
2. Lateral vibrations caused by whirling and rotational unbalance affect the shafts in both systems.
3. Unknown non-linear forces act on the bearings of both systems.
4. Both systems can be analyzed and diagnosed by measuring similar variables, such as shaft displacement, rotational speed, and acceleration.
5. Both systems may exhibit non-linear dynamic behavior, particularly under changing operational conditions and loads, complicating analysis and necessitating advanced modeling approaches.
6. Advanced computational modeling techniques enhance both systems by predicting performance, identifying potential problems, and optimizing design.

The Francis turbine is typically a more complex multi-physics system when compared to the VARM model, as it involves not only dynamic vibrations, but also intricate fluid-structure interactions due to water flow, electrical analysis in the generator, thermal analysis in the bearings, in addition to the lubrication film theory governing the behavior of tilted-pad journal bearings attached to the turbine's shaft.

Consequently, the development of a unified approach for constructing a digital twin of the entire turbine system, encompassing all its diverse subsystems, remains a goal for the future. Our focus was on the internal runner shaft that is attached to bearings at both ends. This shaft exhibits a behavior comparable to the VARM that we introduced in Chapter 5. Therefore, the suggested approach can be extended to analyze the turbine vertical shaft, enabling the estimation of the unknown bearing forces applied to it. These forces are difficult to quantify via sensors and hence must be deduced from data in a way analogous to the method used to infer the bearing forces on the VARM.

In turbines, phenomena such as swirling cavitation, generator unbalance, or uneven water flow resulting from blockage in the guide veins can produce effects akin to the unbalanced mass observed in the VARM. Also, in a standard Francis Turbine configuration, tilted-pad journal bearings are used. According to the literature, the forces exerted by these bearings are represented by equations with unspecified coefficients that can be analyzed using a method similar to the one presented in this project. When optimizing parameters for a Francis turbine, the use of additional sensors facilitates the simultaneous measurement of multiple variables, thereby enhancing accuracy. Among the variables that can be measured are shaft displacements, shaft acceleration, oil temperature, noise, and rotational speed, though this list is not exhaustive.

In summary, the suggested methodology is a preferred strategy for investigating a particular subsystem of the Francis turbine. It lays the groundwork for further examination of a comprehensive approach for constructing digital twins, thereby facilitating a more precise analysis and enhancement of Francis turbine systems.

6.2 Limitations

Despite its advantages, the method presents certain limitations, outlined in the points below:

1. This method is primarily employed to examine a harmonic response within the steady-state regime. This is because solving the problem in the frequency domain fails to capture transient responses, which typically requires time integration for analysis.
2. The approach presumes the governing PDE of the concerned system is explicitly defined, although it contains certain unknown parameters requiring estimation. If the system cannot be characterized by such a PDE, this method is inadequate.
3. The method necessitates that the unknown parameters fall within an anticipated range that encompasses the actual value.
4. The entire approach was designed for discrete excitation frequencies (rotational speeds in the case of the VARM), meaning the excitation frequency could not be parameterized. In Chapter 4, even though sPGD only represented the steady-state amplitude, this constraint arose from employing DEIM to approximate bases in the frequency domain. However, in Chapter 5, although DEIM was not implemented, the method still could not parameterize the excitation frequency. This was because sPGD represented the reduced order representation of a complete space-time response which is typically a sinusoidal signal, and attempting to generate an interpolated model of sinusoidal signals with varying frequencies is not possible with the proposed methods.
5. The approach was employed to examine a system that ended up having a response that can be completely characterized by a single mode, hence it was not evaluated on models with responses that are more complex and require multiple modes for response characterization.
6. Since the method does not rely on time integration, there is a restriction on the optimization techniques that can be applied. Consequently, DDDAS is excluded, preventing parameter adjustment using real-time data. Techniques relying on incremental time integration, such as Kalman filters, are incompatible with the suggested methods.

Additional constraints arose due to the experimental configuration of the VARM. These constraints are enumerated in the following points:

1. The maximum rotational speed of the stepper motor was below the machine's critical speed, hence we were limited to examining the VARM's behavior at speeds under the critical threshold.
2. Both bearings were ball bearings featuring a substantial preload, unsuitable for such a thin shaft, necessitating the use of variable eccentric masses at various rotational speed ranges to ensure the preload is counteracted without the risk of damaging the machine.
3. We had access to just two laser sensors, which enabled us to record the displacement at a single point on the shaft.
4. Following several tests, the shaft exhibited irreversible deformation and required multiple replacements to ensure data reliability.

Nevertheless, these constraints serve as a motivation for future studies in this area, which will be elaborated upon in the subsequent Chapter.

6.3 Main Contributions

This section outlines the key contributions of this thesis and links them to the objectives outlined in Chapter 3.

6.3.1 Objective 1: Development of a PGD framework to solve nonlinear dynamic problems

- Constructed a parameterized reduced-order surrogate model for a nonlinear dynamic system, consisting of a cantilevered beam with a nonlinear spring at its tip and subjected to a harmonic force at the same point.
- The surrogate model employed the sPGD technique to provide instant predictions of the parameterized system response.
- Integrated DEIM with the HMH solver to develop a faster solver through the construction of a reduced basis for nonlinear functions. This accelerated solver facilitated the production of the sparse dataset required for building the surrogate model using sPGD.
- The proposed strategy substantially decreased the time needed to compute the system response for a single parameter set in comparison to high-fidelity techniques like Newton-Raphson, albeit at the cost of carrying out the offline computations just once. Additionally, this method demonstrated greater efficiency for systems with a large number of DOFs.

6.3.2 Objective 2: ROM of the VARM using PGD

- Created a parameterized reduced-order surrogate model for a VARM using the methodology outlined in the preceding subsection.
- Applied two distinct methods to represent the unknown bearing forces: a geometric approach and an expansion approach, both characterized by nonlinear functions of displacements, velocities, and additional parameters.
- Demonstrated that the suggested approach could be extended to a more practical system, specifically the VARM, which included a greater number of unknown parameters.

6.3.3 Objective 3: Data-Driven Parameter Estimation towards a complete Digital Twin

- Used collected data to approximate the unknown parameters required for calculating the challenging-to-measure bearing forces.
- Developed a digital twin of the VARM using the optimized parameters capable of forecasting the machine's response for any set of parameters.

In conclusion, the contributions of this thesis successfully addressed certain gaps, providing novel solutions, and demonstrating practical applications.

CHAPTER 7 CONCLUSION

7.1 Summary of Works

In summary, this thesis presents a recipe for developing digital twins via reduced-order modeling and data assimilation, carried out in two phases. Initially, a global rapid solver is introduced that produces a reliable low-rank approximation of the system solution. This solver creates a sparse offline dataset, which is subsequently compressed to formulate a surrogate model capable of real-time predictions of the parameterized dynamic response utilizing sPGD. An academic example was used to validate the efficiency and reliability of the method. The approach was proved to be reliable and significantly reduces computational time after the offline computation phase.

Subsequently, the devised methodology was enhanced with data integrated from an experimental configuration of a VARM, which was exposed to non-linear bearing forces, rotational unbalance, and differing loading amplitudes coupled with rotational speeds, to construct its digital twin. This digital twin estimated the unknown parameters of the VARM and developed an extensive model capable of predicting the VARM response in diverse scenarios.

This approach can be applied to various engineering systems that have analogous motion equations, including Francis turbines. While the resulting Reduced-Order Model will be specific to the particular issue being addressed, the overall methodology remains consistent. A library of parameterized solutions is considered a vital element of virtual twins, which, when enhanced with real-time experimental data, can transform into a robust digital twin.

7.2 Future Research

Suggested future work ideas based on this thesis include the following points:

1. Including transient regime analysis which requires using an extra technique, contingent upon time integration, which can be added as an additional block within the general methodology. This may increase computational costs compared to the HMH approach; however, such costs are confined to the offline stage, as the online phase of sPGD consistently remains efficient. This is crucial for hydro-turbine applications because their operation necessitates frequent starting and stopping, leading to repetitive transient phases.
2. For components of the system that are challenging to depict with a PDE, machine

learning methods can be used to model these particular sections and integrate the outcomes into the sPGD framework.

3. The parametrization of excitation frequency offers a broad area for investigation. Adaptive interpolation techniques, like sliding window interpolation, can substitute sPGD to accomplish this.
4. Other optimization techniques may be tested.
5. Evaluate the approach using a more complex version of the VARM experimental setup incorporating these modifications:
   - A stronger motor to explore the behavior beyond the critical speed.
   - Various bearing types leading to distinct bearing force models.
   - Additional sensors to record additional parameters such as speed and acceleration.
   - A high-speed camera to document motion at all points along the shaft, resulting in a more reliable data set.
   - Altering the location of the eccentric mass introducing multiple modes into the solution.
6. Applying the method to solve for a system whose response needs multiple modes to be fully represented.
7. Applying the method to address a system comprising two or more blocks (multiple primary interacting operations), as seen in the situation with a Francis turbine.

REFERENCES

Adams, M. L. (2010). *Rotating Machinery Vibration: From Analysis to Troubleshooting.* CRC Press, Boca Raton, FL, second edition.

Aguado, J. V., Chinesta, F., Leygue, A., Cueto, E., and Huerta, A. (2013). Deim-based pgd for multi-parametric nonlinear model reduction. In *XII International Conference on Computational Plasticity: Fundamentals and Applications COMPLAS XII.*

Aguilera, D. C. (2017). *Stratégies numériques avancées pour la simulation efficace de procédés de soudage conventionnels et non conventionnels : Une approche de réduction de modèles.* Doctoral thesis, L'université Bretagne Loire, Rennes, France.

Ammar, A. (2009). The proper generalized decomposition: A powerful tool for model reduction. *International Journal of Material Forming*, 3:89–102.

Ammar, A., Mokdad, B., Chinesta, F., and Keunings, R. (2006). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139:153–176.

Ammar, A., Mokdad, B., Chinesta, F., and Keunings, R. (2007). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. part ii: Transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics*, 144:98–121.

Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. (2004). An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667–672.

Barricelli, B. R., Casiraghi, E., and Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7.

Bellman, R., Corporation, R., and Collection, K. M. R. (1957). *Dynamic Programming.* Rand Corporation research study. Princeton University Press.

Bickford, J., Bossuyt, D. L. V., Beery, P., and Pollman, A. (2020). Operationalizing digital twins through model-based systems engineering methods. *Systems Engineering*, 23:724–750.

Blasch, E., Ravela, S., and Aved, A. (2018). *Handbook of Dynamic Data Driven Applications Systems.* Springer International Publishing, Switzerland.

Boucinha, L., Gravouil, A., and Ammar, A. (2013). Space time proper generalized decompositions for the resolution of transient elastodynamic models. *Computer Methods in Applied Mechanics and Engineering*, 255:67–88.

Brunton, S. L. and Kutz, J. N. (2019). *Data-Driven Science and Engineering.* Cambridge University Press.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 113.

Burnett, D. S. (1988). *Finite element analysis: From concepts to applications.* Addison-Wesley.

Cameron, T. M. and Griffin, J. H. (1989). An alternating frequency/time domain method for calculating the steady-state response of nonlinear dynamic systems. *Journal of Applied Mechanics*, 56.

Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764.

Chen, W. J. and Gunter, E. J. (2007). *Introduction to Dynamics of Rotor Bearing Systems.* Eigen Technologies.

Chinesta, F., Ammar, A., and Cueto, E. (2010a). Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. archives of computational methods in engineering. *Archives of Computational Methods in Engineering*, 17.

Chinesta, F., Ammar, A., Cueto, E., Chinesta, F., and Ammar, A. (2010b). Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, pages 83–91.

Chinesta, F. and Cueto, E. (2014). *PGD-Based Modeling of Materials, Structures and Processes.* Springer.

Chinesta, F., Cueto, E. G., Abisset-Chavanne, E., Duval, J. L., and Khaldi, F. A. (2019). Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. *Archives of Computational Methods in Engineering.*

Chinesta, F., Keunings, R., and Leygue, A. (2014). *The Proper Generalized Decomposition for Advanced Numerical Simulations.* Springer International Publishing.

Chinesta, F., Ladevèze, P., and Cueto, E. (2011). A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18:395–404.

Chinesta, F., Leygue, A., Bordeu, F., Cueto, E., González, D., Ammar, A., and Huerta, A. (2013). Pgd-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20:31–59.

Clough, R. W. and Penzien, J. (2003). *Dynamics of Structures*. Computers and Structures, Inc.

Cremonesi, M., Néron, D., Guidault, P., Ladevèze, P., and Guidault, P.-A. (2013). A pgd-based homogenization technique for the resolution of nonlinear multiscale problems. *Computer Methods in Applied Mechanics and Engineer-ing*.

Cueto, E., González, D., and Alfaro, I. (2016). *Proper Generalized Decompositions - An Introduction to Computer Implementation with Matlab*. Springer International Publishing.

Diamond, T. W., Sheth, P. N., Allaire, P. E., and He, M. (2009). Identification methods and test results for tilting pad and fixed geometry journal bearing dynamic coefficients - a review. *Shock and Vibration*, 16:13–43.

Diamond, T. W., Younan, A., and Allaire, P. (2011). A review of tilting pad bearing theory. *International Journal of Rotating Machinery*, 2011.

Dumon, A., Allery, C., and Ammar, A. (2010). Proper generalized decomposition method for incompressible flows in stream-vorticity formulation. *European Journal of Computational Mechanics*, 19:591–617.

EERE (2021). How hydropower works. Office of Energy Efficiency and Renewable Energy. Accessed: November 29, 2021.

Evangeline, P. and Anandhakumar (2020). *Digital twin technology for "smart manufacturing"*, volume 117, pages 35–49. Academic Press Inc.

Fasi Ur Rahman (2021). Francis turbines - its components, working, and applications. The Constructor. Accessed: December 8, 2021.

Fasshauer, G. E. (2007). *Meshfree Approximation Methods with Matlab*. World Scientific.

Fletcher, R. (1971). Modified marquardt subroutine for non-linear least squares. Technical report, United Kingdom Atomic Energy Authority, United Kingdom.

Friswell, M. I., Penny, J. E. T., Garvey, S. D., and Lees, A. W. (2010). *Dynamics of Rotating Machines*. Cambridge University Press.

Fuller, A., Fan, Z., Day, C., and Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8.

Gavin, H. P. (2020). The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. Technical report, Department of Civil and environmental Engineering, Duke University, Durham, NC. accessed: March 15, 2022.

Germoso, C., Duval, J. L., and Chinesta, F. (2020). Harmonic-modal hybrid reduced order model for the efficient integration of non-linear soil dynamics. *Applied Sciences*, 10(19).

Ghnatios, C. (2021). A hybrid modeling combining the proper generalized decomposition approach to data-driven model learners, with application to nonlinear biphasic materials. *Comptes Rendus. Mécanique*, 349(2):259–273.

Ghnatios, C., Alfaro, I., González, D., Chinesta, F., and Cueto, E. (2019). Data-driven generic modeling of poroviscoelastic materials. *Entropy*, 21(12):1165.

Ghnatios, C., di Lorenzo, D., Champaney, V., Ammar, A., Elias, C., and Chinesta, F. (2023a). Optimal trajectory planning combining model-based and data-driven hybrid approaches. This is a preprint; it has not been peer reviewed by a journal.

Ghnatios, C., Kestelyn, X., Denis, G., Champaney, V., and Chinesta, F. (2023b). Learning data-driven stable corrections of dynamical systems—application to the simulation of the top-oil temperature evolution of a power transformer. *Energies*, 16(15):5790.

Ghnatios, C., Masson, F., Huerta, A., Leygue, A., Cueto, E., and Chinesta, F. (2012). Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213-216:29–41.

Ghnatios, C., Simacek, P., Chinesta, F., and Advani, S. (2020). A non-local void dynamics modeling and simulation using the proper generalized decomposition. *International Journal of Material Forming*, 13.

Glaessgen, E. H. and Stargel, D. S. (2012). The digital twin paradigm for future nasa and u.s. force vehicles. In *53rd Structures, Structural Dynamics and Materials Conference: Special Session on the Digital Twin*, Virginia, USA.

González, D., Cueto, E., and Chinesta, F. (2014). Real-time direct integration of reduced solid dynamics equations. *International Journal for Numerical Methods in Engineering*, 99:633–653.

Grieves, M. (2015). Digital twin: Manufacturing excellence through virtual factory replication. accessed: February 22, 2022.

Grolet, A. and Thouverez, F. (2012). On the use of the proper generalised decomposition for solving nonlinear vibration problems. In *International Mechanical Engineering Congress and Exposition*, Houston, United States.

Guilhaumon, C., Hascoët, N., Chinesta, F., Lavarde, M., and Daim, F. (2024). Data augmentation for regression machine learning problems in high dimensions. *Computation*, 12(2).

Gustavsson, R. K., Lundstrom, M. L., and Aidanpaa, J.-O. (2005). Determination of journal bearing stiffness and damping at hydropower generators using strain gauges. In *Proceedings of ASME Power Conference 2005*, Chicago, Illinois.

Halabi, F. E., González, D., Sanz-Herrera, J. A., and Doblaré, M. (2016). A pgd-based multiscale formulation for non-linear solid mechanics under small deformations. *Computer Methods in Applied Mechanics and Engineering*, 305.

Haris, M. M., Borzacchiello, D., Aguado, J. V., and Chinesta, F. (2018). Advanced parametric space-frequency separated representations in structural dynamics: A harmonic-modal hybrid approach. *Comptes Rendus Mécanique*, 346:590–602.

Haslam, A. H., Schwingshackl, C. W., and Rix, A. I. J. (2020). A parametric study of an unbalanced jeffcott rotor supported by a rolling-element bearing. *Nonlinear Dynamics*, 99:2571–2604.

He, J. and Fu, Z.-F. (2001). *Modal Analysis.* Butterworth-Heinemann.

Henneron, T., Benabou, A., and Clenet, S. (2012). Non linear proper generalized decomposition method applied to the magnetic simulation of a smc microstructure. *IEEE Transactions on Magnetics, Institute of Electrical and Electronics Engineers*, 48:3242–3245.

Hydro-Québec (2021a). Our mission and activities. Hydro-Quebec. Accessed: November 29, 2021.

Hydro-Québec (2021b). Quebec hydropower: clean, renewable and low in ghg emissions. Hydro-Quebec. Accessed: November 29, 2021.

Hydro-Québec (2021c). Turbines. Hydro-Quebec. Accessed: November 29, 2021.

Ibáñez, R., Abisset-Chavanne, E., Ammar, A., González, D., Cueto, E., Huerta, A., Duval, J. L., Chinesta, F., et al. (2018). A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. *Complexity*, 2018.

Ishida, Y. and Yamamoto, T. (2012). *Linear and Nonlinear Rotordynamics*. Wiley-VCH.

ISO (2020). *Surveillance et diagnostic d'état des machines-Groupes de production hydroélectrique Condition monitoring and diagnostics of machines-Hydroelectric generating units*. Switzerland.

Johannessen, K. (2015). The duffing oscillator with damping. *European Journal of Physics*, 36(6):065020.

JPE - High Tech Engineering (2025). Structural damping properties of mechanical systems. JPE. Accessed: 2025-01-27.

Kapteyn, M. G. and Willcox, K. E. (2020). Predictive digital twins: where dynamic data-driven learning meets physics-based modeling. In *Dynamic Data Driven Application Systems, Third International Conference*, Boston, MA, USA.

Karhunen, K. (1946). Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34.

Knight, D. (2003). Data driven design optimization methodology, a dynamic data driven application system. In *Proceedings of the 2003 International Conference on Computational Science*, pages 329–336.

Korsch, H. J., Jodl, H.-J., and Hartmann, T. (2008). *Chaos: A Program Collection for the PC*, chapter 8 - The Duffing Oscillator, pages 157–184. Springer Berlin Heidelberg.

Ladevèze, P. (1985). Sur une famille d'algorithmes en mécanique des structures. *Comptes-rendus des séances de l'Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l'univers, sciences de la terre*, 300(2):41–44.

Ladevèze, P. (1999). *Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation*. Springer-Verlag.

Ladevèze, P. and Chamoin, L. (2011). On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 200:2032–2047.

Ladevèze, P., Passieux, J.-C., and Neron, D. (2010). The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199:1287–1296.

Lauzeral, N., Borzacchiello, D., Kugler, M., George, D., Rémond, Y., Hostettler, A., and Chinesta, F. (2019). A model order reduction approach to create patient-specific mechanical models of human liver in computational medicine applications. *Computer Methods and Programs in Biomedicine*, 170:95–106.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer International Publishing.

Leygue, A. and Verron, E. (2010). A first step towards the use of proper general decomposition method for structural optimization. *Archives of Computational Methods in Engineering*, 17.

Loeve, M. (1948). Fonctions aléatoires du second ordre. *Lévy, Processus Stochastiques et Mouvement Brownien, Gauthier-Villars, Paris.*

Marghitu, D. B., Raju, P., and Mazilu, D. (2001). 6 - theory of vibration. In Marghitu, D. B., editor, *Mechanical Engineer's Handbook*, Academic Press Series in Engineering, pages 339–444. Academic Press, San Diego.

Meirovitch, L. (2001). *Fundamentals of Vibrations*. McGraw-Hill.

Meyrand, L., Sarrouy, E., Cochelin, B., and Ricciardi, G. (2019). Nonlinear normal mode continuation through a proper generalized decomposition approach with modal enrichment. *Journal of Sound and Vibration*, 443:444–459.

Montagud, S., Aguado, J. V., Chinesta, F., and Joyot, P. (2020). Parametric inverse impulse response based on reduced order modeling and randomized excitations. *Mechanical Systems and Signal Processing*, 135:1–15.

Nasselqvist, M. (2009). Simulation and characterization of rotordynamic properties for hydropower units. Master's thesis, Lulea University of Technology, Lulea, Sweden.

Nasselqvist, M., Gustavsson, R., and Aidanpaa, J.-O. (2014). Experimental and numerical simulation of unbalance response in vertical test rig with rilting-pad bearings. *International Journal of Rotating Machinery*, 2014.

Néron, 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. *International Journal for Numerical Methods in Engineering*, 103:275–292.

Niroomandi, S., Alfaro, I., Cueto, E., Alfaro, I., González, D., and Chinesta, F. (2013). Model order reduction in hyperelasticity: A proper generalized decomposition approach. *International Journal for Numerical Methods in Engineering*, 96.

Pruliere, E., Chinesta, F., and Ammar, A. (2010). On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. *Mathematics and Computers in Simulation*, 81:791–810.

Qin, Z., Talleb, H., and Ren, Z. (2016). A proper generalized decomposition-based solver for nonlinear magnetothermal problems. *IEEE Transactions on Magnetics*, 52.

Quaranta, G., Martin, C. A., Ibanez, R., Duval, J. L., Cueto, E., and Chinesta, F. (2019). From linear to nonlinear pgd-based parametric structural dynamics. *Comptes Rendus - Mecanique*, 347:445–454.

Rishmawi, S., Rodriguez, S., Chinesta, F., and Gosselin, F. P. (2024). Harmonic-modal hybrid frequency approach for parameterized non-linear dynamics. *Computers and Structures*, 301:107461.

Rodriguez, S., Di Lorenzo, D., Chinesta, F., Monteiro, E., Rebillat, M., and Mechbal, N. (2023a). Hybrid twin applied to structural health monitoring. In *X ECCOMAS Thematic Conference on Smart Structures and Materials (SMART 2023)*.

Rodriguez, S., Monteiro, E., Mechbal, N., Rebillat, M., and Chinesta, F. (2023b). Hybrid twin of rtm process at the scarce data limit. *International Journal of Material Forming*, 16(4):40.

Rotordynamic-Seal Research (2015). Rotordynamic-seal research.

Salas, A. H. S., Hernández, J. E. C., and Hernández, L. J. M. (2021). The duffing oscillator equation and its applications in physics. *Mathematical Problems in Engineering*, 2021.

Salinas, P., Pavlidis, D., Xie, Z., Adam, A., Pain, C. C., and Jackson, M. D. (2017). Improving the convergence behaviour of a fixed-point-iteration solver for multiphase flow in porous media. *International Journal for Numerical Methods in Fluids*, 84(8):466–476.

Sancarlos, A., Cameron, M., Abel, A., Cueto, E., Duval, J.-L., and Chinesta, F. (2021a). From rom of electrochemistry to ai-based battery digital and hybrid twin. *Archives of Computational Methods in Engineering*, 28:979–1015.

Sancarlos, A., Cameron, M., Le Peuvedic, J.-M., Groulier, J., Duval, J.-L., Cueto, E., and Chinesta, F. (2021b). Learning stable reduced-order models for hybrid twins. *Data-Centric Engineering*, 2.

Sancarlos, A., Champaney, V., Duval, J.-L., Cueto, E., and Chinesta, F. (2021c). Pgd-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit.

Sancarlos, A., Cueto, E., Chinesta, F., and Duval, J. L. (2021d). A novel sparse reduced order formulation for modeling electromagnetic forces in electric motors. *SN Applied Sciences*, 3:1–19.

Schilders, W. H. A., van der Vorst, H. A., and Rommes, J. (2008). *Model Order Reduction: Theory, Research Aspects and Applications*. Springer.

Serroud, S. (2023). Design of a vertical axis rotating machine for the development of a digital twin. Master's thesis, ÉcolePolytechnique de Montréal, Montréal, QC, Canada.

Shi, M., Wang, D., and Zhang, J. (2012). Nonlinear dynamic analysis of a vertical rotor-bearing system. *Journal of Mechanical Science and Technology*, 27:9–19.

Sohlberg, B. and Jacobsen, E. W. (2008). Grey box modeling - branches and experiences. In *Proceedings of the 17th World Congress - The International Federation of Automatic Control*, Seoul, Korea.

Tiwarti, R. (2006). A brief history of rotor dynamics and recent trends. In *51st Congress of Indian Society of Theoretical and Applied Mechanics (ISTAM) 2006*, Vishakhapatanam, India.

Torregrosa, S., Champaney, V., Ammar, A., Herbert, V., and Chinesta, F. (2022). Hybrid twins based on optimal transport. *Computers & Mathematics with Applications*, 127:12–24.

Torregrosa, S., Champaney, V., Ammar, A., Herbert, V., and Chinesta, F. (2024). Physics-based active learning for design space exploration and surrogate construction for multiparametric optimization. *Communications on Applied Mathematics and Computation*.

Turaga, R., Sekhar, A. S., and Majumdar, B. C. (1999). Comparison betwee linear and nonlinear transient analysis techniques to find the stability of a rigid rotor. *Journal of Tribology*, 121:198–201.

Vella, C., Gosselet, P., and Prudhomme, S. (2024). An efficient pgd solver for structural dynamics applications.

Vella, C. and Prudhomme, S. (2022). Pgd reduced-order modeling for structural dynamics applications. *Computer Methods in Applied Mechanics and Engineering*, 402.

Vidal, P., Gallimard, L., and Polit, O. (2012). Composite beam finite element based on the proper generalized decomposition. *Computers and Structures*, 102-103:76–86.

Vidal, P., Gallimard, L., and Polit, O. (2013). Proper generalized decomposition and layer-wise approach for the modeling of composite plate structures. *International Journal of Solids and Structures*, 50:2239–2250.

Wagg, D. J., Worden, K., Barthorpe, R. J., and Gardner, P. (2020). Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 6.

Water Turbine (2025). Water turbine — Wikipedia, the free encyclopedia. [Online; accessed 24-February-2025].

Wygant, K., Flack, R., and Barrett, L. (2004). Measured performance of tilting-pad journal bearings over a range of preloads - part ii: dynamic operating conditions. *Journal of Tribology Transactions*, 47:585–593.

Yan, S., Xu, X., Lyu, P., and Ren, Z. (2020). Application of pod and pgd for efficient parameter sweeping in frequency-domain full-wave problems. *IEEE Transactions on Magnetics*, 56.

Zlotnik, S., Diez, P., Modesto, D., and Huerta, A. (2014). Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING Int. J. Numer. Meth. Engng*, 00.

Zou, X., Conti, M., Diez, P., and Auricchio, F. (2018). A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113:230–251.

APPENDIX A BENDING VIBRATIONS OF BEAMS

The transverse vibrations of a cantilevered beam are determined by a boundary-value problem, which involves a partial differential equation with boundary conditions specified at both ends of the beam. Bending beams are intricate systems, characterized by a fourth-order differential equation and boundary conditions that include spatial derivatives up to the third order. The derivation of this differential equation is extensively discussed in the book by Meirovitch (2001).

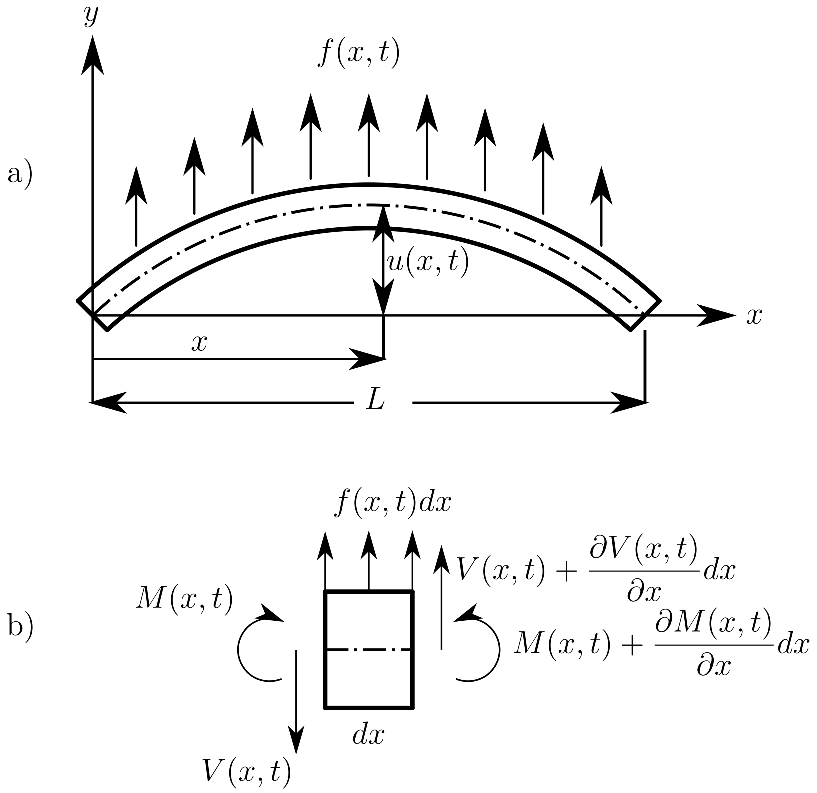

a)
y
$f(x,t)$
$u(x,t)$
x
x
L
b)
$f(x,t)dx$
$V(x,t) + \frac{\partial V(x,t)}{\partial x}dx$
$M(x,t)$
$M(x,t) + \frac{\partial M(x,t)}{\partial x}dx$
dx
$V(x,t)$

Figure A.1 (a) Beam in Bending Vibration, (b) Free-body Diagram of a Beam Element (Meirovitch, 2001)

We consider the beam in bending vibrations shown in Figure A.1a, where $u(x,t)$ is the transverse displacement induced by $f(x,t)$ which is the transverse force per unit length.

The beam we are studying is simple; it has a constant mass per unit length $\hat{m}$ and a constant flexural rigidity EI along its length, where E is the Young's modulus of the beam material, and I is the cross-sectional area moment of inertia.

The partial differential equation will be derived using the Newtonian approach. So, we will consider the free-body diagram of the beam differential element, shown in Figure A.1b. $M(x,t)$ represents the bending moment, and $V(x,t)$ represents the shear force.

Because of the high length-to-depth ratio of the beam, we can apply the Euler-Bernoulli beam theory. In this theory, we can ignore the rotation of the differential element since it is negligible in comparison to its translation. In addition, we can ignore the angular distortion due to shear, as it is relatively small compared to the bending deformation.

Using the free-body diagram shown in Figure A.1b, the force equation of motion in the vertical direction can be written as:

$$\left[V(x,t) + \frac{\partial V(x,t)}{\partial x}dx\right] - V(x,t) + f(x,t)dx = \hat{m}dx\frac{\partial^2 u(x,t)}{\partial t^2}, \qquad 0 < x < L. \tag{A.1}$$

The moment equation of motion about the z-axis, normal to x and y and passing through the center of the cross-sectional area is:

$$\begin{aligned}\left[M(x,t) + \frac{\partial M(x,t)}{\partial x}dx\right] - M(x,t) + &\left[V(x,t) + \frac{\partial V(x,t)}{\partial x}dx\right]dx \\ &+ f(x,t)dx\frac{dx}{2} = 0, \qquad 0 < x < L.\end{aligned} \tag{A.2}$$

Now, we eliminate all terms with higher orders of dx, and simplify equation (A.2) to get:

$$\frac{\partial M(x,t)}{\partial x} + V(x,t) = 0, \qquad 0 < x < L. \tag{A.3}$$

The next step is to divide equation (A.1) by dx, and substitute the second derivative of equation (A.3) with respect to x in it, to get:

$$-\frac{\partial^2 M(x,t)}{\partial x^2} + f(x,t) = \hat{m}\frac{\partial^2 u(x,t)}{\partial t^2}, \qquad 0 < x < L. \tag{A.4}$$

We are looking for an equation that exclusively relate the applied force $f(x,t)$ to the transverse displacement $u(x,t)$. However, we know from the mechanics of materials that the bending moment can be expressed as:

$$M(x,t) = EI\frac{\partial^2 u(x,t)}{\partial x^2}. \tag{A.5}$$

Using equations (A.3) and (A.5), we can express the shear force as follows:

$$V(x,t) = -EI\frac{\partial^3 u(x,t)}{\partial x^3}. \tag{A.6}$$

Substituting equation (A.5) in (A.4) we get the partial differential equation of bending vibrations of a beam in the general case:

$$-\frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 u(x,t)}{\partial x^2}\right] + f(x,t) = \hat{m}\frac{\partial^2 u(x,t)}{\partial t^2}, \qquad 0 < x < L, \tag{A.7}$$

which can be rewritten as follows:

$$\hat{m}\frac{\partial^2 u(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left[EI\frac{\partial^2 u(x,t)}{\partial x^2}\right] = f(x,t), \qquad 0 < x < L. \tag{A.8}$$

For the static case, we just need to eliminate the transverse acceleration term which gives:

$$\frac{d^2}{dx^2}\left[EI\frac{d^2 u(x)}{dx^2}\right] = f(x), \qquad 0 < x < L. \tag{A.9}$$

APPENDIX B FINITE ELEMENT SHAPE FUNCTIONS

One Dimensional Beam Shape Function

Considering a beam element with two nodes, where each node has two degrees of freedom; the beam deflection u and the slope of the deflection curve which we will denote by θ, we can express the deflection by a polynomial of the third degree as follows:

$$u^*(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \tag{B.1}$$

which can be written in matrix form as:

$$u^*(x) = \begin{Bmatrix} 1 & x & x^2 & x^3 \end{Bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix}. \tag{B.2}$$

The slope of the deflection curve can be expressed as:

$$\theta(x) = \frac{du^*(x)}{dx} = \begin{Bmatrix} 0 & 1 & 2x & 3x^2 \end{Bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix}. \tag{B.3}$$

The nodal values can then be calculated using equations (B.2) and (B.3):

$$\begin{gathered} u_1 = u^*(0) = a_0, \\ \theta_1 = \theta(0) = a_1, \\ u_2 = u^*(l) = a_0 + a_1 l + a_2 l^2 + a_3 l^3, \\ \theta_2 = \theta(l) = a_1 + 2a_2 l + 3a_3 l^2, \end{gathered} \tag{B.4}$$

which can be written in matrix form as:

$$\begin{Bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & 1 & 2l & 3l^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix}. \tag{B.5}$$

Now, using equation (B.5) we can express the coefficients of the polynomial in equation (B.1) as:

$$\begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & 1 & 2l & 3l^2 \end{bmatrix}^{-1} \begin{Bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{Bmatrix}. \tag{B.6}$$

By substituting the vector of coefficients in equation (B.2) we get:

$$u^*(x) = \begin{Bmatrix} 1 & x & x^2 & x^3 \end{Bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & 1 & 2l & 3l^2 \end{bmatrix}^{-1} \begin{Bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{Bmatrix}. \tag{B.7}$$

Since $u^*(x)$ can be expressed as $\{N_i(x)\}^T\{\Delta_i\}$, where $\{N_i(x)\}^T$ is the vector of shape functions and $\{\Delta_i\}$ is the vector of nodal values, then the shape functions can be expressed as:

$$\{N_i(x)\} = \begin{Bmatrix} 1 - 3\left(\frac{x}{l}\right)^2 + 2\left(\frac{x}{l}\right)^3 \\ l\left[\left(\frac{x}{l}\right) - 2\left(\frac{x}{l}\right)^2 + \left(\frac{x}{l}\right)^3\right] \\ 3\left(\frac{x}{l}\right)^2 + 2\left(\frac{x}{l}\right)^3 \\ l\left[-\left(\frac{x}{l}\right)^2 + \left(\frac{x}{l}\right)^3\right] \end{Bmatrix}. \tag{B.8}$$

Shape functions make it possible to interpolate between discrete values, here the nodal values.

One Dimensional Linear Shape Functions

If we consider a simple element with two nodes, each with a degree of freedom, we can define the interval between the two nodes by an interpolation function derived from a polynomial of the first degree as follows:

$$u^*(x) = a_0 + a_1 x, \tag{B.9}$$

which can be written in matrix form as:

$$u^*(x) = \begin{Bmatrix} 1 & x \end{Bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix}. \tag{B.10}$$

The nodal values can be calculated using equation (B.9):

$$\begin{aligned} u_1 &= u^*(0) = a_0, \\ u_2 &= u^*(l) = a_0 + a_1 l, \end{aligned} \tag{B.11}$$

which can be written in matrix form as:

$$\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix}. \tag{B.12}$$

Now, using equation (B.12) we can express the coefficients of the polynomial in equation (B.9) as:

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix}^{-1} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}. \tag{B.13}$$

Substituting the vector of coefficients in equation (B.10) we get:

$$u^*(x) = \begin{Bmatrix} 1 & x \end{Bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix}^{-1} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}. \tag{B.14}$$

Since $u^*(x)$ can be expressed as $\{N_i(x)\}^T\{\Delta_i\}$, where $\{N_i(x)\}^T$ is the vector of shape functions and $\{\Delta_i\}$ is the vector of nodal values, then the shape functions can be expressed as:

$$\{N_i(x)\} = \begin{Bmatrix} 1 - \dfrac{x}{l} & \dfrac{x}{l} \end{Bmatrix}^T. \tag{B.15}$$

APPENDIX C MODAL ANALYSIS OF VIBRATION SYSTEMS AND SOLVING THEM IN THE FREQUENCY DOMAIN

A comprehensive overview of the modal analysis is given by He and Fu (2001). A modal analysis framework is extensively used to demonstrate the technique discussed in the article presented in Chapter 4. Consequently, we will briefly review its primary components.

The motion of an undamped vibration system with multiple degrees of freedom can be expressed by the following equation:

$$\mathbf{M}\{\overrightarrow{\ddot{\mathbf{u}}}\} + \mathbf{K}\{\overrightarrow{\mathbf{u}}\} = \{\overrightarrow{\mathbf{F}}\}, \tag{C.1}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrices respectively, $\{\overrightarrow{\mathbf{u}}\}$ and $\{\overrightarrow{\ddot{\mathbf{u}}}\}$ are the displacement and acceleration vectors containing elements corresponding to each Degree of Freedom (DOF), and $\{\overrightarrow{\mathbf{F}}\}$ is the vector representing the external force(s) acting on the system.

This equation leads to the following eigenvalue problem:

$$(-\omega_j^2\mathbf{M} + \mathbf{K})\{\overrightarrow{\phi_{\mathbf{j}}}\} = \{\overrightarrow{\mathbf{0}}\}. \tag{C.2}$$

The eigenvalue problem yields n eigenvalues ω_j^2 and n eigenvectors $\{\overrightarrow{\phi_{\mathbf{j}}}\}$ with j ranging from 1 to n. The square root of the eigenvalues is the natural frequency of the system, and the eigenvectors are its mode shapes.

It is noteworthy that the mode shapes are not exclusive, as multiples of $\{\overrightarrow{\phi_{\mathbf{j}}}\}$ also fulfill equation (C.2). Consequently, $\{\overrightarrow{\phi_{\mathbf{j}}}\}$ are generally referred to as the primary modes or normal modes of the system. Furthermore, since the natural frequencies of a vibratory system are distinct and non-zero, the mode shapes associated with them can be considered independent.

The eigenvalues and eigenvectors of a system can be expressed in matrix form. The eigenvalues are typically listed in ascending order, and the eigenvectors are placed in columns in the same order. This creates a frequency matrix and a mode shape matrix:

$$\mathbf{\Omega^2} = \begin{bmatrix} \omega_1^2 & 0 & \dots & 0 \\ 0 & \omega_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_n^2 \end{bmatrix}, \tag{C.3}$$

$$\mathbf{\Phi} = \left[\{\overrightarrow{\phi_{\mathbf{1}}}\} \quad \{\overrightarrow{\phi_{\mathbf{2}}}\} \quad \ldots \quad \{\overrightarrow{\phi_{\mathbf{n}}}\} \right]. \tag{C.4}$$

Using equations (C.3) and (C.4), equation (C.2) can be written as follows:

$$\mathbf{K\Phi} = \mathbf{M\Phi\Omega^2}. \tag{C.5}$$

Similarly,

$$\mathbf{\Phi}^T\mathbf{K} = \mathbf{\Omega^2\Phi}^T\mathbf{M}. \tag{C.6}$$

The mode shape matrix is employed to link the spatial model to the modal model. According to He and Fu (2001), the mode shapes are mutually orthogonal with respect to the matrices **M** and **K**, implying that:

$$\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{M}\{\overrightarrow{\phi_{\mathbf{k}}}\} = 0 \quad \ldots \quad j \neq k, \tag{C.7}$$

$$\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{K}\{\overrightarrow{\phi_{\mathbf{k}}}\} = 0 \quad \ldots \quad j \neq k, \tag{C.8}$$

and

$$\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{K}\{\overrightarrow{\phi_{\mathbf{j}}}\} = \omega_j^2\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{M}\{\overrightarrow{\phi_{\mathbf{j}}}\} \quad \ldots \quad j = 1, 2, \ldots, n. \tag{C.9}$$

Let

$$\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{M}\{\overrightarrow{\phi_{\mathbf{j}}}\} = m_j, \tag{C.10}$$

and

$$\{\overrightarrow{\phi_{\mathbf{j}}}\}^T\mathbf{K}\{\overrightarrow{\phi_{\mathbf{j}}}\} = k_j, \tag{C.11}$$

then

$$\omega_j^2 = \frac{k_j}{m_j} \quad \ldots \quad j = 1, 2, \ldots, n. \tag{C.12}$$

In equations (C.10), (C.11), and (C.12) m_j and k_j are defined as the modal mass and the modal stiffness, respectively.

To sum up, the orthogonality of the system's mode shapes with respect to its mass and stiffness matrices enables the system to be diagonalized, making it simpler to solve.

The system response $\{\overrightarrow{\mathbf{u_t}}\}$ can be expressed in terms of the system mode shapes as follows:

$$\{\overrightarrow{\mathbf{u}}\} = \mathbf{\Phi}\{\overrightarrow{\mathbf{y}}\}, \tag{C.13}$$

where $\{\overrightarrow{\mathbf{y}}\}$ is a vector describing the modal displacements.

Substituting equation (C.13) in equation (C.1) and pre-multiplying by $\mathbf{\Phi}^T$ we get:

$$\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}\{\overrightarrow{\ddot{\mathbf{y}}}\} + \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}\{\overrightarrow{\ddot{\mathbf{y}}}\} = \mathbf{\Phi}^T\{\overrightarrow{\mathbf{F_t}}\}, \tag{C.14}$$

which can be written as follows in light of equations (C.10) and (C.11):

$$\begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m_n \end{bmatrix}\{\overrightarrow{\ddot{\mathbf{y}}}\} + \begin{bmatrix} k_1 & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_n \end{bmatrix}\{\overrightarrow{\mathbf{y}}\} = \mathbf{\Phi}^T\{\overrightarrow{\mathbf{F_t}}\}. \tag{C.15}$$

The matrices in equation (C.15) are diagonal, resulting in a system of n one-dimensional equations that can be solved directly. The response $\{\overrightarrow{\mathbf{u}}\}$ can then be obtained from equation (C.13). This same approach can be used to solve the equation of motion in either the time domain or the frequency domain.

Haris et al. (2018) demonstrated that the frequency domain approach is more attractive to implement in this case than time domain methods, as it eliminates the requirement of using time integration techniques.

We must employ the Fast Fourier Transform ($\mathcal{FFT}$) to convert the partial differential equation into the frequency domain to use the frequency method. This technique is also advantageous when a forced response is necessary, as the forcing term can be expressed as a combination of the frequencies taken into account. It is always possible to restore the time domain solution by using Inverse Fourier Transforms ($\mathcal{IFFT}$).

APPENDIX D DISCRETE EMPIRICAL INTERPOLATION METHOD (DEIM)

Constructing a reduced-order model of a non-linear system can be a difficult task due to the complexity of computing higher-order non-linear terms, which can be as hard as solving the initial problem before reduction. The empirical interpolation method (EIM) proposed by Barrault et al. (2004) and its discrete version, the discrete empirical interpolation method (DEIM) introduced by Chaturantabut and Sorensen (2010), address this issue by combining projection and interpolation.

The DEIM approach is based on constructing an approximation of the non-linear function $\{\overrightarrow{\mathbf{N}}\} \in \mathbb{R}^{n\times 1}$ using a reduced basis $\Xi_{\mathbf{N}p} \in \mathbb{R}^{n\times p}$, where p is much smaller than n. This reduced basis is used to interpolate the non-linear function, and a point selection operator $\mathbf{P} \in \mathbb{R}^{n\times p}$ is employed to select a few points in the domain where the non-linear function is evaluated.

Using DEIM, a non-linear function $f(x)$ can be represented as:

$$\{\overrightarrow{\mathbf{N}}\} = \Xi_{\mathbf{N}p}\{\overrightarrow{\mathbf{c}}\}, \tag{D.1}$$

where $\{\overrightarrow{\mathbf{N}}\} \in \mathbb{R}^{n\times 1}$ is the discrete version of the non-linear function $f(x)$, and $\{\overrightarrow{\mathbf{c}}\} \in \mathbb{R}^{p\times 1}$ is a vector of unknown amplitudes. These amplitudes can be determined once the reduced basis $\Xi_{\mathbf{N}p}$ is known.

According to the framework presented by Brunton and Kutz (2019), a non-linear function $\{\overrightarrow{\mathbf{N}}\}$ is calculated on a set of n_s states $\mathcal{X} = [\mathbf{x_1}, \cdots, \mathbf{x_{n_s}}]$ randomly chosen to span the domain of x. The evaluations of the non-linear function $\mathcal{N} = [\mathbf{N}(\mathbf{x_1}), \cdots, \mathbf{N}(\mathbf{x_{n_s}})]$ are then assembled into a matrix $n \times n_s$, referred to as a snapshot matrix. A lower-rank representation $\mathcal{N} \in \mathbb{R}^{n\times n_s}$ can be obtained by using Singular Value Decomposition (SVD) as follows:

$$\mathcal{N} = \Xi_{\mathbf{N}}\mathbf{\Sigma}_{\mathbf{N}}\mathbf{V}_{\mathbf{N}}{}^{*}. \tag{D.2}$$

The matrices $\Xi_{\mathbf{N}}$ and $\mathbf{V}_{\mathbf{N}}$ are both $n\times n$ and $n_s\times n_s$ in size, respectively, and have orthonormal columns. The operator * stands for the complex conjugate transpose. Additionally, $\mathbf{\Sigma}_{\mathbf{N}}$ is a $n \times n_s$ diagonal matrix with real positive entries in descending order.

A basis of rank p can be obtained from $\Xi_{\mathbf{N}}$, which contains the optimal basis that spans the non-linear function, forming:

$$\Xi_{\mathbf{N}p} = [\xi_{\mathbf{N_1}}, \xi_{\mathbf{N_2}}, \cdots, \xi_{\mathbf{N_p}}]. \tag{D.3}$$

The matrix $\Xi_{\mathbf{N}p}$ is a $n \times p$ matrix of real numbers. It should have a large enough number of columns, p, which is much smaller than the number of rows, n, in order to capture all the important features of the non-linear function.

Now that the matrix $\Xi_{\mathbf{N}p}$ is known, the system in equation (D.1) is highly over-determined. To determine the value of c, only p rows need to be taken into account. The purpose of the DEIM is to decide which p rows should be chosen.

The DEIM algorithm constructs a projection matrix $\mathbf{P}_p \in \mathbb{R}^{n \times p}$ with p columns, each of which is the γ_j^{th} column of an $n \times n$ identity matrix $\{\overrightarrow{\mathbf{e}_{\gamma_\mathbf{j}}}\} \in \mathbb{R}^n$. The matrix $\mathbf{P}_p = [\mathbf{e}_{\gamma_1}, \mathbf{e}_{\gamma_2}, \cdots, \mathbf{e}_{\gamma_p}]$ is chosen so that $\mathbf{P}_p^T \Xi_{\mathbf{N}p}$ is non-singular.

Now, pre-multiplying equation (D.1) by $\mathbf{P}_p^T$ and rearranging gives:

$$\{\overrightarrow{\mathbf{c}}\} = (\mathbf{P}_p^T \Xi_{\mathbf{N}p})^{-1} \mathbf{P}_p^T \{\overrightarrow{\mathbf{N}}\}. \tag{D.4}$$

Then substituting the value of $\{\overrightarrow{\mathbf{c}}\}$ in equation (D.1) gives:

$$\{\overrightarrow{\mathbf{N}}\} = \Xi_{\mathbf{N}p} (\mathbf{P}_p^T \Xi_{\mathbf{N}p})^{-1} \mathbf{P}_p^T \{\overrightarrow{\mathbf{N}}\}. \tag{D.5}$$

The benefit of this outcome is that the expression $\mathbf{P}_p^T \{\overrightarrow{\mathbf{N}}\}$ necessitates the calculation of the non-linear function $\{\overrightarrow{\mathbf{N}}\}$ at a much lower number of points, p, than before, thus drastically reducing the time and cost of computation.

The DEIM provides a methodical greedy approach to pick the indices of the previously mentioned p points. This algorithm is outlined in Algorithm 4.

Algorithm 4: DEIM algorithm to calculate the approximating basis of a non-linear function and its interpolation indices.

```
Input: A set of $n_s$ states randomly chosen to span the domain $x$.
Output: The projection matrix $\mathbf{P}_p$

procedure INITIALIZATION $(j = 1)(\mathcal{X})$

  $\mathcal{X} \leftarrow [\mathbf{x_1}, \cdots, \mathbf{x_{n_s}}]$
  $\mathcal{N} \leftarrow [\mathbf{N}(\mathbf{x_1}), \cdots, \mathbf{N}(\mathbf{x_{n_s}})]$
  $\mathcal{N} \leftarrow \mathbf{\Xi_N \Sigma_N V_N}^*$
  $\mathbf{\Xi_N}_p \leftarrow [\xi_{\mathbf{N_1}}, \xi_{\mathbf{N_2}}, \cdots, \xi_{\mathbf{N_p}}]$
  $\gamma_1 \leftarrow \max |\ \xi_{\mathbf{N_1}}\ |$
  $\mathbf{P}_1 \leftarrow \mathbf{e}_{\gamma_1}$

procedure INTERPOLATION INDICES $(j = 2, \cdots, p)(\mathbf{\Xi_N}_p)$

  for $j \leftarrow 2$ to $p$

    do
      $\mathbf{P}_j^T \mathbf{\Xi_N}_j \{\overrightarrow{\mathbf{c}}\}_j = \mathbf{P}_j^T \xi_{\mathbf{N_{j+1}}}$
      $\mathbf{R}_{j+1} \leftarrow \xi_{\mathbf{N_{j+1}}} - \mathbf{\Xi_N}_j \{\overrightarrow{\mathbf{c}}\}_j$
      $\gamma_j \leftarrow \max |\ \mathbf{R}_{j+1}\ |$
      $\mathbf{P}_{j+1} \leftarrow [\mathbf{P}_j, \mathbf{e}_{\gamma_j}]$

return $(\mathbf{P}_p)$
```

APPENDIX E SPARSE PROPER GENERALIZED DECOMPOSITION (SPGD)

Ibáñez et al. (2018) proposed the concept of sPGD by considering a continuous function of two parameters, $g(r, s) \in \mathbb{R}^2$, which needs to be reconstructed from sparse data. Within this context, the function $g(r, s)$ has its value determined for only a few sparse combinations of the parameters r and s. The objective is to create a surrogate model of this function, enabling interpolation to estimate its value for any combination of these two parameters. The initial step is to use Galerkin projection in the following way:

$$\int_{\Omega} w^*(r, s)(u(r, s) - g(r, s))drds = 0. \tag{E.1}$$

The domain $\Omega \subset \mathbb{R}^2$ is the two-dimensional area on which the function $g(r, s)$ is defined and $w^*(r, s)$ is an arbitrary test function. To approximate the unknown function $g(r, s)$, a set of two separate one-dimensional functions is constructed within a PGD context, resulting in $u(r, s)$:

$$u(r, s) = u^N(r, s) = \sum_{k=1}^{N} R^k(r)S^k(s). \tag{E.2}$$

The sPGD algorithm seeks to identify the shape of the pairs $R^k(r)S^k(s)$ for $k = 1, \cdots, N$, by projecting them onto a finite element basis and using a greedy approach in which the approximation up to the $(N-1)^{th}$ order is already known and the N^{th} order term is calculated in the current iteration with a fixed-point algorithm:

$$\begin{gathered} u^N(r, s) = u^{N-1}(r, s) + R^N(r)S^N(s) = \\ \sum_{k=1}^{N-1} R^k(r)S^k(s) + R^N(r)S^N(s). \end{gathered} \tag{E.3}$$

It is important to remember that the function $g(r, s)$ is only defined at a few points, so the product of the test function $w^*(r, s)$ and the function $g(r, s)$ is only calculated at the few known data points.

To create an approximation of $g(r, s)$, sPGD proposes projecting and then interpolating. If data is known at P sampling points (r_i, s_i), where $i = 1, \cdots, P$, then it is sensible to express the test function not in a finite element context but as a set of Dirac delta functions located at the sampling points, like this:

$$\begin{aligned} w^*(r,s) &= u^*(r,s)\sum_{i=1}^{P}\delta\left(r-r_i, s-s_i\right) \\ &= \Big(R^*(r)S^N(s)+R^N(r)S^*(s)\Big)\sum_{i=1}^{P}\delta\left(r-r_i, s-s_i\right). \end{aligned} \tag{E.4}$$

Then, equation (E.1) can be expressed as:

$$\begin{aligned} &\int_\Omega w^*(r,s)(u(r,s)-g(r,s))drds \\ &= \int_\Omega u^*(r,s)\sum_{i=1}^{P}\delta\left(r-r_i, s-s_i\right)\left(u(r,s)\right. \\ &\qquad\qquad \left.-g(r,s)\right)drds = 0, \end{aligned} \tag{E.5}$$

The last concept necessary to comprehend the entire sPGD concept is the finite element projection of the one-dimensional functions $R^k(r)$ and $S^k(s)$ with $k = 1, \cdots, N$ in equation (E.2). These functions are represented in matrix form as follows:

$$R^k(r) = \sum_{j=1}^{C} Q_j^k(r)\alpha_j^k = \Big[Q_1^k(r)\dots Q_C^k(r)\Big]\begin{bmatrix}\alpha_1^k \\ \vdots \\ \alpha_C^k\end{bmatrix}, \tag{E.6}$$

and

$$S^k(s) = \sum_{j=1}^{C} Q_j^k(s)\beta_j^k = \Big[Q_1^k(s)\dots Q_C^k(s)\Big]\begin{bmatrix}\beta_1^k \\ \vdots \\ \beta_C^k\end{bmatrix}, \tag{E.7}$$

The interpolant shape functions $Q_j^k(r)$ and $Q_j^k(s)$ can vary from standard piece-wise linear functions to global non-linear shape functions, maximum entropy interpolants, splines, and kriging functions. The coefficients α_j^k and β_j^k for $j = 1, \cdots, C$ must be determined using the greedy fixed-point algorithm. The quantity of shape functions C is equivalent to the amount of control points that are selected along each one-dimensional domain. This number influences the convergence of the sPGD algorithm. More details on the sPGD algorithm can be found in the work of Ibáñez et al. (2018).

APPENDIX F LEVENBERG-MARQUARDT OPTIMIZATION TECHNIQUE (LM)

In this section, we explain the detailed derivations of the Gradient-Descent and Gauss-Newton techniques for optimization and combine them to obtain the Levenberg-Marquardt formulation in light of Gavin (2020)'s work, where a suggested numerical implementation with codes can be found.

Assuming we have a model function $\hat{y}(x; \boldsymbol{p})$ of an independent variable x and a vector of n parameters $\boldsymbol{p}$, and a set of m measured data points $(x_i, y(x_i))$, then the least square error is represented by the sum of the weighted squares of the errors between the data and the function $\hat{y}(x; \boldsymbol{p})$:

$$\mathcal{L}(\boldsymbol{p}) = \sum_{i=1}^{m} \left[\frac{y\left(x_i\right) - \hat{y}\left(x_i; \boldsymbol{p}\right)}{\sigma_{y_i}} \right]^2, \tag{F.1}$$

where σ_{y_i} is the measurement error in $y(x_i)$.

Equation (F.1) can be expressed in a simplified matrix form as:

$$\mathcal{L}(\boldsymbol{p}) = (\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p}))^T \boldsymbol{W} (\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p})), \tag{F.2}$$

where $\boldsymbol{W}$ is defined as the weighting matrix, which is a diagonal matrix where $W_{ii} = \dfrac{1}{\sigma_{y_i}}$. In other words, $\boldsymbol{W}$ is set to be the inverse of the measurement error covariance matrix if it is known, or the values of its elements may be set to accomplish other curve fitting goals.

The minimization problem involves an iterative procedure to find the parameters $\boldsymbol{p}$ that minimize the loss function expressed in equation (F.1). This is traditionally done using the Gradient-Descent technique or the Gauss-Newton technique.

In the Gradient-Descent technique, the function is minimized by updating the parameters in the steepest-descent direction. This is a good method for simple functions with multiple parameters to optimize.

The algorithm starts with an initial guess of the parameter vector $\boldsymbol{p}$, then a parameter update $\boldsymbol{\Delta_{gd}} = \boldsymbol{p}_{i+1} - \boldsymbol{p}_i$ is calculated in each iteration as follows:

$$\boldsymbol{\Delta_{gd}} = \alpha \boldsymbol{J}^T \boldsymbol{W} (\boldsymbol{y} - \hat{\boldsymbol{y}}), \tag{F.3}$$

where $\boldsymbol{J}$ is an $m \times n$ Jacobian matrix $\left[\dfrac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{p}} \right]$ which represents the sensitivity of the function

$\hat{\boldsymbol{y}}$ to variations in the parameters $\boldsymbol{p}$, and α is called the learning rate and it controls the step in the steepest-descent direction.

Although this method can be helpful for multi-parameter cases, it has disadvantages such as being stuck in a local minimum or a saddle point, and being sensitive to initial parameter guesses and learning rates that may cause it to diverge.

In the Gauss-Newton technique, Taylor series expansion is used to estimate the value of the function $\hat{\boldsymbol{y}}$ at the updated parameter values $(\boldsymbol{p} + \boldsymbol{\Delta_{ng}})$ as follows:

$$\hat{\boldsymbol{y}}(\boldsymbol{p} + \boldsymbol{\Delta_{ng}}) \approx \hat{\boldsymbol{y}}(\boldsymbol{p}) + \left[\frac{\partial \hat{y}}{\partial \boldsymbol{p}}\right] \boldsymbol{\Delta_{ng}} = \hat{\boldsymbol{y}} + \boldsymbol{J}\boldsymbol{\Delta_{ng}}. \tag{F.4}$$

Now, substituting equation (F.4) in equation (F.1), $\mathcal{L}(\boldsymbol{p} + \boldsymbol{\Delta_{ng}})$ can be expressed as:

$$\mathcal{L}(\boldsymbol{p} + \boldsymbol{\Delta_{ng}}) = \boldsymbol{y}^T\boldsymbol{W}\boldsymbol{y} + \hat{\boldsymbol{y}}^T\boldsymbol{W}\hat{\boldsymbol{y}} - 2\boldsymbol{y}^T\boldsymbol{W}\hat{\boldsymbol{y}} - 2(\boldsymbol{y} - \hat{\boldsymbol{y}})^T\boldsymbol{W}\boldsymbol{J}\boldsymbol{\Delta_{ng}} + \boldsymbol{\Delta_{ng}}^T\boldsymbol{J}^T W\boldsymbol{J}\boldsymbol{\Delta_{ng}}. \tag{F.5}$$

Our goal is to find $\boldsymbol{\Delta_{ng}}$ that minimizes the loss function $\mathcal{L}(\boldsymbol{p} + \boldsymbol{\Delta_{ng}})$. This is done by evaluating $\boldsymbol{\Delta_{ng}}$ that solves the equation:

$$\frac{\partial}{\partial \boldsymbol{\Delta_{ng}}}\mathcal{L}(\boldsymbol{p} + \boldsymbol{\Delta_{ng}}) = -2(\boldsymbol{y} - \hat{\boldsymbol{y}})^T\boldsymbol{W}\boldsymbol{J} + 2\boldsymbol{\Delta_{ng}}^T\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J} = 0. \tag{F.6}$$

The resulting Gauss-Newton equations for parameter updating is:

$$\left[\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J}\right]\boldsymbol{\Delta_{ng}} = \boldsymbol{J}^T\boldsymbol{W}(\boldsymbol{y} - \hat{\boldsymbol{y}}), \tag{F.7}$$

which can be solved for $\boldsymbol{\Delta_{ng}}$.

Based on that, the Levenberg-Marquardt parameter update equation is expressed as:

$$\left[\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J} + \lambda\boldsymbol{A}\right]\Delta_{\mathrm{LM}} = \boldsymbol{J}^T\boldsymbol{W}(\boldsymbol{y} - \hat{\boldsymbol{y}}), \tag{F.8}$$

where λ is the coefficient that determines which path the algorithm takes. Gradient-Descent is applied using larger values of λ, while Gauss-Newton is applied using smaller values (Fletcher, 1971).

The matrix $\boldsymbol{A}$ can have various forms, the simplest being an identity matrix, and the most efficient containing only diagonal elements of the matrix $\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J}$ (Fletcher, 1971).