Model Reduction of Muscle-Driven Tissue Models

A Thesis Submitted to the

College of Graduate and Postdoctoral Studies

in Partial Fulfillment of the Requirements

for the degree of Master of Science

in the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Erik Widing

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9
Or
Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building
110 Science Place
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Biomechanical simulations are a necessary tool for a proper understanding of biomechanics and hence are subject to intense research. One field that relies on this research is articulatory speech synthesis as it attempts to simulate the physics of the speech production process. Out of the many aspects involved, muscle driven tissue is one of the most important as it is required to simulate the deformable structures of the vocal tract. Modelling of muscle driven tissue requires continuum models of high complexity for the purpose of accuracy. On the other hand, time-efficient models are desirable in order to provide fast simulations which enable the user to test input parameters interactively. These requirements impose limitations on each other as the time-efficiency of a model is reduced with increasing complexity, hence techniques that can bridge the gap between these requirements are needed.

This thesis attempts to bridge this gap through two major contributions. Model reduction techniques, that up until now have only been applied to inactive materials, have been implemented and tested for muscle driven tissue models. The implementation has been made in a general way to ensure that it can be used for biomechanical simulations in other fields than articulatory speech synthesis. In addition, the implementation has been made such that it can handle more advanced simulations than those investigated in this thesis. The simulations show acceptable but not ideal accuracy in both dynamic simulations and in measurements of equilibrium configurations. In addition, the reduced simulations using hyperreduction show good speedup for the more complex models investigated.

Preface

Parts of this thesis have been published elsewhere. The results presented in Sections 4.4.5 and Section 4.4.6 have been published in [32]. For these sections, I performed the simulations and the validation studies. Dr. Lloyd evaluated the computational speed-up and Dr. Stavness determined the muscle activations that were used.

Preliminary results for Section 4.3.4 have been published in [33]. For this paper, I wrote the source code for the reduced model and computed the reduced basis. The training for hyperreduction and the evaluation was performed by Dr. Lloyd.

For Chapter 3, I wrote all the source code for training, both for computation of reduced bases and for hyperreduction. I also wrote all the source code for the various procedures of sampling. In addition, I wrote parts of the source code for model reduction dynamics. The remaining part of the source code was written by Dr. Lloyd and Dr. Stavness.

For Chapter 4, I wrote the source code for the reduced models, performed the sampling and training, and ran all simulations. I also performed all evaluations, except for the computational speed-up (Table 4.5) presented in Section 4.4.5, which was performed by Dr. Lloyd. In addition, I wrote the source code for the modified modelling of muscles for the model presented in Section 4.4. The muscle properties were determined through testing, performed by myself and Dr. Stavness.

Acknowledgements

I would like to thank my supervisor, Dr. Ian Stavness, for giving me the opportunity of being a graduate student under his supervision. His constant encouragements and patience have made every day of work more productive and, in the case of bad days, more endurable. The lessons he has taught me about the importance of enthusiasm will be of great value to me for the rest of my life.

Throughout my work I have had the privilege of collaborating with Dr. John Lloyd, a researcher whose knowledge and guidance has been of vital importance to my work. It has been a true pleasure to collaborate with and learn from him.

I have acquired many great memories throughout my time at the Biomedical and Interactive Graphics (BIG) Lab and as a member of the OPAL modelling group. I thank all the members for the conversations and company that has made hard work a genuine pleasure.

To Agnes, whose adventure of education is about to begin

Contents

List of Tables

List of Figures

List of Abbreviations

3D	3-Dimensional
DOF	Degrees of Freedom
FEM	Finite Element Model
LMA	Linear Modal Analysis
MAE	Mean Absolute Error
POD	Proper Orthogonal Decomposition
SVD	Singular-Value Decomposition

List of Symbols

x	current position
$\mathbf{X}$	initial position
ϕ	shape-function
t	time
x_i	position of node i
n	number of nodes
N_i	local shape-function of node i
$\mathbf{X}_j$	rest position of node j
$\mathbf{F}$	shape gradient
$\mathbf{E}$	Green strain tensor
$\mathbf{C}$	right Cauchy-Green deformation tensor
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\mathbf{P}$	first Piola-Kirchhoff stress tensor
J	$det(F)$
$\mathbf{f}_i^{(e)}$	force acting on node i generated by element e
$\underline{\boldsymbol{\sigma}}$	$[\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]^T$
Ψ	strain energy density function
$\mathbf{x}$	position-state
$\mathbf{x}_0$	position-state at rest
$\mathbf{u}$	displacement-state
$\dot{\mathbf{u}}$	velocity-state
$\ddot{\mathbf{u}}$	acceleration-state
$\mathbf{M}$	mass-matrix
$\mathbf{f}$	force
$\mathbf{f}_{ext}$	external force
$\mathbf{f}_{int}$	internal force
$\mathbf{f}_{damping}$	damping force
$\mathbf{K}$	stiffness-matrix
α	mass-damping coefficient
β	stiffness-damping coefficient
$\mathbf{t}$	translation
θ	rotation
$\ddot{\mathbf{t}}$	translational acceleration
$\ddot{\theta}$	angular acceleration
$\mathbf{M}_{tt}$	translational mass-matrix
$\mathbf{M}_{\theta\theta}$	rotational mass-matrix
$\mathbf{M}_{t\theta}$	interaction mass-matrix between translation and rotation
$\mathbf{M}_{tu}$	interaction mass-matrix between translation and deformation
$\mathbf{M}_{\theta u}$	interaction mass-matrix between rotation and deformation
$\mathbf{f}_{ext,\mathbf{t}}$	translational component of external force
$\mathbf{f}_{ext,\theta}$	rotational component of external force
$\mathbf{f}_{qv,\mathbf{t}}$	translational component of centrifugal and Coriolis forces
$\mathbf{f}_{qv,\theta}$	rotational component of centrifugal and Coriolis forces
$\mathbf{f}_{qv}$	centrifugal and Coriolis forces
$\mathbf{f}_e$	force of element e
$\mathbf{K}_e$	stiffness-matrix of element e
$\mathbf{R}_e$	rotation matrix of element e
I_1	first invariant of $\mathbf{C}$
I_2	second invariant of $\mathbf{C}$
$\bar{I}_1$	$J^{-2/3} I_1$

$\bar{I}_2$	$J^{-4/3}I_2$
σ_{total}^{fibre}	total stress generated by muscle-fibre
σ_{max}	maximum stress of muscle-fibre
κ	muscle activation
λ	muscle-fibre stretching
l	muscle-fibre length
L	muscle-fibre length at rest
λ_{ofl}	optimal fibre stretch
f_{total}^{fibre}	normalized force generated by muscle-fibre
$f_{passive}^{fibre}$	passive force generated by muscle-fibre
f_{active}^{fibre}	active force generated by muscle-fibre
λ^*	fibre stretch at which the passive force becomes linear with the stretch
r	reduced degrees of freedom
$\mathbf{U}$	reduced basis
$\mathbf{q}$	reduced coordinates
$\dot{\mathbf{q}}$	reduced velocity
$\ddot{\mathbf{q}}$	reduced acceleration
$\bar{\mathbf{M}}$	reduced mass-matrix
$\bar{\mathbf{f}}$	reduced force
$\bar{\mathbf{f}}_{ext}$	reduced external force
$\bar{\mathbf{f}}_{int}$	reduced internal force
$\bar{\mathbf{f}}_{damping}$	reduced damping force
$\bar{\mathbf{K}}$	reduced stiffness-matrix
$\bar{\mathbf{M}}_{tq}$	reduced interaction mass-matrix between translation and deformation
$\bar{\mathbf{M}}_{\theta q}$	reduced interaction mass-matrix between rotation and deformation
$\bar{\mathbf{f}}_{qv}$	reduced centrifugal and Coriolis forces
$\mathbf{\Lambda}$	diagonal matrix containing eigenvalues from LMA
$\mathbf{U}_{modal}$	modal basis, reduced basis computed with LMA
Ω	volume of FEM
n'	number of elements
$\mathbf{g}$	per element internal force
$\bar{\mathbf{g}}$	per element reduced internal force
w_i	Hyperreduction weight of element i
$\bar{\mathbf{g}}_i^t$	per element reduced internal force of element i and training-sample t
$\bar{\mathbf{f}}_t$	reduced internal force of training-sample t
$\mathbf{A}$	training-matrix containing per element reduced internal forces
$\mathbf{b}$	training-vector containing reduced internal forces
$\mathbf{w}$	vector containing the hyperreduction weights
$\mathbf{X}_{\mathcal{F}}$	alternative training-matrix containing volume-weighted differences between the per element reduced internal forces and the reduced internal forces
W_i	volume of element i
$\mathbf{\Lambda}_{n'\times n'}$	$n' \times n'$-dimensional matrix resulting from SVD of $\mathbf{X}_{\mathcal{F}}$
$\mathbf{\Lambda}_{m\times n'}$	sub-matrix of $\mathbf{\Lambda}_{n'\times n'}$, that only contains force-data from m elements
$\sqrt{\mathbf{W}}_m$	$[\sqrt{W_1} \cdots \sqrt{W_m}]^T$
$\mathbf{J}$	$\begin{bmatrix}\mathbf{\Lambda}_{m\times n'} & \sqrt{\mathbf{W}}_m\end{bmatrix}^T$
$\boldsymbol{\beta}$	$\begin{bmatrix}\mathbf{0}^T & \Omega\end{bmatrix}^T$
$\boldsymbol{\alpha}$	solution to $\mathbf{J}\boldsymbol{\alpha} = \boldsymbol{\beta}$

Chapter 1

Introduction

1.1 Motivation

Biomechanical simulation is an important tool for a proper understanding of biomechanics. As many of the aspects of biomechanics include not just bones but also soft-tissue and, in particular, muscle tissue, biomechanical models need to include techniques for modelling deformable structures as well as rigid structures. Regardless of which biomechanical motion is being modelled, it is usually the result of complex interactions between muscle excitations as well as the rigid motions and deformations of bones and tissues. This means that many of the settings and inputs that are assigned to the model will have to be adjusted, either through automatic techniques or through interactive user input. The latter option however, requires that the simulations are fast in order for quick feedback on how the input influences the output.

Speech is a vital form of communication that is unique to the human species. The production of speech sounds in humans involves complex interactions between aerodynamics and biomechanics. Pressurized air from the lungs interacts with the vocal folds to create acoustical waves. The vocal tract (composed of the larynx, pharynx, palate, tongue, teeth and lips) forms a 3-dimensional (3D) tube that acts like an acoustic filter to transform the acoustical waves produced at the vocal folds into the sound that is radiated from the mouth-opening. The biomechanical side of speech production[20] involves bringing the vocal folds close together in order for them to interact properly with the airflow as well as dynamically shaping the 3D vocal tract through posturing of bones and soft tissues. This task is accomplished by a coordinated activation of many muscles for different organs along the vocal tract.

Speech has been a topic of intensive study for centuries and over the years extensive efforts have been made within speech research to build systems for artificial speech synthesis. This has been done with the use of a variety of techniques such as formant-based[49], concatenative[24] or HMM-based[62] speech synthesis. One of the most challenging and promising methods for speech synthesis that has been proposed is articulatory speech synthesis[4, 63]. The idea of this method is to simulate the physiological processes as well as the physics that occurs inside the human body during speech production. The hope is that this method can provide more meaningful results and more expressive synthetic speech sounds than other, more simple methods. Another advantage with articulatory speech synthesis is that it has the potential to be very speaker specific, for example through the use of medical scans. A great challenge however, is that there is a large number

of parameters, such as muscle activations or geometries of the involved structures, that need to be taken into account in order for the simulations to be accurate. One way to handle this challenge is to use models that run in real-time, which allows for interactive testing of the input parameters. This however, creates the additional challenge that the simulations need to be close to real-time while at the same time having a high level of accuracy for effective testing of the parameters. The tongue, being the most influential organ for the shape of the vocal tract and hence the acoustic filtering, is a central part of articulatory speech synthesis. For this reason, it is critical to have accurate 3D muscle-driven tissue models of the tongue that can be simulated in real-time.

Modelling the biomechanics of speech production requires a number of tools to account for the different biomechanical structures involved. The bones,although deformable, are usually modelled as rigid bodies and are therefore given 6 degrees of freedom (DOF). This can be made under the assumption that the deformations of the bones are small enough to be neglected. For soft tissues, however, this assumption cannot be made as the deformations in this case can be large. For this reason, soft tissues are usually modelled using finite element models (FEMs) which allows for deformations by dividing the model into many small elements for which the internal forces are computed. The modelling of muscles provides different possibilities depending on the structures it controls. One possibility, is to model the muscle as point-to point muscle, essentially acting like a spring whose parameters depend on the muscle activation, which, can be useful for muscles connecting different bones. Another possibility is to use muscle fibres embedded in the elements of an FEM, thus making the stress and stiffness of the FEM depend on the muscle activation.

In the case of speech production, there is a large number of bones and soft tissues involved. Many of the bones, such as the mandible, and the soft tissues, such as the tongue, also require a large number of muscles to be properly controlled. In addition, many of the models influence each other through attachments, direct or via muscles, or contact. All together, this creates complex interactions between many different muscle-controlled models, some being rigid bodies and some being FEMs. This causes models of this type of biomechanics to include large numbers of DOF and input parameters. The computational cost of simulating models with this many DOF is generally high which causes the simulations to be drastically slower as the models become more detailed.

When reducing a biomechanical model related to voice production, it might at first seem suitable to reduce the model into a 2D-model. This has been attempted for the tongue in connection with a static 2D vocal tract geometry[48] in a successful attempt to simulate vowel-vowel sequences. Although tempting, this option faces problems with compatibility since different structures have to match each other. This fact makes it difficult to reduce some models into 2D while letting some models stay in 3D; instead all models would have to be changed into 2D. This also proves problematic, since most structures in the vocal tract, such as the tongue and palate, only would make sense as 2D models in the sagittal plane while for structures like the vocal folds, the only logical option would be to make them into 2D models in the coronal plane. Hence, if one wants to combine all the structures involved in voice production, 3D models are required, and so the

reduction of the models must be made in other ways.

One way to reduce models is to assume that the position- and velocity-states are linear combinations of just a few standard deformations[50], referred to as a reduced basis. This assumption reduces the total DOF of the model to the number of standard deformations included, while still allowing the nodes to move in 3 dimensions. By choosing the reduced basis carefully, one can therefore maintain the accuracy of the model while reducing the DOF substantially which in turn speeds up the simulations. The standard deformations can be determined either through linear modal analysis[55], with the possibility of extending the basis to non-linear deformations[3, 64], or through sampling and analysis of model data at different deformations[31, 65]. Apart from the acceleration through reduction of the DOF, it is possible to accelerate the computation of the internal forces through hyperreduction[1], a technique with which the internal force is only computed in a subset of the elements, which further accelerates the simulations.

Up until now, only the most basic model reduction techniques have been applied to biomechanical simulations of tissue[44, 51] and in those cases, the tissue being simulated has consisted of passive material and has not been driven by muscles. The developed model reduction techniques have only been compared to a limited extent and none of the existing techniques have been tested or compared for muscle driven tissue models. For these reasons, an open source implementation of many of the existing model reduction techniques has been made for this thesis. This implementation has been made in such a way that it handles muscle driven tissue consisting of material of changing material properties. In addition, some of the different techniques have been compared for three different muscle driven tissue models, one 3D FEM beam and two 3D FEM tongue models.

1.2 Problem Description

Up to this point, model reduction has not been applied to muscle driven tissue models. In particular, it has not been applied to muscle driven FEMs, but instead only to FEMs consisting of passive materials. As a consequence, although open-source implementations of model reduction exist, they are not fully applicable to biomechanical simulations. In addition, the existing open source implementations of model reduction only include small subsets of the existing techniques and hence do not include the great diversity of techniques that have been developed. Additionally, comparisons between different model reduction techniques have often been limited and occasionally non-existent, even more so in the case of muscle driven FEMs for which the techniques have yet to be applied.

1.3 Objectives

There are three overall objectives of this thesis. First, to provide an open source implementation of model reduction that includes many of the existing model reduction techniques. Second, to apply model reduction

to FEMs that are driven by muscles instead of being passive which has been the case so far. Third, to compare the validity and effectiveness for some of these techniques when applied to muscle driven FEMs.

1.3.1 Implementation of Model Reduction in ArtiSynth

Although open source implementations of model reduction already exist[57], they only contain small subsets of the many existing techniques. For this reason, an open source implementation of model reduction that includes the many different techniques could be useful for further development of the field. One objective of this thesis is therefore to create an implementation in the biomechanical toolkit ArtiSynth which already includes FEMs and rigid bodies as well as a variety of muscle models. The implementation should include the standard model reduction where all forces, both internal and external, are computed as for a regular FEM while the equations of motion are solved in the reduced space. In addition, the implementation should include the option to compute the internal forces from a small subset of the elements. Since data consisting of displacements and forces are often needed to properly choose which standard deformations and what subsets of elements to use, the implementation should also include functionality to sample these entities from FEMs in an efficient way. In order to choose the standard deformations and the subsets of elements, the sampled data needs to be analysed, so the implementation should include a variety of training-algorithms.

1.3.2 Model Reduction applied to MuscleFemModel

So far, model reduction has only been applied to passive FEMs that have been subject to external perturbation such as gravity or pulling. However, in many biomechanical applications muscles are an important aspect of the simulations. Therefore another objective of this thesis is to make the implementation general enough to apply to a broad variety of biomechanical simulations, and to make it such that it takes muscles into account. The implementation should include functionality for FEMs controlled by muscles, both internal and external. In addition, the possibility of applying hyperreduction to muscle-controlled FEMs should be explored by applying it to muscle driven FEMs in ArtiSynth.

1.3.3 Comparison of Model Reduction Approaches

As new model reduction techniques have been developed, they have been compared to older techniques[64, 65]. The comparisons have focused on dynamic fidelity, or the accuracy of static deformations, or in some cases, the efficiency of the training algorithms. Since model reduction only has been applied to passive FEMs some of these comparisons are not necessarily valid when applying model reduction to muscle driven FEMs. For this reason, another objective of this thesis is to compare some of the implemented techniques for muscle driven FEMs in order to determine which of them are more appropriate for this application.

Chapter 2

Background

The main contribution of this thesis is providing tools for improved articulatory speech synthesis. Through articulatory speech synthesis, input parameters can be tested which in turn provides possibilities of deeper insight and characterization of speech to a larger extent than any other method [47]. The large number of input parameters, such as muscle activations, geometry and lung pressure, makes it desirable to use interactive simulations since that provides quick feedback regarding the effect of the input to the user. This chapter provides a brief review of Finite Element Modelling which is the main tool for modelling the tissues that are involved in speech production. The chapter also reviews model reduction, which is a set of techniques that have been developed to increase the simulation speed of FEMs.

Section 2.1 provides a basic description of FEMs in order for the reader to have a proper understanding about the advantages and limitations of this technique. This description includes the concept behind Finite Element Modelling as a way to simulate deformable models. The section will also cover the mathematical description of FEMs, both the description of the elements that are used to compute the internal forces as well as the formulation of the equations of motion which are used in order to convert the forces into deformations. Finally, the section will cover the complexity of simulating FEMs in order to demonstrate the need for additional techniques that can speed up the simulations.

Section 2.2 focuses on model reduction and provides the reader with the relevant background to understand the content of the following chapters. The section includes a description of the concept behind model reduction, as well as the mathematical description that follows from it. The complexity will be covered, as that is the main issue that drives further development of new techniques in model reduction. The relevant training-algorithms are also covered in order to provide a good understanding of the advantages and issues of the different options when using model reduction.

2.1 Finite Element Modelling

Finite Element Modelling has emerged as one of the most commonly used techniques to simulate deformable objects. The idea of this technique is to divide an object into many small elements of finite size. The elements are limited by nodes, that are the points at the elements corners, and edges that connects adjacent nodes of the elements. The nodes also serve as connections between adjacent elements and are therefore generally

connected to more than one element. By dividing a deformable object like this it is possible to create a discrete formulation of the otherwise continuous displacement-field of the object. This section provides a basic description of the concepts of Finite Element Modelling and overviews some of the limitations of this technique in order to clarify the benefits of model reduction. It also includes description of different material models in order to present a clear motivation for some of the more advanced model reduction techniques in the next section. For further reading, the reader is referred to [7] which provides a much more detailed description of the different aspects of Finite Element Modelling.

2.1.1 Mathematical Formulation

Strain

A central part of simulating FEMs is to compute the internal forces which is usually done through computation of the internal stress. For passive materials, which is the most common type of material, the stress arises from strains in the FEM and the specific relation between the two entities depends on which material model is being used. A brief survey of material models can be found in Section 2.1.2 while here, the focus will be on the mathematical description of strain. The points of an FEM are described by shape-functions that create a mapping between the initial position $\mathbf{X}$ and the current position x of any point of the FEM as shown below

$$x = \phi(\mathbf{X}, t) \tag{2.1}$$

A typical way to formulate the shape-function is to use local shape-functions for each node which allows each point to be described as

$$x = \sum_{i=1}^{n} x_i(t) N_i(\mathbf{X}) \tag{2.2}$$

where x_i is the position of node i, n is the number of nodes and N_i is a local shape-function that meets the condition

$$N_i(\mathbf{X}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{2.3}$$

where $\mathbf{X}_j$ is the rest position of node j. The formulation of the local shape-function depends on the type of element they are applied to.

A key quantity for defining strain is the shape gradient tensor

$$\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \nabla \phi \tag{2.4}$$

which is defined as the gradient of the shape-function.

One of the most commonly used definitions of strain is the Green strain tensor whose definition looks as

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - 1) \tag{2.5}$$

where $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ is the right Cauchy-Green deformation tensor. This tensor is important when mapping the differences in squared lengths before and after deformation. By doing this, the change in length and angles due to deformation can be accurately described simultaneously. The tensor also takes rotations of the FEM into account which makes it useful for more advanced material models that undergo large deformations.

Stress

As already stated, the stresses arising from certain strains depend on the material model of the FEM. Stress is usually defined through the Cauchy stress tensor $\boldsymbol{\sigma}$, which is a symmetric 3×3-matrix, where the diagonal elements are given by normal stresses while the off-diagonal elements are given by shear stresses. Another important measure of stress is the first Piola-Kirchhoff stress tensor which is defined as

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T} \tag{2.6}$$

where $J = det(\mathbf{F})$. This tensor can be loosely interpreted as the force in the deformed state per unit of undeformed area.

Once the Cauchy stress tensor is known for an element it can be used to compute the nodal forces generated by the element through

$$\mathbf{f}_i^{(e)} = \int_\Omega \mathbf{B}_i^T \underline{\boldsymbol{\sigma}} d\Omega_e \tag{2.7}$$

where i is the node, e is the element, $\underline{\boldsymbol{\sigma}} = [\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]^T$ is the vector form of the Cauchy stress tensor and $\mathbf{B}_i$ is given by

$$\mathbf{B}_i \equiv \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{pmatrix} N_i \tag{2.8}$$

In the case of hyperelastic materials, which will be discussed more in Section 2.1.2, the stress is computed through the strain energy density $\Psi(\mathbf{C})$. This class of materials is defined as conservative or path-independent, meaning that the strain energy density only depends on the current deformation. From this property, one can show that

$$\mathbf{P} = \frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{F}} \tag{2.9}$$

which directly relates stress to the strain energy density. By using this in Equation 2.6 on can express the Cauchy stress tensor as

$$\boldsymbol{\sigma} = \frac{1}{J}\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{F}}\mathbf{F}^T \tag{2.10}$$

which then makes it possible to compute the nodal forces from the strain energy density through Equation 2.7.

Dynamics

When simulating an FEM, the position and configuration of the model is updated through updates of the bodies position-state, $\mathbf{x}$, which is a $3n$-dimensional vector containing the positions of all nodes of the model. The position-state can be described by

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{u} \tag{2.11}$$

where $\mathbf{x}_0$ is the position-state at rest and $\mathbf{u}$ is the displacement-state.

In general, the displacement-state is used more in the computations than the position-state since it has a more direct connection to the changes in internal forces. One can also, without loss of generality, use the displacement state when solving the equations of motion due to the fact that $\mathbf{x}_0$ is constant which leads to

$$\dot{\mathbf{x}} = \dot{\mathbf{u}} \tag{2.12}$$

The equations of motion, when written in their most simple form, look as

$$\mathbf{M}\ddot{\mathbf{u}} = \mathbf{f} \tag{2.13}$$

In this equation, $\mathbf{M}$ is a diagonal $3n \times 3n$-matrix containing the masses of all the nodes along its diagonal. $\ddot{\mathbf{u}}$ and $\mathbf{f}$ are both vectors of size $3n$ that contains the acceleration and total force respectively.

The force acting on the FEM can be divided up into the external force, $\mathbf{f}_{ext}$, and the internal force, $\mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}})$. The external force is the sum of all external forces, local or global, acting on the nodes of the FEM. The internal force contains the internal forces, arising from the material properties and current deformation of the FEM, that acts on each node. Dividing the force in this way the equations of motion can be rewritten as

$$\mathbf{M}\ddot{\mathbf{u}} = -\mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}}) + \mathbf{f}_{ext} \tag{2.14}$$

The internal force itself can be divided up further, as shown below, into one part containing the internal force that arises due to deformation, $\mathbf{f}_{int}(\mathbf{u})$, and one part that arises due to damping, $\mathbf{f}_{damping}(\mathbf{u}, \dot{\mathbf{u}})$.

$$\mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{f}_{int}(\mathbf{u}) + \mathbf{f}_{damping}(\mathbf{u}, \dot{\mathbf{u}}) \tag{2.15}$$

$\mathbf{f}_{int}(\mathbf{u})$ is the internal force acting on each node that arises from deformations alone, being static or dynamic. $\mathbf{f}_{damping}(\mathbf{u}, \dot{\mathbf{u}})$ on the other hand is a force acting on each node that is moving, meaning that it only acts during dynamic deformations.

One commonly used model for the damping force in solids is the Rayleigh damping model shown below.

$$\mathbf{f}_{damping}(\mathbf{u}, \dot{\mathbf{u}}) = (\alpha \mathbf{M} + \beta \mathbf{K}(\mathbf{u}))\dot{\mathbf{u}} \tag{2.16}$$

In this model, the damping depends linearly on the velocity of the nodes. The coefficient matrix is given by a linear combination of the mass-matrix and the stiffness-matrix, $\mathbf{K}(\mathbf{u}) = \dfrac{\partial \mathbf{f}_{int}(\mathbf{u})}{\partial \mathbf{u}}$, which is a sparse, symmetric, positive-definite $3n \times 3n$-matrix that contains the stiffness of the FEM. With this formulation, the damping is a combination of mass-damping and stiffness-damping, with two non-negative coefficients α and β, that determine the amount of mass-damping and stiffness-damping respectively.

Multi-Body Dynamics

In the case of biomechanics, many simulations involve more than just one biomechanical structure. In these cases, the different structures are usually coupled either through joints, muscles or contacts. All these situations occur for the biomechanical simulations that are faced through Articulatory Synthesis with joints for example between the mandible and the maxilla, muscles for connecting for example bones to tissue, and contact between for example the tongue and the palate. For coupled multi-body systems like these, where the structures might experience large rigid motions, it can be desirable to write the equations of motion for each structure using the generalized Newton-Euler equations[56, 35] shown below

$$\begin{bmatrix} \mathbf{M}_{tt} & & \mathbf{sym} \\ \mathbf{M}_{t\theta} & \mathbf{M}_{\theta\theta} & \\ \mathbf{M}_{tu} & \mathbf{M}_{\theta u} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{t}} \\ \ddot{\theta} \\ \ddot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ext,\mathbf{t}} \\ \mathbf{f}_{ext,\theta} \\ \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{u}, \dot{\mathbf{u}}) \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{qv,\mathbf{t}} \\ \mathbf{f}_{qv,\theta} \\ \mathbf{f}_{qv} \end{bmatrix} \tag{2.17}$$

In this equation, the rigid motion of the body is separated from the deformations through the introduction of $\mathbf{t}$ and θ which are the translation and rotation of the body respectively. The matrix on the left-hand side of the equation is a symmetric $(6 + 3n) \times (6 + 3n)$-matrix where $\mathbf{M}_{tt} \in \mathbb{R}^{3\times3}$ is a diagonal matrix with each diagonal entry equal to the mass of the body, $\mathbf{M}_{\theta\theta} \in \mathbb{R}^{3\times3}$ is a symmetric matrix containing the moment of inertia of the body, $\mathbf{M}_{t\theta} \in \mathbb{R}^{3\times3}$ is a matrix containing the coupling between translation and rotation and $\mathbf{M}_{tu} \in \mathbb{R}^{3n\times3}$ and $\mathbf{M}_{\theta u} \in \mathbb{R}^{3n\times3}$ are matrices containing the coupling of deformation with translation and rotation respectively. On the right-hand side $\mathbf{f}_{ext,\mathbf{t}}$ and $\mathbf{f}_{ext,\theta}$ are the external forces of translation and rotation respectively while $\begin{bmatrix} \mathbf{f}_{qv,\mathbf{t}} & \mathbf{f}_{qv,\theta} & \mathbf{f}_{qv} \end{bmatrix}^T$ contains centrifugal and Coriolis forces.

2.1.2 Material Models

There are many different material models in use for different FEMs. The choice between the different models depends mainly on what type of behaviour the FEM is required to include. The first, and most simple

material, is linear material which is only a useful model when the FEM does not undergo large deformations. In this model the stiffness-matrix is assumed to be constant while at the same time it is assumed that the internal force is given by the negative product between the stiffness-matrix and the displacement-vector. From these assumptions, Equation 2.14 can be rewritten into

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} + (\alpha\mathbf{M} + \beta\mathbf{K})\dot{\mathbf{u}} = \mathbf{f}_{ext} \tag{2.18}$$

As already mentioned, linear material is limited to FEMs that only experience small deformations. The main reason for this is that elements that undergo large rotations start to experience large distortions that make the simulations quite inaccurate. One technique to solve this is co-rotated[39, 40, 42, 25, 43] linear material which takes the rotation of elements into account when evaluating the forces on the nodes. This is done by first finding the rotation matrix $\mathbf{R}_e$ of each element through polar decomposition of the deformation gradient. Once the rotation matrix has been obtained, the element force can be computed through the expression below.

$$\mathbf{f}_e = -\mathbf{R}_e\mathbf{K}_e\mathbf{R}_e^T\mathbf{x} + \mathbf{R}_e\mathbf{K}_e\mathbf{x}_0 \tag{2.19}$$

Out of the many existing material models one particular class of models will be mentioned here without going into the specific formulation. This class of models is hyperelastic models and it is mentioned here because of its extensive use when modelling tissue. This class of models includes among others St. Venant-Kirchhoff Material, Neo-Hookean Material[45], Mooney-Rivlin Material[38, 53], Yeoh Matrial[66], Ogden Material[46] and Fung Material[17]. Unlike Linear materials these models define the relation between stress and strain by formulating the energy density functions in terms of the Green strain. St. Venant-Kirchhoff Material is the simplest of these materials as it extends linear materials to apply for non-linear deformations while still keeping the material linear. The other materials are more advanced and are non-linear both in terms of deformation and material properties.

One common feature to model when working with FEMs is incompressible materials; that is materials whose volume does not change under deformation. This can be modelled with a variety of techniques that will be mentioned briefly here. These techniques all rely on computation of the volumes of all elements at every timestep. One option is to create element forces on the nodes directed such that the volume is kept constant. This would require registering small changes in the volume and the computation of forces that attempt to remove these changes. Another option is to create constraint equations that are solved at the same time as Equation 2.14. These constraints will automatically enforce the incompressibility as the node-displacements will be set by solving the equations.

Generalized Rivlin Material

Since all the models explored in this thesis (see Chapter 4) consist of versions of Generalized Rivlin Material[52], this material will be presented with somewhat more depth here. Generalized Rivlin Material (also called Poly-

nomial Hyperelastic Model) is a phenomenological material model of rubber elasticity. The strain energy density function for the incompressible case of this model is expressed as

$$\Psi(\mathbf{C}) = \sum_{p,q=0}^{M_1} C_{pq}(I_1 - 3)^p (I_2 - 3)^q \tag{2.20}$$

where $C_{00} = 0$, $I_1 = Tr(\mathbf{C})$ and $I_2 = tr(\mathbf{C}^T\mathbf{C})$ are the first and second invariants of $\mathbf{C}$, respectively. In the case of a compressible material, the strain energy density due to volumetric change is taken into account by modifying Equation 2.20 into

$$\Psi(\mathbf{C}) = \sum_{p,q=0}^{M_1} C_{pq}(\bar{I}_1 - 3)^p (\bar{I}_2 - 3)^q + \sum_{m=1}^{M_2} D_m (J-1)^{2m} \tag{2.21}$$

where $\bar{I}_1 = J^{-2/3} I_1$ and $\bar{I}_2 = J^{-4/3} I_2$, with $J = det(\mathbf{F})$ measuring the volumetric change.

Many of the different hyperelastic materials are simply special cases of this material. In each of these cases, the strain energy density is formulated based on subsets of the terms of Equation 2.20 or Equation 2.21, depending on if the material is compressible or not. For the case when only the linear terms of the first sum and the quadratic term of the second sum are considered one ends up with a Mooney-Rivlin Material. If, in addition, the linear term including I_2 is ignored, one ends up with a Neo-Hookean Material. Yeoh Material, on the other hand, corresponds to the case where all terms including I_2 are ignored. In addition, a case that is commonly used in biomechanics that is referred to as five parameter Mooney-Rivlin Material[41], includes the linear and quadratic terms only from both sums of Equation 2.21.

2.1.3 Muscle Control

For the application of tissue simulations, it is often desirable to have FEMs that are controlled by muscles. This type of control can be modelled in a variety of ways. A brief overview will be provided here for improved clarity. The simplest form of muscle control is external muscles that attach to the FEM. In this case the muscle force will be added to the external force of one or more nodes of the FEM. For internal muscles there are mainly two options, point-to-point muscles that connects different pairs of nodes or muscle fibres that are embedded inside elements. In the first case, the muscle forces can once again be added to the external forces of the nodes that are being connected by the muscles. In the other case, the muscles are modelled with transversely isotropic material that generates additional stress in the fibre direction of the muscle[11]. Since this model adds muscle forces as internal stresses of the elements, the muscle forces appear in the internal force and stiffness-matrix of the FEM.

The relation between the muscle activation and the stress generated by the muscle depends on the model of the muscle material. The simplest way to model the muscle material is to assume that the muscle stress of each fibre, σ_{total}^{fibre}, is proportional to the muscle activation as

$$\sigma_{total}^{fibre} = \sigma_{max}\kappa \tag{2.22}$$

where σ_{max} is the maximal muscle stress and κ is the muscle activation that ranges from 0 to 1.

Among the more advanced muscle material models, focus will be on the Blemker Muscle model[5] that is used for most of the models explored in this thesis. This model also takes into account the stretch, $\lambda = \frac{l}{L}$ where l is the current length and L is the length of the fibre at rest. The stress for each fibre is given by

$$\sigma_{total}^{fibre}(\lambda, \kappa) = \sigma_{max} f_{total}^{fibre}(\lambda, \kappa)\lambda/\lambda_{ofl} \tag{2.23}$$

where f_{total}^{fibre} is the normalized fibre force and λ_{ofl} is the optimal fibre stretch.

The normalized fibre force is divided into two parts as

$$f_{total}^{fibre}(\lambda, \kappa) = f_{passive}^{fibre}(\lambda) + \kappa f_{active}^{fibre}(\lambda) \tag{2.24}$$

where $f_{active}^{fibre}(\lambda)$ is the active muscle force that arises from activation but also depends on the stretch of the muscle and is given by

$$f_{active}^{fibre}(\lambda) = \begin{cases} 9(\lambda/\lambda_{ofl} - 0.4)^2 & \lambda \leq 0.6\lambda_{ofl} \\ 1 - 4(1 - \lambda/\lambda_{ofl})^2 & 0.6\lambda_{ofl} < \lambda < 1.4\lambda_{ofl} \\ 9(\lambda/\lambda_{ofl} - 1.6)^2 & \lambda \geq 1.4\lambda_{ofl} \end{cases} \tag{2.25}$$

$f_{passive}^{fibre}(\lambda)$ is the passive muscle force that arises solely from stretching of the muscle and is given by

$$f_{passive}^{fibre}(\lambda) = \begin{cases} 0 & \lambda \leq \lambda_{ofl} \\ P_1(e^{P_2(\lambda/\lambda_{ofl}-1)} - 1) & \lambda_{ofl} < \lambda < \lambda^* \\ P_3\lambda/\lambda_{ofl} + P_4 & \lambda \geq \lambda^* \end{cases} \tag{2.26}$$

where λ^* is the fibre stretch at which point the passive muscle force becomes linear with the stretch. The constants, P_3 and P_4, in Equation 2.26 are set with respect to P_1 and P_2 such that both $f_{passive}^{fibre}(\lambda)$ and its first derivative are continuous for all values of λ.

2.1.4 Complexity

Simulating FEMs dynamically require that Equation 2.13 be solved for every timestep. Given that the system is sparse, the complexity of solving it is $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$, which would have been the case if the system were dense. The consequence of this is that the solve time at each timestep increases quadratically with the number of nodes. This puts a limitation on the possibilities of performing fast simulations of detailed models and has therefore been an important topic in the field of computer graphics.

2.2 Model Reduction

As already stated in the previous section, the complexity of FEMs has been an important topic within the fields of computer graphics and computational engineering. One technique that has shown great success in handling this issue is model reduction. This section covers the basic formulation of this technique and how simulations of FEMs gain from it. It also covers the limitations of model reduction and how the technique has been developed to make further improvements. Model reduction relies on different types of training, and so this section will provide an overview of the different types of training that can be used.

2.2.1 Mathematical Formulation

This technique is based on the idea that the displacement-state of an FEM can be expressed as a linear combination of r standard deformations[50] where $r \ll n$. This can be expressed in matrix form as done below

$$\mathbf{u} = \mathbf{U}\mathbf{q} \tag{2.27}$$

where $\mathbf{U}$, referred to as the reduced basis, is a $3n \times r$-matrix whose column vectors contain the standard deformations, $\mathbf{q}$ is an r-dimensional vector containing the reduced coordinates. Through this formulation the DOF for the FEM is reduced from $3n$ to r.

Generally it is assumed that $\mathbf{U}$ is time-independent which makes it possible to formulate the reduced velocity-state of the FEM as is done below

$$\dot{\mathbf{u}} = \mathbf{U}\dot{\mathbf{q}} \tag{2.28}$$

By simply computing the time-derivative of this equation in the same way, one can compute the reduced acceleration as shown below

$$\ddot{\mathbf{u}} = \mathbf{U}\ddot{\mathbf{q}} \tag{2.29}$$

This formulation makes it possible to transform Equation 2.13 into the space of reduced coordinates. This is done by first inserting the expression of Equation 2.29 into Equation 2.13 and then multiplying all terms by $\mathbf{U}^T$ from the left side. From this, the reduced equations of motion can be written as done below

$$\mathbf{U}^T\mathbf{M}\mathbf{U}\ddot{\mathbf{q}} = \mathbf{U}^T\mathbf{f} \tag{2.30}$$

From this expression, it is possible to express the reduced mass-matrix $\bar{\mathbf{M}} = \mathbf{U}^T\mathbf{M}\mathbf{U}$ and the reduced force $\bar{\mathbf{f}} = \mathbf{U}^T\mathbf{f}$. By defining the quantities like this Equation 2.30 can be rewritten into

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} = \bar{\mathbf{f}} \tag{2.31}$$

Just like the reduced force, the reduced external force, $\bar{\mathbf{f}}_{ext} = \mathbf{U}^T\mathbf{f}_{ext}$, and the reduced internal force, $\bar{\mathbf{f}}_{int}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{U}^T\mathbf{f}_{int}(\mathbf{Uq}, \mathbf{U}\dot{\mathbf{q}})$, are both defined by multiplying the unreduced force by the transposed reduced basis from the left side. Likewise, the reduced internal force force due to deformations, $\bar{\mathbf{f}}_{int}(\mathbf{q}) = \mathbf{U}^T\mathbf{f}_{int}(\mathbf{Uq})$, and the reduced damping force, $\bar{\mathbf{f}}_{damping}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{U}^T\mathbf{f}_{damping}(\mathbf{Uq}, \mathbf{U}\dot{\mathbf{q}})$, are defined through the same transformation. By defining the reduced stiffness-matrix as

$$\bar{\mathbf{K}}(\mathbf{q}) = \frac{\partial \bar{\mathbf{f}}_{int}(\mathbf{q})}{\partial \mathbf{q}} = \mathbf{U}^T \frac{\partial \mathbf{f}(\mathbf{Uq})}{\partial \mathbf{q}} = \mathbf{U}^T\mathbf{K}(\mathbf{Uq})\mathbf{U} \tag{2.32}$$

the reduced damping force can be rewritten into

$$\bar{\mathbf{f}}_{damping}(\mathbf{q}, \dot{\mathbf{q}}) = (\alpha\bar{\mathbf{M}} + \beta\bar{\mathbf{K}}(\mathbf{q}))\dot{\mathbf{q}} \tag{2.33}$$

The reduced mass- and stiffness-matrix are both symmetric, dense $r \times r$-matrices which changes the computational cost of solving the equations of motion. While the complexity of solving Equation 2.13 is $\mathcal{O}(n^2)$ the complexity of solving Equation 2.30 is $\mathcal{O}(r^3)$, which is significantly lower for a proper choice of r in the case of complex models.

Multi-Body Dynamics

Just as for non-reduced FEMs, being able to simulate systems of coupled FEMs is an important topic for model reduction. It should be noted that it is possible to include rigid motions in the reduced basis in order to include rigid motions in reduced simulations. On the other hand, it is possible to formulate the reduced equations of motion in the same way as in Section 2.1.1 with the difference that the parts that relate to deformation need to be reduced. The reduced form[30] of Equation 2.17 is shown below

$$\begin{bmatrix} \mathbf{M}_{tt} & & \mathbf{sym} \\ \mathbf{M}_{t\theta} & \mathbf{M}_{\theta\theta} & \\ \bar{\mathbf{M}}_{tq} & \bar{\mathbf{M}}_{\theta q} & \bar{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{t}} \\ \ddot{\theta} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ext,\mathbf{t}} \\ \mathbf{f}_{ext,\theta} \\ \bar{\mathbf{f}}_{ext} - \bar{\mathbf{f}}_{int}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{qv,\mathbf{t}} \\ \mathbf{f}_{qv,\theta} \\ \bar{\mathbf{f}}_{qv} \end{bmatrix} \tag{2.34}$$

where the parts relating to rigid motion alone have not been changed. The blocks of the mass-matrix that handle coupling between rigid motion and deformation are given by $\bar{\mathbf{M}}_{tq} = \mathbf{U}^T\mathbf{M}_{tu}$ and $\bar{\mathbf{M}}_{\theta q} = \mathbf{U}^T\mathbf{M}_{\theta u}$, making the mass-matrix a $(6 + r) \times (6 + r)$-matrix. In the same way, the reduced deformation force related to centrifugal and Coriolis forces is defined as $\bar{\mathbf{f}}_{qv} = \mathbf{U}^T\mathbf{f}_{qv}$. Through this transformation of the Equation 2.17, one can still profit from the powerful formulation of the generalized Newton-Euler equations, while at the same time keeping all advantages of model reduction.

2.2.2 Reduced Basis

One of the key issues for model reduction is finding the reduced basis. Determining this basis can be done in a variety of ways depending on the characteristics of the model at hand. Generally, the different techniques

for constructing the reduced basis all create a basis that is orthonormal but it should be noted that it is only required that the modes included in the reduced basis are linearly independent.

Linear Material

The process of finding a reduced basis for an FEM consisting of linear material is far simpler than the respective process for other material. This is mainly because of the simplicity of the material model as well as the fact that FEMs of linear materials are not constructed to handle large deformations. The simplicity of linear materials make it possible to find the reduced basis for an FEM of such material by using Linear Modal Analysis (LMA)[55]. The first step of this analysis is to solve the generalized eigenvalue problem for the stiffness- and mass-matrix shown below

$$\mathbf{KU} = \mathbf{MU\Lambda} \tag{2.35}$$

After that, the reduced basis can be constructed from the eigenvectors corresponding to the smallest positive eigenvalues making it a purely modal basis. The amount of eigenvectors included in the reduced basis depend on the desired sensitivity. Because of the characteristics of the mass- and stiffness-matrix all the eigenvalues will be non-negative, in fact they will all equal the square of the natural frequency of their respective mode. If the FEM is constrained in any way, all the eigenvalues will be positive while if the FEM is unconstrained the six smallest eigenvalues will be 0 and correspond to rigid motion rather than deformation.

Because of the limitations of linear material models, LMA has some limitations of its own. The first limitation is directly related to the the fact that linear material models are not applicable to large deformations. For a modal basis, that means that the modes that include any rotation shows large distortions as their amplitude increases. Another limitation is that a modal basis is only applicable for its respective FEM as long as the FEM still uses a linear material model. In principle, it is possible to find a reduced basis for an FEM of any material model by using LMA on the stiffness-matrix at rest, but for a non-linear model, the stiffness-matrix will change as the FEM deforms, which will make the reduced basis invalid. Generally, the effect is that the amplitude of the motions would decrease, while at the same time the frequency would increase, making the FEM stiffer compared to its original characteristics. It is also worth noting that a modal basis when applied to large deformations of an FEM of linear material can be expected to reproduce the linear behaviour of the original model, unrealistic as it may be.

In the case of co-rotated linear materials, the same issues apply as those for non-linear material models when using a modal basis. However, since the material in this case is linear, it is possible to modify a modal basis so that it is able to preserve the original behaviour of the model. This is done with techniques called Modal Warping, which was first proposed by [10]. This technique uses the fact that by taking the curl of each mode, one can derive infinitesimal rotations for each node related to the activation of each mode. A more recent and more efficient variation of Modal Warping[23] uses the fact that the shape gradient can be decomposed into a symmetric part, the symmetric strain tensor, and an antisymmetric part representing a

small rotation. These tensors can then be used to transform a modal basis into a reduced basis that is able to handle large rotations.

Non-Linear Materials

Modal bases have difficulties handling large deformations as well as non-linear materials. Therefore significant efforts have been invested in developing techniques that can create reduced bases that are applicable for those cases. The development has essentially taken two approaches, one being to compute modal bases and then modify and expand them, and the other being to simulate FEMs and sample their states, and then create reduced bases based on the sampled data. It should be mentioned that the approach of modifying and extending a modal basis essentially makes it more tolerant to non-linear deformations, but does not necessarily make it tolerant to the type of deformations that non-linear material can experience. A reduced basis created from sampled data on the other hand, can be expected to handle the most significant deformations included in the sampling regardless if they are large or not.

The original idea on how to improve a modal basis suggested that the basis could be expanded by modes, called Modal Derivatives, that corresponds to the derivative of the stiffness-matrix [3]. The basic idea is that the modal basis corresponds to the linear term in a Taylor-expansion of the deformation-space and that the basis can be improved by adding modes corresponding to the second order term of the same expansion. These terms can be found by considering a case where the FEM is subject to a static load along its modal basis vectors as shown below

$$\mathbf{f}_{int}(\mathbf{u}(\mathbf{q})) = \mathbf{M}\mathbf{U}_{modal}\mathbf{\Lambda}\mathbf{q} \tag{2.36}$$

In this equation, $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues corresponding to each of the modes of the modal basis $\mathbf{U}_{modal}$. One can then take the derivative of this with respect to the reduced coordinates and obtain

$$\frac{\partial \mathbf{f}_{int}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{q}} = \mathbf{M}\mathbf{U}_{modal}\mathbf{\Lambda} \tag{2.37}$$

At $\mathbf{q} = \mathbf{0}$, one can identify $\frac{\partial \mathbf{f}_{int}}{\partial \mathbf{u}} = \mathbf{K}$ and compare the equation to Equation 2.35. By doing this, one realizes that $\frac{\partial \mathbf{u}}{\partial \mathbf{q}} = \mathbf{U}_{modal}$ for $\mathbf{q} = \mathbf{0}$, which confirms the idea that the modal basis correspond to linear terms in a Taylor-expansion.

By taking the derivative of Equation 2.37 with respect to $\mathbf{q}$, for which the right-hand side equals 0, one can rearrange the terms and once again set $\mathbf{q} = \mathbf{0}$ to obtain

$$\mathbf{K}\frac{\partial^2 \mathbf{u}}{\partial \mathbf{q}^2} = -(\frac{\partial \mathbf{K}}{\partial \mathbf{u}}\mathbf{U}_{modal})\mathbf{U}_{modal} \tag{2.38}$$

By solving this equation for $\frac{\partial^2 \mathbf{u}}{\partial \mathbf{q}^2}$, one can obtain the Modal Derivatives, a new set of basis vectors that correspond to the second order terms of the Taylor-expansion. It is worth noting that because of the

symmetry of second order derivatives, a modal basis with r modes will be expanded to a reduced basis with $\frac{r(r+1)}{2}$ modes through this procedure. A limitation of this technique is that the derivative of the stiffness-matrix is required. This is usually solved by perturbing the different reduced coordinates and computing the derivative of the stiffness-matrix numerically. This however, causes a limitation since now the model has to be simulated in order for a basis to be computed.

In later work, an alternative technique[64] for how to expand modal bases has been presented. This technique takes a more geometrical approach, as it recognizes that the node motions of each mode of a modal basis can be transformed with 9 linearly independent affine 3D-transformations. Through this, one can produce a basis with 9 times the number of modes of the modal basis. After that, a Gram-Schmidt process can be applied to the basis to eliminate redundant modes and to make sure that the new reduced basis is orthonormal. Compared to modal derivatives, this is a simpler process that does not require any more knowledge about the FEM beyond its modal-basis. It is also worth noting that for high values of r, this technique produces fewer new modes than what the modal derivatives do. Finally, when the two techniques have been compared, this geometrical approach has proved to be more accurate for the same number of modes.

For the approach of sampling model states, the reduced basis is generally referred to as a proper orthogonal decomposition (POD). There are mainly two techniques for how to perform the actual sampling, while at the same time there are mainly two options for what data to sample. The simplest sampling technique is to simulate the FEM with similar input to what will be used when simulating the reduced model. The samples are taken by simply taking "snapshots" of the model at different timesteps[31]. A more advanced technique is to use interactive sketching[3, 27] in which case the FEM is highly damped to prevent free oscillations. This causes the model to be at equilibrium as long as no additional forces are applied and allows the user to perturb the FEM interactively and sample model states whenever a desired configuration is achieved.

The far most common type of data to sample is the displacements of all the dynamic nodes of the FEM. In this case, the reduced basis is computed through Singular-Value Decomposition (SVD), where the modes with the largest significance are included in the basis. A more recent approach[65] is to sample not only displacements, but also the internal forces of all dynamic nodes as well as the stiffness-matrix of the FEM for every sample. The reason to sample the internal forces and stiffness-matrices, is to compute the tangent spaces between the different samples using the Hermitian Lanczos algorithm[54] and add the tangents to the displacement-data. After the tangent spaces have been added to the displacement-data, the reduced basis can be computed through SVD in the same way as for data consisting of only displacements. An advantage of this approach is that fewer samples are needed (typically less than 10) to produce the reduced basis since more information is extracted for each sample.

Apart from the approaches already mentioned, a technique[29] has also been presented where the FEM is simulated as a non-reduced FEM at first and as the simulation goes along, a reduced basis is assembled from the different configurations produced during the simulation. With the reduced basis being assembled, the

FEM is simulated by use of model reduction for most timesteps while being simulated as a non-reduced FEM for just a few timesteps. The reduced basis is continuously updated by sampling the displacement-state at the timesteps where the non-reduced formulation is used, and then performing modified Gram-Schmidt[21]. The number of reduced steps being taken between each non-reduced step is determined based on the relative error of the velocity-state each time the reduced basis is being updated. With this approach, some modes will have to be removed from the reduced basis continuously for it to maintain an efficient size. This is done by performing an SVD on the most recent reduced states and then transforming the reduced basis into the most significant directions of those states.

2.2.3 Hyperreduction

The reduction of a FEM removes $\mathcal{O}(n^2)$ from the computational cost by changing the complexity of solving the equations of motion. Beneficial as this may be, it does not remove all dependency of n from the computational cost, mainly because of the computation of the reduced internal force. The reduced internal force is computed through

$$\bar{\mathbf{f}}_{int}(\mathbf{q}) = \mathbf{U}^T \mathbf{f}_{int}(\mathbf{U}\mathbf{q}) = \mathbf{U}^T \int_{\Omega} \mathbf{g}(\mathbf{X}, \mathbf{U}\mathbf{q}) d\Omega_{\mathbf{X}} = \mathbf{U}^T \sum_{i=1}^{n'} \mathbf{g}(\mathbf{X}_i, \mathbf{U}\mathbf{q}) \tag{2.39}$$

where Ω is the volume of the FEM, n' is the number of elements and $\mathbf{g}(\mathbf{X}, \mathbf{u})$ is the internal force density for point $\mathbf{X}$ and displacement $\mathbf{u}$. By moving $\mathbf{U}^T$ into the sum and defining the reduced internal force density as $\bar{\mathbf{g}}(\mathbf{X}, \mathbf{q}) = \mathbf{U}^T \mathbf{g}(\mathbf{X}, \mathbf{U}\mathbf{q})$ the reduced internal force can be expressed as

$$\bar{\mathbf{f}}_{int}(\mathbf{q}) = \sum_{i=1}^{n'} \mathbf{U}^T \mathbf{g}(\mathbf{X}_i, \mathbf{U}\mathbf{q}) = \sum_{i=1}^{n'} \bar{\mathbf{g}}(\mathbf{X}_i, \mathbf{q}) \tag{2.40}$$

Since both these expressions for the reduced internal force use sums over all the n' elements, and $n' \sim n$, there is still a complexity of $\mathcal{O}(n)$ for every timestep. This creates a limitation on the speed-up that can be made with model reduction for complex FEMs and has hence been a topic of research for some time.

For linear material models, this is hardly an issue since, the force computations are fast for such a simple model. If the material is not co-rotated, the reduced stiffness-matrix can be computed in advance of the simulations and therefore the unreduced force does not have to be computed. In the case of St.Venant-Kirchhoff material, it has been shown that the strain energy for any deformation is a fourth order polynomial of the components of the displacement-state. Since the internal force equals the gradient of the strain energy, the reduced internal force can be written as a third order polynomial of the reduced coordinates[3]. The coefficients of the polynomial can be obtained by applying standard FEM formulas for the unreduced internal forces of St.Venant-Kirchhoff material[9] and then reducing the obtained coefficients through pre-multiplication by $\mathbf{U}^T$. Unfortunately, this technique has a computational cost of $\mathcal{O}(r^4)$, which makes it unsuitable for reduced FEMs with many DOF. Furthermore, these techniques are bound to be limited to their specific material, meaning that a need for more general techniques has been developed.

Because of the limitations of the techniques for computing the reduced internal force already described, efforts have been made to find more general techniques with low computational cost. Hyperreduction[1, 13, 14, 15] is a technique that fulfills these criteria in that it is applicable to any material and offers a solution to reducing the time to compute the reduced internal force. This technique takes inspiration from Key-Point Subspace Acceleration (KPSA)[36] which is used to accelerate animations without force-computations, as well as Precomputed Acoustic Transfer (PAT)[26] which is used to to simulate sound-generation from vibrating objects. The idea behind this technique is that since the FEM through model reduction is reduced to r DOF, it should be enough to compute the internal force for a subset of the elements where the size of the subset is proportional to r. The reduced internal force of the reduced FEM can then be computed through

$$\bar{\mathbf{f}}_{int}(\mathbf{q}) \approx \sum_{i=1}^{m} w_i \bar{\mathbf{g}}(\mathbf{X}_i, \mathbf{q}) \tag{2.41}$$

where w_i is the respective, non-negative weight of each element in the subset. Likewise, the reduced stiffness-matrix can be computed with the same set of elements using the expression below

$$\bar{\mathbf{K}}(\mathbf{q}) = \frac{\partial \bar{\mathbf{f}}_{int}(\mathbf{q})}{\partial \mathbf{q}} \approx \sum_{i=1}^{m} w_i \frac{\partial \bar{\mathbf{g}}(\mathbf{X}_i, \mathbf{q})}{\partial \mathbf{q}} \tag{2.42}$$

Since the unreduced stiffness-matrix is positive definite, it is important that all weights are non-negative in order to preserve this property for the reduced stiffness-matrix. With $m \propto r$, the $\mathcal{O}(n)$-behaviour is removed from the computation of the reduced internal force and almost entirely removed from the computational cost at every timestep. In fact, the most computationally expensive computation is the computation of the stiffness-matrix, which is $\mathcal{O}(r^3)$ with this formulation.

More recently, this concept has been developed further and been applied to self-collision[61] as well. In this case, the hyperreduced FEM has used a small subset of surface points to detect self-collision and to compute the reduced collision force. Interestingly, the attempt to use a universal subset of surface points for all cases of self-collision was not successful. Instead, a case-based hyperreduction was developed, where different subsets of surface-points were used depending on the the current reduced state.

Training-Algorithms

The main issue for hyperreduction is what subset of elements to use and what weights to assign the different elements. Training-algorithms designed for this issue all involve solving either a non-negative least-squares problem or just a least-squares problem where the matrix and the right-hand side vector consist of sampled force-data. Generally each column of the matrix corresponds to a specific element and consists of reduced internal force densities for different training samples. The right-hand side vector generally consist of reduced internal forces.

Greedy Cubature[1], which is the first algorithm that was proposed along with the idea of hyperreduction, is analogous to the algorithm used for PAT-training[26]. This algorithm formulates the non-negative least-

squares problem shown below

$$\mathbf{A}\mathbf{w} = \mathbf{b}, \quad \mathbf{A} = \begin{pmatrix} \frac{\bar{\mathbf{g}}_1^{(1)}}{\|\bar{\mathbf{f}}_1\|} & \cdots & \frac{\bar{\mathbf{g}}_i^{(1)}}{\|\bar{\mathbf{f}}_1\|} & \cdots & \frac{\bar{\mathbf{g}}_m^{(1)}}{\|\bar{\mathbf{f}}_1\|} \\ \vdots & & \vdots & & \vdots \\ \frac{\bar{\mathbf{g}}_1^{(t)}}{\|\bar{\mathbf{f}}_t\|} & \cdots & \frac{\bar{\mathbf{g}}_i^{(t)}}{\|\bar{\mathbf{f}}_t\|} & \cdots & \frac{\bar{\mathbf{g}}_m^{(t)}}{\|\bar{\mathbf{f}}_t\|} \\ \vdots & & \vdots & & \vdots \\ \frac{\bar{\mathbf{g}}_1^{(T)}}{\|\bar{\mathbf{f}}_T\|} & \cdots & \frac{\bar{\mathbf{g}}_i^{(T)}}{\|\bar{\mathbf{f}}_T\|} & \cdots & \frac{\bar{\mathbf{g}}_m^{(T)}}{\|\bar{\mathbf{f}}_T\|} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \frac{\bar{\mathbf{f}}_1}{\|\bar{\mathbf{f}}_1\|} \\ \vdots \\ \frac{\bar{\mathbf{f}}_t}{\|\bar{\mathbf{f}}_t\|} \\ \vdots \\ \frac{\bar{\mathbf{f}}_T}{\|\bar{\mathbf{f}}_T\|} \end{pmatrix}, \quad w_i \geq 0 \tag{2.43}$$

where T is the number of training-samples. Since both $\bar{\mathbf{f}}$ and $\bar{\mathbf{g}}$ are r-dimensional vectors, $\mathbf{A}$ is a $rT \times m$-dimensional matrix and $\mathbf{b}$ is a vector of size rT. The algorithm works in iterations where one element is added to the subset at every iteration. At the start of each iteration, the residual of the solution from the previous iteration is computed. A set of randomly selected elements are compared, and the one which will decrease the residual the most, that is the element whose column vector of $\mathbf{A}$ is most parallel to the residual, is added to the subset. At the end of each iteration, the weights are computed by solving Equation 2.43 and the next iteration will only start if the relative error, given by $|\mathbf{A}\mathbf{w} - \mathbf{b}|/|\mathbf{b}|$, is above a specified limit. This algorithm has also been used to reassign the subset of elements after a change of the reduced basis in the middle of simulations[29]. It is worth noting that with this algorithm elements can only be added to the subset, not removed. A new algorithm, Large Sampling Cubature, introduced a slight modification to this aspect and was attempted for advection in fluid simulations[28]. In this case, a randomized set of elements are added to the subset at each iteration. Since some of the weights computed from Equation 2.43 might end up being equal to 0, this algorithm removes any element with such weights at the end of every iteration.

One algorithm that attempts to assign the weights by solving the problem in Equation 2.43 is Non-Negativity-Constrained Hard Thresholding Pursuit (NN-HTP)[64]. This algorithm takes inspiration from Normalized Iterative Hard Thresholding (NIHT)[6], which is a subset selection algorithm whose development was triggered by the field of Compressed Sensing. NIHT was later improved by Hard Thresholding Pursuit (HTP)[16], which is an acceleration method that adds a second step to NIHT. This algorithm requires a specified number of elements to select for the subset. The algorithm uses a filter function that only keeps the largest weights and sets the rest to 0. By keeping the same number of weights as the specified number of elements, this filter process always makes sure that the specified number of elements will be selected. First, the algorithm creates an initial guess, consisting of randomly generated weights for all elements. Second, the filter function is applied to the initial guess setting most of the weights to 0. This create a set of elements before the algorithm starts iterating to find a more optimal solution. In each iteration, the gradient of the residual is computed and used together with the filter function to find a new set of weights. If at any point the same set of selected elements is repeated, or if the the norm of the gradient is smaller than some specified value, the iteration is stopped and the subset of elements and their respective weights will not be changed any more.

One algorithm[22] that has been developed in the field of computational engineering formulates the

problem a bit differently. This algorithm starts by forming the $n' \times rT$-matrix, $\mathbf{X}_{\mathcal{F}}$, shown below

$$\mathbf{X}_{\mathcal{F}} = \begin{pmatrix} \sqrt{W_1}(\bar{\mathbf{g}}_1^{(1)} - \bar{\mathbf{f}}_1/\Omega)^T & \cdots & \sqrt{W_1}(\bar{\mathbf{g}}_1^{(t)} - \bar{\mathbf{f}}_t/\Omega)^T & \cdots & \sqrt{W_1}(\bar{\mathbf{g}}_1^{(T)} - \bar{\mathbf{f}}_T/\Omega)^T \\ \vdots & & \vdots & & \vdots \\ \sqrt{W_i}(\bar{\mathbf{g}}_i^{(1)} - \bar{\mathbf{f}}_1/\Omega)^T & \cdots & \sqrt{W_i}(\bar{\mathbf{g}}_i^{(t)} - \bar{\mathbf{f}}_t/\Omega)^T & \cdots & \sqrt{W_i}(\bar{\mathbf{g}}_i^{(T)} - \bar{\mathbf{f}}_T/\Omega)^T \\ \vdots & & \vdots & & \vdots \\ \sqrt{W_n}(\bar{\mathbf{g}}_n^{(1)} - \bar{\mathbf{f}}_1/\Omega)^T & \cdots & \sqrt{W_n}(\bar{\mathbf{g}}_n^{(t)} - \bar{\mathbf{f}}_t/\Omega)^T & \cdots & \sqrt{W_n}(\bar{\mathbf{g}}_n^{(T)} - \bar{\mathbf{f}}_T/\Omega)^T \end{pmatrix} \tag{2.44}$$

where W_i is the volume of element i. The next step of the algorithm is to decompose $\mathbf{X}_{\mathcal{F}}$ using SVD, resulting in $\mathbf{X}_{\mathcal{F}} = \mathbf{\Lambda}_{n' \times n'} \mathbf{\Sigma}_{\Lambda} \mathbf{V}_{\Lambda}^T$, where $\mathbf{\Lambda}_{n' \times n'}$ is a $n' \times n'$-matrix that spans the column-space of $\mathbf{X}_{\mathcal{F}}$. From this point, the algorithm follows the procedure of the Greedy Cubature-algorithm with one key difference. Instead of solving the problem shown in Equation 2.43, the algorithm attempts to solve the problem shown below

$$\mathbf{J}\boldsymbol{\alpha} = \boldsymbol{\beta}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{\Lambda}_{m \times n'} & \sqrt{\mathbf{W}}_m \end{bmatrix}^T, \quad \boldsymbol{\beta} = \begin{bmatrix} \mathbf{0}^T & \Omega \end{bmatrix}^T \tag{2.45}$$

which is a unconstrained least-squares problem where $\mathbf{\Lambda}_{m \times n'}$ is a sub-matrix of $\mathbf{\Lambda}_{n' \times n'}$, which only contains data from m elements. Instead of using a non-negativity constraint for the weights, this problem puts a constraint on a weighted sum of the weights through the last row of the problem, where $\sqrt{\mathbf{W}}_m = [\sqrt{W_1} \cdots \sqrt{W_m}]^T$. After solving for $\boldsymbol{\alpha}$, the final weights are computed through $w_i = \sqrt{W_i}\alpha_i$. It was empirically observed that this formulation produced non-negative weights, and hence that the produced weights would meet the requirements that the formulation of hyperreduction puts on them. In addition, this formulation relates the weights of the elements to their volumes, which would make sense for many models.

Chapter 3

Model Reduction

3.1 Introduction

One of the objectives of this thesis is to provide an open source implementation of model reduction that allows it to be used with muscle driven biomechanical models. The model reduction has been applied to cases that are relevant to articulatory speech synthesis and so the implementation has been made to handle those cases. However, the objective has also been to make the implementation useful for other types of biomechanical simulations, and so the implementation has been made as general as possible with this objective in mind. This chapter reports the details of the implementation of model reduction in ArtiSynth.

3.1.1 ArtiSynth

The implementation has been made within the Java-based biomechanical toolkit ArtiSynth (`www.artisynth.org`, `https://github.com/artisynth/artisynth_core.git`)[34]. This software package was originally created for the purpose of articulatory speech synthesis and therefore includes a large variety of interactive features. ArtiSynth has also proven useful for many other applications within the field of biomechanical modeling. ArtiSynth includes a variety of different models, such as rigid bodies that are used to model bones as well as FEMs that are used to model tissues. The inclusion of both these entities allows for the creation of hybrid models containing both rigid bodies and FEMs. Different types of muscle models are also included, both in the form of point-to point muscles and muscles embedded within the elements of FEMs. Since the biomechanics of voice production includes many situations of contact, such as tongue to palate or colliding vocal folds, ArtiSynth also includes techniques for detection of contact as well as computation of contact forces.

One of the key-features of ArtiSynth is its interactive nature that allows the user to try out input parameters with real-time visual feedback. For many models, this has partially been achieved by keeping the resolution of FEMs low in order to have a low amount of elements to compute forces in and to have fewer equations of motion. Some models, however, require high resolution for the sake of accuracy. In other cases, different models are combined, which also brings up the complexity of the model as a whole. For both of these cases, the general effect is that the simulations of the models become slower compared to simpler models. For the models that experience this loss of fast simulation, the interactivity of ArtiSynth is partly lost. Model

reduction, which has the potential to significantly speed-up simulations of complex models, offers the solution to this problem and can therefore make even complex models in ArtiSynth interactive.

3.2 Implementation

3.2.1 Overview

In order to have a connection between model reduction and the various FEMs in ArtiSynth, a class, `ReducedFemModel`, has been created. This class is a subclass of `FemMuscleModel`, which is the lowest class in the FEM-hierarchy in ArtiSynth. Through its location in the hierarchy, `ReducedFemModel` inherits all relevant features that FEMs in ArtiSynth have, while adding the functionality of model reduction. If the model reduction is not invoked, the `ReducedFemModel` will simply keep acting like an unreduced FEM instead of being reduced. The central class of the implementation of model reduction in ArtiSynth is `ReducedFemBody`. A FEM is reduced by making it a `ReducedFemModel` and assigning a `ReducedFemBody` to it. At this point, the nodes of the `ReducedFemModel` are made non-dynamic and hence do not react directly to the forces applied to them. Instead, their positions and velocities are controlled through the reduced coordinates of the `ReducedFemBody`.

The dynamics of the nodes of a reduced FEM are controlled through ReducedFemAttachment, a subclass of `DynamicAttachment`, which is an abstract class that handles attachment between dynamic components. When a `ReducedFemBody` is assigned to a `ReducedFemModel`, the `ReducedFemBody` generates a set of `ReducedFemAttachments` and assigns an attachment for each node of the `ReducedFemModel`. Optionally, the non-dynamic nodes of the `ReducedFemModel` can be ignored in this step, in which case only the dynamic nodes get a `ReducedFemAttachment` assigned to them. From this point, all computations regarding the nodes are performed through their respective attachments. The reduced mass-matrix is assembled by identifying the relevant rows of the reduced basis and applying them to the node-mass for each node. In the same way, external forces are projected through the attachments onto the reduced external force of the `ReducedFemBody`.

In order for the model reduction to be able to handle multi-body problems as well as rigid motion, `ReducedFemBody` has been made a subclass of `Frame`, which is a class that is used to handle floating coordinate frames. By constructing `ReducedFemBody` this way, the reduced basis does not have to include rigid motion and should therefore optimally only include deformations. If no reduced basis is assigned to the `ReducedFemBody`, the `ReducedFemModel` is simply reduced to rigid motions. For the case of FEMs with constraints that prohibit them from rigid motion, it is possible to disable the rigid motion for the `ReducedFemBody`. Since the rigid motion is handled through the `Frame` dynamics, the reduced form of the generalized Newton-Euler equation2.34 is used to simulate the dynamics. In the current implementation, the blocks of the mass-matrix that handles the coupling between rigid motion and deformation, $\bar{M}_{tq}$ and $\bar{M}_{\theta q}$ in equation 2.34, are set to 0 so the mass-matrix is given by:

$$\begin{bmatrix} \mathbf{M}_{tt} & & \mathbf{sym} \\ \mathbf{M}_{t\theta} & \mathbf{M}_{\theta\theta} & \\ \mathbf{0} & \mathbf{0} & \bar{\mathbf{M}}. \end{bmatrix} \tag{3.1}$$

In addition, the current implementation does not support constraints between reduced FEMs and other structures. The main reason for this is that the implementation is still under development, where the current focus is on having the model reduction working properly for simpler cases, such as static `Frame`s, while being formulated in a way that makes it applicable for future, more advanced cases, such as multi-body dynamics.

The computation of the reduced internal forces are performed in `ReducedFemBodyAll`, a subclass of `ReducedFemBody`. In this class, the reduced internal forces are computed by first computing the internal force for each node. The internal forces are then assembled into a vector that is transformed into the reduced internal force through multiplication by the transpose of the reduced basis. The computation of internal forces of the nodes is done very much in the same way as it is done for non-reduced FEMs in ArtiSynth. One of the main differences is that incompressibility is handled differently. Instead of applying forces or constraints to enforce incompressibility, it is assumed that the reduced basis will naturally keep incompressible models from changing volumes. For a reduced basis based on sampled data, this is a reasonable assumption provided that the reduced coordinates do not exceed the range they experienced during the sampling. In the case of a reduced basis from LMA with a possible extension, this assumption is not necessarily reasonable since the reduced basis has not been generated from actual deformations. Another important difference is that the internal forces can optionally be computed using just one integration point for each element instead of the standard number of integration points for each element. In this case, the single integration point that is used is located in the middle of the element. Apart from the computation of reduced internal forces, `ReducedFemBodyAll` also contains two methods for computing reduced bases: LMA and the extended modal basis technique using affine transformations [64]. The latter technique was chosen in preference to modal derivatives because of its relative simplicity and relatively high accuracy shown in previous work. One issue for the implementation regarding LMA is that the tools in ArtiSynth for eigendecomposition do not support the generalized eigenvalue problem. Since $\mathbf{M}$ is invertible, it is possible to rewrite Equation 2.35 as a standard eigenvalue problem as follows:

$$\mathbf{M}^{-1}\mathbf{K}\mathbf{U} = \mathbf{U}\mathbf{\Lambda} \tag{3.2}$$

Unfortunately, the symmetry of $\mathbf{M}$ and $\mathbf{K}$ is not preserved by this operation meaning that $\mathbf{M}^{-1}\mathbf{K}$ will not be symmetric, and hence that the mass-orthogonality of the solution will be lost. This property is not required but it is the main reason for even including the mass-matrix in the process of finding the reduced basis. In addition, the eigenvalues of Equation 3.2 are not necessarily non-negative, leading to difficulty when selecting which eigenvectors to include in the reduced basis based on their respective eigenvalues. For this reason, the implementation of LMA in ArtiSynth neglects the mass-matrix and instead solves the standard

eigenvalue problem:

$$\mathbf{KU} = \mathbf{U\Lambda} \tag{3.3}$$

As a consequence, the LMA implementation does not take into account the masses of the nodes when constructing the modal basis, but for FEMs with relatively uniform resolution and uniform density, the node-masses do not differ much. Most importantly, the standard eigenvalue problem is sufficient to produce a reduced basis with normalized, linearly independent basis vectors.

For those cases where sampling is the best option, there are different approaches for how to perform sampling. The first option is to perturb the model with external forces or muscle activations and record the displacements after a certain time has elapsed or once the model has reached equilibrium. A class, `ModalProbesBatchWorker`, has been implemented for this approach. This class can be called while running ArtiSynth in batch-mode. When this is done, a specified number of simulations are run with probabilistic or combinatorial input, depending on the user input. The second approach is to take snapshot samples of the model while running the same type of simulations that one intends to run for the reduced model. For this approach a class, `StateSampler`, a subclass of `MonitorBase` has been implemented. By being a subclass of `MonitorBase`, `StateSampler` will be called at each timestep immediately after the model has been advanced. `StateSampler` takes snapshots of its associated FEM at every timestep and saves all the snapshots to a file. As mentioned in the previous chapter, there are two different options for the data that can be saved during the sampling process. The first option is to save the displacements, either for all nodes or just for the dynamic nodes. The other option is to also save the internal forces and masses for all nodes, as well as the stiffness-matrix of the entire FEM for every snapshot. `ModalProbesBatchWorker` supports both of these options. The main difference is that more samples are needed when the samples just consist of displacements. Generally, the option of saving displacements, masses and internal forces of all the nodes as well as the stiffness-matrix of the entire model does not require many samples, and for that reason sampling at every timestep with `StateSampler` would create far more samples than required. For this reason, `StateSampler` only supports the option of saving displacements of the nodes for every sample.

Once the data has been sampled, it requires analysis in order to create a POD that can be used as a reduced basis for the reduced model. A class, `computeProperOrthogonalDecomposition`, has been created for the purpose of performing this analysis. When running the script, the user specifies what algorithm to use, the path to the data and the required parameters for the algorithm. In the case when the data just contains the displacements of the nodes, the algorithm should be specified to be displacement based, in which case the POD is simply computed through a SVD. In the case when the data also contains the internal forces and masses of the nodes, as well as the stiffness-matrices of the FEM, the algorithm should be specified to be based on the tangent space[65]. In this case, the internal forces, masses and stiffness-matrices of each sample will be used to compute the tangent spaces between the different samples. These tangent spaces are then added to the displacement-data, and the POD is finally computed through a SVD of the combined data.

3.2.2 Hyperreduction

For the case of Hyperreduction, a subclass of `ReducedFemBodyAll`, `ReducedFemBodyCubature`, has been created. In order to compute the reduced internal force using hyperreduction, this class requires a set of elements and a set of associated weights. If the sets are empty, `ReducedFemBodyCubature` will simply compute the reduced internal force through its superclass. If, on the other hand, the sets do contain elements and weights, the internal force is computed for only the elements with their respective weights within the sets. The reduced internal force is then computed by transforming the internal force of each node that is connected to an element in the set. By only accessing the nodes connected to elements in the set, the $\mathcal{O}(n)$-cost is avoided for this process entirely. The reduced stiffness-matrix is computed in the same way through the use of the same sets of element and weights.

In order to perform hyperreduction, it is essential to be able to select a proper subset of the elements and assign weights to them. This process requires sampling as well as training of the model through analysis of the data. Two different classes have been implemented for the sampling process. The first option is `ModalForcesProbesBatchWorker`, which is a subclass of `ModalProbesBatchWorker`, and can be called when running ArtiSynth in batch-mode. The input for the simulations in this case works in the same way as it does for `ModalProbesBatchWorker`. At the end of each simulation, the reduced internal force is being computed and saved to a file. In addition, the reduced internal force generated by each individual element is being computed and saved as well. The other class is `ForceSampler`, which is a subclass of `MonitorBase`, and therefore, just like `StateSampler`, is called immediately after advancing the simulation at each timestep. `ForceSampler` takes snapshots of the model at a specified rate and saves all the snapshots. The data from the snapshots have the same format as the data recorded by `ModalForcesProbesBatchWorker`.

The algorithms that are used to analyse the sampled data, in order to determine the optimal subset of elements and their respective weights, have been implemented in the class `optimizedCubature`. Four different algorithms have been implemented in the class, all of which find the elements and their respective weights by formulating and solving a least-squares problem. The implemented algorithms are Greedy Cubature[1] as well as its volume-constrained version[22], Large Sampling Cubature[28] and Non-Negativity-Constrained Hard Thresholding Pursuit (NN-HTP)[64]. Besides the data, all four algorithms mainly use two number arguments. One of the numbers is used as an error tolerance for all four algorithms. For the Greedy Cubature and Large Sampling Cubature this number sets a limit for the relative error, while for NN-HTP it limits the gradient of the absolute error. At the end of each iteration, the number is used for its respective measure, and if the measure is smaller than the specified number the algorithms finish and will not run more iteration. The other number argument fills different functions for each of the algorithms. For Greedy Cubature, it is used for the number of candidate points out of which the next selected element of the subset is selected at each iteration. For Large sampling Cubature, it is used to determine the number of elements to be added to the subset of elements at each iteration. For NN-HTP, where the size of the subset of elements is fixed throughout the entire training, the integer is used to set the size of the subset of elements.

3.2.3 Reduction Procedure

The purpose of this section is to clarify the procedure of reducing FEMs in ArtiSynth. This is needed both for an improved understanding of how the different parts of the implementation relate to each other and for potential users attempting to apply model reduction to their models in ArtiSynth. The procedure is presented as a list of steps, each consisting of a list of sub-steps that explain how different scenarios should be handled.

1. Generate Reduced Basis

   In order to reduce the FEM, a reduced basis is required and so the first step is to compute such a basis. When computing the reduced basis there are two different options listed below.

   (a) Linear Modal Analysis

   If one wishes to use a reduced basis based on linear computations, one can compute using linear modal analysis. This option requires that a `ReducedFemBody` has been assigned to the FEM as described in step 2. Once the `ReducedFemBody` has been assigned to the FEM, a reduced basis can be computed linearly and assigned to the `ReducedFemBody` by simply calling a method created for this purpose.

   i. Extend Modal Basis

   If one wishes to use a reduced basis that has been computed linearly, but that still can handle non-linear deformations, one can instead call a method that computes a basis linearly and then splits up the different modes in order for the basis to handle non-linear deformations better.

   (b) Compute POD from sampled data

   If one wishes to compute a reduced basis from sampled data, the two steps below are required.

   i. Sampling

   For this procedure, there are two different options. The first option is to simply run simulations of the model and take snapshots at certain times, meaning that data is recorded before allowing the simulation to continue. The second option is to perturb the model and record data once the model has reached equilibrium.

   ii. Analysis

   There are two different options regarding what data to sample and each option requires its own specific analysis. The first option is to only sample node-displacements, in which case the POD is simply computed through a SVD. The second option is to sample displacements, forces, stiffnesses and masses. In this case the tangent space of each sample is computed and added to the displacement-data before computing the POD through a SVD.

2. Assign `ReducedFemBody` to FEM.

   This step requires changes in the actual model in order to reduce the FEM. First, the FEM, being a `FemModel3d` or a `FemMuscleModel`, needs to be changed into a `ReducedFemModel`. Second, a `ReducedFemBody` needs to be created and assigned to the `ReducedaFemModel`.

   (a) Assign Reduced Basis

   In order for the `ReducedFemModel` to be able to deform after the `ReducedFemBody` has been assigned to it, the `ReducedFemBody` needs to have a reduced basis associated with it. If the `ReducedFemBody` does not have a reduced basis associated with it, the `ReducedFemModel` is simply reduced to rigid motion. The reduced basis can either be read from a file or computed after assigning the `ReducedFemBody`, as described in step 1a.

3. Hyperreduction

   Once the `ReducedFemModel` has been reduced, it is possible to speed up the computations of the internal forces through hyperreduction (only sampling a small number of elements to estimate internal forces and stiffness). In order to do this, one has to change the `ReducedFemBodyAll` into a `ReducedFemBodyCubature` before following the steps listed below.

   (a) Sample Reduced Internal Forces

   The first step is to sample reduced internal forces. This can be done with the same procedures as for step 1(b)i. Unlike that step, the data recorded for each sample is the reduced internal force for the entire FEM, as well as the reduced internal force generated by each individual element. This data can be sampled through methods in `ReducedFemBodyCubature` created for this purpose.

   (b) Analyse sampled forces

   Second, the sampled data has to be analysed using a selected training-algorithm that is based on a least-squares scheme. The algorithm generates a small subset of the elements of the `ReducedFemModel` and assigns weights to each element in the subset.

   (c) Assign weights to elements

   Finally, the elements and their respective weights are assigned to the `ReducedFemBodyCubature` by reading them from a file. If no elements are assigned to the `ReducedFemBodyCubature`, the internal forces are computed, as they would for a `ReducedFemBodyAll`.

3.3 Conclusions

This chapter has presented the implementation of model reduction in the open source biomechanical toolkit ArtiSynth. The implementation has been made in such a way that FEMs are unreduced unless otherwise specified. Once a model is specified, it is given reduced DOF depending on its associated reduced basis as

well as its constraints. The implementation also includes hyperreduction which allows for larger speedup of simulations by just using a subset of the elements when computing the reduced internal force. Unless otherwise specified, a reduced model will not use hyperreduction, but instead use all elements of the FEM. The implementation also includes training algorithms for computing reduced bases as well as determining subsets of what elements and weights to use for hyperreduction. In addition, tools have been created for sampling of force- and displacement-data that serves as input for the training-algorithms.

Chapter 4

Results

In this chapter, an empirical evaluation of the modal reduction approaches that have been implemented in ArtiSynth, is provided. FEMs with different complexity were tested, along with different aspects of the model reduction.

4.1 Evaluation Methodology

model reduction was evaluated for three test models. First, model reduction has been applied to a muscle driven FEM-beam consisting of 81 elements that mainly serves as a proof of concept since the complexity is too small for model reduction to result in any significant speedup. Second, model reduction has been applied and evaluated for a FEM tongue model that previously has been used for speech production simulations [60]. This model includes 11 muscles that are modelled as fibres embedded within elements whose material properties therefore change as the muscles are excited. This tongue model was intentionally created with a small number of elements (740) in order to make it simulate faster. Third, model reduction was applied to a high-resolution version of the tongue model with 4255 elements. In previous work this model has been simulated using node-to-node muscles but for this evaluation the model was tested with muscle fibres embedded within the elements for the first time. With this testing being in an early stage, the behaviour of the model is not entirely realistic but still useful for evaluating model reduction. This model includes similar muscle definitions as the original tongue model, a slightly altered shape, and a higher resolution mesh so that the model can capture more flexible tongue shapes. Given the increase in number of elements, the high-resolution tongue model is slower to simulate as a full FEM and therefore provides a more compelling case for model reduction.

For all three models the model reduction was applied using roughly the same procedure. For each of the models, two reduced bases were computed. The first basis was computed through extension of a modal basis consisting of 6 modes making the final reduced basis consist of 54 modes. The second basis was computed by first simulating the models using different types of perturbations and sampling displacements of all dynamic nodes at every timestep. The sampled displacements were then used to compute a POD through SVD. After measuring the accuracy of the reduced simulations using these two bases, a new sampling procedure was used to sample force data in order for training of hyperreduction. This sampling procedure used roughly the same

set of perturbations as the training for the PODs. As will be presented in the following sections, the POD produced far better accuracy than the extended modal basis for each of the models. For this reason, the force sampling was only applied to the reduced model using the POD. Finally the sampled force data was used as input to the NN-HTP algorithm to determine proper subsets of elements, with respective weights.

The FEMs were then simulated, both with and without model reduction, in order to compare the computational speed of each simulation and the accuracy of the reduced model as compared to the non-reduced model (which was considered as ground-truth). Computational speed was measured by measuring the computational time for each timestep throughout different simulations of the models, using a system clock, and then using the average computational time to compare the reduced simulations to their respective non-reduced simulations. Accuracy was measured by computing the Absolute Error as the Euclidean distance between the position of all nodes in the reduced simulations and the non-reduced simulations. The average with respect to the nodes of these Absolute Errors was then plotted as functions for times when testing dynamic accuracy as well as computed for static configurations when testing static accuracy. In addition, the Absolute Errors were used in heatmaps of static configurations in order to investigate where in the models the Absolute Errors were the largest. All simulations that were part of this evaluation study were run on computer using a 2.8 GHz Intel Core i7 processor running Linux.

4.2 Muscle Driven FEM Beam

Our first test case uses a simple rectangular beam shape with muscles along the length of the beam. This `FemMuscleBeam` model already exists in ArtiSynth and is used mainly for the validation of the muscle FEM functionality in ArtiSynth as well as visualization of the functionality for new users. Passive FEM beams are a common test case and have been used in previous model reduction papers, therefore it was appropriate to use a muscle-driven version of the beam as an initial and simple test case here.

4.2.1 Model Description

The beam, shown in Figure 4.1, has dimensions $0.9m \times 0.3m \times 0.3m$ and consists of 160 nodes ($10 \times 4 \times 4$) that are connected by 81 hexahedral elements ($9 \times 3 \times 3$). The 16 nodes on the left end of the beam are static (fixed boundary conditions), making 144 nodes dynamic, which in total gives the model 432 DOF. The material of the model is five parameter Mooney-Rivlin material with $C_{10} = 1037$, $C_{20} = 486$ and $C_{01} = C_{11} = C_{02} = D_1 = 0$. The density is $1000kg/m^2$ and the stiffness- and particle-damping is 0.01 and 6.22, respectively. The model includes 2 muscles that are located at the model's top (blue fibres) and bottom (green fibres) layer of elements, respectively. These have been modelled with two different options, as point-to point muscles between nodes, or as muscle fibres embedded in elements. The validation has been applied to the later option, in which the muscles are modelled with the Blemker Muscle model with $\lambda^* = 1.4$, $\lambda_{ofl} = 1$, $\sigma_{max} = 300000$ Pa, $P_1 = 0.05$ and $P_2 = 6.6$. All simulations of this model were run using implicit

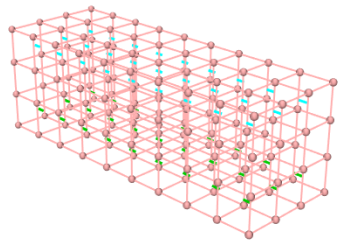
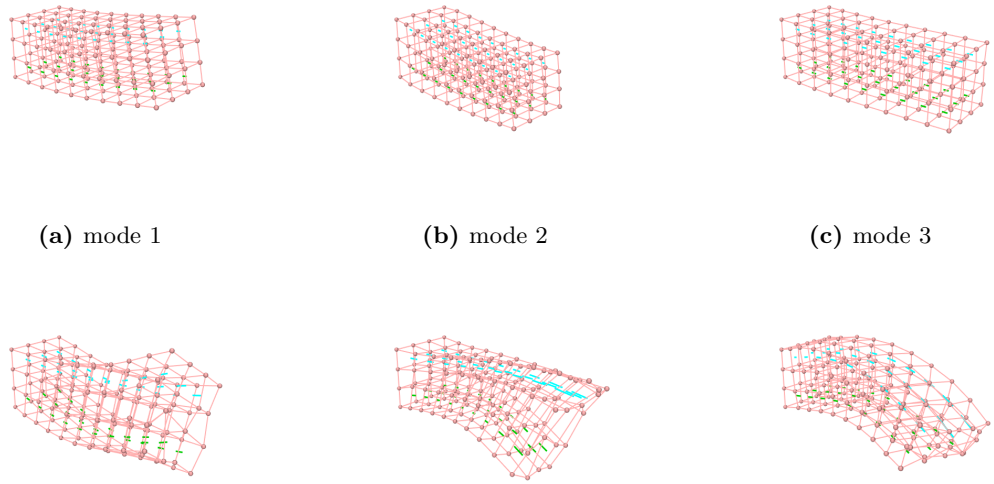


Figure 4.1: The Muscle FEM Beam in its resting state.

integration and a timestep of 10 ms. From the number of elements and nodes in this model, it is obvious that it is not a model for which one can expect a large computational speed-up from model reduction. Therefore this model is included mainly as an example of the reduction procedure and for the purpose of validation.

The model has been reduced according to the procedure described in Section 3.2.3. Two reduced bases were created, one through sampling of displacement-data that was recorded through snapshots throughout simulations, and one through the creation of an extended modal basis. The snapshots were taken at every timestep for the first 5 seconds throughout 2 different simulations. In the first simulation, the top layer of muscle-fibres was excited linearly from 0 to 0.1 in 2.5 seconds and then back to 0 in 1 second. This allowed the snapshots to capture both the deformation caused by the muscle-activation and the free vibration that occurred after the muscle-activation had been reduced to 0. In the second simulation, the model was allowed to deform under gravity until an equilibrium was reached. After that, the interactive pull control in ArtiSynth was used to achieve other deformation than those achieved by gravity or muscle-activations. Through this procedure, a total of 1000 training-samples were generated, each consisting of the 3D displacement of the 144 dynamic nodes. `computeProperOrthogonalDecomposition` was used to compute a POD using SVD, that covered 99.99% of the space spanning all training-configurations. The resulting POD consists of 14 modes, 6 of whom are shown in Figure 4.2. The extended modal basis was created by computing the 6 most significant linear modes of the model, shown in Figure 4.3, and then expanding those six modes into a basis of 54 modes. The nine modes of the extended modal basis that were generated from the first linear mode are shown in Figure 4.4. The modes of the modal basis show the type of behaviour one would expect from linear modal analysis of a model like this. The modes essentially include first order bending (mode 1 and 2), twisting (mode 3), second order bending (mode 4 and 5) and elongation (mode 6). Since the analysis is linear, some distortion can be observed for the twisting mode and the two modes of second order bending because some elements experience quite large rotation for these modes.

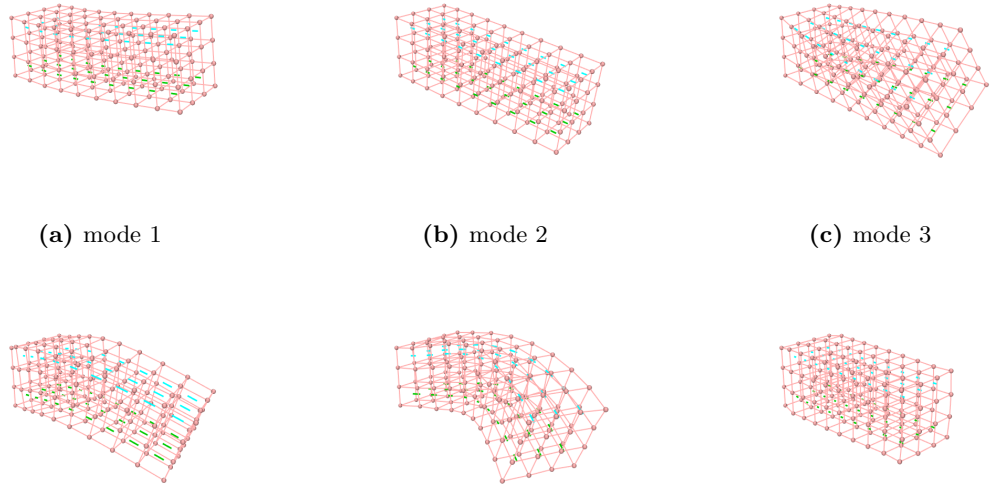
After validating the dynamic behaviour of the reduced model, a second sampling procedure was started



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

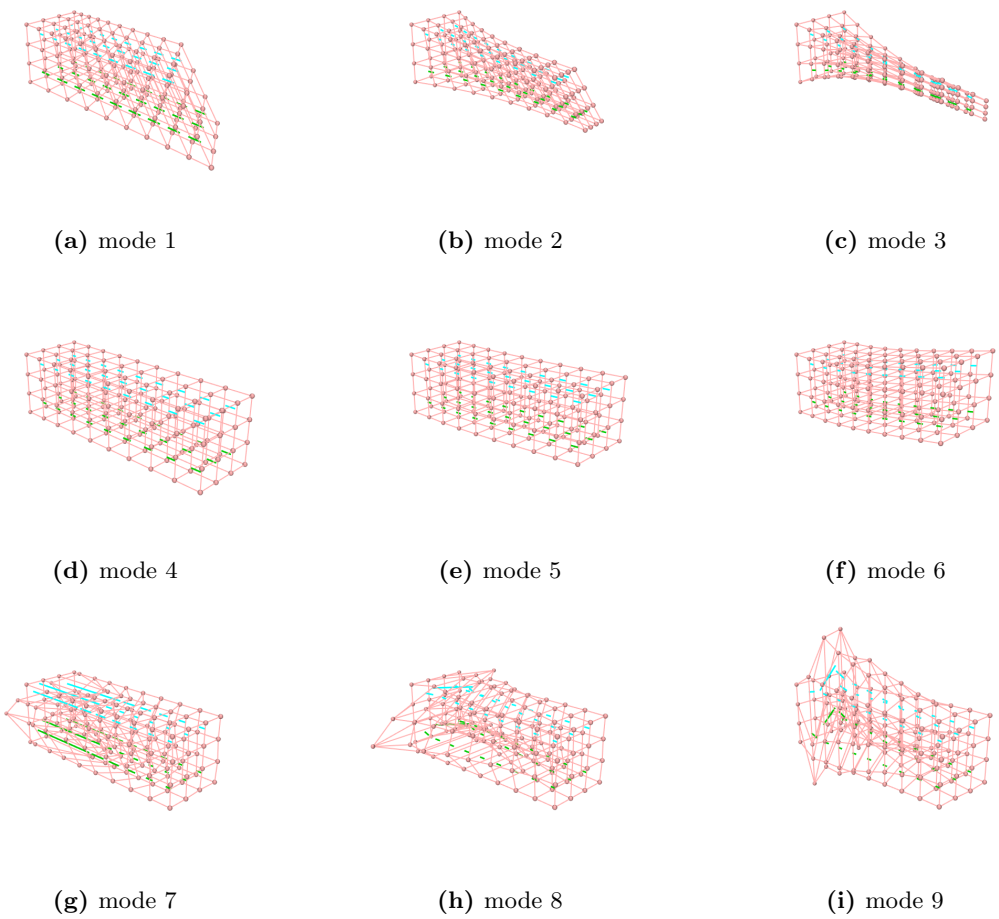
Figure 4.2: The six most significant modes from sampling of Muscle FEM beam deformations.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

Figure 4.3: The six most significant modes from linear modal analysis of the Muscle FEM Beam.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

(g) mode 7 **(h)** mode 8 **(i)** mode 9

Figure 4.4: The nine modes generated from the first linear mode through the extension algorithm.

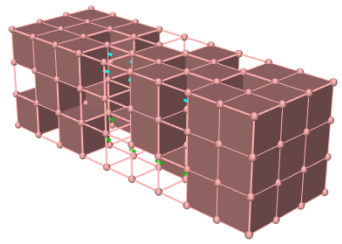
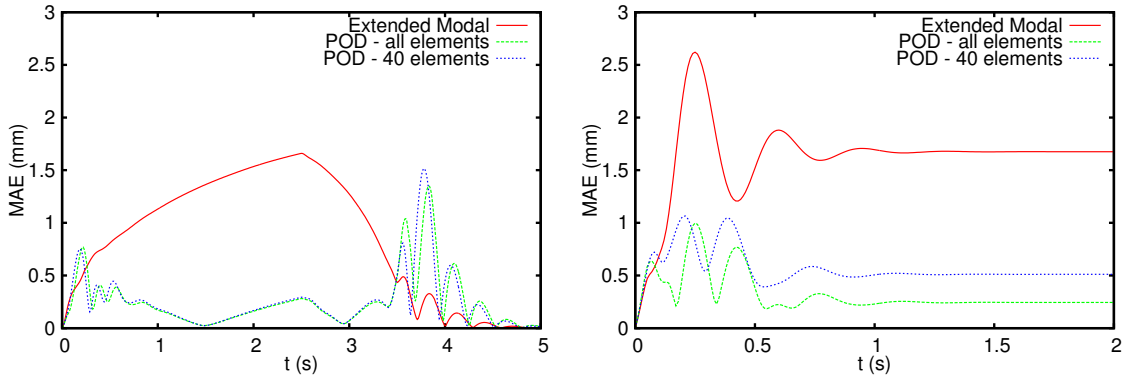


Figure 4.5: The Muscle FEM Beam with all the elements selected for hyperreduction filled in.

in order to train the reduced model for hyperreduction. This sampling was done in a way similar to that of the sampling for the reduced basis; the same set of simulations were run and the reduced internal force was recorded at every second timestep. This procedure generated a total of 500 training-samples, all consisting of the 14-dimensional internal force generated by the entire model as well as the 14-dimensional force density that was generated by each of the 81 individual elements. The NN-HTP algorithm was then used to analyse the data in order to select an optimal subset of the elements and compute their respective weights. In order to ensure that the relative error was small enough, the algorithm was set to select 40 elements which reduced the relative error to 0.11%. The large number of elements that was required in relation to the total amount of elements in the model relates to the point made earlier that this model has too few nodes and elements to have any substantial computational speed-up from model reduction. Figure 4.5 shows the beam where all the elements that have been selected for hyperreduction are filled in. Interestingly, many elements in the two ends of the beam seem to have been selected while the elements selected in the middle of the beam seem to be sparse.

4.2.2 Dynamic Accuracy

In order to validate the reduction of this model, two different simulations were run for each reduced case as well as for the non reduced model. In the first simulation, the top layer muscle-fibres were activated just as they had been during the training (ramped-up linearly from 0 to 0.1 in 2.5 seconds and then ramped-down back to 0 in 1 second). In the second simulation the model was allowed to deform under gravity. In each simulation, the displacements of all dynamic nodes were recorded at every timestep. For each timestep, the displacements were used to compare the reduced simulations to the non-reduced simulations. Figure 4.6 shows the Mean Absolute Error (MAE) as a function of time for each of the reduced simulations for both validation cases. In this case, MAE has been defined as the mean Euclidean distance between the nodes' displacements in the reduced and the non-reduced simulation.



(a) top muscle activation

(b) gravity

Figure 4.6: Plots of the Mean Absolute Error as a function of time for each of the reduced simulations for both validation cases.

Despite including most DOF, the extended modal basis has the least accuracy at almost every timestep in both cases. It performs particularly badly at those times when the deformations are the largest, which can at least partly be explained by the non-linearity in the deformation at those timesteps. The only time where the extended modal basis performs the best consistently is in the muscle-driven simulation just after the muscle activation has been reduced to 0. At this point, the model experiences some free vibrations with small amplitude, which suits a modal basis particularly well. In the muscle-driven case, the hyperreduced simulation follows the simulation using the sampled POD almost perfectly while in the gravity-driven case it has a slightly larger amplitude on its vibrations. The MAE stays below 3 mm for the extended modal basis, and below 2 mm for the sampled basis, both with and without hyperreduction.

4.2.3 Static Accuracy

The oscillatory behaviour of the MAE for the two validation cases show a difference in amplitude, but also a slight difference in frequency. This indicates a difference in stiffness and possibly damping. Since the damping of the model is stiffness-dependent, a difference in stiffness should be transferred into a difference in damping. In an effort to investigate this issue further, the beam was perturbed to three different equilibrium configurations, for which the MAE was measured. The three different configurations, whose MAE are shown in Figure 4.7, are gravity, 5% and 10% activation of the top layer muscle-fibres. In the case of gravity, the figure simply shows the value to which the MAE converged for the different reduction cases. In this case, the extended modal basis has the by far largest MAE, almost reaching 2 mm, while the MAE of the two reduced simulations using the POD are well below 1 mm, with the hyperreduced simulation producing the larger MAE. In the cases of muscle induced equilibriums, the extended modal basis again produces the largest MAE, reaching above 1 mm in both cases, while the two reduced simulations using the POD produce almost identical MAE in both cases.

As indicated in Figure 4.6 and Figure 4.7 there are differences in equilibrium configurations between

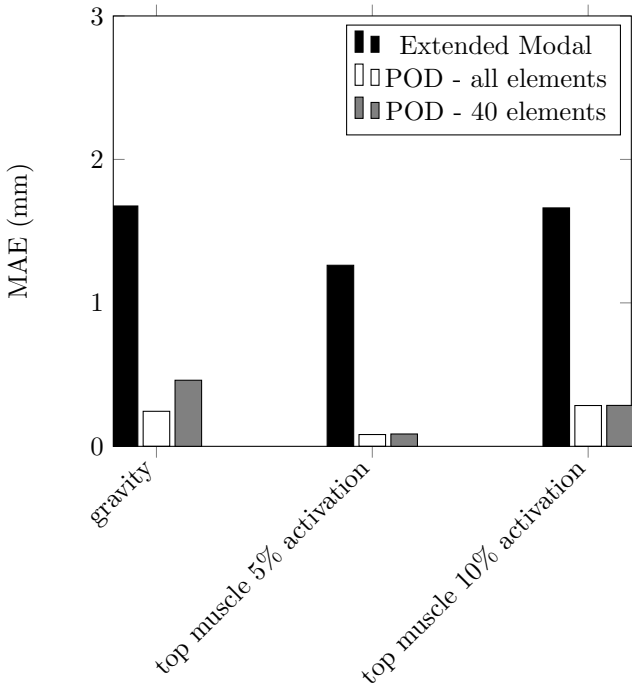
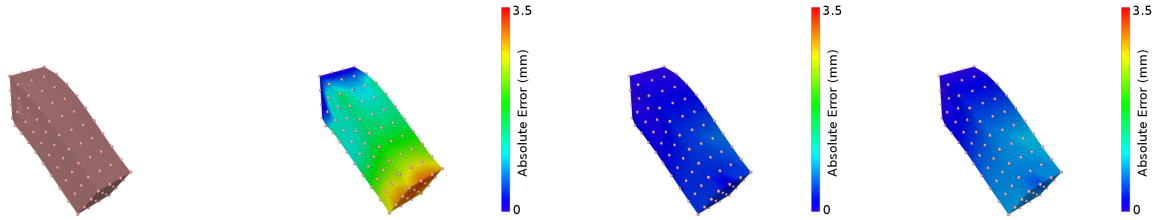


Figure 4.7: Bar chart showing the MAE at equilibrium for gravity and different levels of muscle activation for the different reduction cases.

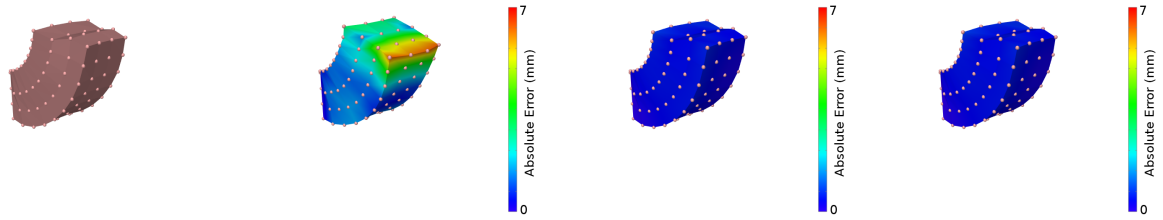
the full and the reduced simulation when the FEM beam is deformed by gravity. Figure 4.8 shows the configurations produced by the different simulations at this equilibrium, where the surface colouring of the reduced simulations are based on the local deviation from the non-reduced simulation. For the simulation using the extended modal basis, there is a continuous increase of the deviation along the beam, with the maximum deviation almost reaching 3 mm. For the two reduced simulations using the POD, the local deviations do not reach that high. For the simulation using all the elements, the local deviation stays below 0.5 mm, while for the hyperreduced simulation, the local deviation almost reaches 1 mm at the deflected end of the beam.

As Figure 4.7 indicates, there is not much difference between the equilibriums of the two reduced simula-



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 40 elements

Figure 4.8: Comparison of non-reduced and the reduced simulations at equilibrium due to gravity. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 40 elements

Figure 4.9: Comparison of full and reduced simulations at equilibrium from 10% activation of top muscles. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.

tions using the POD when the beam is deformed under muscle activation. The configurations at equilibrium of the non-reduced FEM beam and the reduced beam for the 10% muscle activation are shown in Figure 4.9. The reduced simulations that use the POD, both the one using all elements and the hyperreduced one, produce configurations that are very similar to the non-reduced simulation with the local deviation staying below 1 mm. The reduced simulation using the extended modal basis on the other hand, has clear difficulty reproducing the deformation of the beam and reaches a local deviation of almost 7 mm in a small area around the nodes that are most displaced.

4.2.4 Computational Speed

The computational speed-up was measured by the same simulations that were used for training. Instead of recording the displacements at each timestep, the time to perform each timestep was measured and saved to file. The computational time for each timestep as well as the speedup for the two validation cases and for all measured timesteps are presented in Table 4.1. As expected for this model, there is not much speedup for any of the reduced simulations. In fact, the reduced simulations using the extended modal basis are slower than their respective non-reduced simulations. The reduced simulations using the sampled POD are slightly faster than their respective non-reduced simulations, while the hyperreduced simulations show a speedup slightly larger than 50%.

4.3 Muscle Driven Tongue Model

In order to continue the evaluation of the model reduction into realistic biomechanical models, the model reduction has been applied to a model of the tongue that already exists in ArtiSynth. This model was created based on a reference model [8], and has been used in simulations [60] in combination with other biomechanical structures such as the jaw.

Simulation		Full	Reduced					
			Extended Modal		POD - all elements		POD - 40 elements	
Perturbation		time (ms)	time (ms)	speedup	time (ms)	speedup	time (ms)	speedup
top muscle activation	mean	5.0	6.9	0.72	4.5	1.1	3.0	1.7
	std	2.3	2.7		1.8		1.7	
gravity	mean	3.9	6.2	0.63	3.5	1.1	2.6	1.5
	std	1.6	1.3		1.4		0.9	
average	mean	4.7	6.7	0.70	4.2	1.1	2.9	1.6
	std	2.2	2.4		1.8		1.5	

Table 4.1: Summary of the computational time per timestep and speedup for each of the reduced simulations for both perturbations.

4.3.1 Model Description

The tongue model, shown in Figure 4.10, consists of 948 nodes that are connected by 740 hexahedral elements. In the case where the tongue is simulated without interaction with other structures, 117 of the nodes are static, making 831 nodes dynamic, which in total gives the model 2493 DOF. The material of the model is five parameter Mooney-Rivlin material with $C_{10} = 1037$, $C_{20} = 486$ and $C_{01} = C_{11} = C_{02} = D_1 = 0$. These parameters where determined by [8] through comparison of results from different measurements of mechanical properties of the tongue[18, 12]. The density is $1040kg/m^2$, and the stiffness- and particle-damping is 0.03 and 40, respectively. The model includes 11 muscles, which are the posterior genioglossus (GGP), medial genioglossus (GGM), anterior genioglossus (GGA), styloglossus (STY), geniohyoid (GH), mylohyoid (MH), hyoglossus (HG), vertical muscle (VERT), transverse muscle (TRANS), superior longitudinal muscle (SL) and inferior longitudinal muscle (IL). These have been modelled with two different options, as point-to point muscles between nodes, or as muscle fibres embedded in elements. The evaluation has been applied to the latter option, in which the muscles are modelled with the Blemker Muscle model with $\lambda^* = 1.4$, $\lambda_{ofl} = 1$, $\sigma_{max} = 300000$ Pa, $P_1 = 0.05$ and $P_2 = 6.6$. All simulations of this model were run with implicit integration and a timestep of 10 ms.

The model has been reduced according to the procedure described in Section 3.2.3. Two reduced bases were created, one through sampling of displacement-data that was recorded through snapshots throughout simulations, and one through the creation of an extended modal basis. The snapshots were taken at every timestep through 11 different simulations. In each simulation, one selected muscle was excited with left-right symmetry from 0 to 0.3, and then back to 0, with cubic interpolation over a period of 1 second. The model was then allowed to reach equilibrium due to gravity over the next 0.2 seconds, making each simulation 1.2 seconds long. Through this procedure, a total of 1322 training-samples were generated, each consisting of the 3D displacement of the 831 dynamic nodes. `computeProperOrthogonalDecomposition` was used to compute

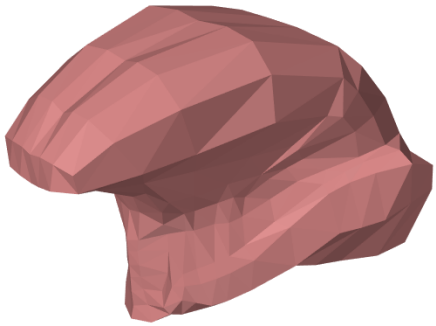
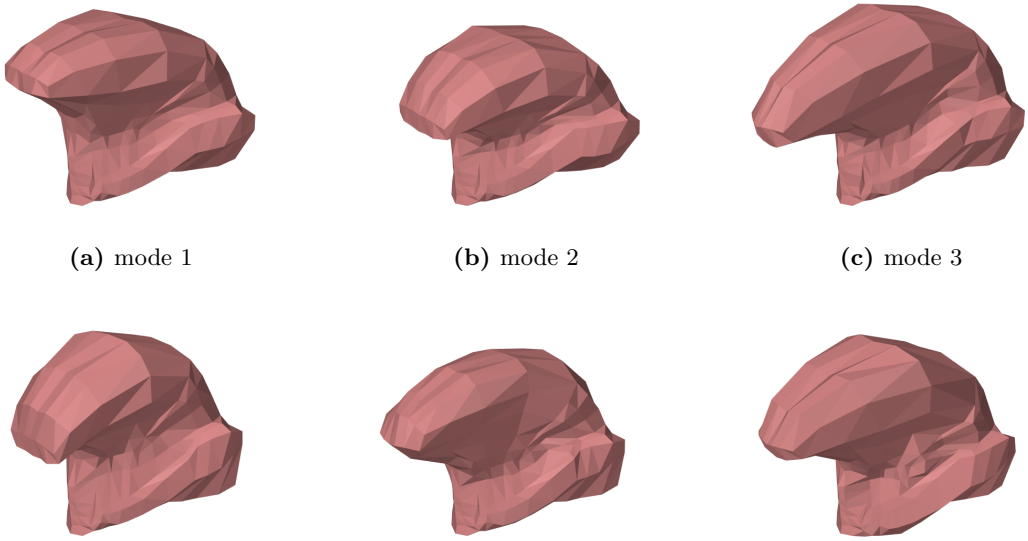


Figure 4.10: The tongue model at rest state.

a POD using SVD, that covered 99.99% of the space spanning all training-configurations. The resulting POD consists of 17 modes, 6 of whom are shown in Figure 4.11. The extended modal basis was created by computing the 6 most significant linear modes of the model, shown in Figure 4.12, and then expanding those six modes into a basis of 54 modes. It is worth noting that the modes computed from sampling has a close resemblance with actual tongue-deformations while the linear modes show some deformations that are quite unrealistic. It is also worth noting that all the modes computed from the sampled data are symmetric around the mid-sagittal plane while four of the linear modes are not symmetric. The 9 modes that corresponds to the first linear mode are shown in Figure 4.13. Most of these modes show little resemblance with actual tongue-deformations. Also, since these modes are derived from the first linear mode that is not symmetric around the mid-sagittal plane, they have inherited this property.

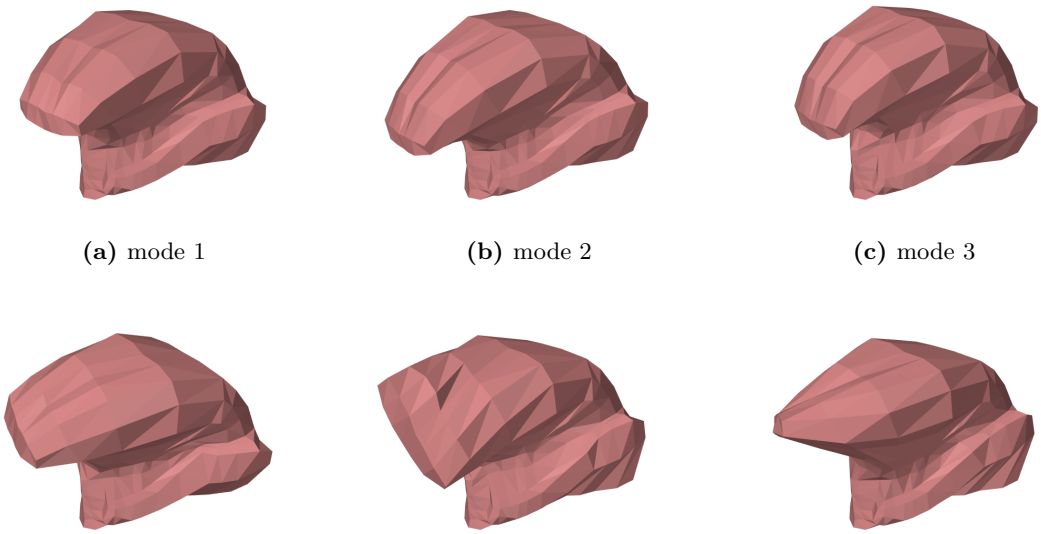
After validating the dynamic behaviour of the reduced model, a second sampling procedure was started in order to train the reduced model for hyperreduction. This sampling was done in a way similar to that of the sampling for the reduced basis; the same set of simulations were run and the reduced internal force was recorded at every 4th timestep. In addition to this, one extra simulation was added where the reduced model reacted to gravity and some external perturbation that was generated interactively. This procedure generated a total of 360 training-samples, all consisting of the 17-dimensional internal force generated by the entire model as well as the 17-dimensional force density that was generated by each of the 740 individual elements. The NN-HTP algorithm was then used to analyse the data in order to select an optimal subset of the elements and compute their respective weights. In order to ensure that the relative error was small, the algorithm was set to select 150 elements, which reduced the relative error to 0.11%. Figure 4.14 shows the Tongue model with all the elements selected for hyperreduction being filled in. It is worth noting that the selected elements seem quite uniformly distributed over the model. Interestingly, the selected set of elements is not symmetric around the mid-sagittal plane. Since the reduced basis is symmetric around the mid-sagittal plane, the motions of the hyperreduced tongue will inherit this behaviour anyway but it is worth noting that



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

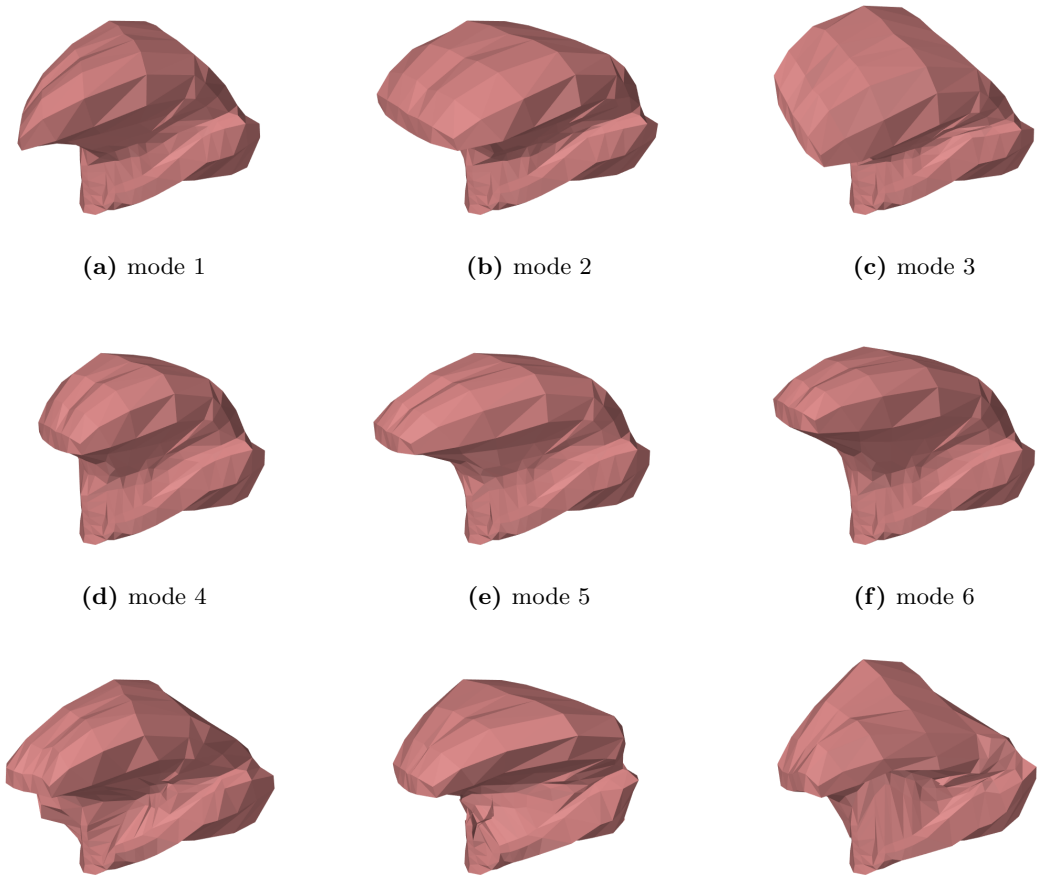
Figure 4.11: The six most significant modes from sampling of tongue deformations. Since the samples were generated through muscle activations with left-right symmetry all these modes are symmetric around the mid-sagittal plane.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

Figure 4.12: The six most significant modes from linear modal analysis of the tongue. Note that modes 1, 4, 5 and 6 are not symmetric around the mid-sagittal plane.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

(g) mode 7 **(h)** mode 8 **(i)** mode 9

Figure 4.13: The nine modes generated from the first linear mode through the extension algorithm.

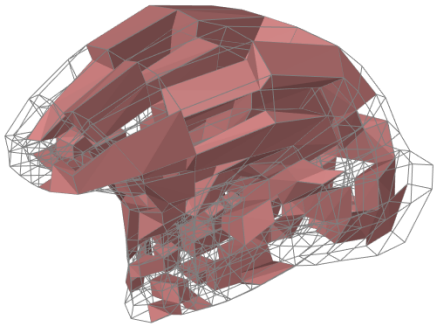


Figure 4.14: The tongue model with all the elements selected for hyperreduction being filled in.

this set of elements would have difficulty producing symmetric motions if it were to be used in combination with a non-symmetric reduced basis.

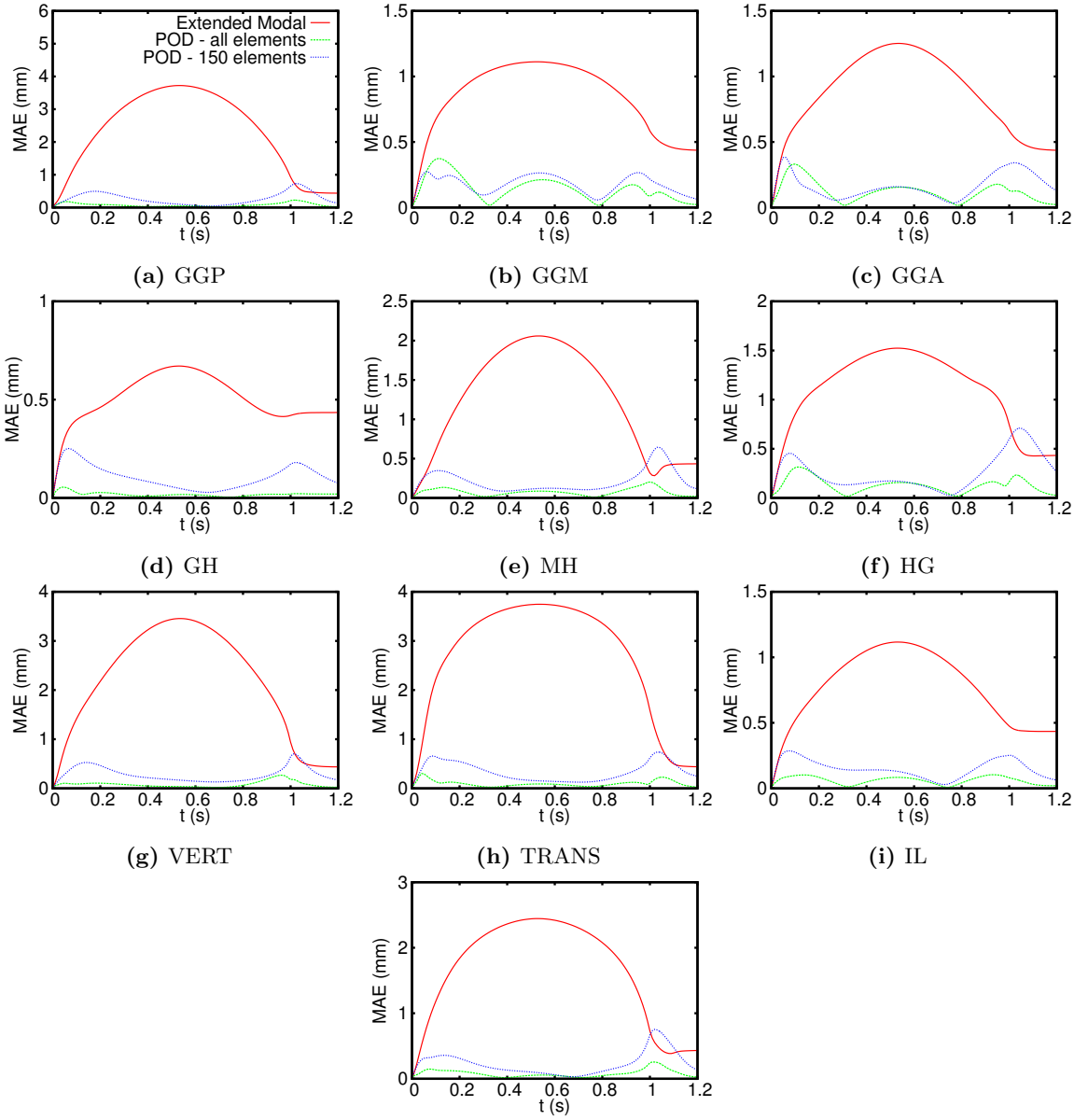
4.3.2 Dynamic Accuracy

In order to validate the reduction of this model, 10 different simulations were run for each reduced case as well as for the non-reduced model. In each simulation, one of the muscles was excited in the same way as during the training, while all other muscles stayed inactive. Since STY includes some muscle fibres outside of the tongue body, it was omitted from the validation study since the model reduction currently cannot handle the forces from external muscles. Just as for the muscle-driven beam, the displacements of all dynamic nodes were recorded at every timestep. Figure 4.15 shows the Mean Absolute Error (MAE) as a function of time for each of the reduced simulations for each validation case.

Again, the extended modal basis has the least accuracy at almost every timestep in every case despite including the most DOF. It is also very clear that the MAE for the extended modal basis and the muscle-activation peaks almost simultaneously in every simulation. This is explained by the fact that this coincides with the peak deformation of the model in all simulations, at which point the non-linearities are the most significant. The only time when the extended modal basis performs best is just after the muscle activation has been reduced to 0 in a few of the simulations (GGP, MH, HG, SL). The worst performance for all the extended modal basis occurs when exciting GGP, VERT and TRANS for which the MAE almost reaches 4 mm at maximum muscle activation.

4.3.3 Static Accuracy

For all the validation cases, the MAE of the hyperreduced simulation experiences a local maximum after around 1 second. This almost coincides with the time at which the muscle activation reaches back to 0. After this maximum, the MAE goes down again and reaches levels that are comparable to the final MAE of the



(a) GGP **(b)** GGM **(c)** GGA

(d) GH **(e)** MH **(f)** HG

(g) VERT **(h)** TRANS **(i)** IL

(j) SL

Figure 4.15: Mean Average Errors as a function of time for the three different reduction cases for ramp-up ramp-down simulations of the tongue under activation of one single muscle, one simulation for each muscle.

simulations of the other reduction cases. The occurrence of this maximum prompted the question whether it occurs as a consequence of a difference in the dynamics of the model or if there is some other cause. To answer this question, a new set of simulations were run where the displacement of all dynamic nodes were recorded as soon as the model had reached static equilibrium. The input used for these simulations were gravity and muscle activations. In the case of muscle activation, the gravity was omitted and the muscle activation was increased with cubic interpolation over 0.5 seconds and held at a specified level. One simulation was run for each reduction case and muscle, except for STY and the activation levels were set to 15% and 30%. In addition, three simulations were run where muscles were combined using activation levels that correspond to the speech sounds /a/, /i/ and /r/. In these simulations, the muscle activations were also increased with a cubic interpolation, and the final muscle activations were chosen based on previous work[58], where muscle activations had been found through interactive simulations. Just as for the dynamic validation the displacements were used to compute the MAE at the equilibrium deformation for each of the simulations.

Figure 4.16 shows the MAE for the reduced simulations when deformed under gravity. The MAE of both the reduced simulations using the POD as well as the difference between the MAE of these simulations are in fractions of milimeters in this case, which indicates that both simulations manage to reach a configuration that is very close to the equilibrium configuration of the non-reduced model. The reduced simulation using the extended modal basis produce a MAE that is far greater, but still stays below 1 mm. The MAE for the equilibrium configurations that were produced using 15% muscle activation are shown in Figure 4.17. In these cases, the MAE of the reduced simulations using the POD stays well below 1 mm and often below 0.5 mm. Again, the differences between the simulations using standard reduction and the simulations using hyperreduction is only in fractions of millimeters which indicates that the two reduction cases reach almost the same configurations. In six of the cases, the MAE of the reduced simulations using the extended modal basis reaches above 1 mm and in the other cases it is still higher than the MAE of the other reduction cases. Figure 4.18 shows the MAE for the equilibrium configurations for 30% muscle activation as well as for the configurations for /a/, /i/ and /r/ for the three different reduction cases. The MAE stays below 1 mm for all cases except for /i/, where it reaches above 3 mm for both reduction cases using the POD. In all the cases, the difference between the MAE of the reduced simulations using the POD is in fractions of milimeters. For the reduced simulations using the extended modal basis, the MAE reaches above 1 mm in nine cases and in the other case it is still higher than the MAE of the other reduction cases. The similarities of the equilibrium configurations between the standard reduced simulations and the hyperreduced ones indicate that the observed differences in behaviour between standard reduction and hyperreduction arise due to differences in the dynamics, with damping being the most likely candidate.

As indicated in Figure 4.18, the largest MAEs for all three reduced simulations are produced for the muscle activations that produce /i/. In addition, activation of GGP, VERT and TRANS produces relatively large MAEs for all three reduced simulations, although smaller than for /i/. The configurations of the non-reduced tongue and the reduced tongue for 30% activation of GGP are shown in Figures 4.19. For the

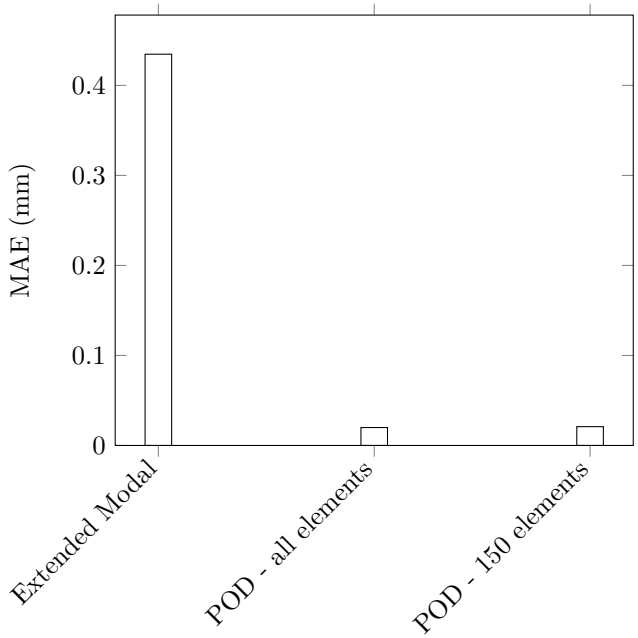


Figure 4.16: Bar chart showing the MAE at equilibrium resulting from gravity for the different reduction cases.

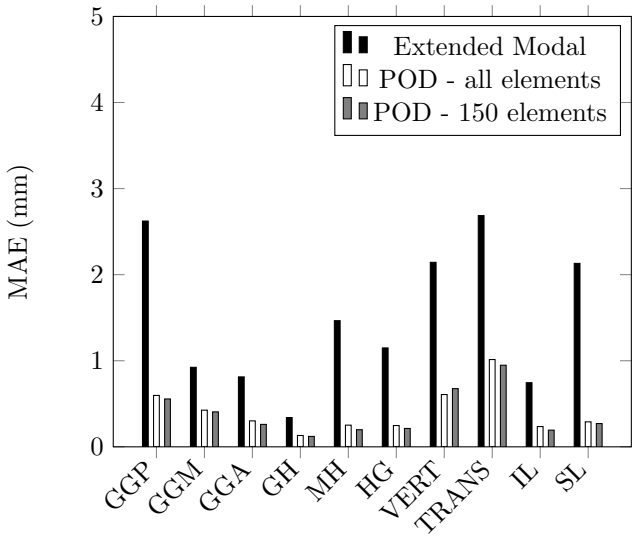


Figure 4.17: Bar chart showing the MAE at equilibrium at 15% muscle activation for the different reduction cases.

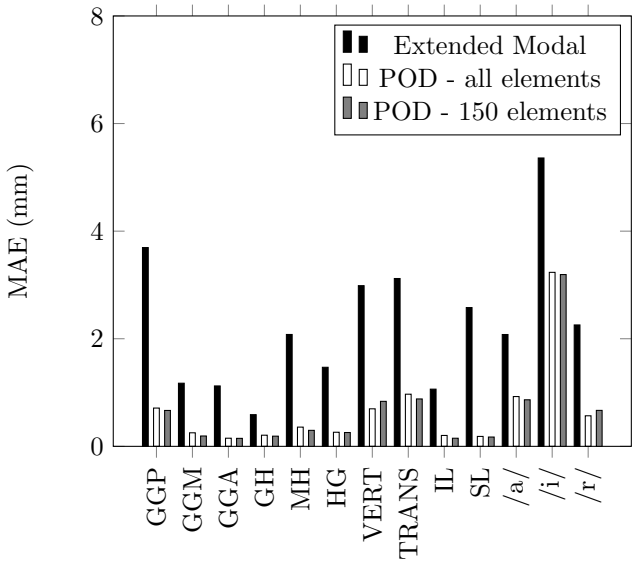
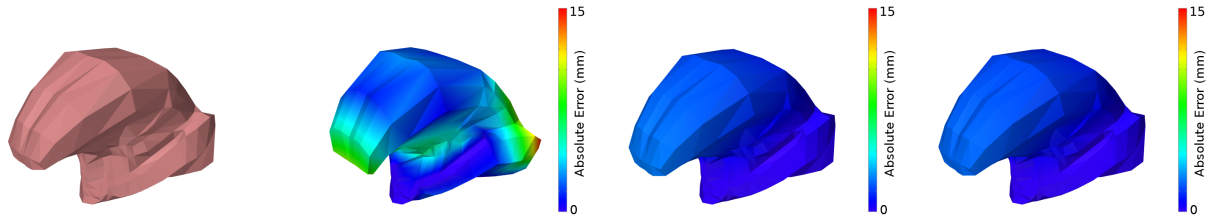


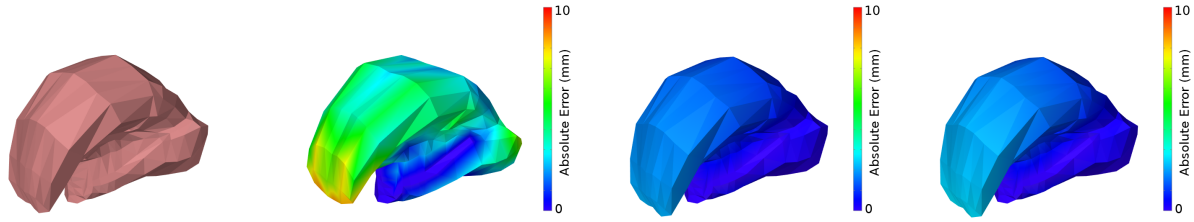
Figure 4.18: Bar chart showing the MAE at equilibrium at 30% muscle activation for the different reduction cases.

reduced simulation using the extended modal basis, the local deviation reaches as high as above 13 mm at the back corners of the root, of the tongue while reaching ca 10 mm at the tongue tip. For the two reduced simulations using the POD, the local deviation stays well below 2 mm across the entire tongue. Figure 4.20 shows the configurations of the non-reduced tongue and the reduced tongue for 30% activation of VERT. For this configuration, the reduced simulation using the extended modal basis reaches a local deviation close to 10 mm at the tongue tip, and above 5 mm at the back corners at the root of the tongue. The two simulations using the POD again show much smaller local deviation, with the simulation using all elements staying below 2 mm across the entire tongue, while the hyperreduced simulation reaches ca 3 mm in the most displaced part of the tongue. The configurations of the non-reduced and the reduced tongue for 30% activation of TRANS are shown in Figure 4.21. The reduced simulation using the extended modal basis produces local deviations around 5 mm across the entire top of the tongue for this configuration, with the largest local deviation being ca 9 mm. The two reduced simulations using the POD again produce lower local deviations, with the simulation using all elements reaching 3 mm in the area around the tip of the tongue. Interestingly, the hyperreduced simulation shows slightly less local deviation in this area, while showing almost the same local deviation across the rest of the tongue, which produces the slightly lower MAE for this muscle activation, as shown in Figure 4.18. Figure 4.22 compares the non-reduced and the reduced simulation for the static configuration produced by the muscle activations that correspond to /i/. The reduced simulation using the extended modal basis produces local deviations above 6 mm across large parts of the tongue, with the largest local deviation being above 13 mm. The two reduced simulations using the POD produce local deviations below 6 mm across most of the tongue, with a continuous increase up to 13 mm at the tip of the tongue.



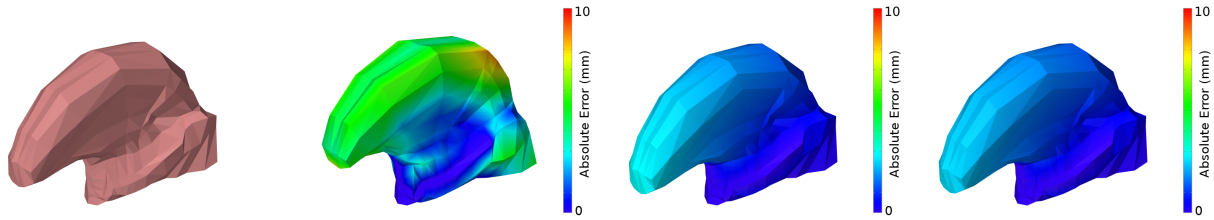
(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD- 150 elements

Figure 4.19: Comparison of equilibriums of full and reduced simulation 30% activation of GGP. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



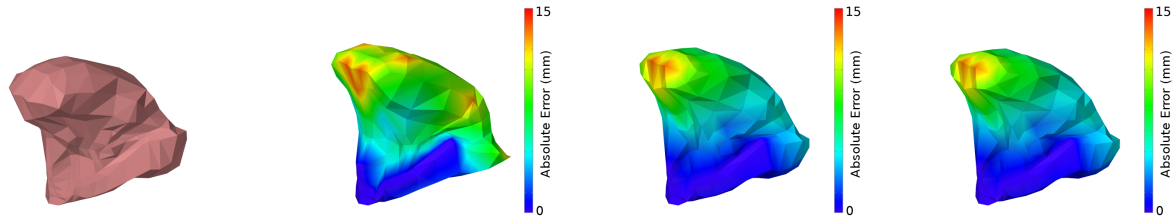
(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 150 elements

Figure 4.20: Comparison of equilibriums of full and reduced simulation 30% activation of VERT. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 150 elements

Figure 4.21: Comparison of equilibriums of full and reduced simulation 30% activation of TRANS. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 150 elements

Figure 4.22: Comparison of configurations of full and reduced simulations for muscle activations corresponding to the vowel /i/. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.

4.3.4 Computational Speed

Just as for the Muscle Driven FEM Beam, the set of simulations used for training were also used for evaluation of the computational speed-up. Table 4.2 shows the computational time per timestep and the speedup for each of the reduced simulations for every validation case. Just as was the case for the Muscle Driven FEM Beam, the reduced simulations using the extended modal basis are slower than the non-reduced simulations, while the reduced simulations using the sampled POD show a small speedup. The hyperreduced simulations, on the other hand, are more than 3 times faster than the non-reduced simulations and mostly have a smaller computational time than the actual timestep meaning that it runs in real-time. Compared to previous work this is a relatively small speedup, which is mainly attributed to the fact that a very large portion of the elements have been included in the hyperreduction.

4.4 Muscle Driven High Resolution Tongue Model

To be able to evaluate the model reduction for a realistic biomechanical model of high complexity, the model reduction has been applied to a tongue model with much higher resolution than the tongue model in the previous section. This model is still under development, and therefore the actual tongue deformations are not as realistic as for the original tongue model. However, the higher-resolution mesh allows the model to undergo larger and more fine-grained deformations, which makes it a good test case for evaluating model reduction regardless of the biomechanical accuracy of the model.

4.4.1 Model Description

The model shown in Figure 4.23 consist of 2961 nodes that are connected by 4255 elements of varying type (predominately hexahedral elements). The attachments to the jaw and hyoid are modelled by setting 88 of the nodes static (fixed boundary conditions), making 2873 of the nodes dynamic, which in total gives the model 8619 DOF. Just like in the previous section, the material of this tongue model is five parameter Mooney-Rivlin material with $C_{10} = 1037$, $C_{20} = 486$ and $C_{01} = C_{11} = C_{02} = D_1 = 0$. The density is $1040 kg/m^2$ and the stiffness- and particle-damping is 0.03 and 40, respectively. The model also uses the same 11 muscles as the tongue model of the previous section. Once again, the evaluation has been applied with the use of muscle-fibres embedded in the elements of the FEM. This time, however, the muscles have been modelled with stress proportional to the muscle activation, where $\sigma_{max} = 30000$ Pa for all muscles except VERT and TRANS, for which $\sigma_{max} = 15000$ Pa. The simulations of this model were run using implicit integration and a timestep of 10 ms.

The model has been reduced according to the procedure described in Section 3.2.3. Two reduced bases were created: one through sampling of displacement-data that was recorded through snapshots throughout simulations, and one through the creation of an extended modal basis. The snapshots were taken at every

Simulation		Full	Reduced					
			Extended Modal		POD - all elements		POD - 150 elements	
Muscle Excited		time (ms)	time (ms)	speedup	time (ms)	speedup	time (ms)	speedup
GGP	mean	39.7	46.3	0.86	33.7	1.2	12.6	3.2
	std	9.7	7.7		6.3		3.1	
GGM	mean	36.0	44.4	0.81	29.7	1.2	9.0	4.0
	std	7.7	7.1		4.4		1.7	
GGA	mean	36.8	43.7	0.8	29.0	1.3	9.8	3.8
	std	9.6	7.1		3.5		2.8	
STY	mean	35.4	42.9	0.83	28.3	1.3	9.0	3.9
	std	5.8	4.8		4.4		1.7	
GH	mean	32.4	44.1	0.73	28.2	1.1	9.3	3.5
	std	5.5	5.6		3.7		2.3	
MH	mean	33.3	45.6	0.73	28.0	1.2	8.5	3.9
	std	10.3	15.8		3.7		1.6	
HG	mean	32.4	45.7	0.71	30.0	1.1	9.3	3.5
	std	5.2	7.3		5.5		4.0	
VERT	mean	31.6	42.9	0.74	29.1	1.1	8.9	3.6
	std	4.9	5.3		4.1		2.0	
TRANS	mean	32.2	42.6	0.76	29.3	1.1	8.6	3.7
	std	5.6	4.4		4.5		1.7	
IL	mean	31.9	49.6	0.64	29.0	1.1	8.4	3.8
	std	4.9	12.4		4.6		2.2	
SL	mean	34.2	42.2	0.81	27.5	1.2	8.5	4.0
	std	9.4	4.8		3.1		1.8	
average	mean	34.2	44.5	0.77	29.3	1.2	9.3	3.7
	std	7.8	8.4		4.7		2.6	

Table 4.2: Summary of the computational time per timestep and speedup for each of the reduced simulations for all tested perturbations.

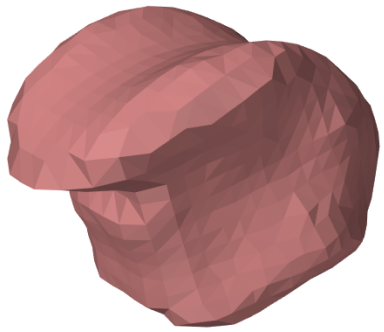
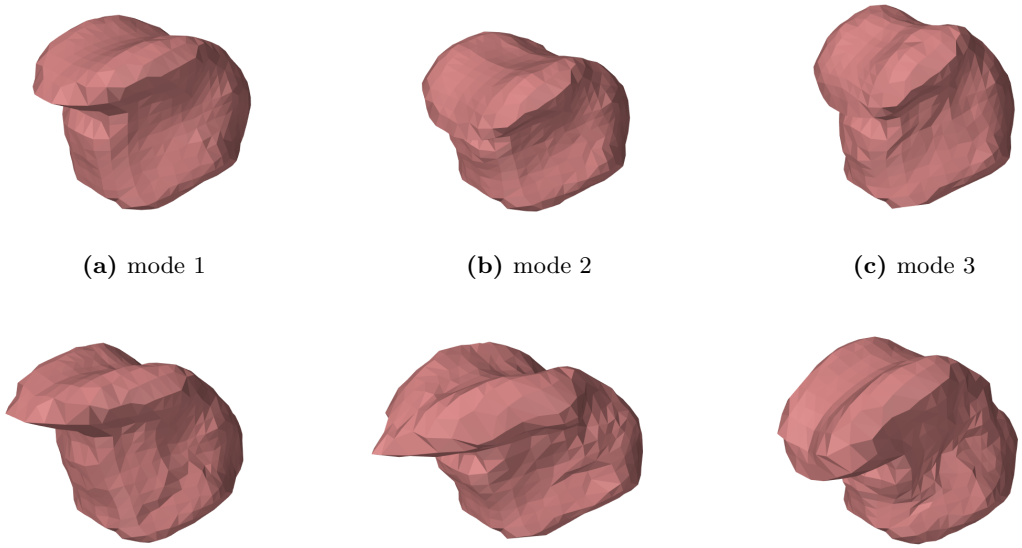


Figure 4.23: The tongue model at rest state.

timestep through 11 different simulations. In each simulation, one selected muscle was excited with left-right symmetry from 0 to 1 and then back to 0 with cubic interpolation over a period of 1 second. The model was then allowed to reach equilibrium due to gravity over the next 0.2 seconds, making each simulation 1.2 seconds long. Through this procedure, a total of 1320 training-samples were generated, each consisting of the 3D displacement of the 2873 dynamic nodes. `computeProperOrthogonalDecomposition` was used to compute a POD using SVD, that covered 99.99% of the space spanning all training-configurations. The resulting POD consists of 20 modes, 6 of whom are shown in Figure 4.24. The extended modal basis was created by computing the 6 most significant linear modes of the model, shown in Figure 4.25, and then expanding those six modes into a basis of 54 modes. It is worth noting that the modes computed from sampling have a close resemblance with actual deformations of the model. Since the model is still under development, some of these modes show behaviour that does not reflect realistic deformations of the tongue. It is also worth noting that all the modes computed from the sampled data are close to symmetric around the mid-sagittal plane, while four of the linear modes are not symmetric. The 9 modes that corresponds to the first linear mode are shown in Figure 4.26. Most of these modes show little resemblance to actual tongue-deformations. Also, since these modes are derived from the first linear mode that is not symmetric around the mid-sagittal plane, they have inherited this property.

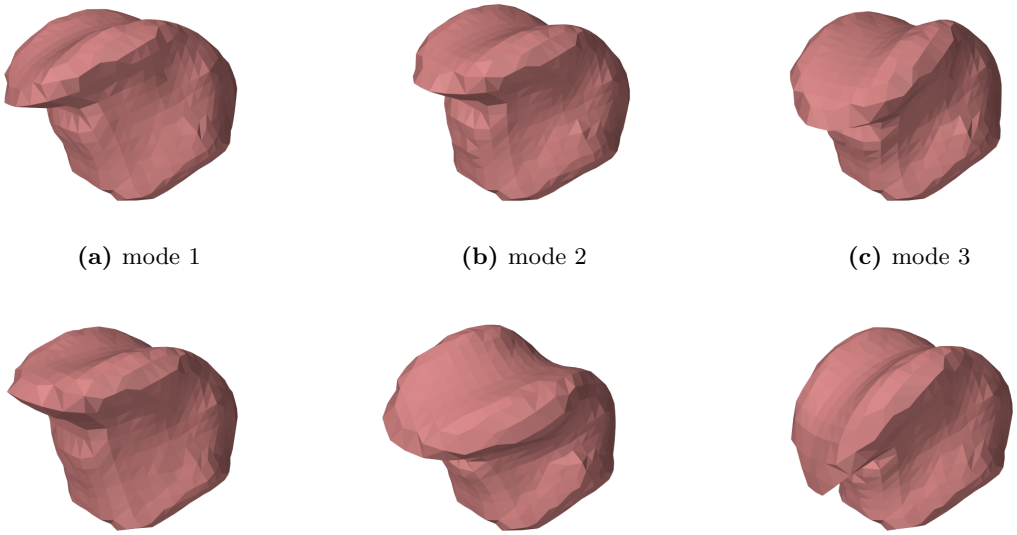
After validating the dynamic behaviour of the reduced model, a second sampling procedure was started in order to train the reduced model for hyperreduction. This sampling was done in a way similar to that of the sampling for the reduced basis; the same set of simulations were run, and the reduced internal force was recorded at every 2nd timestep. In addition to this, one extra simulation was added where the reduced model reacted to gravity and some external perturbation that was generated interactively. This procedure generated a total of 660 training-samples, all consisting of the 20-dimensional internal force generated by the entire model as well as the 20-dimensional force density that was generated by each of the 4255 individual elements. The NN-HTP algorithm was then used to analyse the data in order to select an optimal subset



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

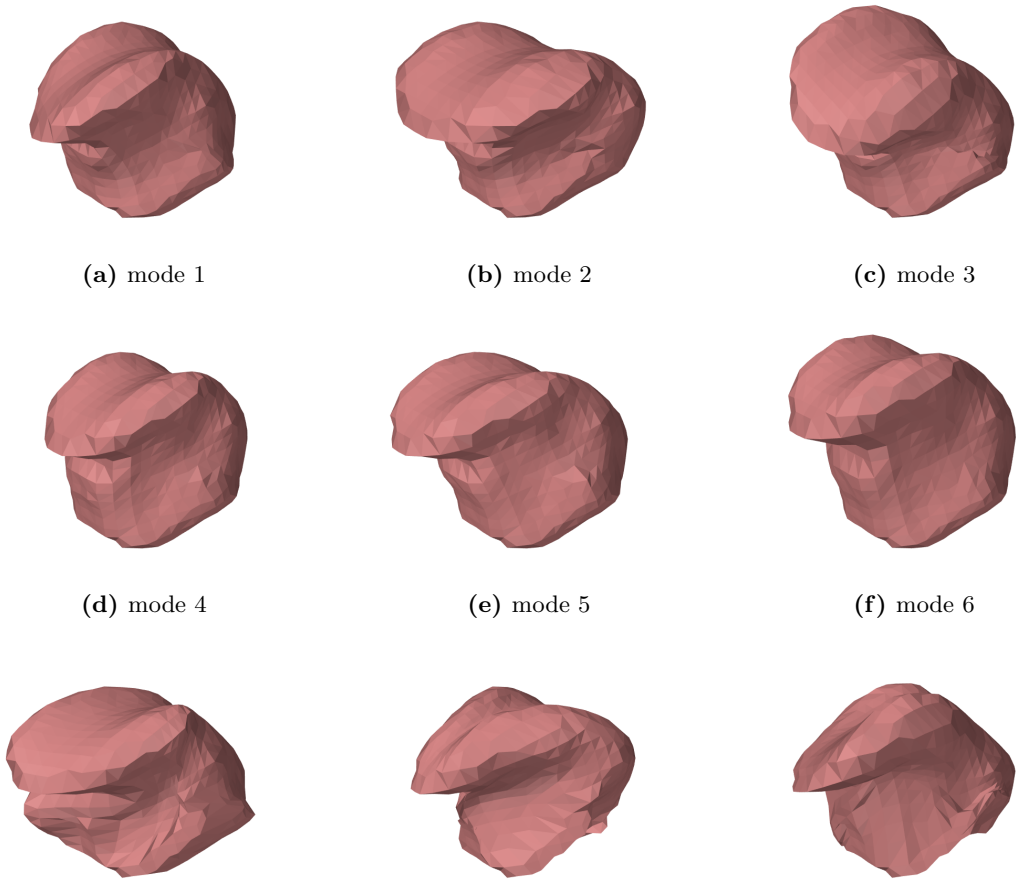
Figure 4.24: The six most significant modes from sampling of tongue deformations. Since the samples were generated through muscle activations with left-right symmetry all these modes are symmetric around the mid-sagittal plane.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

Figure 4.25: The six most significant modes from linear modal analysis of the tongue.



(a) mode 1 **(b)** mode 2 **(c)** mode 3

(d) mode 4 **(e)** mode 5 **(f)** mode 6

(g) mode 7 **(h)** mode 8 **(i)** mode 9

Figure 4.26: The nine modes generated from the first linear mode through the extension algorithm.

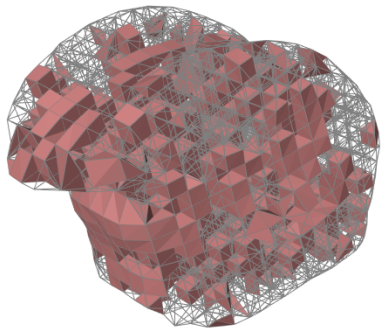


Figure 4.27: The tongue model with all the elements selected for hyperreduction filled in.

of the elements and compute their respective weights. In order to ensure that the relative error was small, the algorithm was set to select 600 elements which reduced the relative error to 0.05%. Figure 4.27 shows the tongue model with all the elements selected for hyperreduction being filled in. It is worth noting that the selected elements seems quite uniformly distributed over the model. Interestingly, the selected set of elements is not symmetric around the mid-sagittal plane. Since the reduced basis is symmetric around the mid-sagittal plane, the motions of the hyperreduced tongue will inherit this behaviour anyway but it is worth noting that this set of elements would have difficulty producing symmetric motions if it were to be used in combination with a non-symmetric reduced basis.

4.4.2 Dynamic Accuracy

In order to validate the reduction of this model, 10 different simulations were run for each reduced case, as well as for the non reduced model. In each simulation, one of the muscles were excited in the same way as during the training, while all other muscles stayed inactive. Since STY includes some muscle fibres outside of the tongue body, it was omitted from the validation study since the model reduction currently cannot handle the forces from external muscles. Just as for the validations in the previous sections, the displacements of all dynamic nodes were recorded at every timestep. Figure 4.28 shows the Mean Absolute Error (MAE) as a function of time for each of the reduced simulations for each validation case.

Again, the extended modal basis has the least accuracy at almost every timestep in every case, despite including the most DOF. It is also very clear that the MAE for the extended modal basis and the muscle-activation peaks almost simultaneously in every simulation. This is explained by the fact that this coincides with the peak deformation of the model in all simulations, at which point the non-linearities are the most significant. The only time where the extended modal basis performs best is just after the muscle activation has been reduced to 0 in some of the simulations (GGP, GGM, HG, VERT, TRANS, IL). The worst performance for all the extended modal basis occurs when exciting GGP and VERT, for which the MAE almost reaches

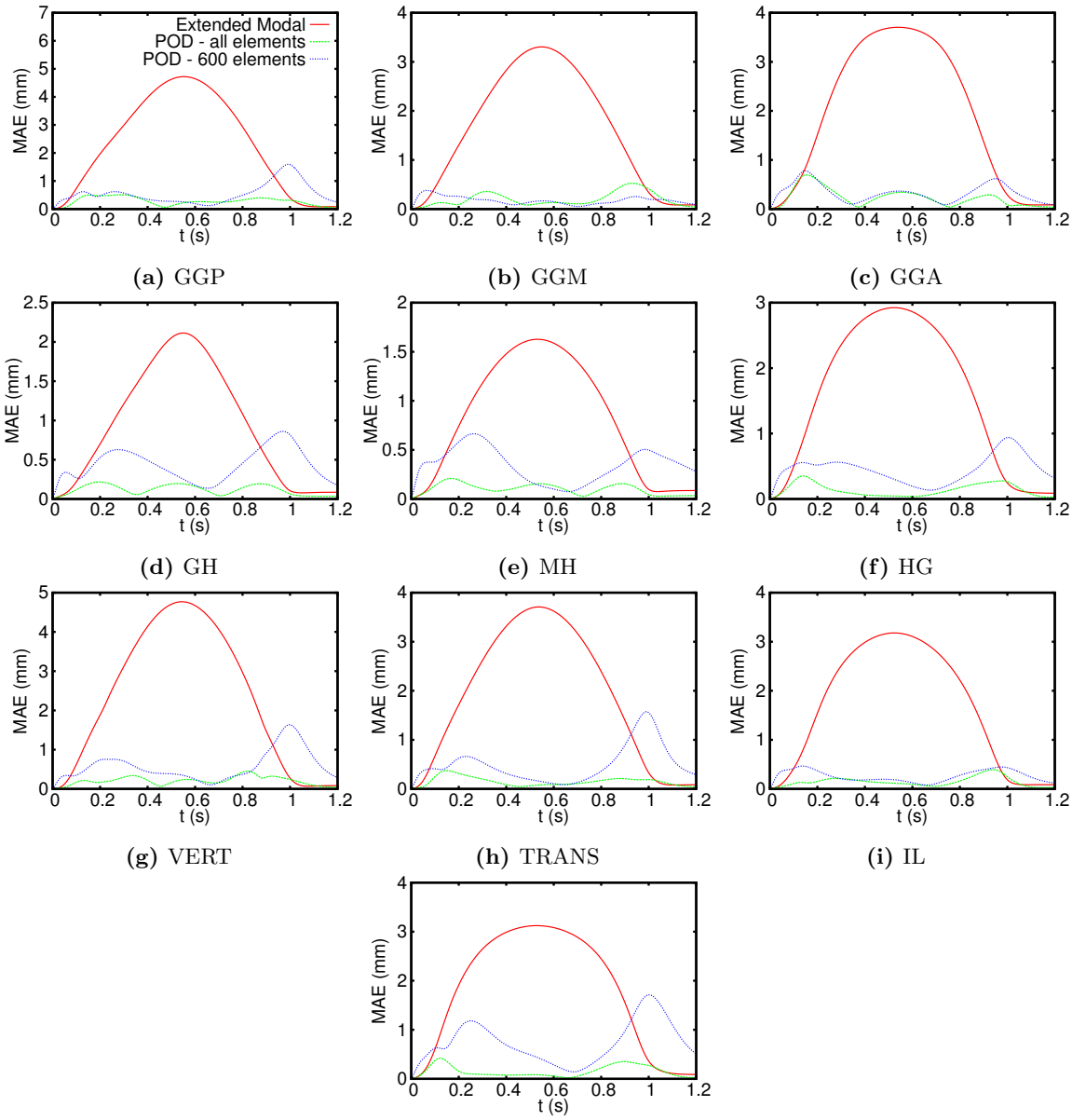
5 mm at maximum activation.

4.4.3 Static Accuracy

Just as for the tongue model investigated in the previous section, the MAE of the hyperreduced simulations experience a local maximum after around 1 second for all the validation cases. Again, this almost coincides with the time at which the muscle activation reaches back to 0, and after this maximum, the MAE goes down again and reaches levels that are comparable to the final MAE of the simulations of the other reduction cases. To further investigate this, a new set of simulations were run where the displacement of all dynamic nodes were recorded as soon as the model had reached equilibrium. The input used for these simulations were the same as that of the tongue model in the previous section, with the difference that the activation levels were held at 50% and 100% in order to assess the accuracy at static equilibrium. In addition, the same set of simulations with combined muscle activation levels that correspond to /a/, /i/ and /r/ as in the previous section were used.

Figure 4.29 shows the MAE for the reduced simulations when deformed under gravity. The MAE of both the simulations using the POD as well as the difference between the MAE of these simulations is in fractions of millimeters in this case, which indicates that both simulations manage to reach a configuration that is very close to the equilibrium configuration of the non-reduced model. The reduced simulation using the extended modal basis produce an MAE that is more than twice as large as that of the other cases. The MAE for the equilibrium configurations that were produced using 50% muscle activation are shown in Figure 4.30. In these cases, the MAE stays well below 1 mm and often below 0.6 mm. Again, the differences between the reduced simulations using the POD is only in fractions of millimeters, which indicates that the two reduction cases reach almost the same configurations. In seven of the cases, the MAE of the reduced simulations using the extended modal basis reaches above 2 mm, and in the other cases it is still higher than the MAE of the other reduction cases. Figure 4.31 shows the MAE for the equilibrium configurations for 100% muscle activation as well as for the configurations for /a/, /i/ and /r/ for the different reduction cases. The MAE stays below 1 mm for all cases, except for /i/, where it reaches above 1.5 mm for both reduced simulations using the POD. In all the cases the difference between the MAE of the reduced simulations using the POD is in fractions of milimeters. For the reduced simulations using the extended modal basis, the MAE reaches above 2 mm in nine cases, and in the other cases it is still higher than the MAE of the other reduction cases.

As indicated in Figure 4.31, the largest MAEs for the two reduced simulation that use the POD are produced for the muscle activations that produce /i/. In addition, activation of GGP, VERT and TRANS produces relatively large MAEs for all three reduced simulations, in particular for the one using the extended modal basis. The configurations of the non-reduced tongue and the reduced tongue for 100% activation of GGP are shown in Figures 4.32. For the reduced simulation using the extended modal basis, the local deviation reaches as high as above 13 mm at the tongue tip, while being around 8 mm across large portions of the tongue. For the two reduced simulations using the POD, the local deviation stays well below 5 mm across



(a) GGP **(b)** GGM **(c)** GGA

(d) GH **(e)** MH **(f)** HG

(g) VERT **(h)** TRANS **(i)** IL

(j) SL

Figure 4.28: Mean Average Errors as a function of time for the three different reduction cases for ramp-up ramp-down simulations of the tongue under activation of one single muscle, one simulation for each muscle.

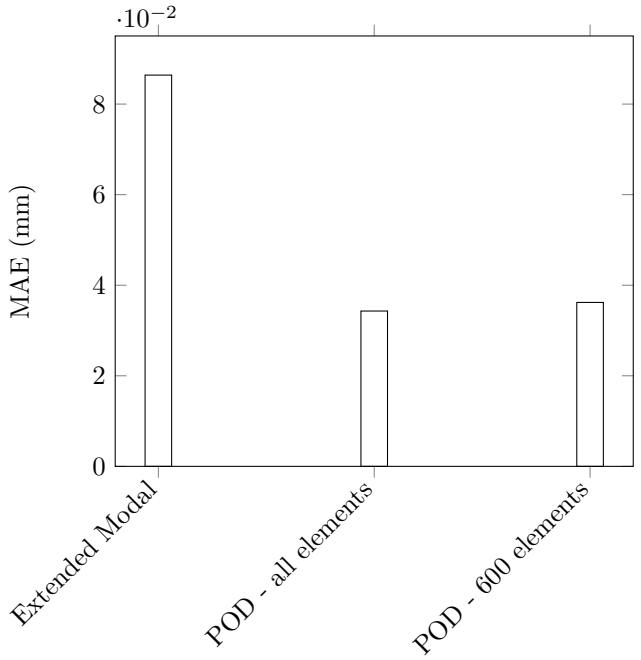


Figure 4.29: Bar chart showing the MAE at equilibrium resulting from gravity for the different reduction cases.

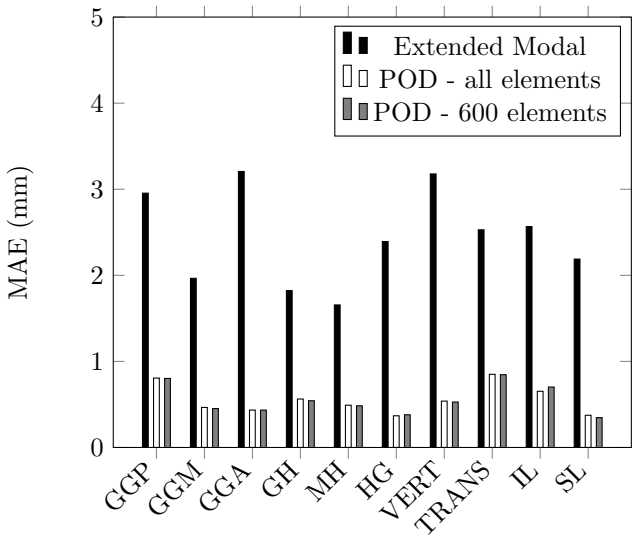


Figure 4.30: Bar chart showing the MAE at equilibrium at 50% muscle activation for the different reduction cases.

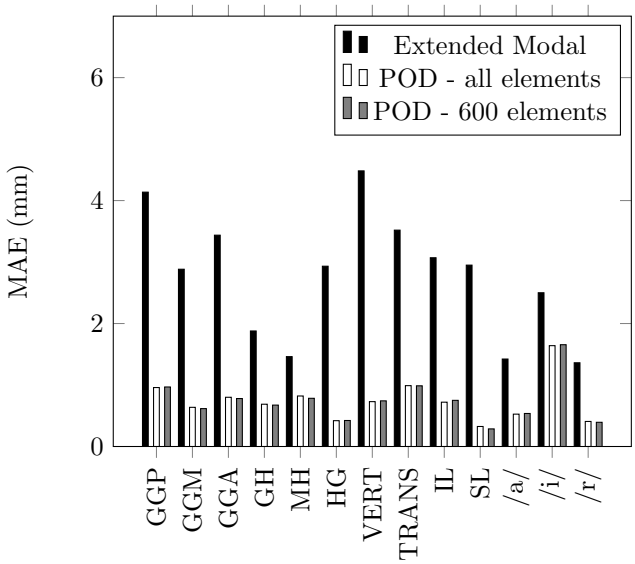
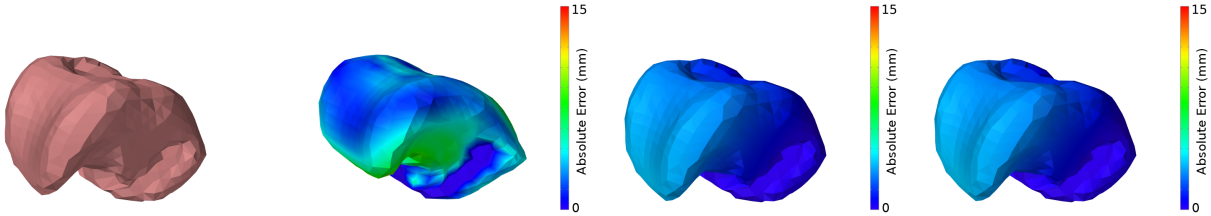


Figure 4.31: Bar chart showing the MAE at equilibrium at 100% muscle activation for the different reduction cases.

the entire tongue, especially at the base where the local deviation does not exceed 3 mm. Figure 4.33 show the configurations of the non-reduced tongue and the reduced tongue for 50% activation of VERT. As can be seen in this figure, the tongue deforms heavily already for 50% activation, and hence this activation was chosen for this illustration. For this configuration, the reduced simulation using the extended modal basis reaches a local deviation close to 13 mm at the tongue tip while staying below 5 mm for most parts of the tongue. The two simulations using the POD again show much smaller local deviation that stays below 3 mm across the entire tongue. The configurations of the non-reduced and the reduced tongue for 100% activation of TRANS are shown in Figure 4.34. The reduced simulation using the extended modal basis produces the largest local deviation at the tongue tip, where it reaches above 15 mm while reaching above 7 mm for large portions of the tongue. The two reduced simulations using the POD again produce lower local deviations, with both simulations staying below 3 mm across the entire tongue. Figure 4.35 compares the non-reduced and the reduced simulations for the static configuration produced by the muscle activations that correspond to /i/. The reduced simulation using the extended modal basis produces local deviations above 3 mm across large parts of the tongue, with the largest local deviation being above 7 mm close to the tip of the tongue. The two reduced simulations using the POD produce local deviations above 3 mm across large parts of the tongue, with the largest local deviation reaching ca 6 mm close to the tip of the tongue.

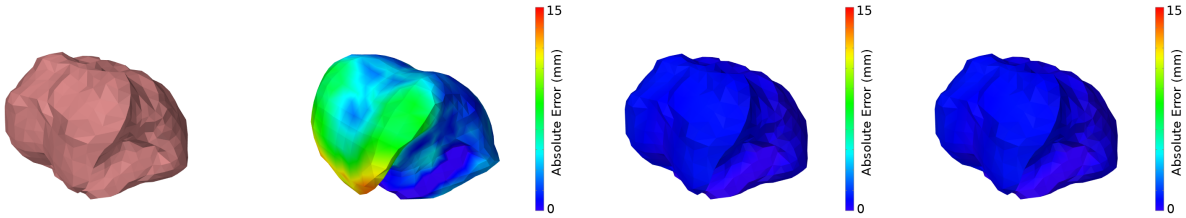
4.4.4 Computational Speed

Just as for the earlier models, the set of simulations used for training were also used for evaluation of the computational speed-up. Table 4.2 shows the computational time per timestep and the speedup for each of the reduced simulations for every validation case. Unlike the models in the previous sections, the reduced



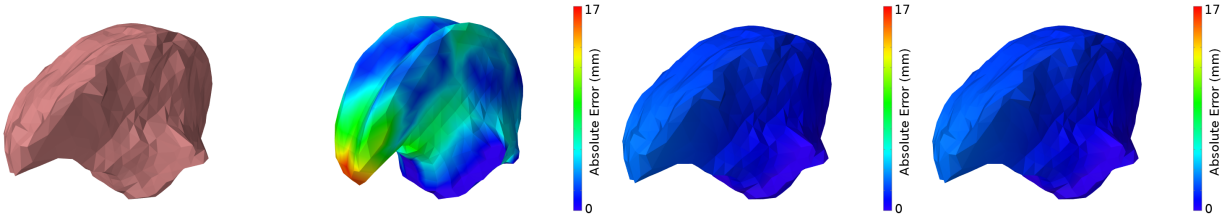
(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 600 elements

Figure 4.32: Comparison of equilibriums of full and reduced simulation 100% activation of GGP. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



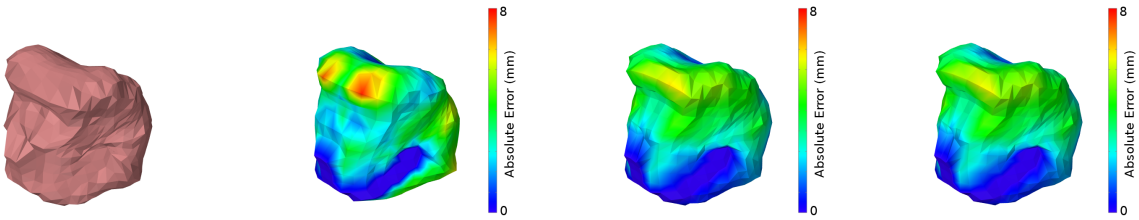
(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 600 elements

Figure 4.33: Comparison of equilibriums of full and reduced simulation 50% activation of VERT. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 600 elements

Figure 4.34: Comparison of equilibriums of full and reduced simulation 100% activation of TRANS. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.



(a) Full **(b)** Extended Modal **(c)** POD - all elements **(d)** POD - 600 elements

Figure 4.35: Comparison of configurations of full and reduced simulations for muscle activations corresponding to the vowel /i/. The colouring of the reduced models is based on the local deviation from the non-reduced simulation measured in millimeters.

simulations using the extended modal basis are slightly faster than the non-reduced simulations. The reduced simulations using the POD show a slightly larger speedup due to the POD having less than half as many DOF as the extended modal basis. The hyperreduced simulations are more than 6 times faster than the non-reduced simulations for all cases, and more than 7.5 times faster on average. Unlike for the previous sections, the hyperreduction does not reduce the computational time to less than the actual timestep. Again, a large portion of the elements were used in the hyperreduction, leading to a limited speedup.

4.4.5 Comparisons of Different Hyperreductions

The number of elements used in hyperreduction for the test cases reported above were chosen to achieve reasonably high accuracy. The resulting speedup of 7% for the high-resolution tongue is reasonable, but it is also interesting to assess how that speed and accuracy change with more severe hyperreduction, i.e., using smaller number of elements. Therefore, the NN-HTP algorithm was used to determine two additional subsets of elements, one containing 300 and one containing 200 elements, and their respective weights. The three subsets of elements were then used in simulations of protrusion, retroflexion and retraction in order to compare their accuracy and speedup. Unlike previously presented simulations, that used muscle activations that ramped-up then ramped-down back to rest, for these simulations we chose to simulate muscle activations as a ramp-and-hold in order to evaluate dynamic accuracy during the ramp phase and static accuracy at the end of the hold phase. For each of the three motions, the muscle activations were increased from 0 to the values presented in Table 4.4 with cubic interpolation over a time period of 0.5 s and then held constant. The MAE as a function of time for the three motions for all three hyperreduction cases are shown in Figure 4.36.

The hyperreduction using 600 elements shows the best accuracy for both protrusion and retroflexion. In the simulation of retraction, the 600 element case shows the best accuracy for the first half of the simulation but after the muscle activation becomes constant the 300 element hypereduction becomes more accurate as the model reaches its static equilibrium. The hyperreduction using 200 elements, on the other hand, is the least accurate for both retroflexion and retraction. In the case of protrusion, the hyperreduction using 200 elements actually shows better accuracy than the one using 300 elements.

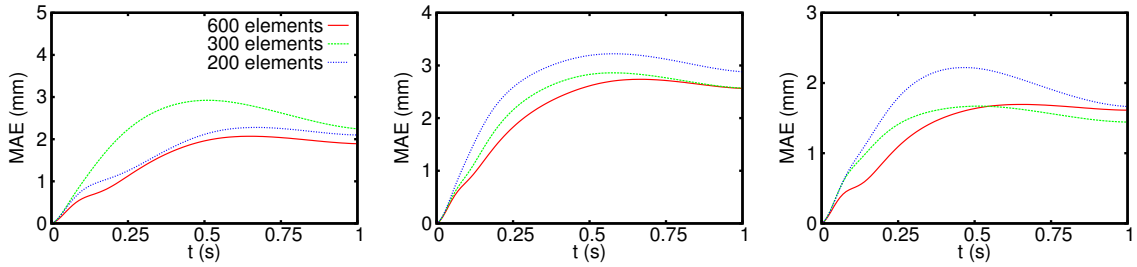
To compare the speedup of the three hyperreduction cases the computational time of each simulation time step was measured. The speedups for the three cases are presented in Table 4.5. The speedup is roughly inversely proportional to the number of elements used which is to be expected as the computations of the reduced internal force is the main remaining bottleneck. This knowledge motivates improvement of the hyperreduction training for the class of models investigated in this thesis, either through modifications of the training algorithms or through different procedures when sampling the reduced internal forces.

Simulation		Full	Reduced					
			Extended Modal		POD - all elements		POD - 600 elements	
Excited Muscle		time (ms)	time (ms)	speedup	time (ms)	speedup	time (ms)	speedup
GGP	mean	281.3	213.4	1.3	133.1	2.1	43.7	6.4
	std	37.9	23.6		17.3		9.1	
GGM	mean	286.7	214.8	1.3	133.3	2.2	39.3	7.3
	std	36.8	25.0		14.9		8.1	
GGA	mean	306.2	215.3	1.4	131.6	2.3	38.9	7.9
	std	56.6	23.4		21.9		11.0	
STY	mean	303.4	214.8	1.4	134.5	2.3	38.0	8.0
	std	52.2	21.9		16.4		5.6	
GH	mean	283.0	217.5	1.3	132.9	2.1	39.0	7.3
	std	44.0	31.5		14.2		7.3	
MH	mean	297.5	212.9	1.4	132.5	2.2	39.5	7.5
	std	50.6	21.5		14.5		7.5	
HG	mean	305.4	234.6	1.3	131.3	2.3	39.8	7.7
	std	56.7	45.9		15.2		10.5	
VERT	mean	296.4	215.0	1.4	130.5	2.3	37.7	7.9
	std	42.0	22.4		18.4		5.5	
TRANS	mean	306.3	228.2	1.3	131.2	2.3	36.8	8.3
	std	55.5	46.0		20.7		4.8	
IL	mean	285.2	214.1	1.3	130.6	2.2	39.0	7.3
	std	35.4	27.9		13.0		5.5	
SL	mean	305.6	217.1	1.4	132.4	2.3	40.6	7.5
	std	53.3	25.3		12.9		6.3	
average	mean	296.1	218.0	1.4	132.2	2.2	39.3	7.5
	std	48.8	30.5		16.5		7.8	

Table 4.3: Summary of the computational time per timestep and speedup for each of the reduced simulations for all tested perturbations.

motion	GGP	TRANS	SL	HG	IL
protrusion	0.4	0.4	0.1	-	-
retroflexion	-	0.4	0.6	-	-
retraction	-	-	-	0.6	0.4

Table 4.4: The peak muscle activations used for tests of protrusion, retroflexion and retraction.



(a) protrusion **(b)** retroflexion **(c)** retraction

Figure 4.36: Mean Average Errors as a function of time for the three different cases of hyperreduction for ramp-and-hold simulations of protrusion, retroflexion and retraction of the tongue.

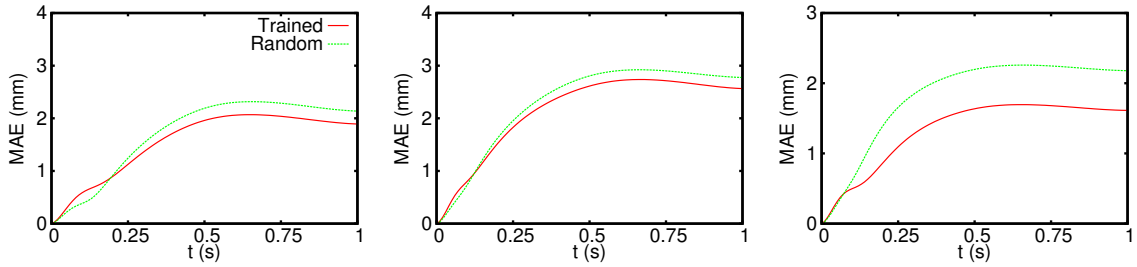
number of elements	600	300	200
speedup	6.8	12.4	15.6

Table 4.5: The computational speedup for different cases of hyperreduction using different number of elements.

4.4.6 Trained vs Random Hyperreduction

The large number of elements used for the main evaluation of hyperreduction of this tongue model provokes the question of whether or not similar accuracy could be achieved with hyperreduction using randomly selected elements. To answer this question, a subset of 600 randomly selected elements was created. The weights were set the same for all the elements and was set so that the sum of the weights equal the number of elements in the entire model. The same set of simulations as in Section 4.4.5 were used, which also allows for comparisons between the randomly selected subset of elements to smaller subsets produced from training.

Figure 4.37 shows the Mean Absolute Error as a function of time for all three motions for the trained and the randomly selected 600 element hyperreductions. The trained hyperreduction produces better accuracy than the randomly selected hyperreduction for each of the motions, but the difference is only large for retraction. It is also worth noting that the randomly selected hyperreduction shows better accuracy for parts of the motions than the trained hyperreduction using 200 elements that was investigated in Section 4.4.5.



(a) protrusion **(b)** retroflexion **(c)** retraction

Figure 4.37: Mean Average Errors as a function of time for trained and random hyperreduction for ramp-and-hold simulations of protrusion, retroflexion and retraction of the tongue.

4.5 Discussion

This chapter has presented the evaluation of model reduction to three different muscle driven tissue models, one beam model and two tongue models. In all three cases two different reduced bases were used, one computed by extension of a modal basis and one POD computed through the SVD of displacement-data. In addition, a hyperreduction was used for the POD for all three models, in order to compare the accuracy and computational speed-up. The extended modal bases showed the least accuracy almost exclusively in all three cases, mainly because this type of base only takes non-linear deformations but not non-linear elasticity into account. The hyperreduced simulations showed slightly less accuracy than their respective standard reduced simulations. In particular the hyperreduced simulations showed their largest errors when muscle activations were decreasing towards 0. Additional static measurements indicate that such errors are due to differences in the dynamics.

One of the reasons that the reduced simulations using the POD showed better accuracy than those using the extended modal basis might be the fact that the models used muscle fibres embedded within the elements. This way of modelling muscles means that the internal stress and stiffness of the model change, not only because of deformation, but also because of muscle activation. This limits the accuracy that a modal basis can produce since LMA only take the passive stiffness at rest into account while this type of muscle model can produce a wide range of stresses and stiffnesses for the same deformation. Another aspect of this issue is that the muscle activations cause a change in stress and stiffness locally in the model which cause deformations that are simply not predictable when using LMA. A POD on the other hand, takes this type of muscle model into account naturally by only focusing on the displacements that are produced throughout simulations, either from muscle activations or from external forces.

As expected the computational speed-up was largest for the high resolution tongue model since the model reduction reduced most DOF in that case. In addition there was also a clear difference in computational speed-up between the different reduced bases because of their inclusion of different DOF. The amount of elements required for the hyperreduction to uphold acceptable accuracy was higher for the models explored here than it has been in previous work where it has been applied to inactive materials. This generally caused a smaller computational speed-up than what had otherwise been the case, although the computational speed-up for the two tongue models is still significant. It is also worth noting that a smaller fraction of the total number of elements was required as the complexity of the models increased, indicating good potential for use of hyperreduction on tissue models of even higher complexity.

Chapter 5

Conclusions

This thesis has described the implementation and evaluation of different model reduction techniques for biomechanical simulation. In particular, finite element models with constitutive equations representing muscle tissue have been investigated. In this chapter, I review the main contributions of the thesis and describe a few promising directions for further work.

5.1 Contributions

There are three overall contributions of this thesis. First, an open source implementation that includes many of the existing model reduction techniques has been created. Second, model reduction has been applied to muscle driven FEMs instead of those consisting of passive materials, as has been the case until now. Third, the validity and effectiveness of some of the implemented techniques have been compared for muscle driven FEMs.

5.1.1 Implementation of Model Reduction in ArtiSynth

The first contribution of this thesis is the creation of an open source implementation of model reduction for muscle-driven biomechanical models. This implementation includes many of the existing techniques for creating a reduced basis and for further improvement of dynamic simulation efficiency through hyperreduction. The included techniques cover aspects of the entire workflow of model reduction generation, either through linear modal analysis or through sampling, to the solution of the reduced equations of motion and the computation of reduced internal force, either through simple projection of the non-reduced entities or through hyperreduction. The included techniques have been selected with the goal that the framework should be able to handle as general simulations as possible. For this reason, the included techniques are not specific to certain geometries, types of elements or material models. The implementation is open source and freely available, with the hope that it is used and extended by the biomechanical modelling community.

5.1.2 Model Reduction applied to MuscleFemModel

The implementation has been tested on muscle driven tissue models in order to test the validity of model reduction for such FEMs. The results show a successful reduction of muscle driven FEMs where the dynamic

fidelity is within acceptable limits. As expected the computational speed-up from model reduction was greater for FEMs of higher complexity and even greater after the addition of hyperreduction. Hyperreduction proved useful for muscle driven FEMs, although it generally required more elements than it usually does for FEMs consisting of inactive material. The large number of elements required to maintain accuracy for hyperreduction of muscle FEMs led to somewhat lower computational speed-up as compared to passive structures alone. Nevertheless, when hyperreduced the high-resolution FEM tongue model ran 7 times faster when using 600 elements and showed a speedup that was roughly inversely proportional to the number of elements used. This will have a significant impact on the usability of that model for articulatory speech synthesis studies.

5.1.3 Comparisons of Model Reduction Approaches

Since model reduction has not been applied to muscle driven FEMs earlier, the performance of the different techniques has not been compared for this case. For this reason, some of the techniques implemented in this work have been compared for muscle driven FEMs. Reduced bases produced by sampling performed better than extended modal bases due to the fact that extended modal bases take non-linear deformations but not non-linear elasticity into account. Some differences regarding dynamic accuracy was noted between hyperreduced simulations and their respective standard reduced simulations. After additional comparisons using static configurations, these differences were concluded to arise from the dynamics, most likely related to damping. For the tongue models, the best case accuracy resulted in visibly similar tongue postures with the reduced model and overall accuracy within 2 mm MAE for almost all cases.

5.2 Future Work

5.2.1 Model Reduction for Vocal Tract Biomechanics

The current implementation of model reduction is able to efficiently handle relatively simple simulations of biomechanics, that is unconstrained FEMs in static frames. With this in mind, the natural next step is to expand the implementation to be able to handle more advanced biomechanical simulations. In the case of articulatory speech synthesis, it could be particularly interesting to apply model reduction to a model[2] that includes all major structures of the vocal tract. This model includes FEMs of the tongue[59], the soft palate[19], the larynx[37] and the pharynx as well as rigid bodies for modelling the skull, jaw and hyoid bones. Applying model reduction to this model would require the implementation to handle multi-body simulations including both constraints and rigid motions. The constraints can be either attachments, such as between the jaw and the tongue, contacts, for example between the tongue and the soft palate, or connections through muscles, such as between the tongue and the maxilla. The rigid motions could, for example, be tongue motion due to jaw motion. In this case, at least parts of the tongue experience purely rigid motion. These types

of situations could be handled by the use of floating frames. The frame of the reduced tongue would then have to be attached to the jaw which would allow it to follow the motions of the jaw while letting the tongue deform at the same time. The nodes of the tongue that are attached to the jaw will not have to be included in the reduced basis but can instead follow the motions of the frame entirely.

5.2.2 Improved Efficiency

For some biomechanical simulations of high complexity, it could be inefficient to create the reduced basis from sampled data since that requires simulation of the non-reduced FEM. It can at the same time be insufficient to use a linear modal basis or an extended modal basis since they, while not requiring any pre-simulations, might not be able to handle non-linearities of the FEM. For this reason, it could be interesting to implement and test techniques for online reduction[29]. This would allow biomechanical simulations to start with non-reduced FEMs and then reduce them as the simulations progresses, with a continuous update of the reduced basis and hyperreduction. The advantage with this approach would be that it potentially combines the accuracy of PODs with the efficiency of not having to pre-simulate the FEMs before reducing them.

Another possible direction to improve the efficiency of model reduction for muscle-driven tissue models is to implement a *case-sensitive* hyperreduction. This has already been successfully attempted for simulations of self-collisions[61] where the detection of collision and computation of collision forces were computed for subsets of surface points. The selection of which subset to use was based on the configuration of the FEM with the possibility of interpolating between subsets. It is possible that this concept could be transferred to muscle-driven tissue simulations by, for example, creating subsets of elements for each individual muscle exciter. The interpolation could, in this case, be made using the muscle excitations as input parameters. Through this construction, it might be possible to use a smaller number of elements when computing the internal force than has been the case in this thesis.

References

[1] Steven S An, Theodore Kim, and Doug L James. Optimizing cubature for efficient integration of subspace deformations. In *ACM transactions on Graphics (TOG)*, volume 27, page 165. ACM, 2008.

[2] Peter Anderson, Negar M Harandi, Scott R Moisik, Ian Stavness, and Sidney Fels. A comprehensive 3d biomechanically-driven vocal tract model including inverse dynamics for speech research. In *Interspeech 2015: 16th Annual Conference of the International Speech Communication Association*, pages 2395–2399, 2015.

[3] Jernej Barbič and Doug L. James. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.*, 24(3):982–990, July 2005.

[4] Peter Birkholz, Dietmar Jackèl, and Bernd J Kroger. Construction and control of a three-dimensional vocal tract model. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE, 2006.

[5] Silvia S. Blemker, Peter M. Pinsky, and Scott L. Delp. A 3d model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38(4):657 – 665, 2005.

[6] Thomas Blumensath and Mike E Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 4(2):298–309, 2010.

[7] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[8] Stéphanie Buchaillard, Pascal Perrier, and Yohan Payan. A biomechanical model of cardinal vowel production: muscle activations and the impact of gravity on tongue positioning. *The Journal of the Acoustical Society of America*, 126(4):2033–2051, 2009.

[9] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. Interactive skeleton-driven dynamic deformations. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 586–593. ACM, 2002.

[10] Min Gyu Choi and Hyeong-Seok Ko. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):91–101, 2005.

[11] John C Criscione, Andrew S Douglas, and William C Hunter. Physically based strain invariant set for materials exhibiting transversely isotropic behavior. *Journal of the Mechanics and Physics of Solids*, 49(4):871–897, 2001.

[12] Francis A Duck. *Physical properties of tissues: a comprehensive reference book*. Academic press, 1990.

[13] C Farhat, J Cortial, and T Chapman. A hyper-reduction method for nonlinear structural dynamics reduced-order models. In *Tenth World Congress on Computational Mechanics (WCCM X), vol. Book of Abstracts, Sao Paolo, Brazil*, 2012.

[14] Charbel Farhat, Philip Avery, Todd Chapman, and Julien Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.

[15] Charbel Farhat, Todd Chapman, and Philip Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering*, 102(5):1077–1110, 2015.

[16] Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[17] Yuan-cheng Fung. *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media, 2013.

[18] Jean-Michel Gérard, Jacques Ohayon, Vincent Luboz, Pascal Perrier, and Yohan Payan. Non-linear elastic properties of the lingual and facial tissues assessed by indentation technique: application to the biomechanics of speech production. *Medical engineering & physics*, 27(10):884–892, 2005.

[19] Bryan Gick, Peter Anderson, Hui Chen, Chenhao Chiu, Ho Beom Kwon, Ian Stavness, Ling Tsou, and Sidney Fels. Speech function of the oropharyngeal isthmus: a modelling study. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2(4):217–222, 2014.

[20] Bryan Gick, Ian Wilson, and Donald Derrick. *Articulatory phonetics*. John Wiley & Sons, 2012.

[21] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[22] J.A. Hernndez, M.A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687 – 722, 2017.

[23] Jin Huang, Yiying Tong, Kun Zhou, Hujun Bao, and Mathieu Desbrun. Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):983–992, 2011.

[24] Andrew J Hunt and Alan W Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, pages 373–376. IEEE, 1996.

[25] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. Tetrahedral and hexahedral invertible finite elements. *Graphical Models*, 68(2):66–89, 2006.

[26] Doug L James, Jernej Barbič, and Dinesh K Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 987–995. ACM, 2006.

[27] Doug L James and Dinesh K Pai. Dyrt: dynamic response textures for real time deformation simulation with graphics hardware. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 582–585. ACM, 2002.

[28] Theodore Kim and John Delaney. Subspace fluid re-simulation. *ACM Transactions on Graphics (TOG)*, 32(4):62, 2013.

[29] Theodore Kim and Doug L James. Skipping steps in deformable simulation with online model reduction. In *ACM transactions on graphics (TOG)*, volume 28, page 123. ACM, 2009.

[30] Theodore Kim and Doug L James. Physics-based character skinning using multidomain subspace deformations. *IEEE transactions on visualization and computer graphics*, 18(8):1228–1240, 2012.

[31] P Krysl, S Lall, and JE Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for numerical methods in engineering*, 51(4):479–504, 2001.

[32] John E Lloyd, Antonio Sánchez, Erik Widing, Ian Stavness, and Sidney Fels. *New Developments on Computational Methods and Visualization in Biomechanics and Biomedical Engineering*, chapter New Techniques for Combined FEM-multibody Anatomical Simulation. Springer.

[33] John E Lloyd, Antonio Sánchez, Erik Widing, Ian Stavness, and Sidney Fels. New techniques for combined fem-multibody anatomical simulation.

[34] John E Lloyd, Ian Stavness, and Sidney Fels. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 355–394. Springer, 2012.

[35] Dimitri Metaxas and Demetri Terzopoulos. Dynamic deformation of solid primitives with constraints. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 309–312. ACM, 1992.

[36] Mark Meyer and John Anderson. Key point subspace acceleration and soft caching. *ACM Trans. Graph.*, 26(3), July 2007.

[37] Scott Reid Moisik and Bryan Gick. The quantal larynx: The stable regions of laryngeal biomechanics and implications for speech production. *Journal of Speech, Language, and Hearing Research*, 60(3):540–560, 2017.

[38] Melvin Mooney. A theory of large elastic deformation. *Journal of applied physics*, 11(9):582–592, 1940.

[39] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM, 2002.

[40] Matthias Müller and Markus Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, pages 239–246. Canadian Human-Computer Communications Society, 2004.

[41] Mohammad Ali Nazari, Pascal Perrier, Matthieu Chabanas, and Yohan Payan. Simulation of dynamic orofacial movements using a constitutive law varying with muscle activation. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(4):469–482, 2010.

[42] Matthieu Nesme, Yohan Payan, and François Faure. Efficient, physically plausible finite elements. In *Eurographics*, 2005.

[43] Wai-Hin Ngan and John Lloyd. Efficient deformable body simulation using stiffness-warped nonlinear finite elements. In *Symposium on Interactive 3D Graphics and Games (i3D)*, 2008.

[44] Siamak Niroomandi, Icíar Alfaro, Elías Cueto, and Francisco Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer methods and programs in biomedicine*, 91(3):223–231, 2008.

[45] Raymond W Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997.

[46] Raymond William Ogden. Large deformation isotropic elasticity–on the correlation of theory and experiment for incompressible rubberlike solids. *Proc. R. Soc. Lond. A*, 326(1567):565–584, 1972.

[47] Pertti Palo. A review of articulatory speech synthesis. *Master's thesis, Helsinki University of Technology, Department of Electrical and Communications Engineering*, 2006.

[48] Yohan Payan and Pascal Perrier. Synthesis of vv sequences with a 2d biomechanical tongue model controlled by the equilibrium point hypothesis. *Speech communication*, 22(2-3):185–205, 1997.

[49] Steve Pearson, Nicholas Kibre, and Nancy Niedzielski. Formant-based speech synthesizer employing demi-syllable concatenation with independent cross fade in the filter parameter and source domains, November 7 2000. US Patent 6,144,939.

[50] Alex Pentland and John Williams. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics*, 23(3):215–222, July 1989.

[51] Annika Radermacher and Stefanie Reese. A comparison of projection-based model reduction concepts in the context of nonlinear biomechanics. *Archive of Applied Mechanics*, 83(8):1193–1213, 2013.

[52] Ronald S Rivlin and DW Saunders. Large elastic deformations of isotropic materials vii. experiments on the deformation of rubber. *Phil. Trans. R. Soc. Lond. A*, 243(865):251–288, 1951.

[53] RS Rivlin. Large elastic deformations of isotropic materials iv. further developments of the general theory. *Phil. Trans. R. Soc. Lond. A*, 241(835):379–397, 1948.

[54] Youcef Saad. *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

[55] Ahmed A Shabana. *Theory of vibration: Volume II: discrete and continuous systems*. Springer Science & Business Media, 2012.

[56] Ahmed A Shabana. *Dynamics of multibody systems*. Cambridge university press, 2013.

[57] Fun Shing Sin, Daniel Schroeder, and Jernej Barbič. Vega: Non-linear FEM deformable object simulator. In *Computer Graphics Forum*, volume 32, pages 36–48. Wiley Online Library, 2013.

[58] Ian Stavness, Bryan Gick, Donald Derrick, and Sidney Fels. Biomechanical modeling of english/r/variants. *The Journal of the Acoustical Society of America*, 131(5):EL355–EL360, 2012.

[59] Ian Stavness, John E Lloyd, and Sidney Fels. Automatic prediction of tongue muscle activations using a finite element model. *Journal of biomechanics*, 45(16):2841–2848, 2012.

[60] Ian Stavness, John E Lloyd, Yohan Payan, and Sidney Fels. Coupled hard–soft tissue simulation with contact and constraints applied to jaw–tongue–hyoid dynamics. *International Journal for Numerical Methods in Biomedical Engineering*, 27(3):367–390, 2011.

[61] Yun Teng, Miguel A Otaduy, and Theodore Kim. Simulating articulated subspace self-contact. *ACM Transactions on Graphics (TOG)*, 33(4):106, 2014.

[62] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1315–1318. IEEE, 2000.

[63] Kees Van Den Doel, Florian Vogt, R Elliot English, and Sidney Fels. Towards articulatory speech synthesis with a dynamic 3d finite element tongue model. In *International Seminar on Speech Production, Ubatuba, Brazil*. Citeseer, 2006.

[64] Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. An efficient construction of reduced deformable objects. *ACM Trans. Graph.*, 32(6):213:1–213:10, November 2013.

[65] Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. Real-time nonlinear shape interpolation. *ACM Transactions on Graphics (TOG)*, 34(3):34, 2015.

[66] OH Yeoh. Some forms of the strain energy function for rubber. *Rubber Chemistry and technology*, 66(5):754–771, 1993.