PARTICLE-BASED MODELING
of
NONLINEAR VISCOELASTIC DEFORMABLE OBJECTS
based on
EXPERIMENTAL DATA

by

Mert Sedef

A Thesis Submitted to the
Graduate School of Computational Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Computational Sciences and Engineering

Koç University
May 2008

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Mert Sedef

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Basdogan

Çağatay Başdoğan, Ph. D. (Advisor)

M. Sozer

Murat Sözer, Ph. D.

Yücel Yemez, Ph. D.

Date: 05.15.2008

To them who have a place for me in their hearts

ABSTRACT

Simulation-based training using Virtual Reality techniques is a promising alternative to traditional training in minimally invasive surgery. Surgical simulators let the trainee touch, feel, and manipulate virtual tissues and organs through the same surgical tool handles used in actual minimally invasive surgery while viewing images of tool-tissue interactions on a monitor as in real laparoscopic procedures. Developing realistic organ-force models for simulating soft-tissue behavior is an integral part of a surgical simulator.

The particle system approach provides a better solution than mesh-based methods to the topological changes encountered in simulation of surgical cutting and tearing. In addition, they are computationally less expensive and easier to implement than the mesh-based methods. However, the material coefficients of each individual mesh element should be calculated and fine tuned to integrate the realistic tissue properties into particle models, which is not trivial.

This thesis presents an end-to-end solution to realistic particle-based simulation of nonlinear viscoelastic tissue behavior based on the experimental data collected by a robotic indenter. First, the strain-dependent nonlinear elastic response and time-dependent viscoelastic response of a tissue-like silicon phantom is measured via static loading and stress relaxation experiments performed by a robotic indenter. The collected experimental data is used to construct a lumped model of the tissue phantom represented by a nonlinear viscoelastic Maxwell Solid. Then, a 3-dimensional particle-based network is developed to mimic the behavior of the lumped Maxwell model. The material coefficients of the individual Maxwell elements connecting the particles are estimated through a set of novel optimization algorithms.

ÖZET

Sanal gerçeklik kullanılarak oluşturulan simülasyon bazlı eğitim, geleneksel minimal invasiv ameliyat eğitim tekniklerine alternatif olacaktır. Cerrahi simülatörler, stajyerlerin, gerçek laporoskopik operasyonlarda olduğu gibi, sanal doku ve organlara dokunmasına, hissetmesine ve hareket ettirmesine, doku ile cerrahi aletler arasındaki etkileşimleri gözlemlemesine, imkan sağlayacaktır. Yumuşak doku davranışlarını iyi taklit edebilmek için, gerçeğe çok yakın doku modelleri geliştirilmesi gerekmektedir.

Parçacık modelleri, yumuşak cisimlerin simulasyonun da oldukça kullanılan bir yaklaşımdır. Bu teknik, cerrahi kesim ve yırtılma gibi biçimsel değişimlerin olduğu konularda ağ bazlı metotlara (örnek: sonlu eleman methodu) göre daha iyi bir yaklaşım sağlar. Ayrıca parçacık metodları, ağ bazlı olanlara göre hesaplama açısından daha basittir ve uygulanması daha kolaydır. Yine de, fiziksel sistemin tepkisini belirlemek için her bir ağ elemanının malzeme katsayıları hesaplanmalıdır. Bu nedenle, gerçekçi doku özelliklerinin parçacık modeline entegre edilmesi kolay değildir.

Bu tezde, bir robot kolu aracılığıyla toplanan deney verilerini kullanan, doğrusal olmayan viskoelastik doku davranışını simule eden bir parçacık modeli geliştirdik. İlk olarak, yumuşak doku özelliği gösteren bir silikon örneğin gerilime bağlı doğrusal olmayan elastik tepkisi ve zamana bağlı viskoelastik tepkisi ölçüm cihazı ile yapılan statik yükleme ve makaslama gevşemesi deneyleriyle elde edilmiştir. Toplanan veriler yardımıyla, silikon örnek doğrusal olmayan viskoelastik Maxwell katısı kullanılarak modellenmiştir. Daha sonra, bu Maxwell katısı ile aynı davranışı gösteren, doğrusal olmayan viskoelastik Maxwell katılarından oluşan 3 boyutlu parçacık bazlı bir ağ kurulmuştur. Parçacıkları birleştiren bireysel Maxwell katılarının malzeme özellikleri, geliştirilen bir dizi özgün optimizasyon algoritması ile belirlenmiştir.

ACKNOWLEDGEMENTS

I consider myself a very lucky student for being able to study under the supervision of Prof. Çağatay Başdoğan for almost five years. I would like to express my sincere gratitude and appreciation to him for his expert guidance, suggestions and constructive criticisms at all phases of this study and many others that we have successfully completed in the past. He has been more than an advisor to me and I learned a lot from him. I thank him for teaching me what I know about research. I am grateful to him also for his generous and continuous support in non-academic phases of my life.

I would like to thank Canberk Manav for the insightful discussions we had and for his collaboration to my thesis; Umut Özcan, Sina Öcal, Baybora Baran, Osman Yoğurtçu, and İsmail İyigünler for editing and carefully proofreading my thesis, and for providing constructive criticism.

During the hardest times, two special people, Serdar İnal and Janice Osborne, were there to catch me whenever I was about to fall. I would like to thank them for being always there for me no matter what happened.

Last but not least, I am very grateful to my family and my dearest friends, Tufan Berk Aras and Fatih Erdoğan, for their cheerful love and support. I could not have made it without them. Without them, the life would not be that beautiful…

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

σ	stress
σ_0	constant stress
ε	strain
t	time
Ψ	creep function
H	viscosity
I_0	instantaneous compliance
I(t)	compliance
Φ	decay of stress
E_∞	long term (final) elastic modulus
E	elastic modulus of spring
η	viscosity of dashpot
L_0	original(initial) length
ΔL	change of length
ε_S	strain in a spring
σ_S	stress in a spring
ε_D	strain in a dashpot
σ_D	stress in a dashpot
τ	relaxation time
A	slope of the ramp
u(t)	Heaviside unit step function
$\delta(t)$	unit-impulse function (dirac delta function)
τ_σ	the relaxation time for constant stress

$\boldsymbol{F}$	deformation gradient tensor
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\boldsymbol{\varepsilon}$	strain tensor
$\boldsymbol{T}$	first Piola-Kirchhoff stress tensor
$\boldsymbol{S}$	second Piola-Kirchhoff stress tensor
$\boldsymbol{E}$	Lagrangian Green's strain tensor
C_{10}, C_{01}	coefficients of Mooney-Rivlin strain-energy function
I_1, I_2, I_3	principle invariants
C_{ij} , μ_{i}	material stiffness constants corresponding to Young's modulus in linear material
λ	principle stretch ratio
L_f	final length
A_0	initial area
$\mathrm{G}(t)$	reduced relaxation function
$\mathrm{T}^{\mathrm{e}}(\lambda)$	elastic response instantaneously generated in the material when a step function of stretching λ is imposed on the material
$\mathrm{E}_{\infty}(\varepsilon)$	strain-dependent relaxation function of the nonlinear spring
$\mathrm{K}_{\infty}(\varepsilon)$	strain-dependent stiffness function of the nonlinear spring
K	stiffness coefficient of the linear spring
b	viscosity coefficient of the linear dashpot
F^t	external force on the element at the current time step
ε^t	unknown strain value of the element at the current time step
F^{t-1}	known or calculated force value in the previous time step
ε^{t-1}	known or calculated strain value in the previous time step
$\mathbf{x_p}$	position vector of a particle p
$\mathbf{x_n}$	position vector of the neighboring particle n

L	length of the Maxwell Solid between particles
ε^t	current strain of the element in the current time step
L^t	length of the element in the current time step
L^0	rest length of the element
$\boldsymbol{F}^{t-1}$	force vector in the previous time step
$\boldsymbol{F}^t$	external force vector on the element
$\boldsymbol{x}$	unknown position vector
$\mathrm{K}_{\infty}(\varepsilon)_{\mathrm{ELEMENT}}$	element stiffness function
$\mathrm{K}_{\infty}(\varepsilon)_{\mathrm{OPTIMUM}}$	optimum element stiffness function
$\mathrm{F}_{\mathrm{SDN}}$	force response of the sdn mesh model
$\mathrm{F}_{\mathrm{LUMPED}}$	force response of the lumped model
$\mathrm{F}_{\mathrm{REFERENCE}}$	reference force response of the lumped model
F_{PL}	peak force of the lumped model
$\mathrm{F}_{\mathrm{SSL}}$	steady-state force of the lumped model
$\mathrm{T}_{\mathrm{SSL}}$	steady-state time of the lumped model
F_{PM}	peak force of the mesh model
$\mathrm{F}_{\mathrm{SSM}}$	steady-state force of the mesh model
$\mathrm{T}_{\mathrm{SSM}}$	steady-state time of the mesh model
F_{PE}	relative error between peak forces of the lumped and mesh models
$\mathrm{F}_{\mathrm{SSE}}$	relative error between steady-state forces of the lumped and mesh models
$\mathrm{T}_{\mathrm{SSE}}$	relative error between steady-state times of the lumped and mesh models

Chapter 1

INTRODUCTION

Simulation-based training using Virtual Reality (VR) techniques is a promising alternative to traditional training in minimally invasive surgery. Surgical simulators let the trainee touch, feel, and manipulate virtual tissues and organs through the same surgical tool handles used in actual minimally invasive surgery while viewing images of tool-tissue interactions on a monitor as in real laparoscopic procedures. Developing realistic organ-force models for simulating soft-tissue behavior is an integral part of a surgical simulator.

The current physically-based approaches for developing organ-force models in surgery simulation literature can be classified as *mesh-based* and *mesh-free* methods. Mesh-based methods consider the deformable object as a continuum and are generally more accurate than the mesh-free techniques. One of the most widely used mesh-based methods in surgery simulation is the Finite Element Method (FEM). One advantage of FEM is that only a few material parameters are required to describe the response of a physical system and it is straightforward to integrate the experimentally determined material parameters into the system. On the other hand, FEM has a heavy computational load and the computational complexity usually increases quadratically with the quality of the underlying mesh. Moreover, if a procedure like cutting is to be simulated where the topology of the object is modified, element mass and stiffness matrices must be recalculated and reassembled during the simulation, which is computationally intensive. Mesh-free methods, on the other hand, utilize cloud of points only for deformation and force computations, and do not make any assumptions about the underlying geometry. The particle system approach is a widely used mesh-free method in surgical simulation. This technique provides a better solution to the topological changes encountered in simulation of surgical cutting and tearing. In addition, they are computationally less expensive and easier to implement than the mesh-based methods. However, the material coefficients of each individual mesh element should be calculated and fine-tuned to describe the response of a physical system. Therefore, the integration of realistic tissue properties into particle models is not trivial.

This thesis presents an end-to-end solution to realistic particle-based simulation of nonlinear viscoelastic tissue behavior based on the experimental data collected by a robotic indenter. First, the strain-dependent nonlinear elastic response and time-dependent viscoelastic response of a tissue-like silicon phantom is measured via static loading and stress relaxation experiments performed by a robotic indenter. The collected experimental data is used to construct a lumped model of the tissue phantom represented by a nonlinear viscoelastic Maxwell Solid. Then, a 3-dimensional particle-based network is developed to mimic the behavior of the lumped Maxwell model. The material coefficients of the individual Maxwell elements connecting the particles are successfully estimated through a set of novel optimization algorithms.

The following chapter introduces the concepts of virtual reality-based surgical simulation for minimally invasive surgery. It presents the development steps of a surgical simulator and provides a detailed literature review for each step. The chapter is a modified version of our survey paper [1]. Chapter 3 introduces the fundamental concepts of solid mechanics. An introduction to viscoelasticity and hyperelasticity is given, and mathematical models for linear and nonlinear viscoelasticity are presented. Chapter 4 is related to the measurement and characterization of nonlinear viscoelastic material properties of a cylindrical tissue phantom. It first gives the design details of a robotic indenter that is originally developed for measuring soft tissue material properties of pig liver during a minimally invasive surgery [2, 5]. Then, the chapter presents the results of material characterization experiments performed on a tissue-like silicon phantom using the indenter. For characterization, first, a lumped model of the silicon sample is constructed using Maxwell Solid with N=1 and N=2 and then the nonlinear and viscoelastic material coefficients of the model are determined using curve fitting. In Chapter 5, a particle-based spring-damper network (SDN) composed of nonlinear viscoelastic Maxwell Solids with N=1 is developed such that its behavior mimics the behavior of the lumped model. For this purpose, novel optimization algorithms are developed to estimate the material properties of the individual elements in the mesh. In Chapter 6, the thesis is concluded with a summary of the performed work and future research directions.

Chapter 2

VIRTUAL REALITY BASED SIMULATORS
for
TRAINING in MINIMALLY INVASIVE SURGERY

Throughout medical history, the training paradigm for surgeons has not changed substantially. Traditionally, surgical training has followed the apprenticeship model: Novice surgeons receive their training over time in small groups of peers and superiors in the course of patient care. The operating room (OR) and the patient comprise the most common, the most readily available, and often the only setting where hands-on training takes place. Novice surgeons acquire skills by observing experienced surgeons in action and then progressively performing additional surgical procedures under varying degrees of supervision as their training advances and skill levels increase. This so-called "see one, do one, teach one" paradigm has proved reasonably effective for more than 2,500 years.

Recently, however, experts, physicians, and the public are examining this training model and questioning its efficiency. According to "To Err is Human," a 1999 report from the Institute of Medicine of the National Academy of Sciences, more people die from medical mistakes each year than from highway accidents, breast cancer, or AIDS. In addition to this devastating human cost, the financial burden is significant. Among the main reasons cited for this situation are the inexperience of beginners, as well as the inexperience of experts with new techniques and rare medical situations. One of the major shortfalls identified in the report is medical education and training.

In particular, minimally invasive surgery (MIS) is a revolutionary surgical technique that poses an immediate need for improved training methods. Physicians have used MIS in various procedures since the early 1960s. This technology involves a small video camera and a few customized surgical instruments [6]. The surgeon inserts the camera and instruments into the body through small skin incisions or natural orifices to explore internal cavities without making large openings. For patients, MIS's major advantages over conventional surgery are a shorter hospital stay, a quicker return to activities, and less pain and scarring. Some common MIS procedures are laparoscopic cholecystectomy (gallbladder removal), appendectomy, and hernia repair. Using minimally invasive techniques is a trend in other procedures as well. We can predict that as instruments get smaller and thus easier for surgeons to handle, new minimally invasive techniques will develop.

In spite of the advantages of MIS over traditional surgery, surgeons are still handicapped by the current technology's limitations, which pose four problems in the OR:

- Visualization of internal organs achieved with a wide-angle camera is monoscopic and limited by the camera's field of view.
- Hand-eye coordination is difficult because surgeons must move the tool around a pivot point, thus inverting the direction of movement inside and outside the body. Moreover, the location of the displayed image is not the actual manipulation site.
- Surgeons receive limited haptic (tactile sensing and force feedback) cues because they must interact with internal organs by means of surgical instruments attached to long, thin tubes.
- The instruments rotate about a fixed entrance point, making it impossible for the surgeon to perform direct translational movements while interacting with organs.

Although the importance of MIS training is widely acknowledged, there is no consensus on the most effective training method. OR time is an expensive and limited resource to use for training surgeons. Many leading academic institutions in the United States and Europe have established training centers with facilities for practicing surgical techniques on both inanimate and animate models. Box trainers, for instance, are inanimate models equipped with real surgical instruments, endoscopic cameras, and plastic tissue models. They provide the trainee with an environment similar to actual surgery settings. However, simulated surgical procedures are usually poor imitations of actual ones. It is not easy to customize these training systems to the trainee's needs. Moreover, it is not easy to measure the trainee's performance with these systems.

Currently, the most realistic training model available is animals. This model is dynamic and approaches real operative conditions. Animal tissues, although not always of the same consistency as human tissues, respond similarly to applied forces. Using animals for training, however, is expensive and controversial. It requires expensive, dedicated facilities, including care and housing of the animals. Only a few trainees (often only one) can practice on the same animal and for only a limited number of times. (The expensive training session generally ends with euthanizing the animal.) Additionally, animal anatomies are different from human anatomies, and ethical issues surround the use of animals for training. Finally, with animal models, quantitative measurement of a trainee's performance is not straightforward, and evaluation (performed by the instructor) is often subjective.

With either inanimate or animate models, conventional MIS training methodologies suffer from the same main drawbacks: the need for an instructor or supervisor, nonstandard feedback methods, and subjective performance evaluation methods. Hence, new training approaches and devices to reduce the risks and constraints of surgical procedures are necessary. To meet this need, VR-based surgical simulators that give the surgeon visual and haptic cues promise to be powerful aids for training medical personnel and monitoring their performance [6]. Computer-based simulation can revolutionize medical education and augment training by quantifying performance and progress, standardizing training regimens independent of patient population, and exposing trainees to unusual cases [7, 8, 9]. Integrating VR-based simulators in medical training would result in better-trained physicians, reducing the likelihood of error and improving patient outcome. Figure 2.1 shows a typical VR-based surgical simulator.

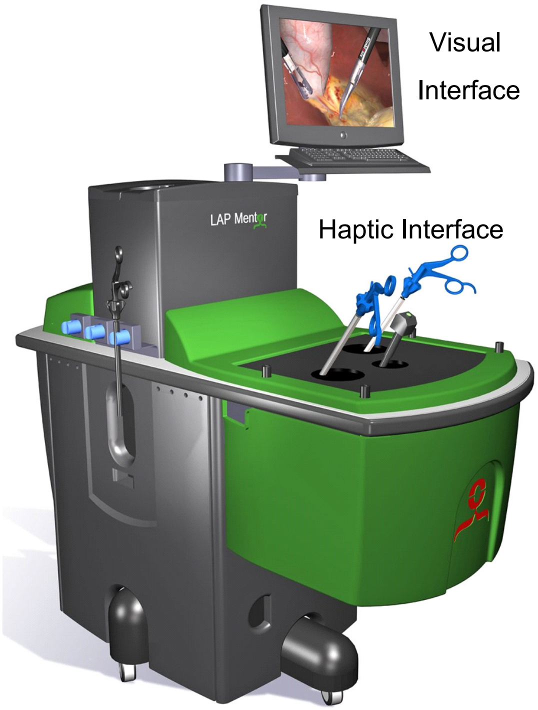


Figure 2. 1: Components of a typical minimally invasive surgery simulator (the Simbionix LAP Mentor laparoscopic surgery simulator) include a visual display and surgical instruments fitted with haptic devices for force feedback [1]

Developing a VR-based MIS simulator requires expertise in systems engineering, materials engineering, robotics engineering, computer science, biomedical engineering, and medicine. As Figure 2.2 shows, simulator development involves six steps. First, the developers use segmentation and reconstruction techniques of computer vision and computer graphics to generate 3D anatomical models of organs from medical images. Second, they measure and characterize the material properties of soft tissues and integrate these properties in organ-force models. Next, they develop collision detection and response techniques to simulate the real-time interactions of simulated surgical instruments and manipulated organs. Then, they integrate the simulator's hardware and software components to form a complete system. Finally, they validate the system and measure training transfer through user studies. The following sections explain each step in detail.

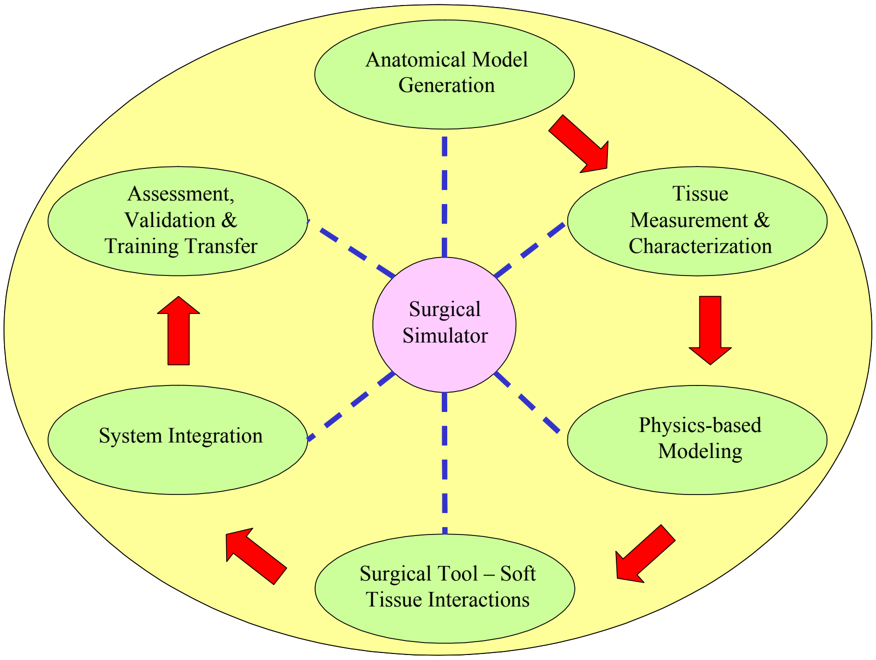

Anatomical Model Generation
Tissue Measurement & Characterization
Physics-based Modeling
Surgical Tool – Soft Tissue Interactions
System Integration
Assessment, Validation & Training Transfer
Surgical Simulator

Figure 2. 2: Simulator development steps for minimally invasive surgery [1]

2.1 Anatomical Model and Training-Scene Generation

Medical applications use various imaging modalities. Anatomical imaging techniques include computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound. Functional techniques include single photon emission computed tomography (Spect), positron emission tomography (PET), and functional MRI. For generating anatomical models, anatomical techniques play the key role – mainly CT and MRI, which provide sufficient resolution. Today's CT scanners, which integrate 64 detector rows, provide nearly isotropic voxels of approximately 0.4 mm. MRI devices provide a spatial resolution of about 1 mm in each direction. Besides different spatial resolutions, the main difference between CT and MRI is their ability to distinguish different tissues types. CT makes it easy to see bone structures, whereas MRI provides superior soft tissue contrast. In recent years, the use of CT for virtual colonoscopy and bronchoscopy has gained importance. These techniques supplement or even replace endoscopic procedures by employing a patient model derived from CT data and displayed with the help of volume-rendering techniques without any preprocessing. Ultrasound, although widely available and inexpensive, suffers from a lower imaging quality than CT and MRI and is therefore not appropriate for anatomical modeling.

The National Library of Medicine's Visible Human (Man and Woman) data sets have become standard sources of medical image scans of the human body. Because of their high resolution and good image quality, several projects have used them for highly accurate anatomical models.

To build 3D models of organs from image data, developers of surgical simulators use either surface or volumetric elements. Surface models represent the external border of the organs. Generating surface models requires extraction of the structure's outer surface, using segmentation algorithms that provide the outer contour. Figure 2.3 shows an example of the segmentation and generation of a 3D surface model.

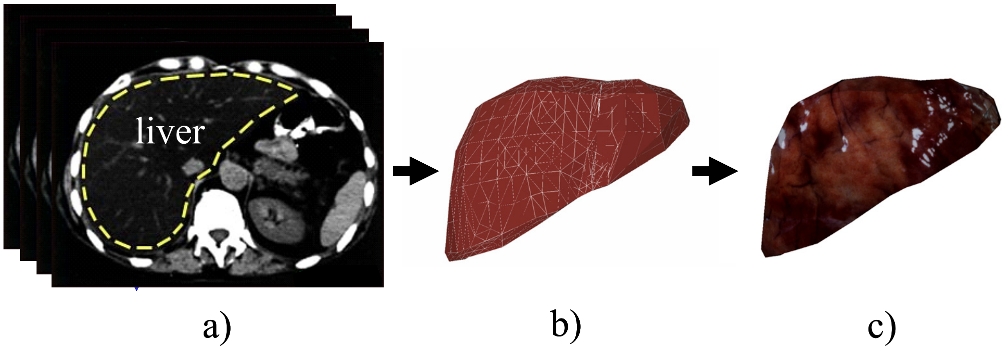


Figure 2. 3: (a) A set of 2D medical images of the abdomen is segmented via filtering techniques for identifying different tissue regions, lesions, and pathologies. (b) Segmented contours in each image are combined to create a 3D surface model of the organ—the liver in this example. (c) Texture mapping over the surface gives the model a more realistic appearance [1]

The segmentation approaches often used for anatomical modeling are simple classification schemes such as thresholding and region growing. These techniques extract isocontours that serve as input to the marching cubes algorithm, which creates a polygonal representation of the structure's surface. Another way to extract organ surfaces is to use active contour models, also known as snakes. This technique obtains a contour by adjusting splines that fit the structure's outer surface, using a physical description of the image data's external and internal forces. Applying the contour found in one 2D slice to the next neighboring slice starts a contour that is gradually refined. This process, called boundary tracking, delivers the organ's 3D contour and thus its surface representation.

An advantage of generating surface models from CT and MRI scans is the reduction in data size. . Also, we can display the triangulated meshes created by this method using the hardware acceleration available from modern graphics boards.

An alternative to surface modeling and surface data visualization is direct volume rendering. Ray casting, a classical volume-rendering technique, provides high-resolution visualization but is rather slow.

Several commercial and free software packages are available for 3D reconstruction from medical images, but most are semiautomated and often require labor- and time-intensive segmentation. Amira[1], Analyze[2], IDL[3], Image-Pro[4], and MEDx[5] are commercial packages for medical and other scientific image data visualization and manipulation, anatomical structure extraction, and surface and tetrahedral model generation. The Visualization Toolkit[6] is a free, open-source image data visualization package with contour extraction and mesh generation algorithms. Another free volume visualization system is VolVis[7]. Mesh generation and manipulation packages developed at academic institutions include TetGen[8] and SUMAA3d[9].

For a realistic visualization of reconstructed organ models, system developers map textures to the models' surfaces. This process involves generating realistic textures and obtaining appropriate texture coordinates. The most basic method of creating organ textures is direct texture painting, which usually requires a medical illustrator to manually draw the texture with appropriate tools. Another way to obtain textures for medical images is to map real volumetric data to surfaces. K.D. Reinig et al. [10] use the Visible Human

[1] http://www.amiravis.com
[2] http://www.mayo.edu/bir/Software/Analyze/Analyze.html
[3] http://www.ittvis.com/idl/
[4] http://www.mediacy.com/
[5] http://medx.sensor.com/products/medx/index.html
[6] http://www.vtk.org
[7] http://www.cs.sunysb.edu/~vislab/volvis_home.html
[8] http://tetgen.berlios.de/
[9] http://www-unix.mcs.anl.gov/sumaa3d

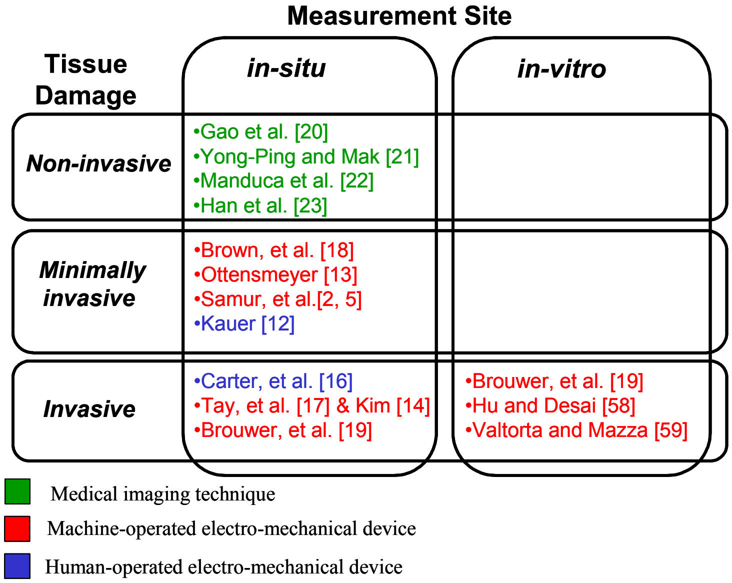
data set to obtain organ textures. Recently, Paget, Harders, and Szekely [11] introduced a fully automatic framework for generating variable textures. The first step in their approach is acquiring in vivo images to form a database. Next, a texture synthesis step creates tileable variable textures from the in vivo images. The final step is mapping the texture to the 3D mesh geometry.

2.2 Soft Tissue Measurement and Characterization

A core component of a VR-based surgical simulation and training system is realistic organ-force models. Realistic organ-force models are virtual representations of soft tissues that display accurate displacement and force response. To develop these models, we must measure and characterize the material properties of organs in living condition and in their native locations. Models with incorrect material properties will result in adverse training effects. Measuring the material properties of soft organ tissues is highly challenging. Soft tissues exhibit complex, nonlinear, anisotropic, nonhomogeneous behavior. Moreover, the tissues are layered, and each layer consists of different materials in varying combinations. Because of this nonhomogeneity, soft tissues have both coordinate- and direction-dependent properties. Time- and rate-dependent behavior caused by viscoelasticity is also common.

Various methods of measuring material properties of organ tissues have appeared in the literature. We categorize these methods in terms of the measurement site and the degree of tissue damage that occurs during measurement (Table 2.1). The two types of measurement sites are ex vivo and in vivo. In the past, most tissue experiments were ex vivo studies [12, 13, 14]. For ex vivo measurements, the tissue's biological functioning has ceased; in other words, the tissue to be measured is dead. Ex vivo measurements can take place within the body (in situ) or outside the body (in vitro). For in vitro measurements, researchers use standard materials-testing methods (tension or compression tests) under well-defined boundary conditions. Typically, they transfer tissue samples in a chemical solution to a laboratory for measurements. Because they carefully decide on sample geometry and experimental conditions in advance, they can easily obtain stress and strain values from the measurement data. However, dead organ and muscle tissues typically stiffen with time, leading to changes in mechanical properties, so the results of in vitro measurements can be misleading. Ottensmeyer et al. [15] show that measurement environment of in-vitro studies does not represent the actual tissue conditions. Therefore, recent research has focused on in vivo, in situ measurement of soft tissues' mechanical properties [2, 5, 12, 13, 16, 17, 18].

Soft-tissue measurement methods incur three levels of tissue damage: invasive, noninvasive, and minimally invasive. In invasive methods, measurement instruments enter the body through a puncture or an incision. Because large openings allow easy insertion of test apparatus into the body, scientists have performed many measurements invasively [16, 17, 19]. However, the experimental devices and procedures used for invasive measurements typically don't match the actual surgical devices and procedures used during MIS. In addition, the invasive approach is unacceptable for conducting human experiments.



	Measurement Site	
Tissue Damage	*in-situ*	*in-vitro*
Non-invasive	•Gao et al. [20] •Yong-Ping and Mak [21] •Manduca et al. [22] •Han et al. [23]	
Minimally invasive	•Brown, et al. [18] •Ottensmeyer [13] •Samur, et al.[2, 5] •Kauer [12]	
Invasive	•Carter, et al. [16] •Tay, et al. [17] & Kim [14] •Brouwer, et al. [19]	•Brouwer, et al. [19] •Hu and Desai [58] •Valtorta and Mazza [59]

🟩 Medical imaging technique

🟥 Machine-operated electro-mechanical device

🟦 Human-operated electro-mechanical device

Table 2. 1: Classification of tissue measurement techniques

In contrast, noninvasive tissue measurements require no incisions. Noninvasive approaches include CT, MRI, and ultrasound [20, 21, 22, 23]. Most of these approaches can measure only linear material properties [13], but soft organ tissues exhibit nonlinear material characteristics as well.

Minimally invasive methods require small incisions, causing much less tissue damage than the invasive method does. A few research groups have recently conducted minimally invasive animal and human experiments to characterize nonlinear and time-dependent material properties of soft tissues [2, 5, 12, 13, 18]. A challenge of this approach is characterizing the measured properties. Determining unknown material properties from the measured system response requires formulating an inverse solution. For this purpose, scientists typically construct a finite-element model of the soft tissue and use it with an optimization method to iteratively match the experimental data to the numerical solution [12, 14].

2.3 Physics-Based Modeling

Developing realistic organ-force models for simulating soft-tissue behavior requires a system that reflects stable forces to the user, displays realistic smooth deformations in real time, and handles various boundary conditions and constraints [6]. The material properties and structure of organ tissues mentioned earlier make developing real-time, realistic organ-force models challenging. In addition, surgical-tool and soft-tissue interactions cause dynamic effects and contact between organs, which are difficult to simulate in real time. Furthermore, simulating surgical operations such as cutting and coagulation requires updating the organ's geometric database frequently and can cause force singularities in the physics-based model at the boundaries.

We classify current physics-based approaches for developing organ-force models as mesh free and mesh based. Mesh-free methods use point clouds (vertices) only for deformation and force computations and make no assumptions about the underlying geometry. Most mesh-based methods consider the deformable object a continuum and are generally more accurate than mesh-free techniques. However, mesh-free techniques provide a better solution to topological changes encountered in simulating surgical cutting and tearing. In addition, they are computationally less expensive and easier to implement than mesh-based methods.

2.3.1 Mesh-based Methods

One of the most widely used mesh-based methods is the finite-element method (FEM) [3, 4, 24, 25, 26, 27]. FEM solves the deformation problem by considering the organ a continuous body that is trying to minimize its potential energy under the influence of external forces. To implement this method, we divide the geometric model of an organ into surface or volumetric elements, formulate each element's properties, and combine the elements to compute the organ's deformation states under the forces applied by the surgical instruments [6]. A major advantage of FEM is that it uses continuum mechanics and has a solid mathematical foundation. On the basis of the partial differential equations and the constitutive relation used, FEM can accurately approximate static and dynamic deformations of an object with linear and nonlinear material properties [3]. Another advantage is that FEM requires only a few material parameters to describe a physical system's response.

However, FEM also has some drawbacks. It has a heavy computational load, and its computational complexity usually increases quadratically with the underlying mesh's quality. Moreover, while simulating a procedure such as cutting, in which the object's topology is modified, we must recalculate and reassemble element mass and stiffness matrices, which is computationally intensive. Precomputation and condensation have been suggested as remedies to these problems [25].

Another mesh-based method based on continuum mechanics is the boundary element method (BEM). BEM discretizes an object's surface or boundary into elements and patches and relies on surface integral equations to calculate displacements at the boundary [28]. On the assumption of linear elasticity, BEM computes small deformations accurately. Extending this approach to large deformation analysis is not straightforward. Another drawback is that direct solution of BEM is computationally too expensive to execute in real time. Yet, precomputation and superposition make it possible to execute a linear deformable model at haptic update rates [28] BEM can model changes in topology resulting from procedures such as cutting by using iterative solvers that update precomputed data to approximate the modified topology.

The long-element method (LEM) is another mesh-based approach. It is based on Hooke's law, Pascal's principle, and volume conservation as the boundary condition [29]

LEM discretizes the object into a set of two-dimensional long elements filled with an incompressible fluid. During deformation computation, these elements reach equilibrium under the effect of bulk variables including pressure, density, volume, and stress. One advantage of LEM is that the parameters such as pressure, density, and volume are easy to identify. The elements filled with an incompressible fluid can represent nonhomogeneous material properties. Because the method intrinsically preserves volume, it supports topological changes such as cutting. On the other hand, LEM produces accurate results for small deformations only. It yields inconsistent results for large deformations. The element deformations must be reevaluated when the object undergoes large deformations, a bottleneck for real-time performance.

The tensor-mass model (TMM) is also a mesh-based approach. Its methodology lies between the continuum mechanics and particle-based approaches. Cotin, Delingette, and Ayache [30] developed TMM as a continuum model based on linear elasticity. The model discretizes the object into tetrahedrons, and the tensors are stored at the edges of the tetrahedrons. Like particle-based approaches, the object's mass is stored in the nodes of the tetrahedrons as lumped mass points. However, unlike particle-based approaches, TMM computes deformation and force through energy-based continuum mechanics, and the computations are independent of mesh topology. One of TMM's main advantages is that the model can handle topological modifications; hence, we can use it to simulate tissue cutting and tearing. In addition, TMM's time complexity is linear and lower than that of the standard FEM approach. The initially proposed TMM approach could simulate small deformations only. Later, Picinbono, Delingette, and Ayache [31] extended TMM to simulate large deformations as well.

2.3.2 Mesh-free Methods

The particle system approach, also called the mass-spring model (MSM), is a widely used mesh-free method in surgical simulation [24, 32, 36, 60, 61, 62, 63, 64]. This approach models the object as point masses connected to each other with springs and dampers [6]. Each point mass is represented by its own position, velocity, and acceleration and moves under the influence of inertial and damping forces and the forces applied by the surgical instrument. This technique is relatively easy to implement because the motion equations need not be constructed explicitly. Hence, the technique's computational complexity allows real-time simulation. However, the integration of realistic tissue properties into particle models is not trivial. In addition, the resulting physical behavior depends on the point masses' connectivity. The construction of an optimal spring network in 3D is a complicated process, and MSM can become oscillatory or unstable under certain conditions.

The point-associated finite-field (PAFF) approach, also called the finite-spheres method, is a newer meshless FEM approach applied to surgical simulation [33]. This method, like TMM, resides between the continuum mechanics and particle-based approaches. Like MSM, it is a point-based approach, using only the nodes of a 3D object for the displacement and force calculations. PAFF approximates the displacement field by using nonzero functions over small spherical neighborhoods of nodes. Like FEM, this technique uses a Galerkin formulation to generate the discretized versions of the partial differential equations governing the deformable medium's behavior. PAFF supports simulation of large deformations as well as topology modifications such as cutting. PAFF can also be used to simulate other procedures involving particles, such as smoke generation during cauterization. Although the technique's brute-force implementation is computationally intensive, users can generate localized solutions in real time [33].

2.4 Simulating Tool-Tissue Interactions

Simulating interactions between surgical tools and soft tissues involves graphical rendering of computer-generated models of surgical instruments, detecting collisions between instruments and deformable organ models, and haptic rendering of the collision response in the procedure to be simulated [6].

We classify MIS tools on the basis of their functionality:

- long, thin, straight tools for palpation, puncture, and injection (for example, palpation probes and puncture and injection needles) and
- articulated tools for grasping, pulling, clamping, cutting, and coagulating (biopsy and punch forceps, grasping forceps, hook scissors, and coagulation hooks).

For realistic visual display during simulations, we render 3D graphical models of surgical tools in exact dimensions and shape using several polygons. But we typically assume that the models consist of a set of geometric primitives such as points or connected line segments for fast detection of collisions between instruments and organs in real time. Collision detection algorithms developed in computer graphics cannot be used directly in rendering force interactions between instruments and organs. Nevertheless, to achieve real-time update rates, haptic rendering algorithms can take advantage of computer graphics rendering techniques:

- space partitioning (partitioning the space that encloses an object into smaller subspaces for faster detection of the first contact),
- local search (searching only the neighboring primitives for possible new contacts), and

- hierarchical data structures (constructing hierarchical links between primitives constituting the object for faster access to the contacted primitive).

In point-based haptic interactions of surgical instruments with organs, only the instrument's end point interacts with virtual organs. Each time the user moves the surgical instrument fitted with haptic devices in physical space, the collision detection algorithm checks whether the end point is inside the virtual organ. This approach provides users with similar force feedback as they would feel when exploring organs in real surgery settings with the tip of an instrument only. However, actual MIS instruments have long, slender bodies, so point-based rendering methods are not sufficient to render realistic tool-tissue interactions.

In ray-based haptic interaction models, the probe is a finite line segment whose orientation the detection algorithm takes into account while checking for collisions between the line segment and the objects. This technique has several advantages over point-based rendering. In addition to displaying forces, users can feel torques if they are using an appropriate haptic device, which is not possible with point-based approaches. For example, they can feel the coupling moments generated by contact forces at the instrument tip and the forces at a trocar's pivot point. Second, users can detect side collisions between the simulated tool and 3D organ models. Third, users can render multiple tissue layers by virtually extending the ray representing the simulated surgical probe to detect collisions with an organ's internal layers. Finally, they can touch and feel multiple objects simultaneously.

Once the algorithm detects contact between an instrument and tissue, the tool-tissue interaction problem centers on collision response. This involves a realistic graphical and haptic display of tissue behavior according to instrument type and the surgical task the user chooses to perform. Tissue deformation is the most generic collision response. Simulation of basic surgery skills such as palpating, grasping, stretching, translocating, and clip applying mainly involves tissue deformation. Simulation of surgical cutting such as transsection, dissection, and coagulation fall into a different category, in which tool-tissue interactions modify the geometry and the underlying model. Figure 2.4 shows several of these simulated interactions.

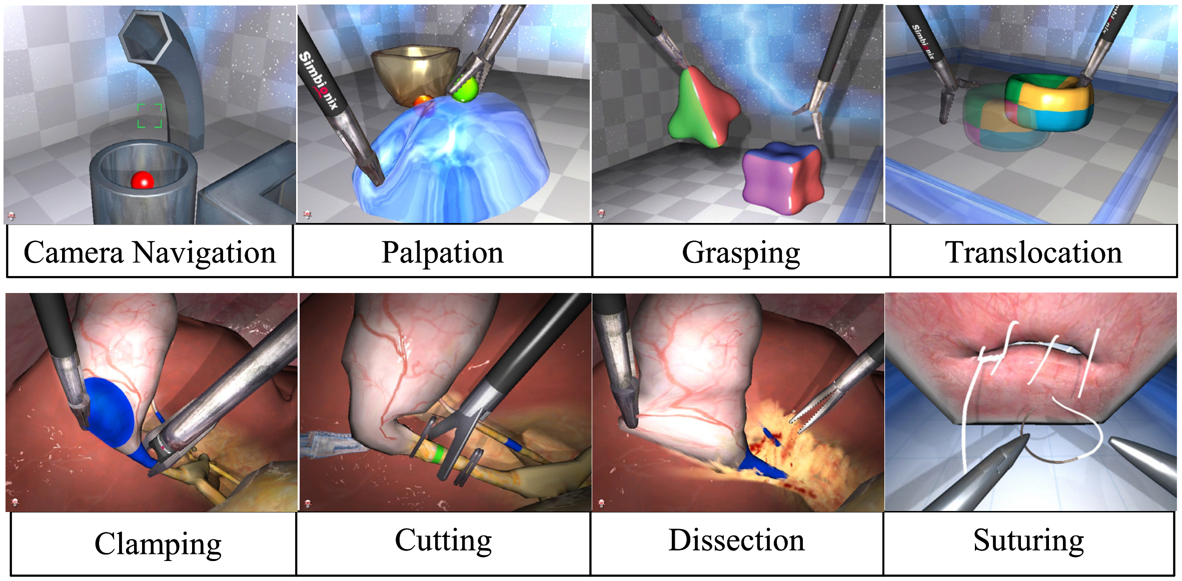

Camera Navigation
Palpation
Grasping
Translocation
Clamping
Cutting
Dissection
Suturing

Figure 2. 4: Typical simulations of MIS tasks: basic (first row) and advanced skills (second row) [1]

Realistic graphical and haptic simulation of cutting is a requirement in any surgical simulator. Research in this area has focused mainly on the graphical display of cut and tissue separation, but some recent studies report the development of mechanistic models for displaying forces during cutting. Cutting approaches for graphical simulation include straightforward element deletion, mesh subdivision, and topology adaptation.

Cotin, Delingette, and Ayache [30] have applied straightforward deletion of mesh entities to remove the elements contacted by a cutting tool. Unfortunately, this method leads to visual artifacts because it cannot approximate the cutting path accurately.

Achieving acceptable visual quality would require very high resolution meshes. Moreover, the method violates the physical principle of mass conservation.

Mesh subdivision methods have produced better visual representations of incisions. Bielser et al. [34] discuss the use of a state machine to keep track of incisions in tetrahedral meshes. All the described mesh subdivision approaches considerably increase the element count. Moreover, introducing new mesh elements often necessitates extensive model recalculations – for instance, in using implicit FEM. Another negative factor is reduction in element size. Researchers have also reported deformation stability problems in the simulation of tissue cutting. This required a significant reduction of the time step, thus rendering real-time simulation intractable. Finally, Molino, Bao, and Fedkiw [35] report on work in which they decoupled the simulation and visualization domains. Their virtual node algorithm copies nodes and elements so that no new elements are created. Elements are decomposed only in the visualization domain. A tetrahedron cannot be cut more than three times, however, and the surface resolution depends on the resolution of the underlying tetrahedral mesh.

Topology adaptation approaches can ameliorate some of the problems [36, 37]. Their central idea is to approximate a cutting path with existing vertices, edges, and polygons of the geometric model. This enables mesh incisions without large increases in element count and also without reductions in element size. Unfortunately, problems arise from degenerated elements, which can appear with unconditional node displacement in the mesh. Also, the initial mesh resolution limits incision approximation quality. In addition, topology adaptation approaches require an update of the undeformed mechanical model's mesh parameters, which can be difficult if the displacements are large. Steinemann et al. [38] recently proposed a hybrid cutting approach for tetrahedral meshes. It combines topological update via subdivision with adjustments of the existing topology. In addition, after the initial cut, local mesh regularization improves element quality. Moreover, the mechanical and the visual models are decoupled, allowing different resolutions for the underlying mesh representations. This method can closely approximate an arbitrary, user-defined cut surface, while avoiding the creation of small or badly shaped elements, thus preventing stability problems in the subsequent deformation computation.

In addition to graphical rendering, cutting simulation research involves the development of a mechanistic model of the cut for realistic force feedback to the user. Most of these models are based on experimental data collected from tissue samples, Chanthasopeephan, Desai, and Lau [39] developed an instrumented hardware-software system for characterization of soft tissue's mechanical response during cutting. They performed ex vivo cutting experiments with porcine liver using various cutting speeds and angles. They showed that although force displacement behavior for different combinations of cutting speed and angle has a characteristic pattern (loading, then a sudden puncture, then unloading), deformation resistance changes with the instrument's speed and angle. Mahvash and Hayward [40] developed a cutting model based on a fracture mechanics approach. They model cutting in three steps: deformation, tearing, and cutting. Their model assumes that energy is recoverable during deformation, it is zero during tearing, and it is not recoverable during fracture generation.

In tissue cutting during MIS, smoke and bleeding can occur – for example, when a coagulation hook tears apart the membrane tissue around organs. A coagulation simulation requires realistic smoke generation in real time. Kuhnapfel, Cakmak, and Maab [24] use animation techniques to simulate smoke generation and fading away of the generated smoke after coagulation. They integrate this approach in their MIS training environment Kismet. De et al. [33] simulate smoke generation using the PAFF method, which they developed for simulating deformable objects. They use the Lagrangian form of the Navier-Stokes formulation as the governing equation to simulate smoke formation. Like smoke simulation, real-time simulation of bleeding is a recent surgical simulation research area.

Basdogan, Ho, and Srinivasan [36] developed a surface flow algorithm based on Navier-Stokes equations for bleeding simulation. They generate an auxiliary mesh for the blood flow and project it to the incision area. Kuhnapfel, Cakmak, and Maab [24] developed a mass-spring model for simulation of arterial bleeding, irrigation, and suction. Zatonyi et al. [41] introduce a real-time approach based on computational fluid dynamics for simulating blood flow in the fluid-filled uterine cavity during hysteroscopy.

Advanced surgical skills such as suturing, thread handling, and knot tying show similarities to cutting in many ways. Brown et al. [32] implemented a suturing simulation environment for training in microsurgical skills. The user manipulates virtual blood vessels, sutures them together, and then ties the knot. MSM is the underlying deformation model for the vessels. Berkley et al. [27] simulate suturing and knot tying in real time on a 3D model of a hand. They use the banded-matrix approach, a fast finite-element modeling technique developed for real-time deformation simulation, as the underlying deformation model. During simulation, they visually display the highly stressed areas on the hand model to prevent the user from damaging the tissue during needle interactions.

2.5 System Integration

A VR-based surgical simulator is actually a human-computer interface consisting of a network of high-end hardware and software components. Providing a realistic training environment in which trainees act as if they are operating on an actual patient is the simulator's essential goal. Selection and design of simulator components depend on the type of training the simulator will serve.

A part-task trainer is a simulator system that is designed to train a particular surgical task. A part-task trainer's hardware components typically include a computer with a 3D graphics accelerator for visualization of virtual organs, force feedback devices to simulate haptic sensations, and auditory interfaces to guide the trainee. Force-reflecting devices and actuators fitted with surgical tool handles are embedded in a mannequin in a manner similar to standard MIS settings where surgical tools are inserted to the body through small incisions. During simulations, the user manipulates actual surgical instruments attached to the force feedback devices in the mannequin to interact with computer-generated anatomical organs. The computer monitor displays the organ manipulations (as the video monitor would do in MIS), and the haptic interfaces feed the reaction forces back to the user. Using this set-up, a trainee can learn to execute a specific procedure. The part-task trainer can monitor and record the trainee's performance during the session for further analysis.

A full-task (team) trainer is designed to train one or more trainees at the same time on a full range of surgical operations in a simulated OR. Compared with part-task trainers, full-task trainers require a larger space and enriched sensory feedback to simulate the OR environment. Sensors and mechanical actuators (such as mechanical lungs, voice output, and drug delivery systems) sensitive to the trainee's actions are placed in the mannequin and around the table to create a realistic OR environment for team training. We can envision a more sophisticated team trainer in which multiple trainees equipped with stereo glasses, head trackers, and exoskeleton haptic devices enter an immersive room such as CAVE, a projection-based VR system, which provides a large-scale virtual environment. Visual images seamlessly projected on the walls would display a 3D anatomical model, information charts, and a floating virtual patient's vital signs.

Integrating a part-task trainer's software components typically requires the construction of hierarchical data structures for storing objects' geometric and material properties, a client-server model, and multithreading and multiprocessing programming techniques to separate visual and haptic servo loops. Each sensory loop has its own requirements and demands a CPU time accordingly. Although a graphics update rate of 30

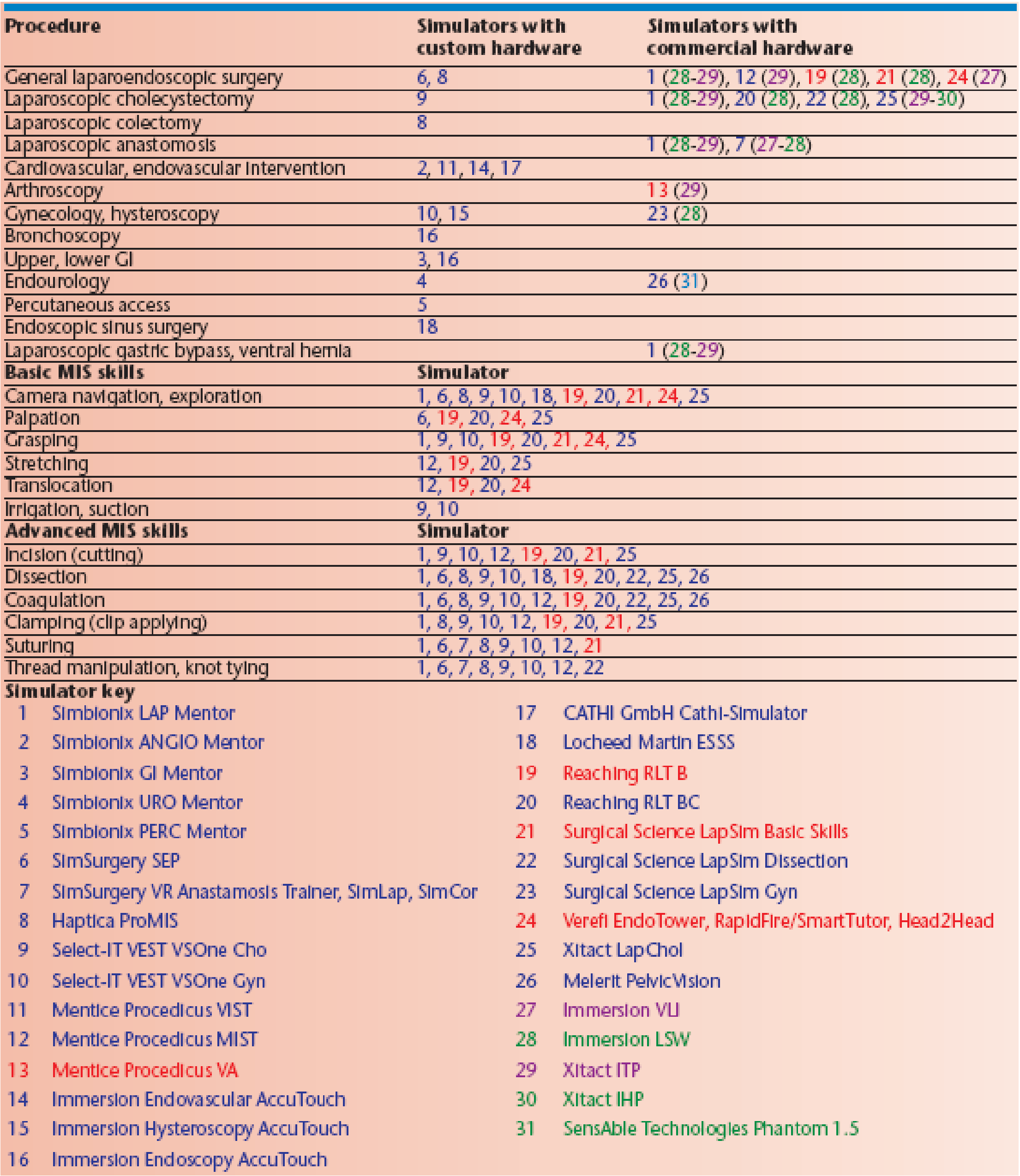
Hz is sufficient for the human sensory system to perceive a flawless display of visual images, the human sensory system is far more sensitive to the haptic update rate. For a realistic sense of touch and stable force interactions, the haptic update rate should be as high as 1 kHz. Moreover, displaying surface textures and vibrations requires even more demanding rates: 5 to 10 kHz.

In a multithreading structure, a separate thread can be assigned to each sensory modality, and a priority level for each thread can be set in advance or dynamically during simulations. Threads can share the same database, but their proper synchronization in accessing the shared database is important for achieving real-time graphical- and haptic-rendering update rates. However, there is always a trade-off between realism and real-time performance. For example, as tissue models get more sophisticated, update rates can easily drop below acceptable levels. To achieve real-time update rates, adaptive subdivision techniques can progressively increase the model's resolution.

To simulate procedures that require topological modifications (such as cutting), we can decouple visual and haptic models or implement hybrid approaches. For example, we can use more accurate but computationally more demanding deformation models such as FEM for areas where topology is preserved. At the same time, to model cutting, we can use less accurate approaches such as MSM, TMM, or PAFF, which allow easier topology changes [30, 38]. In addition to being efficient, the simulator should provide realistic visual images. Instead of standard, static texture-mapping techniques, we can use view-dependent texture-mapping techniques to simulate the complex glistening effect of the movements of the endoscopic-camera light.

Another issue in system integration is standardization of software architectures and languages that bind the various simulator components efficiently. Although many languages and formats exist for geometric representation of 3D objects, there are no standards for representation of physics-based deformable objects. Because there are different approaches to modeling the behavior of deformable organs, and material properties must be integrated uniformly in these models, the need for a generic modeling architecture is obvious. Recently, Chabanas and Promayon [42] have presented the idea of developing a standard language called Physical Model Language (PML), based on Extensive Markup Language (XML), for unified representation of continuous and discrete deformable models for surgical simulation. Cavusoglu, Goktekin, and Tendick [43] have developed the General Interactive Physical Simulation Interface (GiPSi), an open-source and open-architecture software development framework for surgical simulation. GiPSi provides a shared development environment and a standard API to ensure modularity. With GiPSi, users can generate scenarios in which they can simulate different organs with different deformation models. A related endeavor is the Simulation Open Framework Architecture (SOFA) project, a concerted activity of several groups involved in surgical simulation. It targets an extendible, open-source framework for easy exchange of algorithmic blocks between research groups.

Currently, many research institutions are developing simulators, and several medical simulation companies are offering integrated commercial systems. Leskovsky, Harders, and Szekely [44] provide an overview of existing simulator systems developed in academia. Table 2.2 lists current commercial MIS part-task simulators. Whereas some companies offer a complete system consisting of customized software and hardware modules, others develop only software solutions, often in partnership with companies that provide the supplementary hardware interface. Basic-skills simulators aim at training in fundamental skills such as navigation, hand-eye coordination, and basic tool-tissue



Procedure	Simulators with custom hardware	Simulators with commercial hardware
General laparoendoscopic surgery	6, 8	1 (28-29), 12 (29), 19 (28), 21 (28), 24 (27)
Laparoscopic cholecystectomy	9	1 (28-29), 20 (28), 22 (28), 25 (29-30)
Laparoscopic colectomy	8	
Laparoscopic anastomosis		1 (28-29), 7 (27-28)
Cardiovascular, endovascular intervention	2, 11, 14, 17	
Arthroscopy		13 (29)
Gynecology, hysteroscopy	10, 15	23 (28)
Bronchoscopy	16	
Upper, lower GI	3, 16	
Endourology	4	26 (31)
Percutaneous access	5	
Endoscopic sinus surgery	18	
Laparoscopic gastric bypass, ventral hernia		1 (28-29)
Basic MIS skills	**Simulator**	
Camera navigation, exploration	1, 6, 8, 9, 10, 18, 19, 20, 21, 24, 25	
Palpation	6, 19, 20, 24, 25	
Grasping	1, 9, 10, 19, 20, 21, 24, 25	
Stretching	12, 19, 20, 25	
Translocation	12, 19, 20, 24	
Irrigation, suction	9, 10	
Advanced MIS skills	**Simulator**	
Incision (cutting)	1, 9, 10, 12, 19, 20, 21, 25	
Dissection	1, 6, 8, 9, 10, 18, 19, 20, 22, 25, 26	
Coagulation	1, 6, 8, 9, 10, 12, 19, 20, 22, 25, 26	
Clamping (clip applying)	1, 8, 9, 10, 12, 19, 20, 21, 25	
Suturing	1, 6, 7, 8, 9, 10, 12, 21	
Thread manipulation, knot tying	1, 6, 7, 8, 9, 10, 12, 22	

Simulator key

1	Simbionix LAP Mentor	17	CATHI GmbH Cathi-Simulator
2	Simbionix ANGIO Mentor	18	Locheed Martin ESSS
3	Simbionix GI Mentor	19	Reaching RLT B
4	Simbionix URO Mentor	20	Reaching RLT BC
5	Simbionix PERC Mentor	21	Surgical Science LapSim Basic Skills
6	SimSurgery SEP	22	Surgical Science LapSim Dissection
7	SimSurgery VR Anastamosis Trainer, SimLap, SimCor	23	Surgical Science LapSim Gyn
8	Haptica ProMIS	24	Verefi EndoTower, RapidFire/SmartTutor, Head2Head
9	Select-IT VEST VSOne Cho	25	Xitact LapChol
10	Select-IT VEST VSOne Gyn	26	Melerit PelvicVision
11	Mentice Procedicus VIST	27	Immersion VLI
12	Mentice Procedicus MIST	28	Immersion LSW
13	Mentice Procedicus VA	29	Xitact ITP
14	Immersion Endovascular AccuTouch	30	Xitact IHP
15	Immersion Hysteroscopy AccuTouch	31	SensAble Technologies Phantom 1.5
16	Immersion Endoscopy AccuTouch		

Table 2. 2: Procedure and skill classification of commercial MIS part-task simulators. Blue: procedural simulator; red: basic-skills simulator; green: hardware interface with haptic feedback; purple: hardware interface without haptic feedback [1]

interactions. Procedure simulators, on the other hand, provide an environment for training in more complex surgical skills. The training scenarios provided by procedure systems vary from teaching complex tool-tissue interactions such as cutting and suturing to teaching a complete surgical procedure such as a laparoscopic cholecystectomy.

2.6 Assessment, Validation and Training Transfer

Medical education focuses on knowledge-based and skill-based training. In knowledge-based training, trainees become familiar with surgical instruments and their functionality. They learn to find anatomical landmarks, to differentiate healthy and pathological organs through visual cues such as color and texture, and to track physiological changes such as heart rate and blood pressure. Training sessions guide them through the steps of surgical procedures such as cutting, suturing, and coagulation. Skill-based training involves enhancement of the trainee's visio-spatial, perceptual, and motor skills such as hand-eye coordination including depth perception, navigation, aiming, and manipulation. Hand-eye coordination is especially difficult in MIS because the laparoscopic camera reflects 2D mirror images of hand movements and locations of anatomical landmarks [6]. For good coordination, surgeons must use cues such as the sense of touch and the reflection of camera light from organs. Using the virtual counterparts of these cues in simulators, trainees can practice as much as necessary to develop good hand-eye coordination. For example, during a laparoscopic training session, trainees learn to aim the laparoscopic forceps at a target and move the forceps in the abdominal cavity to learn the allowable range of applied movements, and manipulate organs to examine the allowable range of forces and torques applied by the forceps. In addition, an expert surgeon's movements in a procedure can be recorded in advance and played back to the trainee through haptic devices [6]. If the trainee moves out of the expert surgeon's trajectory, force feedback can return the trainee to that path. Additional guidance from visual cues and auditory feedback strengthens the learning effect.

Concerns that simulators lack validity have adversely affected the adoption of this technology in medical training. Medical boards and councils' growing interest in VR-based training has recently given rise to validation studies [8, 45]. Validation is the verification of training effectiveness. We can investigate a VR-based simulator's validity, or training effectiveness, at several levels [9].

Face validity is the level of resemblance between the simulated and real procedures. The factor contributing most to face validity is the fidelity of the organ-force models, as discussed earlier.

Content validity verifies that methods and metrics used for skill assessment are appropriate.

Construct validity examines whether the assessment methods can differentiate expert surgeons from novices. To compare novice and expert performance, we must define performance metrics. Unlike traditional medical training approaches, VR-based simulators can provide objective measurement and assessment of technical competence. In conventional methods, performance measurement and assessment depend on a supervisor's qualitative, subjective evaluation. VR-based simulators, on the other hand, use quantitative, concrete metrics [8]. During a training session, the system can record movements and applied forces and then evaluate the trainee immediately using the performance metrics. Quantitative performance measures include

- task completion time,
- operational accuracy,
- hand motion economy,
- path length,

- work done by trainee (force times displacement),
- number of tasks completed successfully,
- amount of unnecessary tissue damage, and
- excessive use of surgery material (for example, clamps during clip applying).

Stylopoulos et al. [46] developed a standard assessment methodology that uses several of these measures. This methodology merges the recorded values to quantify overall performance by a single number after the training session. In addition, VR simulators can support proficiency evaluation by generating learning curves based on multiple training sessions.

Concurrent validity is the correlation between a trainee's simulator performance and his or her OR performance. Finally, *predictive validity* is a prediction of the correlation between a trainee's present simulator performance and his or her future OR performance. The concurrent and predictive validations are related to training transfer, also called VR-to-OR proof, which refers to the success of simulator training in actual performance, or how well simulator training transfers to the real world [9].

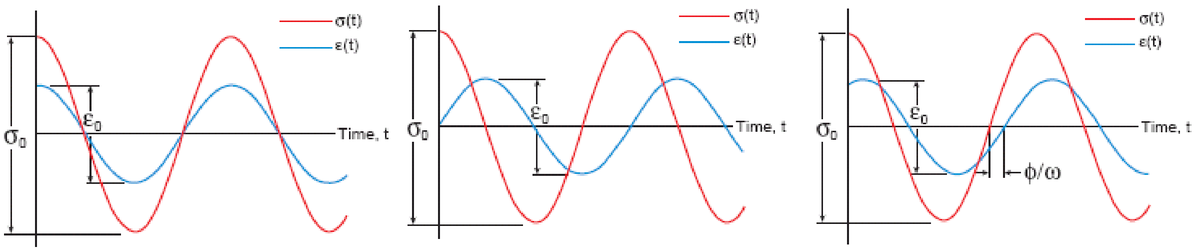
Chapter 3

INTRODUCTION to SOLID MECHANICS

Fung [47] has shown that mechanical properties of soft tissues do not meet the definition of an elastic body. Similar to some materials such as elastomers, and polymers, soft tissues can undergo large deformations, and hence exhibit highly nonlinear behavior. There are three major sources of nonlinear behavior: geometric nonlinearity, material nonlinearity, and boundary nonlinearity. The fact that the strain is a nonlinear function of the displacement is known as the geometric nonlinearity. The changes in the deformed shape are taken into account in the strain-displacement relation. In material nonlinearity, the constitutive equation (i.e. stress-strain relation) is no longer linear. Material behavior depends on current deformation state and possibly past history of the deformation which may give rise to very complex phenomena such as *path dependence*, and *hysteresis* [48]. For some materials, the curve of stress versus strain in dynamic loading is not elliptical, but has pointed ends; this behavior is a manifestation of material nonlinearity [49]. The loop is called a *hysteresis* loop which represents a lag between cause and effect. Material nonlinearity can be grouped into two classes. The first one is time-independent behavior including hyperelasticity and plasticity, and the second is time-dependent behavior such as viscoelasticity and viscoplasticity. The following sections give details of viscoelasticity and hyperelasticity, and nonlinear viscoelasticty. Finally, boundary nonlinearity arises from dependency of applied forces and displacements on deformation of structure. Contact is a typical example of nonlinear boundary condition.

3.1 Viscoelasticity

A viscoelastic material is a combination of the characteristic properties of liquids (viscous dissipative losses) and solids (storage of elastic energy) [50]. For a purely elastic material, all the energy stored in the sample during loading is returned when the load is removed. As a result, loading and response curves for elastic materials move completely in phase (Figure 3.1.a). A purely viscous material does not return the energy stored during loading (Figure 3.1.b). All the energy is lost as “pure damping” once the load is removed. These materials have only damping component and no stiffness component. Viscoelasticity is concerned with materials which exhibit both elastic and viscous behavior. Some of the energy stored in a viscoelastic system is recovered upon removal of the load, and the remainder is dissipated in the form of heat. Therefore, a phase difference occurs between loading and response curves (Figure 3.1.c) [51].



a) b) c)

Figure 3. 1: Cyclic loading and response curves for various materials: a) Elastic material, b) Viscous material, and c) Viscoelastic material [51]

The mechanical properties of viscoelastic materials are characterized by performing simple laboratory tests. Once the properties are characterized, one can integrate them into mathematical models to model and simulate the characterized viscoelastic behavior.

3.1.1 Mechanical Characterization Experiments

Since time plays an important role in the behavior of viscoelastic materials, they are also called time-dependent materials. This time-dependency is explained by the phenomena of creep under constant stress and stress relaxation under constant strain.

3.1.1.1 Creep

Creep is the deformation produced by the sudden application of a constant stress σ_0 applied at an initial moment of time $t = 0$. In general, the function of strain, $\varepsilon(t)$, consists of three components:

$$\varepsilon(t) = \varepsilon_0(t, \sigma_0) + \Psi(t, \sigma_0) + \frac{\mathrm{t}}{\eta(\sigma_0)}\sigma_0 \tag{3.1}$$

where ε_0 is the instantaneous deformation, Ψ is a function of the delayed deformation development, and η is a viscosity function [50]. If the creep experiment is being performed on a viscoelastic solid, the viscosity function becomes unlimitedly high and the last term in Equation 3.1 disappears:

$$\varepsilon(t) = \varepsilon_0(t, \sigma_0) + \Psi(t, \sigma_0) \tag{3.2}$$

Equation 3.2 can also be written in the following form

$$\mathrm{I}(t) = \frac{\eta(t)}{\sigma} = \mathrm{I}_0(\sigma) + \psi(t, \sigma) \tag{3.3}$$

where I(t) is compliance, I_0 is the instantaneous compliance and ψ is the creep function. A viscoelastic solid is called *linear viscoelastic* if I_0 and ψ do not depend on stress [50]. Otherwise, it is called *hyperviscoelastic* (i.e. nonlinear viscoelastic). Figure 3.2 presents the typical deformation behavior of a viscoelastic solid in a creep experiment.

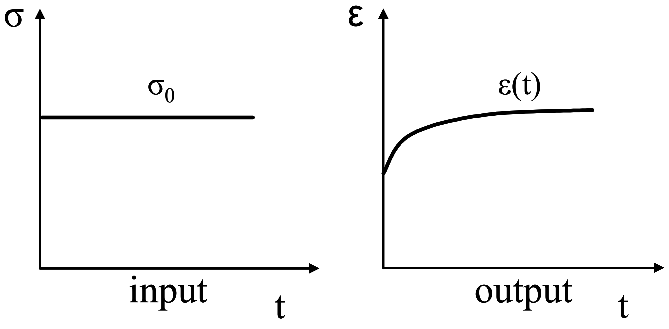


Figure 3. 2: Creep under constant stress

3.1.1.2 Stress Relaxation

Changing the roles of σ and ε in the creep experiment, the stress relaxation experiment can be obtained in which the decaying stress response is monitored, which is produced by the sudden application of a constant strain ε_0 applied at an initial moment of time $t = 0$ (step input). The function of stress, σ(*t*), consists of two components [50]:

$$\sigma(t) = \Phi(t, \varepsilon_0) + E_\infty(\varepsilon_0)\varepsilon_0 \tag{3.4}$$

where Φ is the function representing the decay of stress and E_∞ is the long term (final) elastic modulus. In case of a viscoelastic liquid, E_∞ is zero. If the experiment is being performed on a viscoelastic solid, then E_∞ has a nonzero positive value. Dividing both sides of Equation 3.4 by ε_0 gives the elastic relaxation function:

$$E(t) = \frac{\sigma(t)}{\varepsilon_0} = \phi(t, \varepsilon) + E_\infty(\varepsilon) \tag{3.5}$$

where ϕ is a function of stresses. A viscoelastic solid is called linear viscoelastic if ϕ and E_∞ do not depend on the deformation [50]. Otherwise, it is called hyperviscoelastic. Figure 3.3 presents the typical stress relaxation behavior of a viscoelastic solid under constant strain.

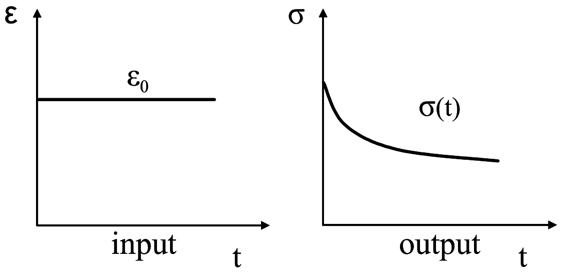


Figure 3. 3: Stress relaxation under constant strain

3.1.2 Mathematical Models for Linear Viscoelasticity

Viscoelastic materials somehow keep a record of their response history and they are said to possess memory. This memory can clearly be seen in the constitutive relationship between the stress and strain. As a result, mathematical models of viscoelastic behavior take the form of partial differential Volterra equation problems [52].

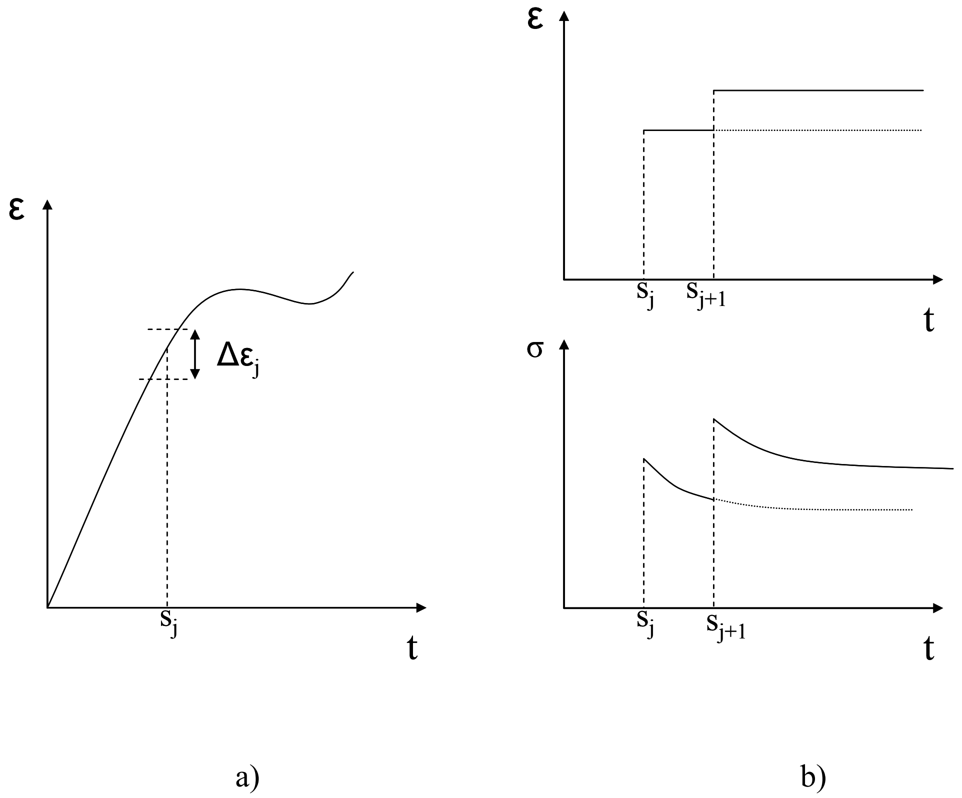


Figure 3. 4: a) Arbitrary strain input, and b) Boltzmann superposition principle

One way of deriving a constitutive relationship for linear viscoelastic materials is to assume that a Boltzmann superposition of strain increments can be applied to viscoelastic materials [52]. Consider an arbitrary strain input which is obtained through superposition of small strain increments (Figure 3.4) [53].

$$\varepsilon(t) = \sum_{j=1}^{n} \Delta\varepsilon_j = \int_0^t d\left[\varepsilon(s)\right] \tag{3.6}$$

These strain increments are related to corresponding stress increments by Hooke's law as follows

$$\sigma(t) = \sum_{j=1}^{n} \Delta\sigma_j (t - s_j) = \sum_{j=1}^{n} \mathrm{E}(t - s_j)\Delta\varepsilon_j \qquad (3.7)$$

Each of these stress increments also relax according to the time dependency of E (Figure 3.4.b). By taking appropriate limit, we get the following constitutive law

$$\sigma(t) = \int_0^t \mathrm{E}(t - s) d\left[\varepsilon(s)\right] \qquad (3.8)$$

If the strain history is differentiable, Equation 3.8 reduces to the following form

$$\sigma(t) = \int_0^t \mathrm{E}(t - s) \frac{\partial\varepsilon(s)}{\partial s} ds \qquad (3.9)$$

Time dependency of the stress relaxation function E can be more conveniently given by spring and dashpot models (Figure 3.5) [52].

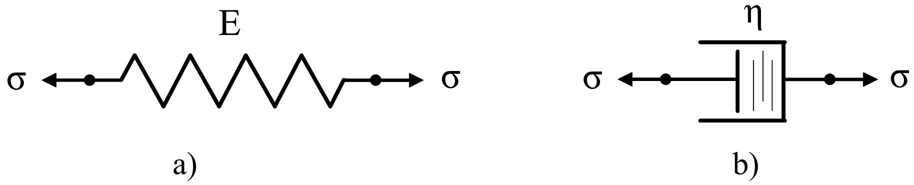


Figure 3. 5: a) Linear (Hookean) spring. E is the spring stiffness, and b) Linear (Newtonian) dashpot. η is the dashpot viscosity

In these models, the stress carried by the spring is proportional to the strain in the spring and is given by Hooke's law

$$\sigma = E\varepsilon \tag{3.10}$$

where E is the stiffness of the spring and ε, the strain in the spring, is the ratio of the change of length of the spring to its original length.

$$\varepsilon = \frac{\Delta L}{L_0} \tag{3.11}$$

σ and ε are analogous to the spring force and displacement, and E is analogous to the Young's modulus [54].The stress carried in the dashpot is proportional to the strain rate and is given by Newton's law of viscosity

$$\sigma = \eta \frac{\partial \varepsilon}{\partial t} \tag{3.12}$$

Viscoelastic materials then can be modeled as combination of springs and dashpots in series or parallel [52]. The most common mathematical models are Maxwell Model, Voigt Model, Maxwell Solid, and Generalized Maxwell Solid.

3.1.2.1 Maxwell Model

The Maxwell model (also referred as Maxwell body [47]) is a series connection of a spring and a dashpot (Figure 3.6).

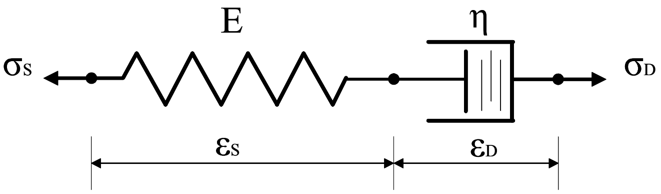


Figure 3. 6: The Maxwell model

ε_S and σ_S denote the strain and stress in the spring alone and ε_D and σ_D denote those in the dashpot alone [52]. Total stress and total strain in the Maxwell model is given by

$$\sigma = \sigma_S = \sigma_D \tag{3.13}$$

$$\varepsilon = \varepsilon_S + \varepsilon_D \tag{3.14}$$

If we differentiate both sides of Equation 3.14 and use Hooke's and Newton's laws, we end up with

$$\frac{\partial \varepsilon}{\partial t} = \frac{1}{\mathrm{E}}\frac{\partial \sigma_S}{\partial t} + \frac{\sigma_D}{\eta} \tag{3.15}$$

Making use of Equation 3.13, Equation 3.15 can be written in the form

$$\frac{\partial\sigma}{\partial t}+\frac{\sigma}{\tau}=\mathrm{E}\frac{\partial\varepsilon}{\partial t} \tag{3.16}$$

where

$$\tau=\frac{\eta}{\mathrm{E}} \tag{3.17}$$

is the *relaxation time*. Using the initial condition $\sigma(0)=\varepsilon(0)=0$, this ordinary differential equation (ODE) can be solved to give

$$\sigma(t)=\mathrm{E}\int_0^t \exp\left(-(t-s)/\tau\right)\frac{\partial\varepsilon(s)}{\partial t}ds \tag{3.18}$$

and this equation is actually Equation 3.9 with the scalar analogue of E given by

$$\mathrm{E}(t)=\mathrm{E}\exp(-t/\tau) \tag{3.19}$$

The creep function of the Maxwell model under a constant initial stress, σ_0, is

$$\varepsilon(t)=\sigma_0\left(\frac{1}{\mathrm{E}}+\frac{t}{\eta}\right) \tag{3.20}$$

[47], and, the stress relaxation function of the Maxwell model under a constant initial strain, ε_0, is

$$\sigma(t) = \varepsilon_0 E \exp(-t/\tau) \tag{3.21}$$

Figure 3.7 and 3.8 present the creep and stress relaxation behavior of the Maxwell model under different constant stress and constant strain inputs respectively.

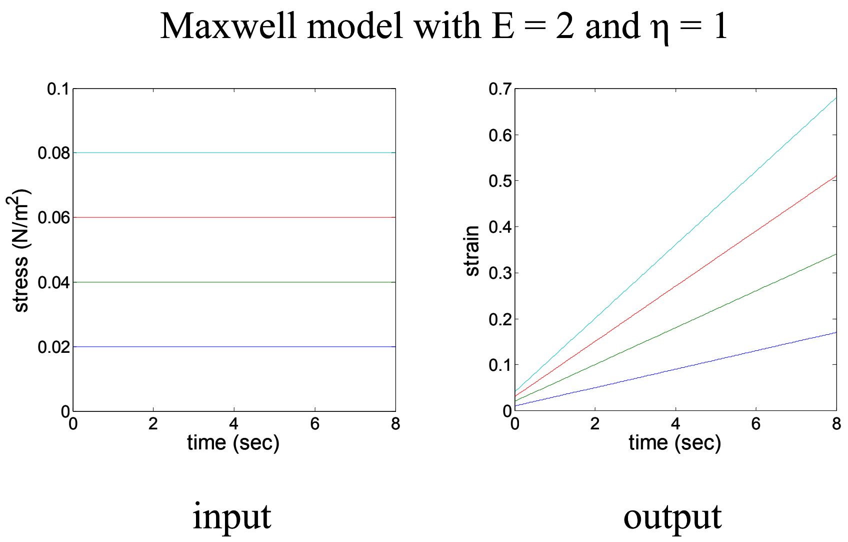


Figure 3. 7: Creep behavior of the Maxwell model under different constant stresses

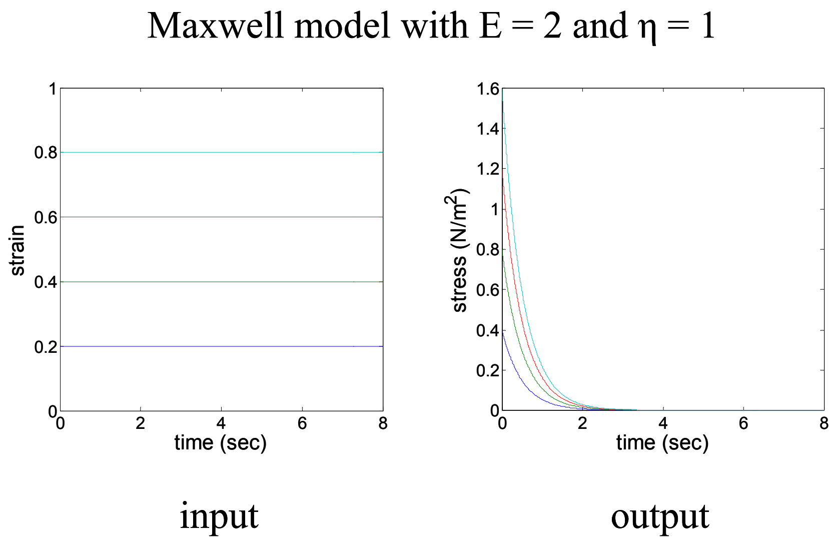


Figure 3. 8: Stress relaxation behavior of the Maxwell model under different constant strains

The stress function of the Maxwell model in response to a ramp input such as $\varepsilon(t) = At$ (Figure 3.9)

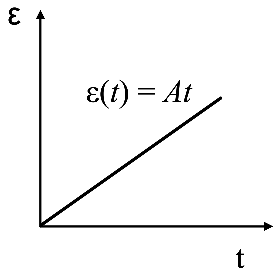


Figure 3. 9: Ramp input

where A is a scalar and the slope of the ramp can easily be derived taking the Laplace transform of both sides of Equation 3.16. Taking the Laplace transform of both sides and substituting $\varepsilon(t) = At$ we get

$$
\begin{aligned}
& s\sigma(s)+\frac{\sigma(s)}{\tau}=\mathrm{E}s\varepsilon(s) \\
& \varepsilon(t)=At \Rightarrow \varepsilon(s)=\frac{A}{s^2} \\
& \Rightarrow s\sigma(s)+\frac{\sigma(s)}{\tau}=\frac{\mathrm{E}A}{s} \\
& \sigma(s)=\frac{\mathrm{E}A}{s\left(s+(1/\tau)\right)}
\end{aligned}
\tag{3.22}
$$

Finally, taking the inverse Laplace transform of both sides in Equation 3.22, we get the stress function of the Maxwell model in response to a ramp strain input as the following:

$$\sigma(t)=\mathrm{E}A\tau\left(1-\exp(-t/\tau)\right) \tag{3.23}$$

The stress function of the Maxwell model in response to a ramp & hold input such as $\varepsilon(t)=A\left(t-(t-c)u(t-c)\right)$ (see Figure 3.10),

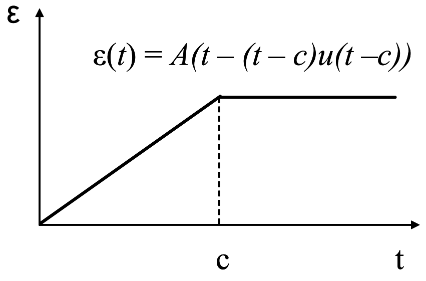


Figure 3. 10: Ramp & hold input

where *u(t-c)* is the Heaviside Unit Step function shifted by c units to the right in the time domain, can also be derived taking the Laplace transform of both sides of Equation 3.16. Taking the Laplace transform of both sides and substituting $\varepsilon(t) = A\big(t-(t-c)u(t-c)\big)$ we get

$$
\begin{aligned}
& s\sigma(s) + \frac{\sigma(s)}{\tau} = \mathrm{E}s\varepsilon(s) \\
& \varepsilon(t) = A\big(t-(t-c)u(t-c)\big) \Rightarrow \varepsilon(s) = \frac{A}{s^2}\big(1-\exp(-cs)\big) \\
& \Rightarrow s\sigma(s) + \frac{\sigma(s)}{\tau} = \frac{\mathrm{E}A}{s}\big(1-\exp(-cs)\big) \qquad (3.24) \\
& \sigma(s) = \frac{\mathrm{E}A}{s\big(s+(1/\tau)\big)} - \frac{\mathrm{E}A\exp(-cs)}{s\big(s+(1/\tau)\big)}
\end{aligned}
$$

Taking the inverse Laplace transform of both sides in Equation 3.24, we get the stress function of the Maxwell model in response to a ramp & hold strain input as the following:

$$
\sigma(t) = \mathrm{E}A\tau\Big(\big(1-\exp(-t/\tau)\big) - u(t-c)\big(1-\exp(-(t-c)/\tau)\big)\Big) \qquad (3.25)
$$

Figure 3.11 and 3.12 present the stress behavior of the Maxwell model under different ramp and ramp & hold strain inputs respectively.

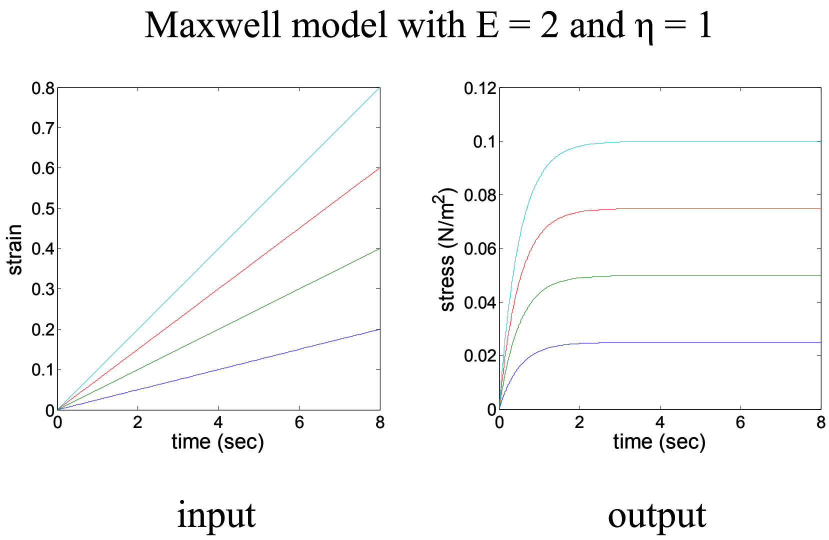


Figure 3. 11: Stress behavior of the Maxwell model under different ramp strain inputs

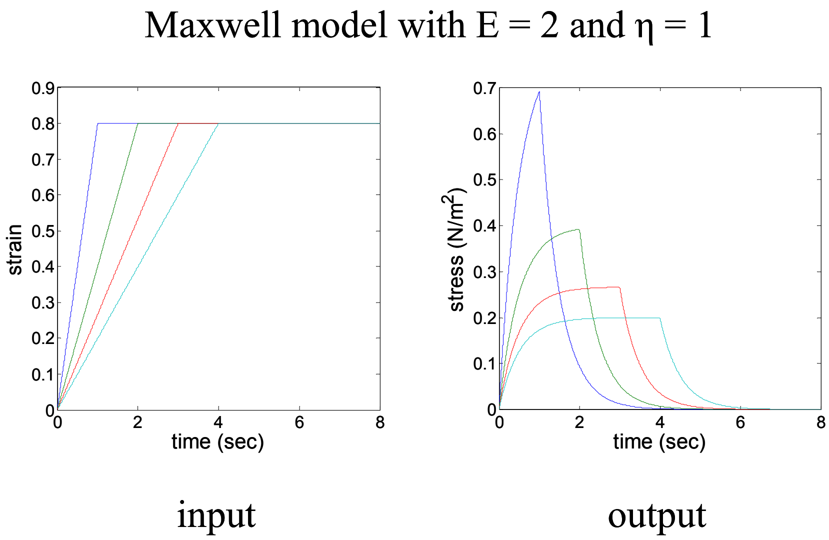


Figure 3. 12: Stress behavior of the Maxwell model under different ramp & hold strain inputs

3.1.2.2 Voigt Model

The Voigt model (also referred as Voigt body [47]) is a parallel connection of a spring and a dashpot (Figure 3.13).

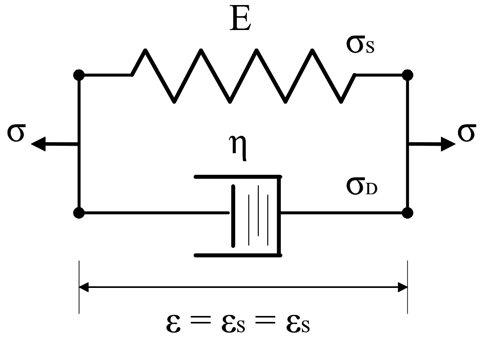


Figure 3. 13: The Voigt Model

In this model $\varepsilon = \varepsilon_S = \varepsilon_D$ and $\sigma = \sigma_S + \sigma_D$ [52]. Therefore,

$$\sigma = \eta \frac{\partial \varepsilon}{\partial t} + E\varepsilon \tag{3.26}$$

Using the initial condition $\sigma(0) = \varepsilon(0) = 0$, this ODE can be solved to give

$$\varepsilon(t) = \frac{1}{\eta} \int_0^t \exp(-(t-s)/\tau)\sigma(s)\,ds \tag{3.27}$$

The creep function of the Voigt model under a constant initial stress, σ_0, is

$$\varepsilon(t) = \frac{1}{\mathrm{E}}\left(1 - \exp\left(-t/\tau\right)\right)\sigma_0 \tag{3.28}$$

[47], and, the stress relaxation function of the Voigt model under a constant initial strain, ε_0, is

$$\sigma(t) = \eta\delta(t) + \mathrm{E}\varepsilon_0 \tag{3.29}$$

where $\delta(t)$ is the *unit-impulse function* or *dirac delta function*.

Figure 3.14 and 3.15 present the creep and stress relaxation behavior of the Voigt model under different constant stress and constant strain inputs respectively.

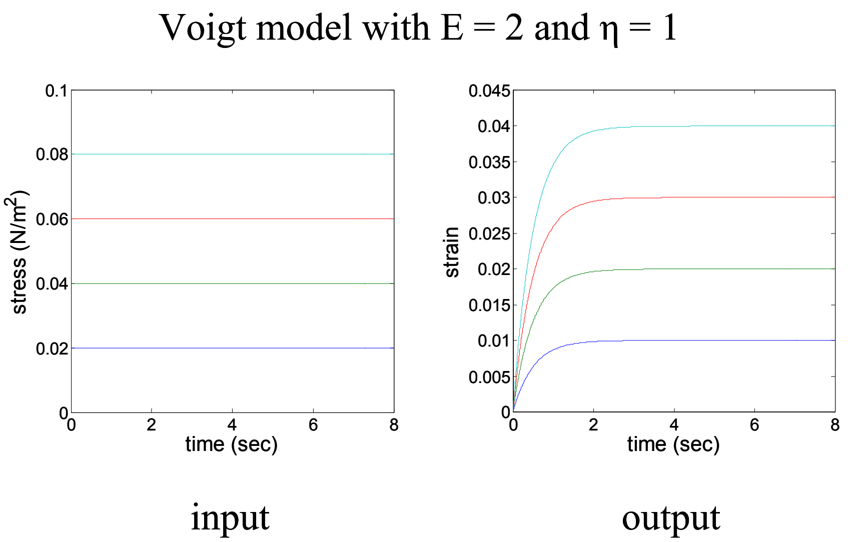


Figure 3. 14: Creep behavior of the Voigt model under different constant stresses

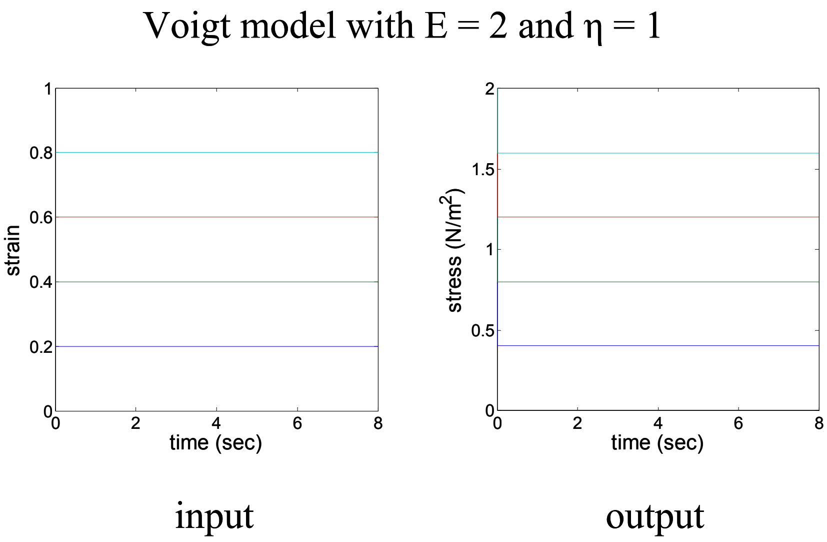


Figure 3. 15: Stress relaxation behavior of the Voigt model under different constant strains

The stress function of the Voigt model in response to a ramp input such as $\varepsilon(t) = At$ is basically

$$\sigma(t) = A(\eta + \mathrm{E}t) \tag{3.30}$$

One can calculate the stress function of the Voigt model in response to a ramp & hold input such as $\varepsilon(t) = A\left(t - (t-c)u(t-c)\right)$ by taking the Laplace transform of both sides of Equation 3.26 and substituting $\varepsilon(t) = A\left(t - (t-c)u(t-c)\right)$ as

$$
\begin{aligned}
&\sigma(s) = \eta s\varepsilon(s) + \mathrm{E}\varepsilon(s) \\
&\varepsilon(t) = A\big(t - (t-c)u(t-c)\big) \Rightarrow \varepsilon(s) = \frac{A}{s^2}\big(1 - \exp(-cs)\big) \\
&\Rightarrow \sigma(s) = \eta\frac{A}{s}\big(1 - \exp(-cs)\big) + \mathrm{E}\frac{A}{s^2}\big(1 - \exp(-cs)\big)
\end{aligned}
\tag{3.32}
$$

Taking the inverse Laplace transform of both sides in Equation 3.32, we get the stress function of the Voigt model in response to a ramp & hold strain input as the following:

$$
\sigma(t) = \eta A\big(1 - u(t-c)\big) + \mathrm{E}A\big(t - (t-c)u(t-c)\big) \tag{3.33}
$$

Figure 3.16 and 3.17 present the stress behavior of the Voigt model under different ramp and ramp & hold strain inputs respectively.

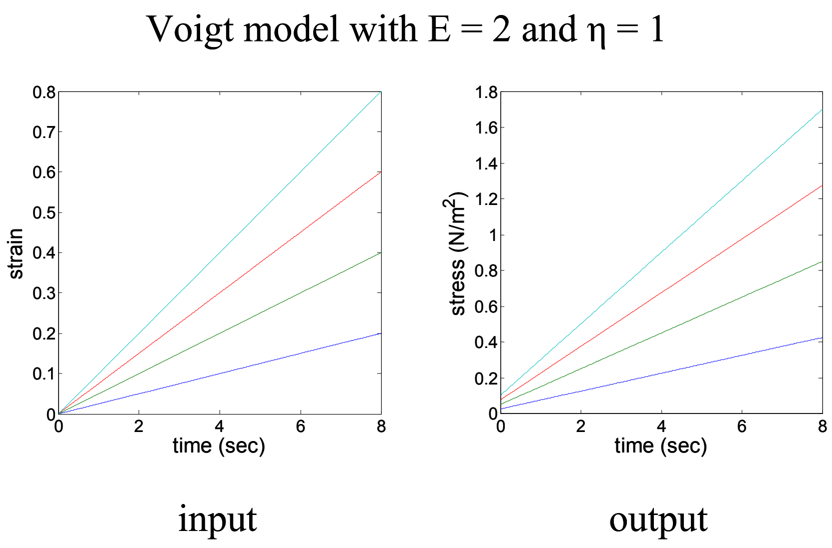


Figure 3. 16: Stress behavior of the Voigt model under different ramp strain inputs

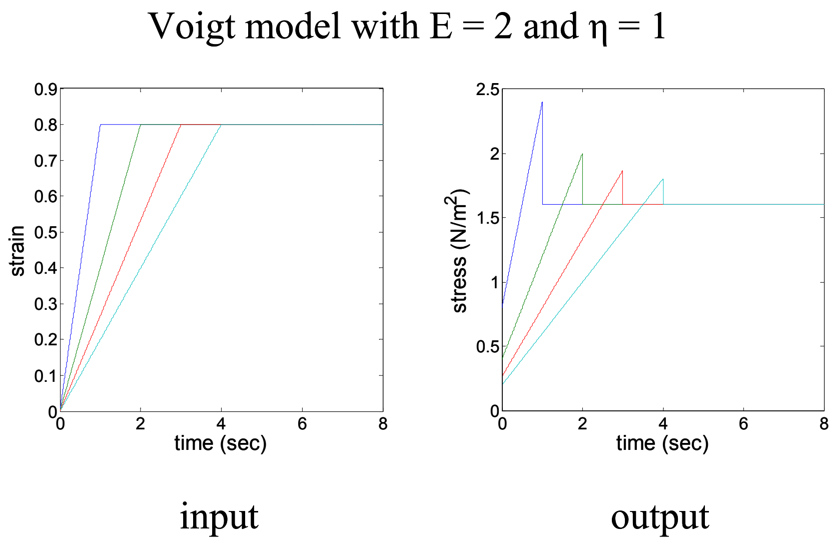


Figure 3. 17: Stress behavior of the Voigt model under different ramp & hold strain inputs

3.1.2.3 Maxwell Solid

The Maxwell solid (also referred as Kelvin body or standard linear solid [47]) is a parallel connection of a Maxwell model and a spring (Figure 3.18).

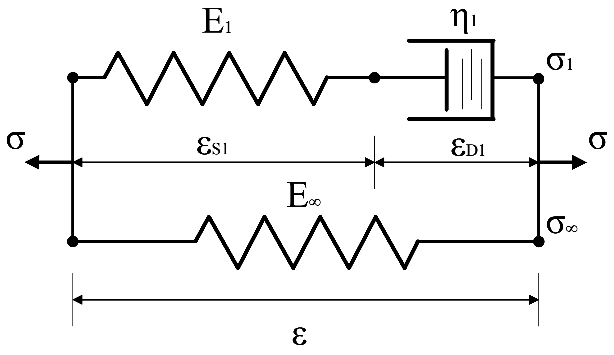


Figure 3. 18: The Maxwell solid

In Maxwell solid the stress in the spring is

$$\sigma_\infty = E_\infty \varepsilon \tag{3.34}$$

and the stress in the Maxwell model is

$$\sigma_1 = E_1 \varepsilon_{S1} = \eta_1 \frac{\partial \varepsilon_{D1}}{\partial t} \tag{3.35}$$

The total strain in the Maxwell solid is

$$\varepsilon = \varepsilon_{S1} + \varepsilon_{D1} \tag{3.36}$$

and total stress is the sum of the stresses in the Maxwell model and spring [47]

$$\begin{aligned} \sigma &= \sigma_\infty + \sigma_1 \\ \sigma &= E_\infty \varepsilon + E_1 \varepsilon_{S1} = \left(E_\infty + E_1\right)\varepsilon - E_1 \varepsilon_{D1} \end{aligned} \tag{3.37}$$

Multiplying the result in Equation 3.37 by (η_1 / E_1) and taking the derivative with respect to time gives:

$$\frac{\eta_1}{E_1}\frac{\partial \sigma}{\partial t} = \frac{\eta_1}{E_1}\left(E_\infty + E_1\right)\frac{\partial \varepsilon}{\partial t} - \eta_1 \frac{\partial \varepsilon_{D1}}{\partial t} \tag{3.38}$$

If we add Equation 3.38 to the result we get in Equation 3.37, we have

$$\sigma+\frac{\eta_1}{E_1}\frac{\partial\sigma}{\partial t}=\left(E_\infty+E_1\right)\varepsilon-E_1\varepsilon_{D1}+\frac{\eta_1}{E_1}\left(E_\infty+E_1\right)\frac{\partial\varepsilon}{\partial t}-\eta_1\frac{\partial\varepsilon_{D1}}{\partial t} \tag{3.39}$$

Finally, replacing the last term in Equation 3.39 by $E_1\varepsilon_{S1}$, and using Equation 3.36, we get

$$\sigma+\tau_1\frac{\partial\sigma}{\partial t}=E_\infty\varepsilon+\eta_1\left(1+\frac{E_\infty}{E_1}\right)\frac{\partial\varepsilon}{\partial t} \tag{3.40}$$

where τ_1 = (η_1 / E_1) [47]. Using the initial condition $\sigma\left(0\right)=\varepsilon\left(0\right)=0$, this ODE can be solved to give [52]

$$\sigma\left(t\right)=E_\infty\varepsilon\left(t\right)+\int_0^t E_1\exp\left(-\left(t-s\right)/\tau_1\right)\frac{\partial\varepsilon(s)}{\partial s}ds \tag{3.41}$$

The creep function of the Maxwell solid under a constant initial stress, σ_0, is

$$\varepsilon\left(t\right)=\frac{1}{E_\infty}\left[1-\left(1-\frac{\tau_1}{\tau_\sigma}\right)\exp\left(-t/\tau_\sigma\right)\right]\sigma_0 \tag{3.42}$$

where τ_σ is called *the relaxation time for constant stress* [47] and is equal to

$$\tau_\sigma=\frac{\eta_1}{E_\infty}\left(1+\frac{E_\infty}{E_1}\right) \tag{3.43}$$

The stress relaxation function of the Maxwell solid under a constant initial strain, ε_0, is

$$\sigma(t) = \mathrm{E}_\infty \left[1 - \left(1 - \frac{\tau_\sigma}{\tau_1} \right) \exp(-t/\tau_1) \right] \varepsilon_0 \tag{3.44}$$

[47]. If we substitute Equation 3.43 for τ_σ in Equation 3.44, we get

$$\sigma(t) = \left[\mathrm{E}_\infty + \mathrm{E}_1 \exp(-t/\tau_1) \right] \varepsilon_0 \tag{3.45}$$

and the relaxation function of the Equation 3.45 is actually the well-known Prony Series expression with N = 1

$$\mathrm{E}(t) = \mathrm{E}_\infty + \sum_{j=1}^{N} \mathrm{E}_j \exp(-t/\tau_j) \tag{3.46}$$

Figure 3.19 and 3.20 present the creep and stress relaxation behavior of the Maxwell Solid under different constant stress and constant strain inputs respectively.

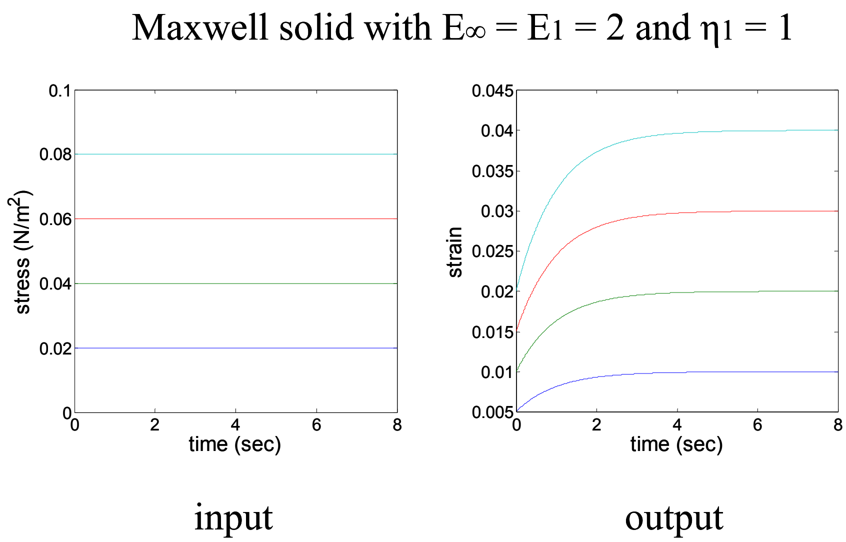


Figure 3. 19: Creep behavior of the Maxwell solid under different constant stresses

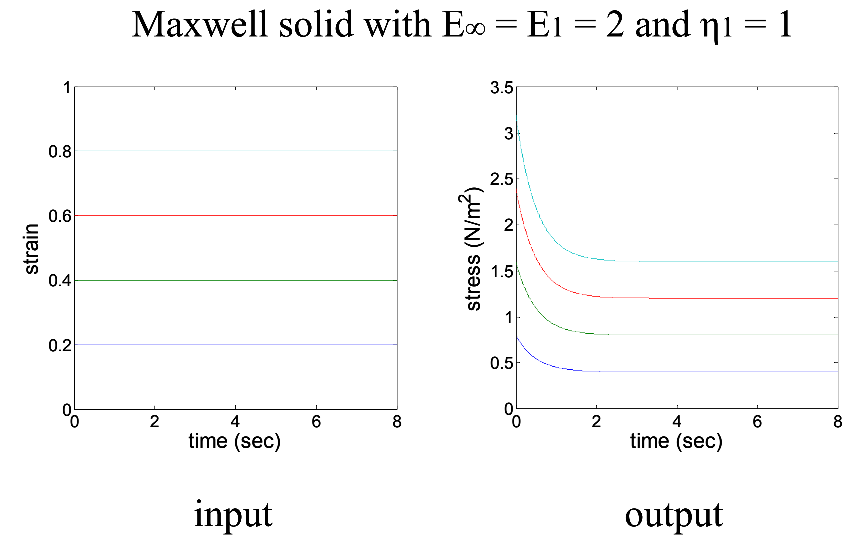


Figure 3. 20: Stress relaxation behavior of the Maxwell solid under different constant strains

The stress function of the Maxwell solid in response to a ramp input such as $\varepsilon(t) = At$ can be derived taking the Laplace transform of both sides of Equation 3.40 and and substituting $\varepsilon(t) = At$

$$\begin{aligned}
&\sigma(s) + \tau_1 s\sigma(s) = E_\infty\varepsilon(s) + \eta_1\left(1 + \frac{E_\infty}{E_1}\right)s\varepsilon(s) \\
&\varepsilon(t) = At \Rightarrow \varepsilon(s) = \frac{A}{s^2} \\
&\sigma(s)(1+\tau_1 s) = E_\infty\frac{A}{s^2} + \eta_1\left(1 + \frac{E_\infty}{E_1}\right)\frac{A}{s} \Rightarrow \sigma(s) = A\left[\frac{E_\infty\frac{1}{s^2} + \eta_1\left(1+\frac{E_\infty}{E_1}\right)\frac{1}{s}}{(1+\tau_1 s)}\right] \\
&\sigma(s) = A\left[\frac{E_\infty + \eta_1\left(1+\frac{E_\infty}{E_1}\right)s}{s^2(1+\tau_1 s)}\right] \Rightarrow \sigma(s) = A\left[\frac{E_\infty}{s^2} + \frac{\eta_1}{s} - \frac{\eta_1}{s+\frac{1}{\tau_1}}\right]
\end{aligned} \tag{3.47}$$

Taking the inverse Laplace transform of both sides of the result in Equation 3.47, we get the stress function of the Maxwell solid in response to a ramp strain input as the following:

$$\begin{aligned}
&\sigma(t) = Az(t) \\
&z(t) = E_\infty t + \eta_1\left(1 - \exp(-t/\tau_1)\right)
\end{aligned} \tag{3.48}$$

Similarly, the stress function of the Maxwell solid in response to a ramp & hold input such as $\varepsilon(t)=A\big(t-(t-c)u(t-c)\big)$ can be derived taking the Laplace transform of both sides of Equation 3.40 and substituting $\varepsilon(t)=A\big(t-(t-c)u(t-c)\big)$

$$\sigma(s)+\tau_1 s\sigma(s)=\mathrm{E}_\infty\varepsilon(s)+\eta_1\left(1+\frac{\mathrm{E}_\infty}{\mathrm{E}_1}\right)s\varepsilon(s)$$

$$\varepsilon(t)=A\big(t-(t-c)u(t-c)\big)\Rightarrow\varepsilon(s)=\frac{A}{s^2}\big(1-\exp(-cs)\big)$$

$$\sigma(s)=A\big(1-\exp(-cs)\big)\left[\frac{\mathrm{E}_\infty+\eta_1\left(1+\dfrac{\mathrm{E}_\infty}{\mathrm{E}_1}\right)s}{s^2\left(1+\tau_1 s\right)}\right] \tag{3.49}$$

$$\sigma(s)=A\big(1-\exp(-cs)\big)\left[\frac{\mathrm{E}_\infty}{s^2}+\frac{\eta_1}{s}-\frac{\eta_1}{s+\dfrac{1}{\tau_1}}\right]$$

Taking the inverse Laplace transform of both sides of the result in Equation 3.49, we get the stress function of the Maxwell solid in response to a ramp & hold strain input as the following:

$$\sigma(t)=A\left[z(t)-z(t-c)u(t-c)\right] \tag{3.50}$$

Figure 3.21 and 3.22 present the stress behavior of the Maxwell solid under different ramp and ramp & hold strain inputs respectively.

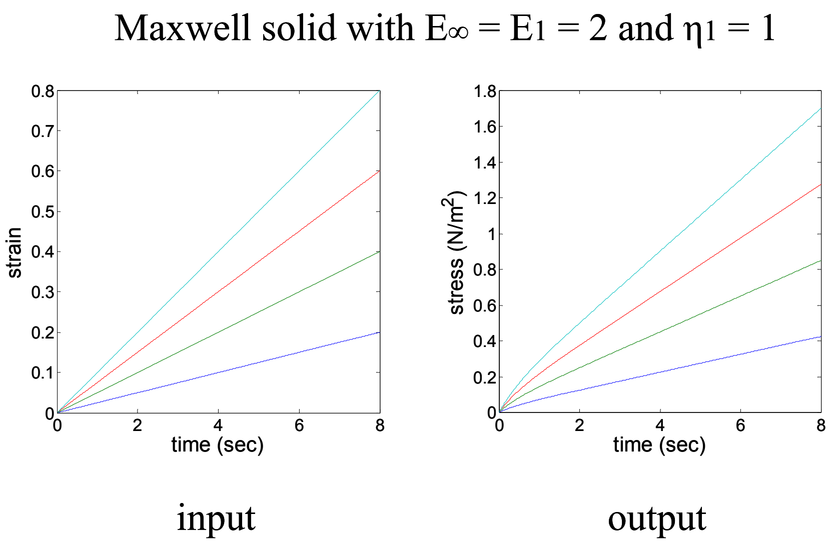


Figure 3. 21: Stress behavior of the Maxwell solid under different ramp strain inputs

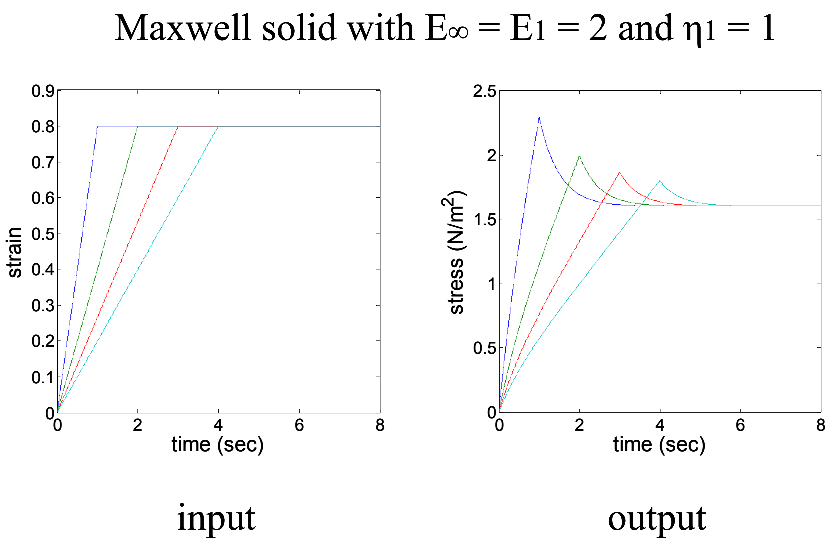


Figure 3. 22: Stress behavior of the Maxwell solid under different ramp & hold strain inputs

3.1.2.4 Generalized Maxwell Solid

The Generalized Maxwell solid is a parallel connection of N Maxwell models and a spring (Figure 3.23).

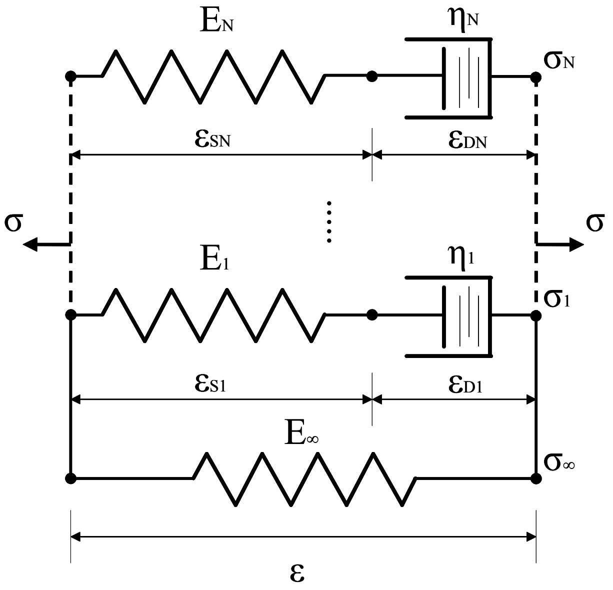


Figure 3. 23: The Generalized Maxwell solid

Using the similar idea performed in the previous section, the constitutive relationship between the stress and strain in the Generalized Maxwell Solid can be found as the following

$$\sigma(t) = \int_0^t \mathrm{E}(t-s)\frac{\partial\varepsilon(s)}{\partial s}ds \tag{3.51}$$

where the relaxation function

$$\mathrm{E}(t) = \mathrm{E}_\infty + \sum_{j=1}^{N} \mathrm{E}_\mathrm{j} \exp\left(-t/\tau_\mathrm{j}\right) \tag{3.52}$$

is the Prony series expression for N elements [52] and

$$\tau_j = \frac{\eta_j}{\mathrm{E}_j} \tag{3.53}$$

Similar to the Maxwell solid, the stress relaxation function of the Generalized Maxwell solid under a constant initial strain, ε_0, is

$$\sigma(t) = \left[\mathrm{E}_\infty + \sum_{j=1}^{N} \mathrm{E}_\mathrm{j} \exp\left(-t/\tau_\mathrm{j}\right)\right]\varepsilon_0 \tag{3.54}$$

Similarly, the stress function of the Generalized Maxwell solid in response to a ramp input such as $\varepsilon(t) = At$ is

$$\begin{gathered} \sigma(t) = Az(t) \\ z(t) = \mathrm{E}_\infty t + \sum_{j=1}^{N} \eta_j \left(1 - \exp\left(-t/\tau_j\right)\right) \end{gathered} \tag{3.55}$$

and the stress function of the Generalized Maxwell solid in response to a ramp & hold input such as $\varepsilon(t) = A\left(t - (t-c)u(t-c)\right)$ is

$$\sigma(t) = A\left[z(t) - z(t-c)u(t-c)\right] \tag{3.56}$$

3.2 Hyperelasticity

The mathematical models for linear viscoelasticity discussed in section 3.1.2 are limited to model the soft tissues in the linear range where the deformations are assumed to be small and the constitutive relationship between stress and strain is linear. However, soft tissues can undergo large deformations and the stress-strain relationship is actually not linear due to the layered and non-homogeneous complex structure of the soft tissues.

Hyperelasticity is a time-independent material nonlinearity where the relationship between stress and strain is nonlinear. A material is called hyperelastic if it has a scalar function from which stresses can be derived. The function is called strain-energy function depending only on deformation gradient tensor, ***F*** [47]. In linear continuum mechanics, the Cauchy stress tensor, **σ,** which is the true stress in the body, and the strain tensor, **ε,** are used to describe the constitutive equation. In nonlinear continuum mechanics, quantities are required that are invariant to rigid-body movements. Two different stress tensors can be defined in addition to Cauchy stress tensor as

$$\begin{aligned} \mathbf{T} &= \mathbf{J}\mathbf{F}^{-1}\boldsymbol{\sigma} \\ \mathbf{S} &= \mathbf{T}\left(\mathbf{F}^{T}\right)^{-1} \end{aligned} \qquad (3.57)$$

where **T** is the *first Piola-Kirchhoff Stress Tensor* and **S** is the *second Piola-Kirchhoff Stress Tensor*. The second Piola-Kirchhoff stresses are directly related to strain-energy function as follows

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} \qquad (3.58)$$

where **E** is the Lagrangian Green's strain tensor

To describe a constitutive equation for hyperelastic materials, Equation 3.58 is used based on several strain-energy functions. If isotropy and incompressibility are assumed for soft tissues, hyperelasticity can be modeled using different strain-energy functions such as, neo-Hookean, Mooney-Rivlin, polynomial, Yeoh, Ogden and Arruda-Boyce. Some of the most common strain-energy functions are as follows

Neo-Hookean model: $$W = C_{10}(I_1 - 3) \tag{3.59}$$

Mooney-Rivlin model: $$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \tag{3.60}$$

Polynomial form: $$W = \sum_{i=0, j=0}^{N} C_{ij}(I_1 - 3)^i (I_2 - 3)^j \tag{3.61}$$

Yeoh model: $$W = \sum_{i=0}^{N} C_{i0}(I_1 - 3)^i \tag{3.62}$$

Ogden model: $$W = \sum_{i=1}^{n} \frac{\mu_i}{\alpha_i}(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3) \tag{3.63}$$

where C_{ij} and μ_i are material stiffness constants corresponding to Young's modulus in linear material. Except the Ogden model, above models are based on the principle invariants and the strain energy function depends on the first and the second principle invariants (i.e. $W = f(I_1, I_2)$) due to incompressibility conditions. The Ogden model is directly based on principle stretch ratios, λ.

3.2.1 Mooney-Rivlin Force Formulation of a Cylindrical Phantom under Uniaxial Compression

A constitutive relation can be derived between the applied force and the resulting strain of a hyperelastic cylindrical phantom going under uniaxial compression based on a Mooney-Rivlin strain-energy function. The three principle invariants for a cylindrical phantom represented in Figure 3.24 are as follows

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2 \\ I_3 &= \lambda_1^2\lambda_2^2\lambda_3^2 \end{aligned} \tag{3.64}$$

where λ_1, λ_2, and λ_3 are the principle stretch ratios in the direction of cartesian coordinate axes.

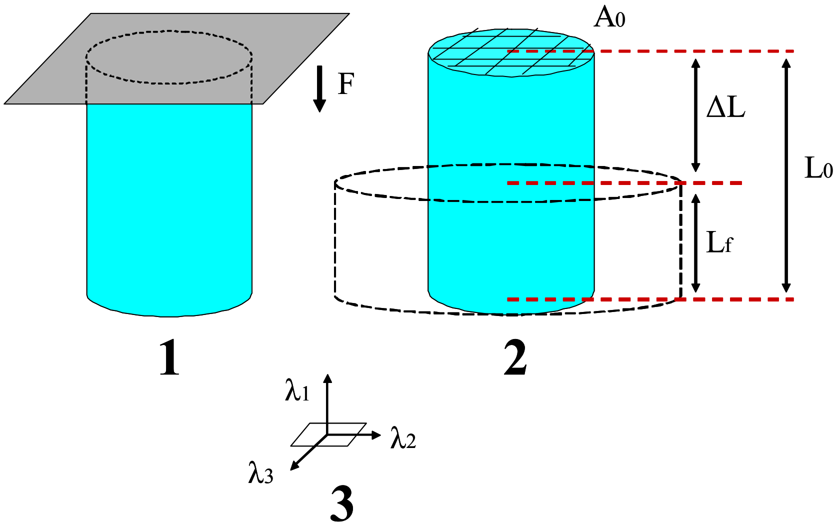


Figure 3. 24: Uniaxial compression of a cylindrical phantom. The phantom is compressed from the top by applying a nonlinear external force F (1) in an amount ΔL (2). The principle stretch ratios are congruent with the Cartesian coordinates (3).

Stretch ratio is a measure of deformation defined as

$$\lambda = \frac{L_f}{L_0} \tag{3.65}$$

in one dimension where L_0 is the rest length and L_f is the final length of the cylindrical phantom. If we assume incompressibility, then the multiplication of the principle stretch ratios is equal to 1.

$$\lambda_1 \lambda_2 \lambda_3 = 1 \tag{3.66}$$

Also, uniaxial compression implies that the stretch ratios other than the one in the compression direction are equal to each other because of the symmetry of the cylinder's geometry in the horizontal.

$$\lambda_2 = \lambda_3 \tag{3.67}$$

Substituting Equation 3.67 into Equation 3.66, one can get

$$\begin{gathered} \lambda_1 \lambda_2 \lambda_2 = 1 \\ \lambda_2^2 = \frac{1}{\lambda_1} \\ \lambda_2 = \lambda_3 = \sqrt{\frac{1}{\lambda_1}} \end{gathered} \tag{3.68}$$

λ_1, stretch ratio along the compression direction, can be written as

$$\lambda_1 = \frac{L_f}{L_0} = \frac{L_0 - \Delta L}{L_0} = 1 - \frac{\Delta L}{L_0} = 1 - \varepsilon \tag{3.69}$$

Substituting the calculated values of the stretch ratios into Equation 3.64, and substituting *λ* for *λ₁* for a more general representation, we get

$$\begin{aligned} I_1 &= \lambda_1^2 + \left(\sqrt{\frac{1}{\lambda_1}}\right)^2 + \left(\sqrt{\frac{1}{\lambda_1}}\right)^2 = \lambda_1^2 + \frac{2}{\lambda_1} = \lambda^2 + \frac{2}{\lambda} \\ I_2 &= \lambda_1^2\left(\sqrt{\frac{1}{\lambda_1}}\right)^2 + \lambda_1^2\left(\sqrt{\frac{1}{\lambda_1}}\right)^2 + \left(\sqrt{\frac{1}{\lambda_1}}\right)^2\left(\sqrt{\frac{1}{\lambda_1}}\right)^2 = 2\lambda_1 + \frac{1}{\lambda_1^2} = 2\lambda + \frac{1}{\lambda^2} \\ I_3 &= 1 \end{aligned} \tag{3.70}$$

Engineering stress is a measure of the force per unit area within a body

$$\sigma = \frac{F}{A_0} \tag{3.71}$$

where A_0 is the initial area of the cylinder before compression (Figure 3.24). Engineering stress is also defined as the rate of change of strain energy with respect to strain, which can also be represented in terms of the stretch ratio

$$\sigma = \frac{\partial W}{\partial \varepsilon} = \frac{\partial W}{\partial \lambda}\frac{\partial \lambda}{\partial \varepsilon} \tag{3.72}$$

Two most common Mooney-Rivlin strain-energy formulations are 2- and 5-term Mooney-Rivlin formulations (MR2 and MR5 respectively)

$$\begin{aligned} W_{\mathrm{MR2}} &= C_{10}(I_1-3)+C_{01}(I_2-3) \\ W_{\mathrm{MR5}} &= C_{10}(I_1-3)+C_{01}(I_2-3)+C_{20}(I_1-3)^2 \\ &\quad +C_{11}(I_1-3)(I_2-3)+C_{02}(I_2-3)^2 \end{aligned} \tag{3.73}$$

If the calculated values of principle invariants from Equation 3.70 are substituted into Equation 3.73, one can get the strain energy formulations in terms of the stretch ratio:

$$\begin{aligned} W_{MR2} &= C_{10}(\lambda^2+\frac{2}{\lambda}-3)+C_{01}(2\lambda+\frac{1}{\lambda^2}-3) \\ W_{MR5} &= C_{10}(\lambda^2+\frac{2}{\lambda}-3)+C_{01}(2\lambda+\frac{1}{\lambda^2}-3)+C_{20}(\lambda^2+\frac{2}{\lambda}-3)^2 \\ &\quad +C_{11}(\lambda^2+\frac{2}{\lambda}-3)(2\lambda+\frac{1}{\lambda^2}-3)+C_{02}(2\lambda+\frac{1}{\lambda^2}-3)^2 \end{aligned} \tag{3.74}$$

From Equation 3.69, the derivative of the stretch ratio with respect to the strain is

$$\frac{\partial\lambda}{\partial\varepsilon}=\frac{\partial(1-\varepsilon)}{\partial\varepsilon}=-1 \tag{3.75}$$

Taking the derivative of the strain-energy functions in Equation 3.74 with respect to the stretch ratio and substituting in Equation 3.72 along with Equation 3.75, one can get

$$\sigma = \frac{\partial W}{\partial \varepsilon} = \frac{\partial W}{\partial \lambda}\frac{\partial \lambda}{\partial \varepsilon}$$

$$\sigma_{MR2} = \left[\frac{\partial \left(C_{10}\left(\lambda^2 + \frac{2}{\lambda} - 3 \right) + C_{01}\left(2\lambda + \frac{1}{\lambda^2} - 3 \right) \right)}{\partial \lambda} \right] \left[-1 \right]$$

$$\sigma_{MR5} = \left[\frac{\partial \begin{pmatrix} C_{10}\left(\lambda^2 + \frac{2}{\lambda} - 3 \right) + C_{01}\left(2\lambda + \frac{1}{\lambda^2} - 3 \right) + C_{20}\left(\lambda^2 + \frac{2}{\lambda} - 3 \right)^2 \\ + C_{11}\left(\lambda^2 + \frac{2}{\lambda} - 3 \right)\left(2\lambda + \frac{1}{\lambda^2} - 3 \right) + C_{02}\left(2\lambda + \frac{1}{\lambda^2} - 3 \right)^2 \end{pmatrix}}{\partial \lambda} \right] \left[-1 \right] \qquad (3.76)$$

$$\sigma_{MR2} = -2\left[C_{10}A_1 + C_{01}A_2 \right]$$
$$\sigma_{MR5} = -2\left[C_{10}A_1 + C_{01}A_2 + 2C_{20}A_1A_3 + C_{11}\left(A_1A_4 + A_2A_3 \right) + 2C_{02}A_2A_4 \right]$$

where

$$A_1 = \lambda - \frac{1}{\lambda^2} = \left(1-\varepsilon\right) - \frac{1}{\left(1-\varepsilon\right)^2}$$
$$A_2 = 1 - \frac{1}{\lambda^3} = 1 - \frac{1}{\left(1-\varepsilon\right)^3}$$
$$A_3 = \lambda^2 + \frac{2}{\lambda} - 3 = \left(1-\varepsilon\right)^2 + \frac{2}{\left(1-\varepsilon\right)} - 3 \qquad (3.77)$$
$$A_4 = 2\lambda + \frac{1}{\lambda^2} - 3 = 2\left(1-\varepsilon\right) + \frac{1}{\left(1-\varepsilon\right)^2} - 3$$

Finally, substituting Equation 3.71 into the result achieved in Equation 3.76, one can get a constitutive relationship between the applied force and the resulting strain based on MR2 and MR5 formulations during the uniaxial compression of a cylindrical phantom as

$$\begin{aligned}
\sigma_{MR2} &= \frac{F}{A_0} = -2\left[C_{10}A_1 + C_{01}A_2\right] \\
F_{MR2} &= -2A_0\left[C_{10}A_1 + C_{01}A_2\right] \\
\sigma_{MR5} &= \frac{F}{A_0} = -2\left[C_{10}A_1 + C_{01}A_2 + 2C_{20}A_1A_3 + C_{11}\left(A_1A_4 + A_2A_3\right) + 2C_{02}A_2A_4\right] \\
F_{MR5} &= -2A_0\left[C_{10}A_1 + C_{01}A_2 + 2C_{20}A_1A_3 + C_{11}\left(A_1A_4 + A_2A_3\right) + 2C_{02}A_2A_4\right]
\end{aligned} \tag{3.76}$$

3.3 Nonlinear Viscoelasticity

Fung [47] has modified the standard linear viscoelastic constitutive relationship in Equation 3.9 to come up with the *quasi-linear viscoelasticity* (QLV) which is able to model the viscoelastic materials undergoing large deformations accurately. According to Fung's QLV theory, the relaxation function can be separated as follows:

$$\begin{aligned}
\sigma(t) &= \int_0^t \mathrm{E}(t-s)\frac{\partial\varepsilon(s)}{\partial s}ds \\
\mathrm{E}(t) &= \mathrm{G}(t)\mathrm{T}^{\mathrm{e}}\left(\lambda\right)
\end{aligned} \tag{3.77}$$

where G(t) is the reduced relaxation function and $\mathrm{T}^{\mathrm{e}}(\lambda)$ is the *elastic response* instantaneously generated in the material when a step function of stretching λ is imposed on the material [47]. Pryse et al. [55] states that Fung's QLV model can be considered as a specific case of a more general nonlinear viscoelastic formulation having a constitutive relationship as the following:

$$\sigma(t) = \int_0^t \mathrm{E}(t-s,\varepsilon)\frac{\partial\varepsilon(s)}{\partial s}ds \tag{3.78}$$

where the relaxation function E(t,ε) is both time and strain dependent. The relaxation function can also be separated into a discrete relaxation notation

$$\mathrm{E}(t,\varepsilon) = \mathrm{E}_\infty(\varepsilon) + \sum_{j=1}^{N} \mathrm{E}_j(t,\varepsilon)\exp\left(-t/\tau_j(\varepsilon)\right) \tag{3.79}$$

where $\tau_j(\varepsilon) = \dfrac{\eta_j(\varepsilon)}{\mathrm{E}_j(\varepsilon)}$ are strain-dependent time constants and $\mathrm{E}_j(\varepsilon)$ are strain-dependent amplitudes corresponding to $\tau_j(\varepsilon)$ [55]. Actually the nonlinear viscoelastic constitutive relation given in Equation 3.79 corresponds to the Generalized Maxwell solid with N elements having nonlinear springs and dashpots presented in Figure 3.25 [55]

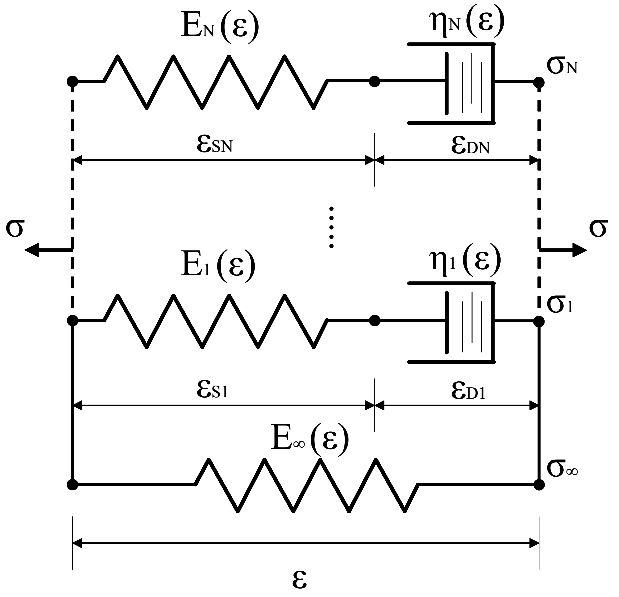


Figure 3. 25: The nonlinear viscoelastic model proposed by Pryse et al. [55].

The Generalized Maxwell solid with the strain-dependent nonlinear springs and dashpots in Figure 3.25 is able to model the nonlinear viscoelastic materials that undergo large deformations.

Chapter 4

MEASUREMENT and CHARACTERIZATION of NONLINEAR VISCOELASTIC MECHANICAL PROPERTIES of a CYLINDRICAL TISSUE PHANTOM

This chapter first provides an overview of a robotic indenter developed for measuring the mechanical properties of soft tissues and deformable objects. The strain-dependent nonlinear elastic response and time-dependent viscoelastic response of a tissue-like silicon phantom are measured via static loading and ramp & hold experiments using this robotic indenter. A lumped nonlinear viscoelastic Maxwell Solid with N=1 and N=2 is used to model the behavior of the phantom and also to estimate its material properties.

4.1 Robotic Indenter

4.1.1. Design Considerations

One method to measure complex material behavior of soft tissues is to make mechanical indentations on tissue and record force response with respect to indentation depth and elapsed time for further analysis. In addition to the classification discussed in Section 2.2, there are also two forms of measuring soft tissue response during a minimally invasive surgery: a) free-form measurements and b) robotic measurements, which are both considered as possible candidates for our design. A “free-form” measurement typically involves the use of a hand-held probe equipped with position and force sensors. An operator holds the instrument and indents organ surface manually to measure displacement and force response [12, 16]. Major benefit of using a hand-held probe over a robotic arm is safety. Since the operator manually derives the probe, unexpected and risky movements are unlikely to happen. However, there are two problems with this design. First, the measurements are made manually and not repeatable. Carter et al. [16] used a load-cell triggered by the operator to make more controlled indentations, which provides a partial solution to the repeatability problem. The second difficulty is the identification of a reference point for the displacement measurements. Linear Variable Differential Transformers (LVDTs) have been used as position sensors for relative measurement of tissue displacement with respect to a reference point. However, it is not easy to keep the probe stationary during the measurements and hence the reference point changes as the probe moves.

The second group of measurements involves the use of a robotic device for providing better controlled stimuli [2, 5, 13, 17, 18, 57, 58, 59]. If a robotic arm is used for the measurements, the problems related to the actuation and position sensing can be solved. More controlled indentations can be performed on tissue surface by a pre-programmed robotic arm and an indenter attached to the arm. In addition, a robotic arm can be programmed to generate different types of stimuli. Thus, dependency to a user is eliminated and repeatability is achieved. Besides, the tip coordinates of the indenter can be acquired using encoders of the robotic arm with respect to the fixed coordinate frame.

4.1.2. Design Details

Based on the design requirements discussed above, we have developed a robotic indenter and measured the material properties of soft tissues in the abdominal region during a MIS in one of our previous studies [2, 5]. The indenter was specifically designed to be operated in MIS. We made minor modifications on the indenter to measure the material properties of the silicon tissue phantom via compression experiments in laboratory conditions. The major components of our system include a robotic arm which can be programmed to make indentations on tissue, a flat aluminum plate that compresses the tissue phantom from the top surface of the phantom, and a force sensor attached to the plate for the measurement of force response of the tissue (Figure 4.1).

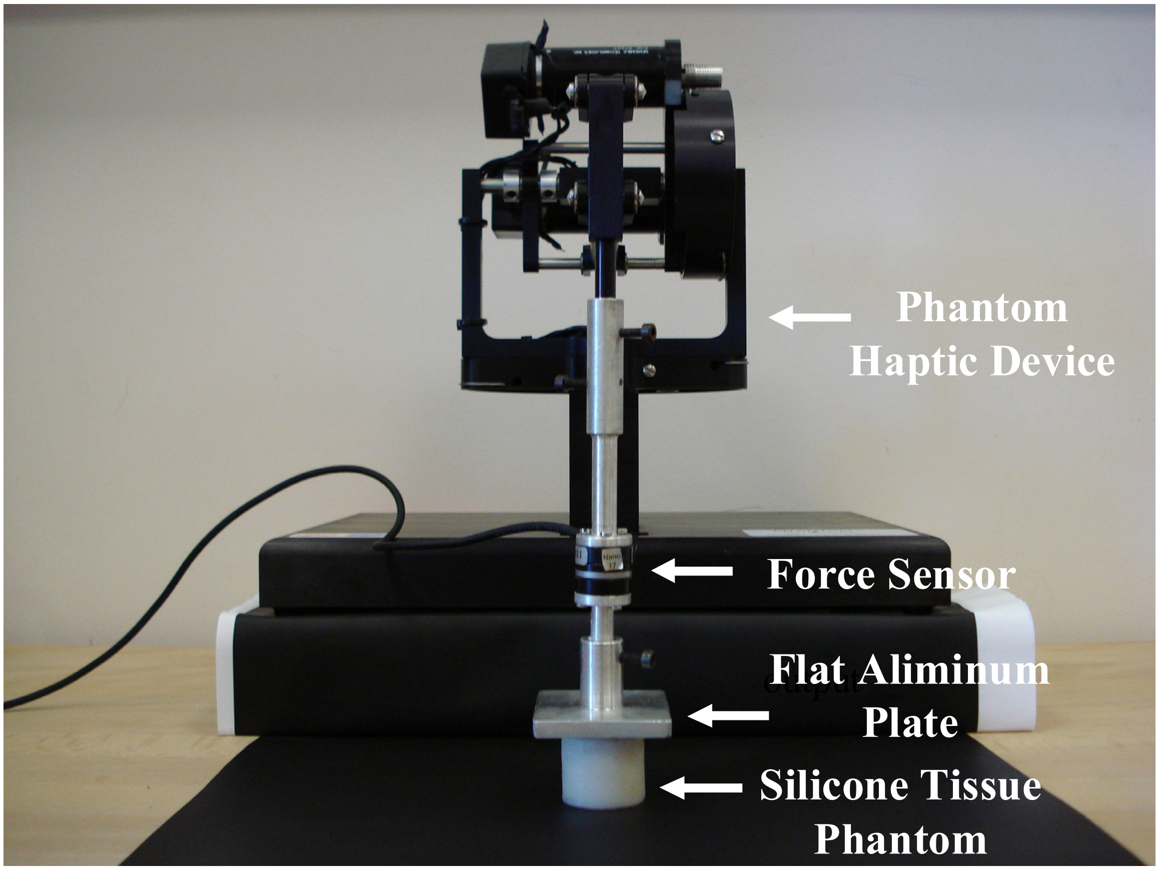


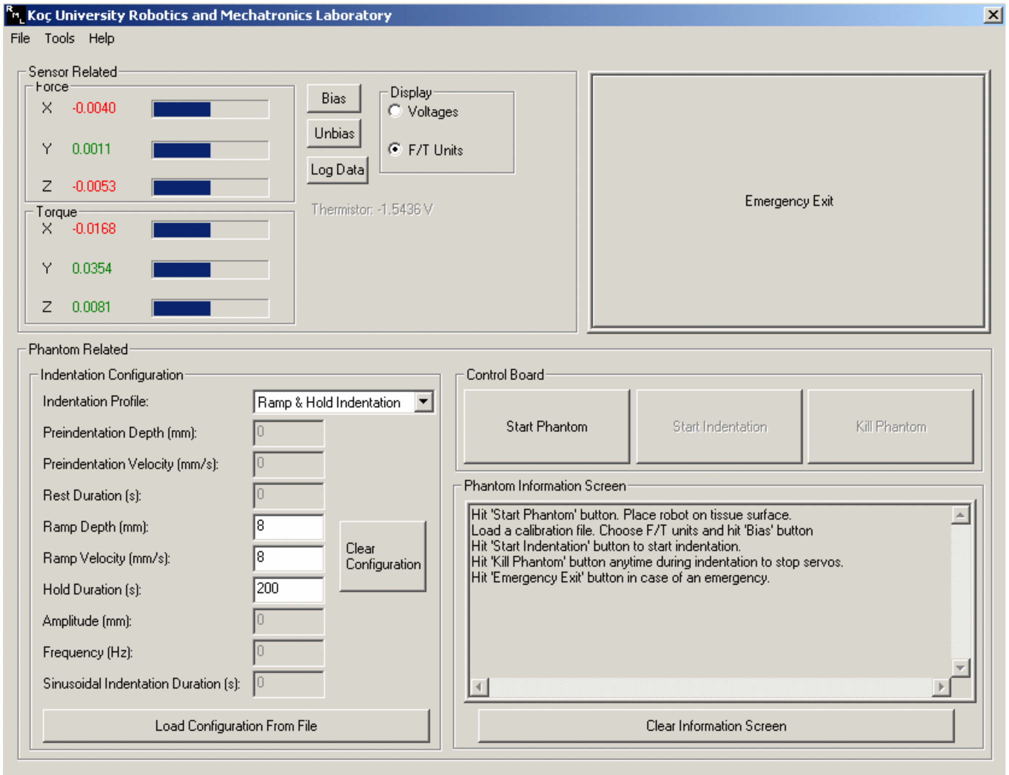
Figure 4. 1: Our robotic indenter and its components.

Using this system, we can make large indentations up to 10 mm. The robotic arm can be programmed to make cyclic indentations or indentations along a user-defined straight path with a given velocity. We used the Phantom Premium haptic device (Model 1.0A from Sensable Technologies Inc.) as our robotic manipulator due to its versatility in position and velocity control in 3D space. The Phantom haptic device has a 13 x 18 x 25 mm workspace and a nominal position resolution of 0.03 mm. The resonant frequencies of the device for x, y and z axes are 90 Hz, 60 Hz and 65 Hz, respectively [56]. A force-torque transducer (Nano 17 from ATI Industrial Automation) [69] is used for the purpose of measuring force response. The Nano 17 has a force range of ± 50 N in the x and y directions and ± 70 N in the z direction and has a resolution of 1/1280 N along each of the three orthogonal axes when attached to a 16-bit A/D converter. Data acquisition unit includes a 16-bit analog input card NI PCI-6034E (National Instruments) [65] with a maximum sampling rate of 200 kS/s.

4.1.3 Controller Design and GUI

A motion control algorithm must be implemented to make indentations on tissue surface. Using the software library of the Phantom haptic device and its position encoders, position of the end-effector point can be acquired in 3D space and appropriate torque commands can be sent to the actuators to control its motion. Using a PID controller, the tip point can be programmed to follow a straight line path with a specified rate. The proper selection of controller gains (i.e. PID tuning) is important for stable output response and the minimization of the steady-state error. We tuned our controller gains on six different material samples having varying softness before the experiments. This process enabled us to construct a look-up table for the feasible set of controller gains to be used in the experiments.

We also developed a graphical user interface (GUI) to record current time, displacement, and force data in a text file following each experiment (Figure 4.2). The GUI was developed in MS Visual C++ environment using an ActiveX component to acquire force values from the force sensor attached to the indenter and the position values from the encoders of the robotic arm at 1 kHz. Using the GUI and the controller, static and dynamic stimuli can be generated easily. Using the GUI, the values of stimulus parameters such as pre-indentation depth, rate of indentation, final indentation depth, and stimulation time can be entered.



Force
X -0.0040
Y 0.0011
Z -0.0053

Torque
X -0.0168
Y 0.0354
Z 0.0081

Display: ○ Voltages ● F/T Units

Thermistor: -1.5436 V

Indentation Profile: Ramp & Hold Indentation
Preindentation Depth (mm): 0
Preindentation Velocity (mm/s): 0
Rest Duration (s): 0
Ramp Depth (mm): 8
Ramp Velocity (mm/s): 8
Hold Duration (s): 200
Amplitude (mm): 0
Frequency (Hz): 0
Sinusoidal Indentation Duration (s): 0

Hit 'Start Phantom' button. Place robot on tissue surface.
Load a calibration file. Choose F/T units and hit 'Bias' button
Hit 'Start Indentation' button to start indentation.
Hit 'Kill Phantom' button anytime during indentation to stop servos.
Hit 'Emergency Exit' button in case of an emergency.

Figure 4. 2: The graphical user interface for generating a stimulus and recording experimental data.

4.2 Characterization Experiments and Experimental Results

4.2.1 Tissue Phantom

The tissue phantom is a circular cylinder with a radius of 12 mm and a height of 24 mm. It is made of Smooth-Sil 910. Smooth-Sil 910 is a two-component silicone rubber (base plus curative; Part A and Part B) and available in hardness range of very soft to medium. It can be cured with either a platinum or a tin catalyst [66]. 30 units of Part A and 1 unit of Part B (in terms of mass) are mixed to obtain a tissue-like softness.

4.2.2 Static Indentation

Nonlinear elastic response of the phantom is characterized via static loading. In the static loading experiment, the phantom is compressed to a predefined depth slowly to eliminate the influence of viscoelastic and inertial effects. The phantom is compressed to 9.76 mm at a rate of 0.1 mm/sec until the servos of robotic indenter warms up and the computer returns an error message of “high temperature error”. Figure 4.3 shows the static compression of the silicon phantom at different stages of loading. The experimental data (Figures 4.4 and 4.5) indicates that the force response of the tissue phantom is nonlinear.

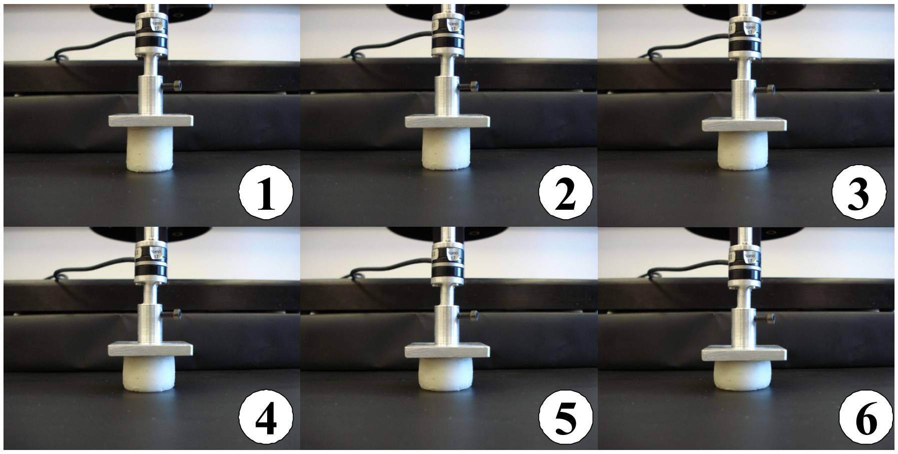


Figure 4. 3: Snapshots from the static loading experiment

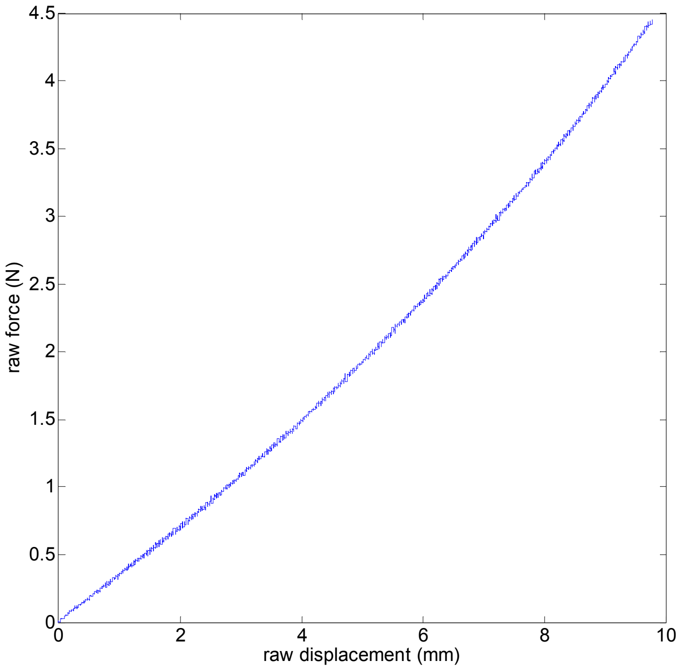


Figure 4. 4: Force-displacement curve obtained from the static compression experiment (raw data)

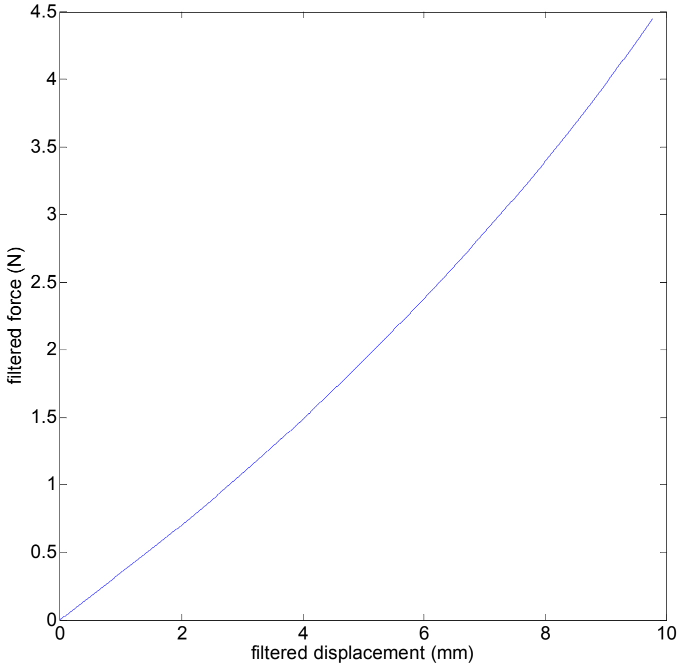


Figure 4. 5: Force-displacement curve obtained from the static compression experiment (filtered data)

4.2.3 Ramp & Hold Indentation

In each of the ramp & hold indentation experiments, the tissue phantom is ramped to the depths of 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, and 7 mm in one second. After the fast indentation to the desired depth, the indenter was held there for 200 seconds to characterize the relaxation response of the tissue phantom. The raw and filtered relaxation responses of the phantom for different ramp depths are shown in Figures 4.6, 4.7, 4.8 and 4.9. Note that, we could not collect data for 200 seconds during the 7 mm experiment. The servos of the Phantom device warmed up, the computer returned an error message of “high temperature error”, and the robotic indenter shutted itself down around 150th seconds. Due to the same reason, ramp & hold experiments requiring a ramp depth larger than 7 mm could not be performed.

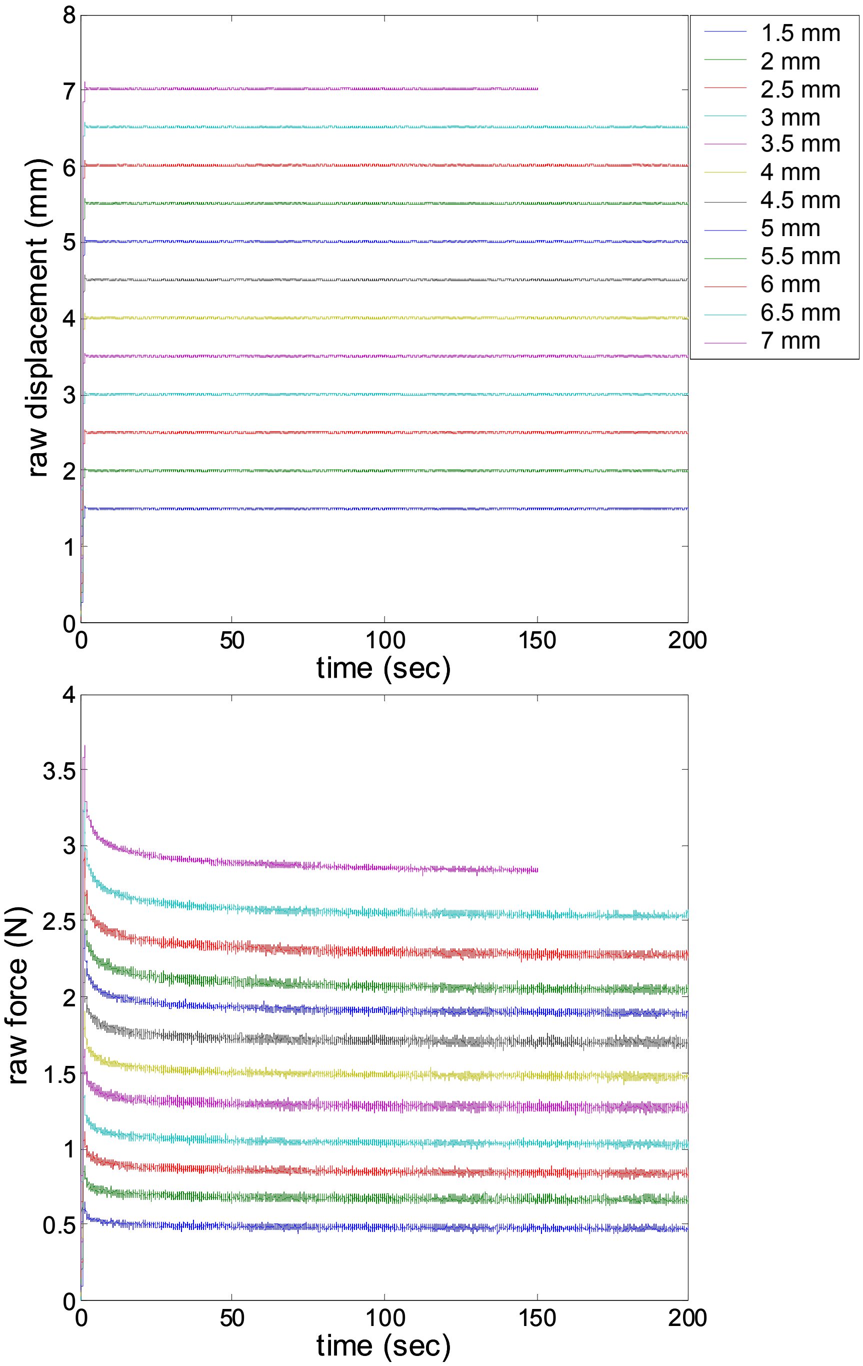


Figure 4. 6: Force-relaxation curves for different ramp depths (raw data)

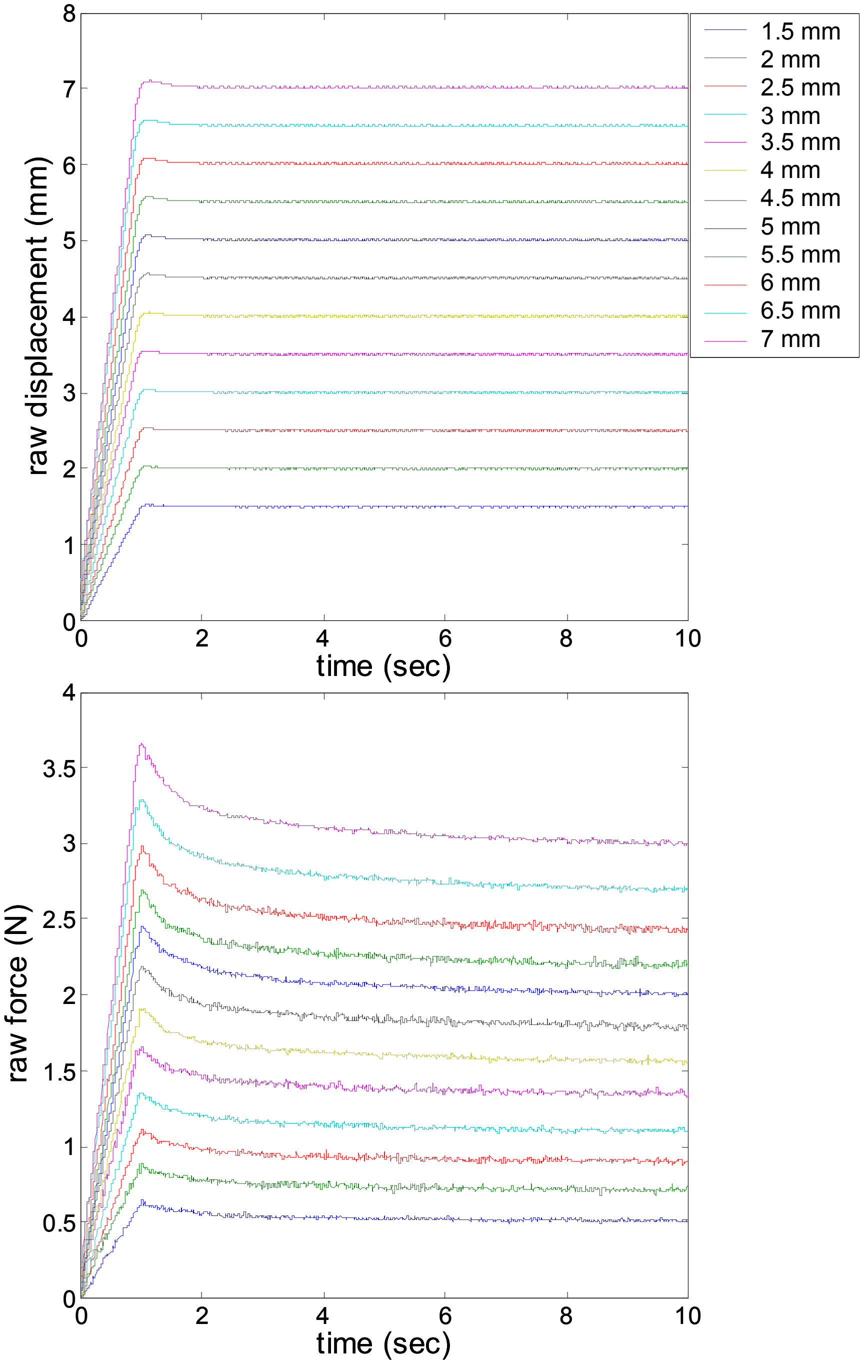


Figure 4. 7: Force-relaxation curves for different ramp depths (raw data, the first 10 seconds)

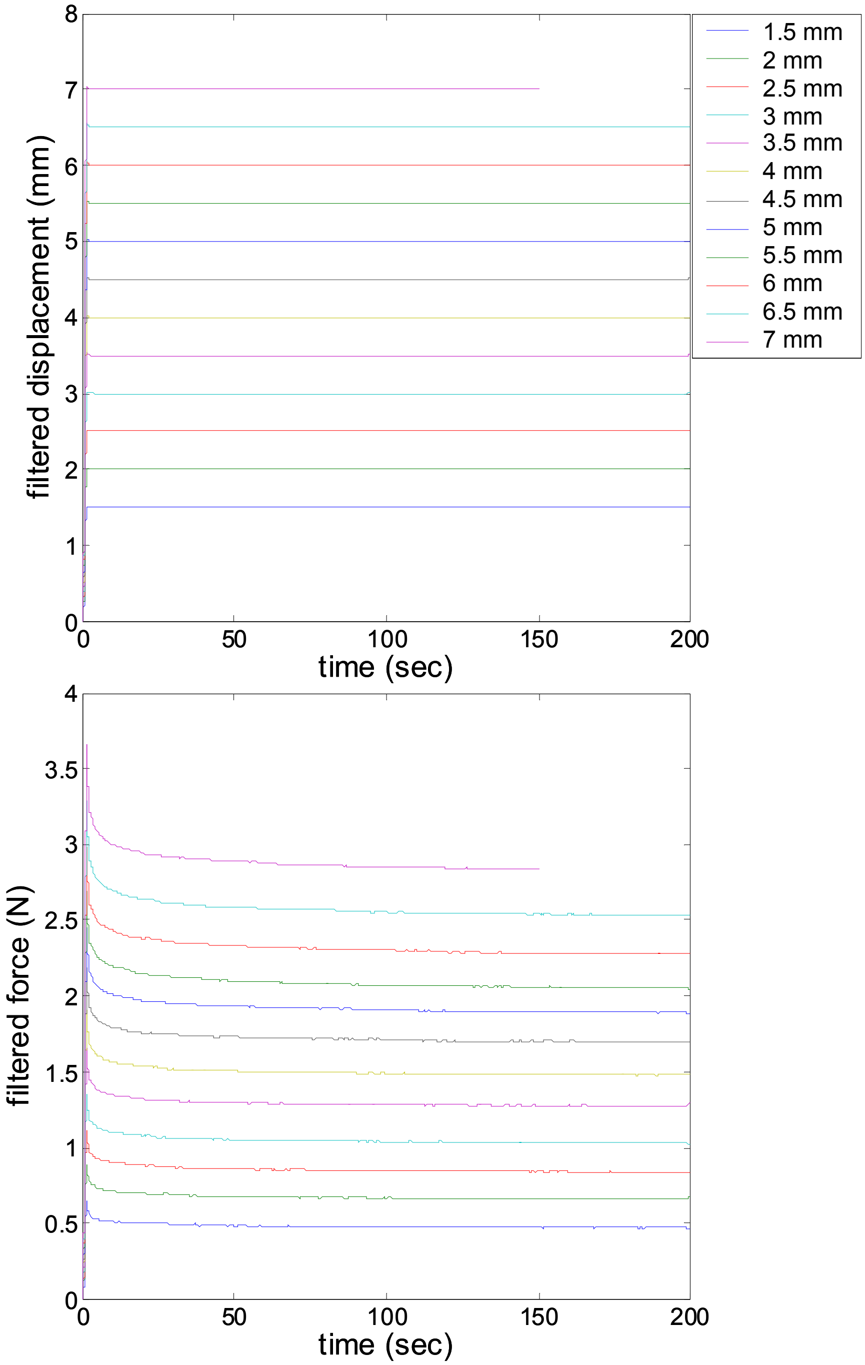


Figure 4. 8: Filtered force-relaxation curves for different ramp depths

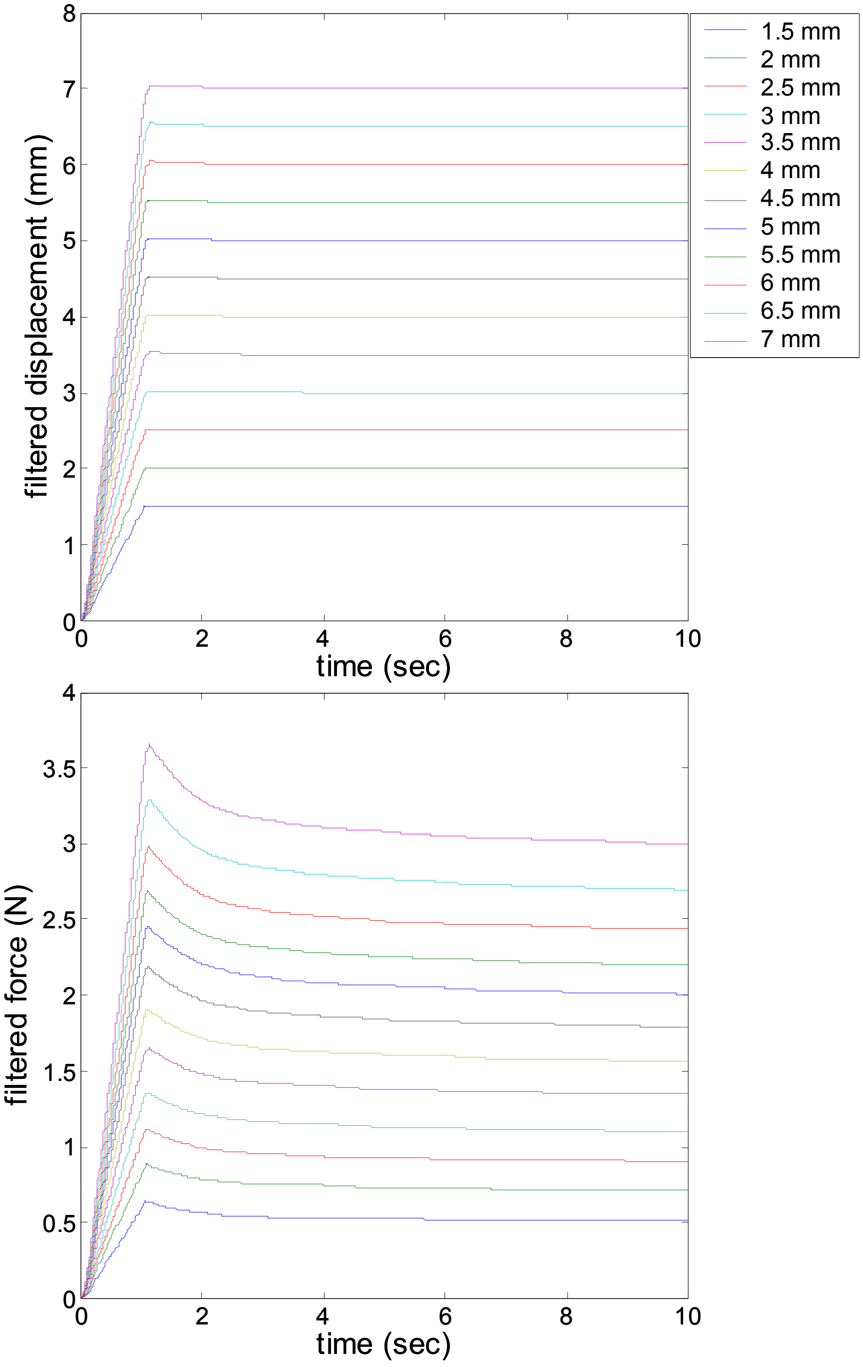


Figure 4. 9: Filtered force-relaxation curve for different ramp depths (the first 10 seconds)

4.3 Characterization

4.3.1 Lumped Nonlinear Viscoelastic Model

The tissue phantom is modeled using a Generalized Maxwell solid by connecting a nonlinear spring in parallel to N linear Maxwell elements (Figure 4.10).

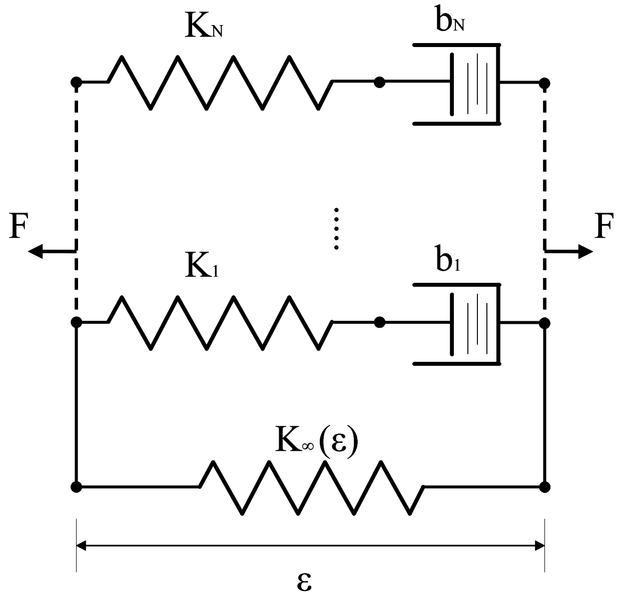


Figure 4. 10: The nonlinear viscoelastic model of the tissue phantom. The model is a Generalized Maxwell solid, which is constructed by connecting a nonlinear spring in parallel to N linear Maxwell elements (note that a spring and a dashpot are connected in series to form one Maxwell element).

The material properties of the phantom are characterized using this lumped model for N=1 and N=2.

4.3.2 Characterization of Nonlinear Elasticity

The nonlinear strain-dependent material properties of the phantom are characterized using its force response under the static loading. Since the phantom is compressed very slowly during the static loading, all time-dependent effects of the Maxwell elements on the force response are void. Hence, the force response of the phantom during static loading is governed by the strain-dependent nonlinear spring only. Therefore, the characterization of nonlinear material properties of the phantom involves the estimation of the strain-dependent stiffness coefficient $K_\infty(\varepsilon)$.

For the nonlinear spring, the relation between the force and the displacement along the direction of static compression can be written as

$$F = K_\infty(\varepsilon)\,\varepsilon \qquad (4.1)$$

The experimental force-displacement response presented in Figure 4.5 can be converted to force-strain representation (Figure 4.11) by dividing the displacement to the original height of the cylindrical phantom, which is 24 mm.

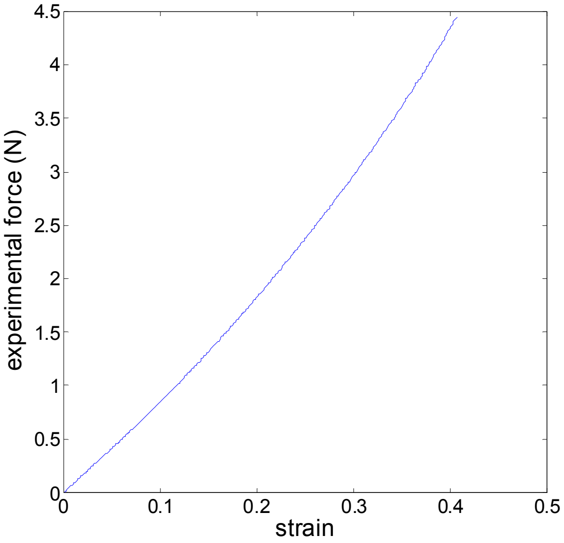


Figure 4. 11: Experimental force-strain curve

Polynomial functions with degrees of 2, 3, 4 and 5 (Figures 4.12, 4.13, 4.14 and 4.15, also see Table 4.1) are fitted to the experimental force-strain data using the “polyfit” function of MATLAB.

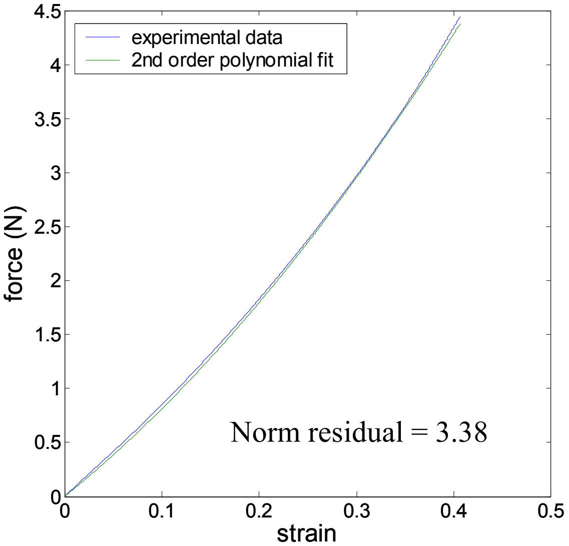


Figure 4. 12: Curve fitting to experimental static indentation data using a 2^{nd} order polynomial

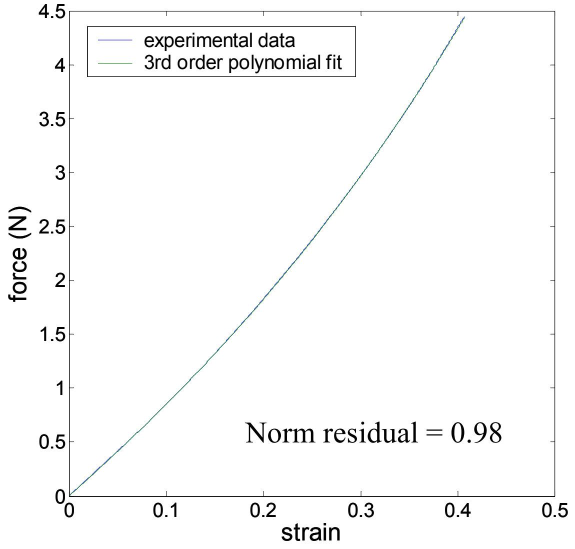


Figure 4. 13: Curve fitting to experimental static indentation data using a 3rd order polynomial

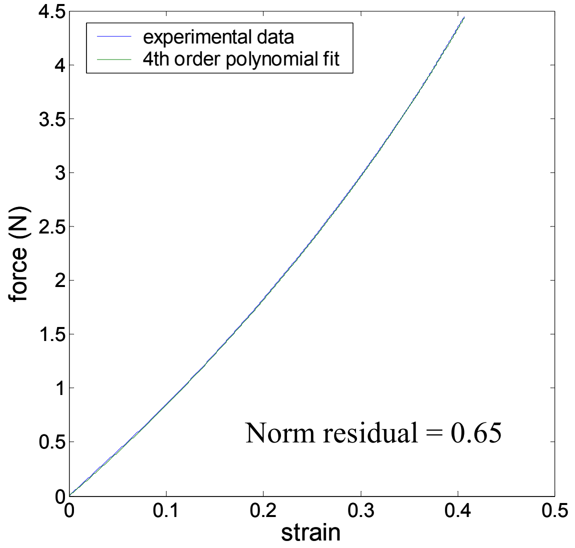


Figure 4. 14: Curve fitting to experimental static indentation data using 4th order polynomial

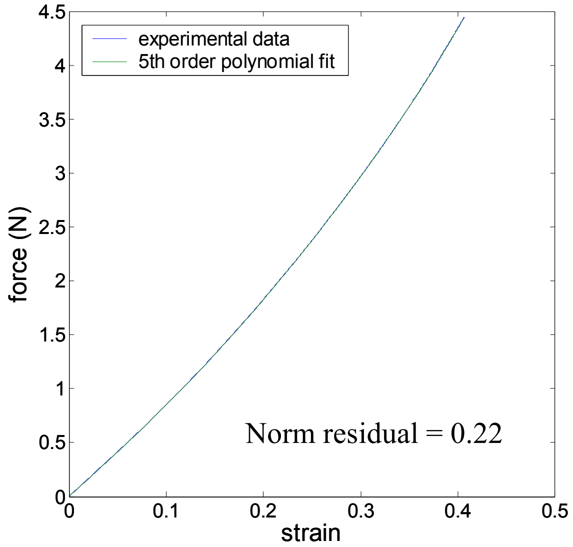


Figure 4. 15: Curve fitting to experimental static indentation data using 5th order polynomial

Degree	Polynomial	Norm residual
2	$F = 8.66\varepsilon^2 + 7.24\varepsilon$	3.38
3	$F = 8.13\varepsilon^3 + 3.69\varepsilon^2 + 8.05\varepsilon$	0.98
4	$F = 18.07\varepsilon^4 - 6.58\varepsilon^3 + 7.54\varepsilon^2 + 7.7\varepsilon$	0.65
5	$F = 146.89\varepsilon^5 - 131.36\varepsilon^4 + 47.47\varepsilon^3 - 0.70\varepsilon^2 + 8.18\varepsilon$	0.22

Table 4. 1: Fitted polynomials of different degrees

Among the polynomials that we have tested, the 5^{th} degree polynomial (Figure 4.15) fits best to the experimental data. The coefficients of this polynomial are

$$F = 146.89\varepsilon^5 - 131.36\varepsilon^4 + 47.47\varepsilon^3 - 0.70\varepsilon^2 + 8.18\varepsilon \qquad (4.2)$$

By dividing Equation 4.2 with strain, the strain-dependent stiffness coefficient of the nonlinear spring is calculated as

$$K_{\infty}\left(\varepsilon\right) = 146.89\varepsilon^{4} - 131.36\varepsilon^{3} + 47.47\varepsilon^{2} - 0.70\varepsilon + 8.18 \tag{4.3}$$

Figure 4.16 shows the variation of K_{∞} as a function of ε.

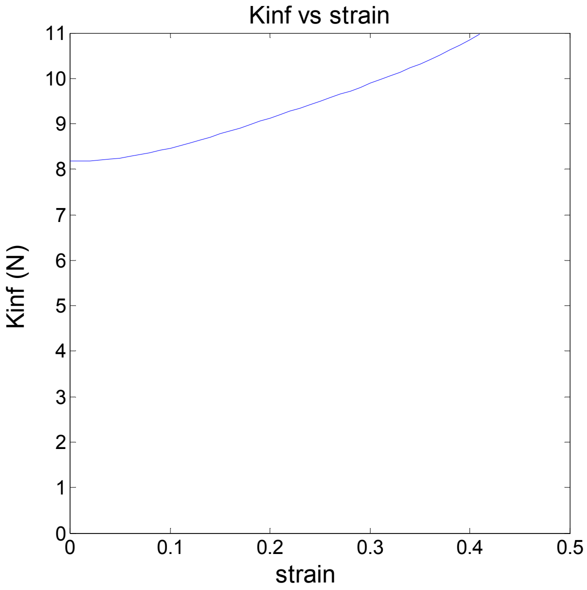


Figure 4. 16: Nonlinear behavior of $K_{\infty}(\varepsilon)$

It is important to emphasize here that the stiffness coefficient $K_{\infty}(\varepsilon)$ given in Equation 4.3 is in units of Newton and independent of the original height of the tissue phantom used in the experiment. Hence, one can simply divide the stiffness coefficient $K_{\infty}(\varepsilon)$ given in Equation 4.3 with the original length of any sample to calculate its stiffness coefficient.

4.3.3 Characterization of Viscoelasticity

The characterization of viscoelastic response of the phantom involves the calculation of the spring and damper coefficients in the Maxwell elements for N=1 and N=2. We calculate these coefficients by curve fitting to the experimental force relaxation data collected via the ramp and hold experiments.

The stress function of a generalized Maxwell Solid for the ramp & hold input of $\varepsilon(t) = A\left(t-(t-c)u(t-c)\right)$ was derived in Section 3.1.2.3 as in Equation 3.56 earlier.

$$\sigma(t) = A\left[z(t) - z(t-c)u(t-c)\right] \tag{3.56}$$

where A is the slope of the ramp, c is the starting time for the hold, *u(t-c)* is the heaviside unit step function shifted by c units to the right in the time domain and *z(t)* was given in Equation 3.55 earlier.

$$z(t) = \mathrm{E}_{\infty}t + \sum_{j=1}^{N} \eta_j\left(1-\exp\left(-t/\tau_j\right)\right) \tag{3.55}$$

where

$$\tau_j = \frac{\eta_j}{\mathrm{E}_j} \tag{3.53}$$

We are interested in the relaxation (hold) component of the Equation 3.56. Therefore, for t > c, the equation becomes

$$\sigma\left(t\right) = A\left[z(t) - z(t-c)\right] \tag{4.4}$$

Plugging *z(t)* into the Equation 4.4 and then rearranging the terms, the following relation for stress is obtained

$$\begin{aligned}&\sigma\left(t\right) = A\left[\mathrm{E}_{\infty}c - \sum_{j=1}^{N}\eta_j \exp\left(-t/\tau_j\right)\left(1-\exp\left(c/\tau_j\right)\right)\right]\\&\tau_j = \frac{\eta_j}{\mathrm{E}_j}\end{aligned} \tag{4.5}$$

Assuming a constant cross sectional area, the elastic moduli E_{∞}, and E_j, and the damping coefficient η_j are replaced by the K_{∞}, Kj and b_j respectively to obtain

$$\begin{aligned}&F\left(t\right) = A\left[\mathrm{K}_{\infty}c - \sum_{j=1}^{N}b_j \exp\left(-t/\tau_j\right)\left(1-\exp\left(c/\tau_j\right)\right)\right]\\&\tau_j = \frac{b_j}{\mathrm{K}_j}\end{aligned} \tag{4.6}$$

After including the nonlinear strain-dependent stiffness coefficient, Equation 4.6 becomes

$$F\left(t\right) = \mathrm{K}_{\infty}\left(Ac\right)Ac - A\sum_{j=1}^{N}b_j \exp\left(-t/\tau_j\right)\left(1-\exp\left(c/\tau_j\right)\right) \tag{4.7}$$

The formulation given in Equation 4.6 is used for curve fitting to the experimental force relaxation data of 12 different ramp & hold experiments. For curve fitting, LSQNONLIN function of MATLAB is used. In addition to estimating the viscoelastic coefficients, we also allow the curve fitting function to estimate the strain-dependent stiffness coefficient K_∞ for each ramp depth. Tables 4.2 and 4.3 list the estimated K_∞ and the viscoelastic coefficients (also, see Figures 4.17 , 4.18 and 4.19).

Ramp depth (mm)	**Ramp depth (strain)**	**K_∞ (N)**	**K_1 (N)**	**b_1 (Ns)**
1.5	0.0625	7.672	1.1217	19.701
2	0.083333	8.0268	1.143	21.2056
2.5	0.104167	8.12448	1.1435	21.6366
3	0.125	8.3152	1.2065	18.3739
3.5	0.145833	8.797714	1.2247	15.3627
4	0.166667	8.9454	1.2338	15.7101
4.5	0.1875	9.129067	1.2878	14.5289
5	0.208333	9.15696	1.2123	18.0238
5.5	0.229167	9.024	1.1983	21.8755
6	0.25	9.1844	1.2189	20.2069
6.5	0.270833	9.430523	1.2838	18.3432
7	0.291667	9.809829	1.3421	15.8973

Table 4. 2: Estimated material coefficients for Maxwell Solid with N=1

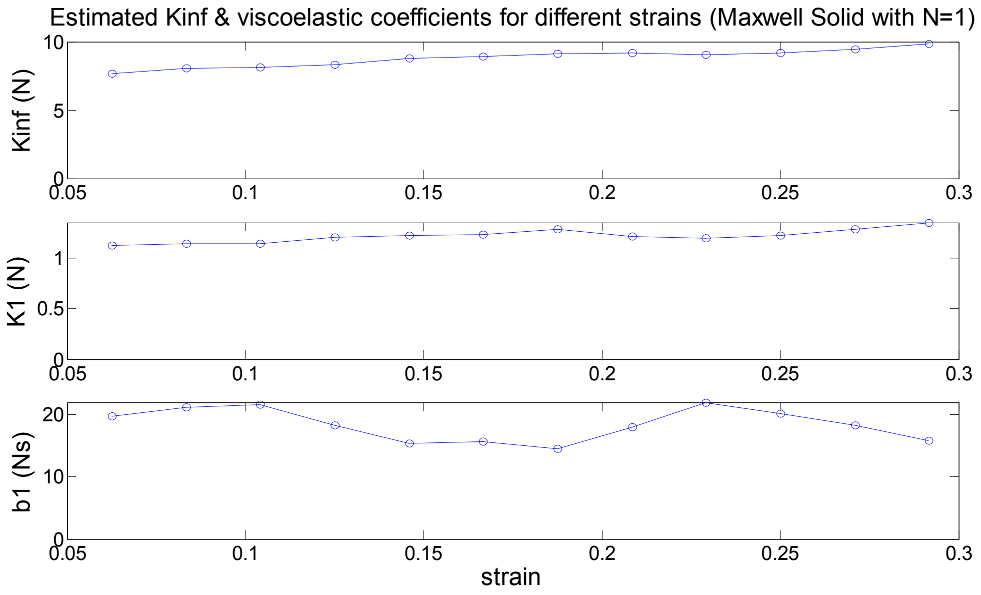


Figure 4. 17: Estimated material coefficients for Maxwell Solid with N=1

Ramp depth (mm)	**Ramp depth (strain)**	**K_∞ (N)**	**K_1 (N)**	**b_1 (Ns)**	**K_2 (N)**	**b_2 (Ns)**
1.5	0.0625	7.608	2.3573	3.8118	0.634	29.8469
2	0.083333	7.9824	2.2852	3.52	0.7026	27.4352
2.5	0.104167	8.06304	2.2897	3.7776	0.6817	31.2822
3	0.125	8.2736	2.276	3.5098	0.682	23.8653
3.5	0.145833	8.756571	2.4121	3.5052	0.6194	20.6645
4	0.166667	8.904	2.3032	3.5462	0.6254	21.0861
4.5	0.1875	9.088533	2.3816	3.5891	0.6125	19.6925
5	0.208333	9.10992	2.4522	3.716	0.6553	24.4958
5.5	0.229167	8.972945	2.378	3.692	0.7268	29.2931
6	0.25	9.1376	2.5257	3.6713	0.7127	26.714
6.5	0.270833	9.389169	2.513	3.6146	0.7123	23.6864
7	0.291667	9.749486	2.624	3.5387	0.7144	22.3128

Table 4. 3: Estimated material coefficients for Maxwell Solid with N=2

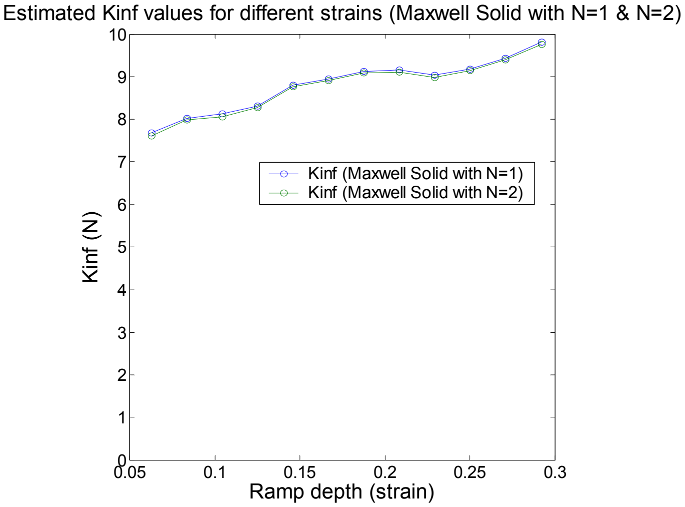


Figure 4. 18: Estimated values of K_∞ for Maxwell Solid with N=1 and N=2

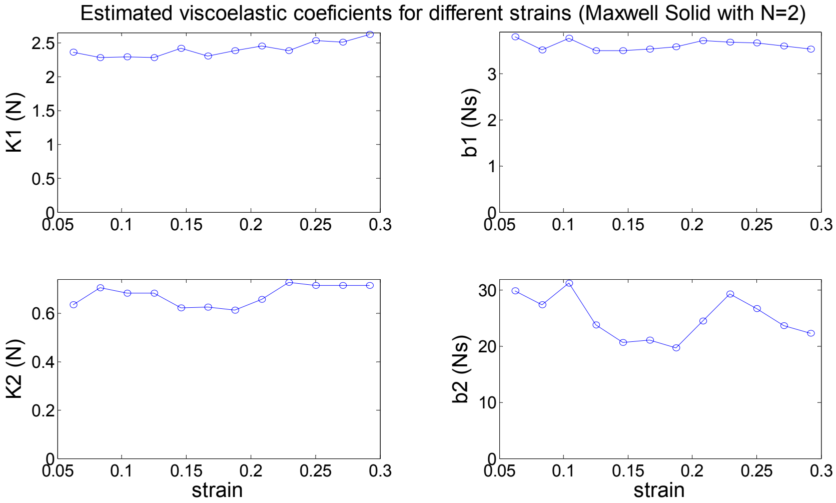


Figure 4. 19: Estimated values of viscoelastic coefficients for Maxwell Solid with N=2

Each relaxation curve, constructed by using the estimated viscoelastic coefficients fits well to the corresponding experimental data at the same depth, but do not necessarily fit well to the experimental data collected at another depth. A procedure is developed to determine a single set of optimum material coefficients that makes the theoretical relaxation function fit reasonably well to all relaxation responses recorded at 12 different depths. The following procedure is developed to estimate the optimum values of the viscoelastic material coefficients for N=1 and N=2.

- The minimum and the maximum values of the viscoelastic coefficients K and b, estimated for 12 different indentation depths through curve-fitting, are determined
- The range for K and b is equally divided into 20 smaller intervals
- All possible combinations of K and b values within the range are inserted into Equation 4.6 along with the corresponding K_∞ value taken from Table 4.2 for N= 1 and Table 4.3 for N = 2 and the total residual is calculated by taking the absolute value of the difference between the experimental and curve-fitted relaxation curves
- The K and b values resulting in the minimum total residual are selected as the optimum viscoelastic coefficients.

The following tables present the optimum viscoelastic coefficients K and b calculated using the above procedure.

K_1 (N)	**b_1 (Ns)**
1.2962987	17.873062

Table 4. 4: Optimum K and b values for N=1

K_1 (N)	**b_1 (Ns)**	**K_2 (N)**	**b_2 (Ns)**
2.5847711	3.6146000	0.68530360	23.964533

Table 4. 5: Optimum K and b values for N=2

And the following figures present the fitted relaxation curves produced by the optimum viscoelastic coefficients for N=1 and N=2. The relaxation curves constructed by using the optimum coefficients fit well to the experimental data of all 12 experiments.

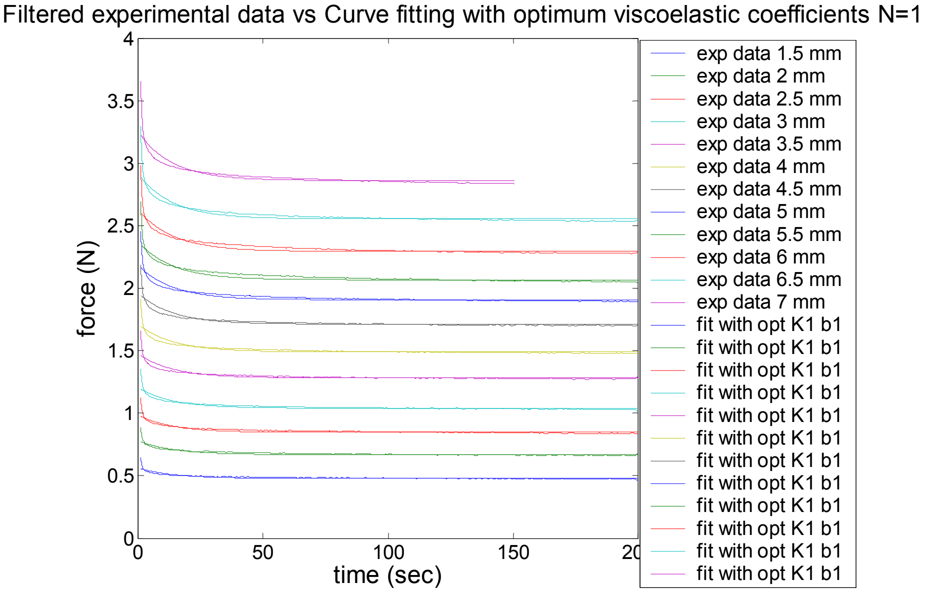


Figure 4. 20: The data collected from the ramp & hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=1

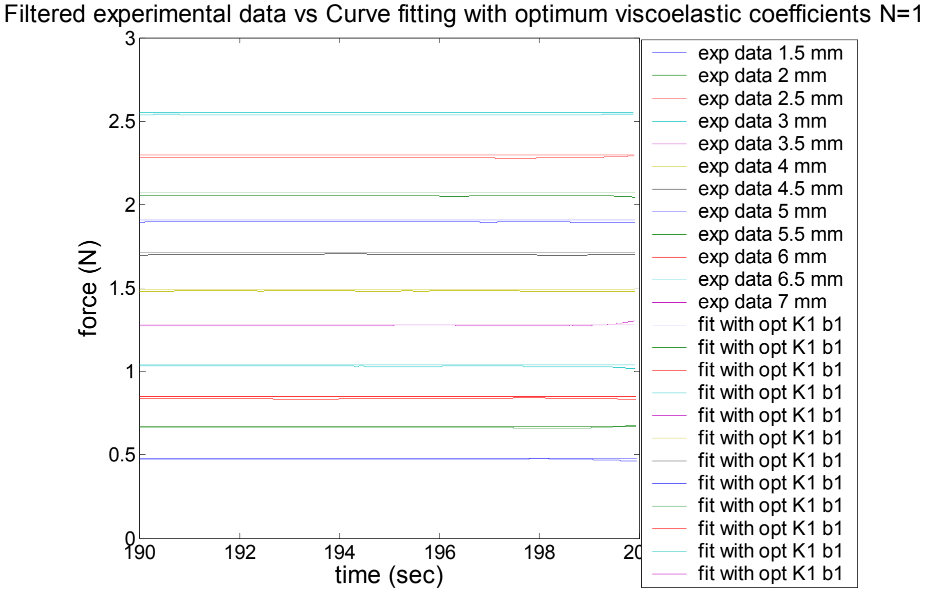


Figure 4. 21: The data collected from the ramp & hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=1 (the last 10 seconds)

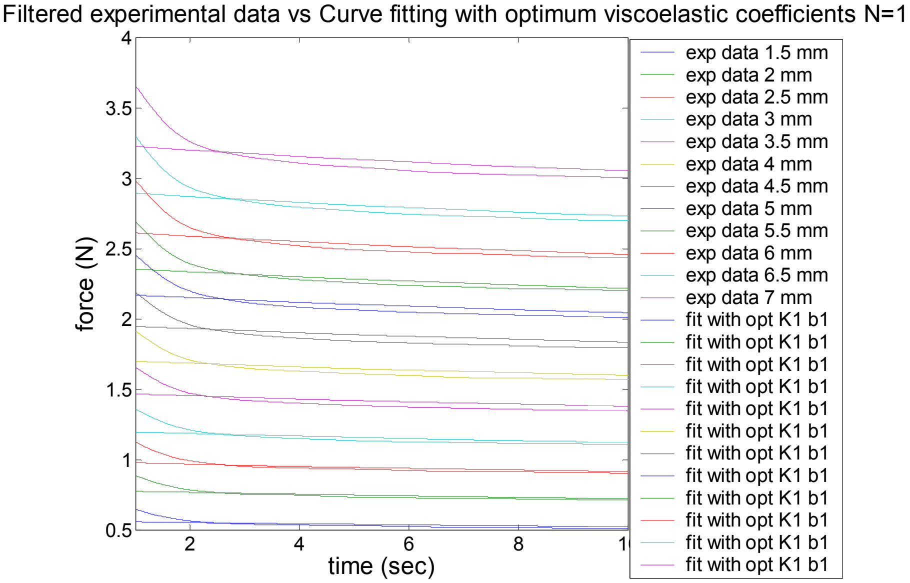


Figure 4. 22: The data collected from the ramp and hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=1 (the first 10 seconds)

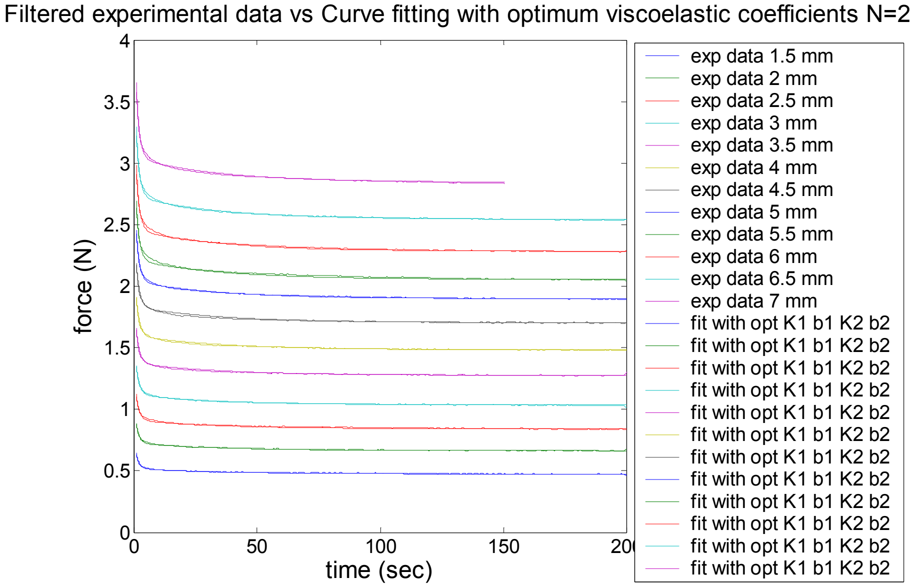


Figure 4. 23: The data collected from the ramp and hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=2

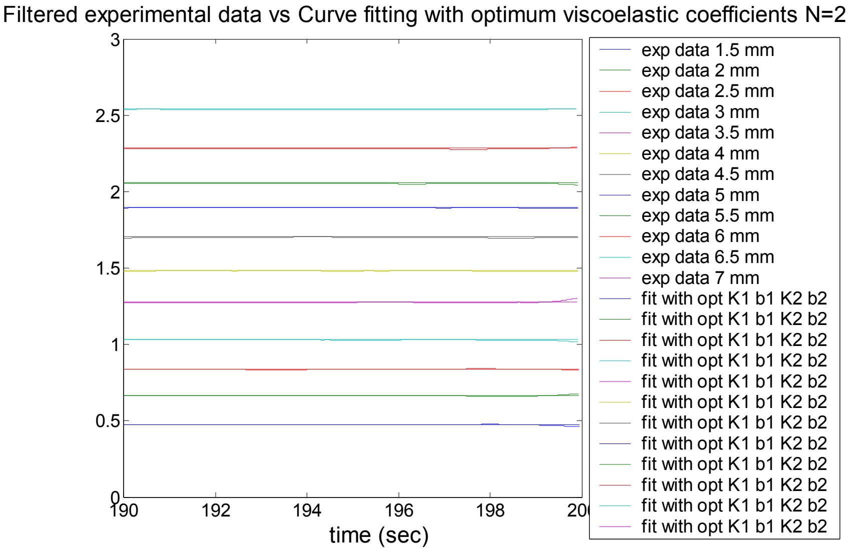


Figure 4. 24: The data collected from the ramp and hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=2 (the last 10 seconds)

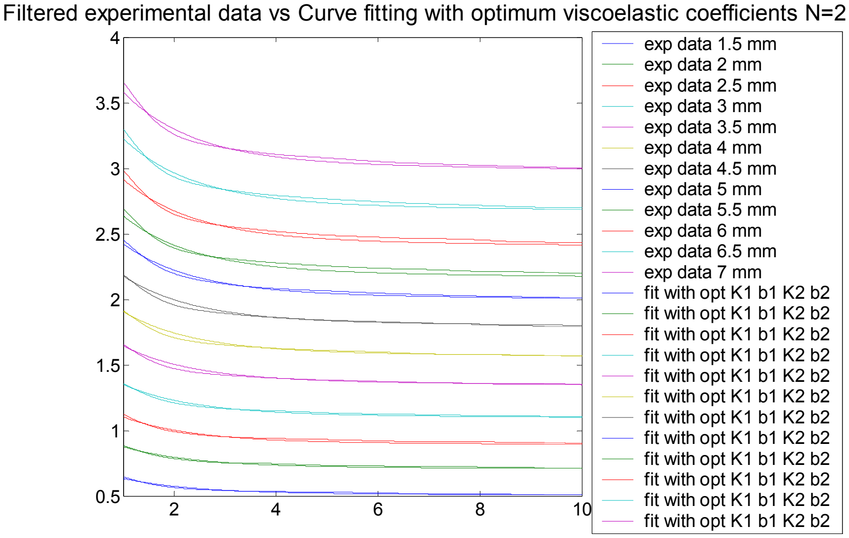


Figure 4. 25: The data collected from the ramp and hold experiments and curve fitting to the experimental data using the optimum viscoelastic coefficients for N=2 (the first 10 seconds)

Figure 4.22 shows that the Generalized Maxwell Solid with N=1 is not able to model the initial relaxation behavior of the viscoelastic phantom well, but N=2 returns acceptable results (see Figure 4.25).

Finally, Figure 4.26 compares the $K_\infty(\varepsilon)$ function derived from the static loading experiments to the similar function derived from the ramp & hold experiments .

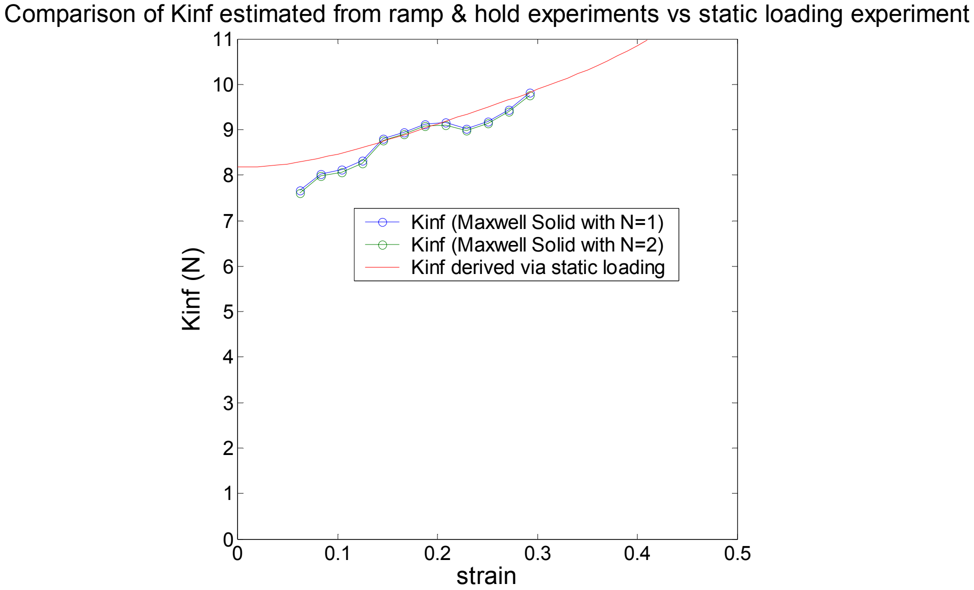


Figure 4. 26: Comparison of $K_\infty(\varepsilon)$ functions derived from the static loading and the ramp & hold experiments

As can be seen from Figure 4.26, the values of K_∞ function derived from the ramp & hold experiments are slightly less than the values of K_∞ function derived from the static loading experiment for the same strain values. While the static compression rate used in the experiments is quite low (0.1 mm/sec), Figure 4.26 suggests that it must be further reduced to eliminate the inertial and viscoelastic effects completely.

Chapter 5

MATERIAL PROPERTY ESTIMATION of a 3-DIMENSIONAL PARTICLE-BASED SYSTEM

This chapter first provides background information on previous studies in literature on parameter estimation techniques developed for particle-based systems. It then explains the development steps and numerical solution of a 3-dimensional (3D) particle based spring-damper network in which the neighboring particles are connected to each other with nonlinear viscoelastic Maxwell Solid elements. The chapter then introduces a set of novel optimization algorithms that are developed to estimate the nonlinear and viscoelastic material coefficients of each element in the network.

5.1 Previous Work on Parameter Estimation Techniques Used in Particle-based Systems

Most of the earlier studies in this area have focused on the estimation of elastic properties of soft objects. In [67], the authors establish a link between discrete mass-spring models and the classical theory of elasticity. By using a finite difference formulation and small deformation assumption, a particle-based model is developed from a continuum model. In [68], a genetic optimization algorithm is used to determine the topology of the mass-spring system. Unlike the previous approaches, the authors consider the topology as a variable and treat the mass distribution and the spring coefficients as the constants of the optimization. The optimization iterations are terminated when the behavior of the topologically altered mass-spring model is sufficiently close to the behavior of a reference mass-spring model in 2D. In [69], this work was extended to 3D. In addition to the topology, spring coefficients are also added to the optimization problem as variables, and the optimization algorithm returns the optimum topology and stiffness coefficients simultaneously. The results of this study suggest that using the same spring constant for all elements in a mesh fail to simulate linear elastic deformations if the sizes of the elements in the mesh are different. In a recent study [70], a 2D mass-spring-damper system is trained to behave like a nonlinear finite element model. The authors use a model-free optimization approach to determine the spring constants. A more advanced approach is taken in [71]. By using a layer of 2D linear springs and dashpots supported by non-linear springs in parallel, the dynamical deformations of the human thigh is simulated. The coefficients of the springs are estimated from the experimental data using a least squares minimization method. To collect the experimental data, the authors measure the force response of the human thigh using a force sensor mounted on a robotic arm. In [72], a method is introduced to derive analytical expressions for the spring coefficients of a particle model from a reference FE model. The main idea is to make the stiffness matrix of the linearized mass-spring system identical to that of the reference FE model in equilibrium. One of the drawbacks of this approach is that these analytical formulas are only valid for specific Poisson ratios.

None of the approaches above attacks the problem of parameter identification for viscoelastic materials. Viscoelasticity is an important characteristic of the soft tissues. Viscoelastic tissues show time-dependent behavior and one way to observe this time-dependent behavior is through the stress relaxation test. Under a constant strain, the response force of the system decreases with time and eventually it converges to a steady-state value. We suggest that a particle-based model can be developed to behave like a nonlinear viscoelastic soft object if the particles are connected to each other using Maxwell

Solids. The following sections present the methods developed to identify of the material coefficients of nonlinear viscoelastic elements in a particle-based system.

5.2 Spring-Damper Network (SDN)

Spring-Damper Network is a network of massless particles (i.e. joints) connected to each other with spring & damper elements in between. Basically, SDN is an ordinary differential equation set where the unknowns are the position of the particles in the network. SDN can be solved using numerical differentiation techniques. Stability and accuracy problems can be solved if an appropriate numerical method and time step for the problem are used and the material coefficients are selected properly. In Section 4.3, we already demonstrated that the nonlinear viscoelastic response the tissue phantom could be modeled using a single Maxwell Solid (i.e., lumped model). In this chapter, we show that a particle network composed of nonlinear viscoelastic Maxwell Solids between the particles can be developed to mimic the behavior of the lumped model.

5.2.1 Numerical Solution of a Spring-Damper Network

The ODE relating stress to strain in a Maxwell Solid with N=1 was given in Equation 3.40 earlier. For a nonlinear viscoelastic Maxwell Solid having a strain-dependent nonlinear $E_\infty(\varepsilon)$ function, Equation 3.40 becomes

$$\sigma + \tau_1 \frac{\partial \sigma}{\partial t} = E_\infty(\varepsilon)\varepsilon + \eta_1 \left(1 + \frac{E_\infty(\varepsilon)}{E_1}\right) \frac{\partial \varepsilon}{\partial t} \tag{5.1}$$

Since the element does not have an area at the tip, the stress in Equation 5.1 can be replaced with the force function as

$$F + \tau_1 \frac{\partial F}{\partial t} = \mathrm{K}_\infty(\varepsilon)\varepsilon + b_1\left(1 + \frac{\mathrm{K}_\infty(\varepsilon)}{\mathrm{K}_1}\right)\frac{\partial \varepsilon}{\partial t} \qquad (5.2)$$

If Equation 5.2 is discretized in time via backward difference, we obtain

$$F^t + \tau_1 \frac{F^t - F^{t-1}}{\Delta t} = \mathrm{K}_\infty(\varepsilon^t)\varepsilon^t + b_1\left(1 + \frac{\mathrm{K}_\infty(\varepsilon^t)}{\mathrm{K}_1}\right)\frac{\varepsilon^t - \varepsilon^{t-1}}{\Delta t} \qquad (5.3)$$

where F^t and ε^t refer to unknown force and strain values of the element at the current time step, and similarly F^{t-1} and ε^{t-1} refer to the known or calculated values in the previous time step. If τ_1 is replaced by its equivalent (b_1 / K_1) in Equation 5.3 and then this equaiton is solved for F^t, the following relation is obtained

$$F^t = \frac{\mathrm{K}_\infty(\varepsilon^t)\varepsilon^t + b_1\left(1 + \frac{\mathrm{K}_\infty(\varepsilon^t)}{\mathrm{K}_1}\right)\frac{\varepsilon^t - \varepsilon^{t-1}}{\Delta t} + F^{t-1}\frac{b_1}{K_1 \Delta t}}{1 + \frac{b_1}{K_1 \Delta t}} \qquad (5.4)$$

Figure 5.1 presents a nonlinear viscoelastic Maxwell Solid model with N=1 between two particles in a SDN.

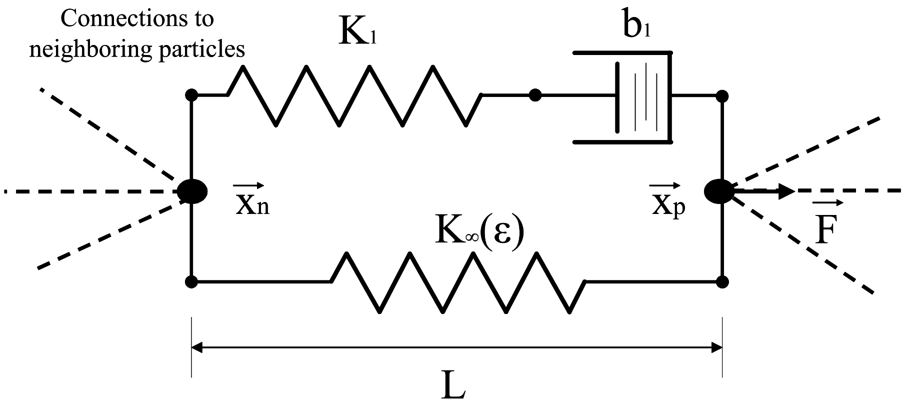


Figure 5. 1: Two arbitrary particles connected to each other with a nonlinear viscoelastic Maxwell Solid with N=1 in a SDN

In this representation, $\mathbf{x_p}$ is the position vector of a particle p, $\mathbf{x_n}$ is the position vector of the neighboring particle n, and *L* is the length of the Maxwell Solid in between. The positions of the particles in the current time step are the unknowns thet we seek to determine.

In the following sections, we derive the differential equations necessary for the numerical solution of a SDN composed of many nonlinear viscoelastic Maxwell Solids such as the one presented in Figure 5.1.

First, the Equation 5.4 can be rewritten as

$$F^t = A + C \tag{5.5}$$

where

$$A = \frac{\mathrm{K}_\infty(\varepsilon^t)\varepsilon^t + b_1\left(1 + \frac{\mathrm{K}_\infty(\varepsilon^t)}{\mathrm{K}_1}\right)\frac{\varepsilon^t - \varepsilon^{t-1}}{\Delta t}}{1 + \frac{b_1}{K_1 \Delta t}}$$

$$C = \frac{\frac{b_1}{K_1 \Delta t}}{1 + \frac{b_1}{K_1 \Delta t}} F^{t-1}$$

(5.6)

ε^t is the current strain of the element at current time step

$$\varepsilon^t = \frac{L^t - L^0}{L^0} = \frac{L^t}{L^0} - 1 \qquad (5.7)$$

where L^t is the length of the element in the current time step and L^0 is the rest length of the element.

$$L^0 = \sqrt{\left(x^0_{px} - x^0_{nx}\right)^2 + \left(x^0_{py} - x^0_{ny}\right)^2 + \left(x^0_{pz} - x^0_{nz}\right)^2} \qquad (5.8)$$

$$L^t = \sqrt{\left(x^t_{px} - x^t_{nx}\right)^2 + \left(x^t_{py} - x^t_{ny}\right)^2 + \left(x^t_{pz} - x^t_{nz}\right)^2} \qquad (5.9)$$

Since we are trying to simulate a SDN in 3D space, Equation 5.5 must be represented in vectorial form. Term C in Equation 5.5 contains the 3D $\boldsymbol{F}^{t-1}$ vector and therefore C can be represented as a vector without any modification. However, term A, is a scalar and can be represented in vectorial form by being projected (via dot product) onto the direction of the Maxwell Solid in space (i.e., $\left(\dfrac{\mathbf{x_p^t} - \mathbf{x_n^t}}{L^t}\right)$). Therefore, Equation 5.5 can be represented in 3D vectorial form as

$$\mathbf{F^t} = A \odot \mathbf{B} + \mathbf{C} \tag{5.10}$$

where

$$A = \frac{\mathrm{K}_\infty(\varepsilon^t)\varepsilon^t + b_1\left(1 + \dfrac{\mathrm{K}_\infty(\varepsilon^t)}{\mathrm{K}_1}\right)\dfrac{\varepsilon^t - \varepsilon^{t-1}}{\Delta t}}{1 + \dfrac{b_1}{K_1 \Delta t}}$$

$$\mathbf{B} = \frac{\mathbf{x_p^t} - \mathbf{x_n^t}}{L^t}$$

$$\mathbf{C} = \frac{\dfrac{b_1}{K_1 \Delta t}}{1 + \dfrac{b_1}{K_1 \Delta t}} \mathbf{F^{t-1}}$$

(5.11)

The Equation 5.10 can be reorganized as

$$\mathbf{F^t} - \left(A \odot \mathbf{B} + \mathbf{C}\right) = \mathbf{0} \tag{5.12}$$

At each time step of the numerical solutions, the vectorial summation of all the forces acting on a particle should be equal to zero. Equation 5.12 actually indicates that the sum of external $\mathbf{F^t}$ vector acting on particle p and the internal forces acting on the same particle is zero. If we decompose Equation 5.12 into x, y, and z degree of freedoms (dof), then we have the following set of equations in each dof:

$$\begin{aligned} I_x &= F_x^t - \left(AB_x + C_x\right) \\ I_y &= F_y^t - \left(AB_y + C_y\right) \\ I_z &= F_z^t - \left(AB_z + C_z\right) \end{aligned} \tag{5.13}$$

where

$$\begin{aligned} A &= \frac{\mathrm{K}_\infty(\varepsilon^t)\varepsilon^t + b_1\left(1 + \dfrac{\mathrm{K}_\infty(\varepsilon^t)}{\mathrm{K}_1}\right)\dfrac{\varepsilon^t - \varepsilon^{t-1}}{\Delta t}}{1 + \dfrac{b_1}{K_1\Delta t}} \\ B_x &= \frac{x_{px}^t - x_{nx}^t}{L^t}, B_y = \frac{x_{py}^t - x_{ny}^t}{L^t}, B_z = \frac{x_{pz}^t - x_{nz}^t}{L^t} \\ C_x &= \frac{\dfrac{b_1}{K_1\Delta t}}{1 + \dfrac{b_1}{K_1\Delta t}} F_x^{t-1}, C_y = \frac{\dfrac{b_1}{K_1\Delta t}}{1 + \dfrac{b_1}{K_1\Delta t}} F_y^{t-1}, C_z = \frac{\dfrac{b_1}{K_1\Delta t}}{1 + \dfrac{b_1}{K_1\Delta t}} F_z^{t-1} \end{aligned} \tag{5.14}$$

Equation 5.13 explains the force relation of the particle p with one of its neighboring particle n only. To develop the same relation between all m particles and all of their neighbors in a SDN, Equation 5.13 can be generalized as follows:

particle p1

$$F_x^t \text{ - } \left(AB_x + C_x\right)_{\text{force coming from neighbor 1}} - \left(AB_x + C_x\right)_{\text{force coming from neighbor 2}} - \ldots = 0$$

$$F_y^t \text{ - } \left(AB_y + C_y\right)_{\text{force coming from neighbor 1}} - \left(AB_y + C_y\right)_{\text{force coming from neighbor 2}} - \ldots = 0$$

$$F_z^t \text{ - } \left(AB_z + C_z\right)_{\text{force coming from neighbor 1}} - \left(AB_z + C_z\right)_{\text{force coming from neighbor 2}} - \ldots = 0$$

particle p2 (5.15)

...

particle pm

...

Using the equation set given in Equation 5.15, the unknown positions of the particles **x** at the current time step t can be computed.

$$\mathbf{x} = \begin{bmatrix} x_{1x}^t \\ x_{1y}^t \\ x_{1z}^t \\ x_{2x}^t \\ x_{2y}^t \\ x_{2z}^t \\ \ldots \\ x_{mx}^t \\ x_{my}^t \\ x_{mz}^t \end{bmatrix} \qquad (5.16)$$

The unknown positions of the particles in Equation 5.15 form a nonlinear set of equations and it can be solved using Newton's Method. To apply Newton's Method to Equation 5.15, we need to take the derivative of this equation with respect to (wrt) the unknown position vector **x** (Equation 5.16), which leads to the Jacobian matrix. Jacobian matrix of Equation 5.15 is a square matrix having the dimensions of N_{dof} x N_{dof}. N_{dof} is the total number of dofs in the system and is equal to 3 x m for a SDN having m particles. Each row of the Jacobian matrix corresponds to the equation of motion of a particle along one positional dof and each column corresponds to an unknown positional dof. If the Equation 5.15 contains a term related to a particular dof, then the corresponding cell in the Jacobian matrix is the derivative of the equation wrt that dof. Otherwise, it is zero.

The following set of equations is the derivatives of the equations given in Equation 5.13

$$\frac{\partial I_x}{\partial x^t_{px}} = -\left(\frac{\partial A}{\partial x^t_{px}} B_x + A\frac{\partial B_x}{\partial x^t_{px}}\right), \frac{\partial I_y}{\partial x^t_{px}} = -\left(\frac{\partial A}{\partial x^t_{px}} B_y + A\frac{\partial B_y}{\partial x^t_{px}}\right), \frac{\partial I_z}{\partial x^t_{px}} = -\left(\frac{\partial A}{\partial x^t_{px}} B_z + A\frac{\partial B_z}{\partial x^t_{px}}\right)$$

$$\frac{\partial I_x}{\partial x^t_{py}} = -\left(\frac{\partial A}{\partial x^t_{py}} B_x + A\frac{\partial B_x}{\partial x^t_{py}}\right), \frac{\partial I_y}{\partial x^t_{py}} = -\left(\frac{\partial A}{\partial x^t_{py}} B_y + A\frac{\partial B_y}{\partial x^t_{py}}\right), \frac{\partial I_z}{\partial x^t_{py}} = -\left(\frac{\partial A}{\partial x^t_{py}} B_z + A\frac{\partial B_z}{\partial x^t_{py}}\right)$$

$$\frac{\partial I_x}{\partial x^t_{pz}} = -\left(\frac{\partial A}{\partial x^t_{pz}} B_x + A\frac{\partial B_x}{\partial x^t_{pz}}\right), \frac{\partial I_y}{\partial x^t_{pz}} = -\left(\frac{\partial A}{\partial x^t_{pz}} B_y + A\frac{\partial B_y}{\partial x^t_{pz}}\right), \frac{\partial I_z}{\partial x^t_{pz}} = -\left(\frac{\partial A}{\partial x^t_{pz}} B_z + A\frac{\partial B_z}{\partial x^t_{pz}}\right)$$

$$\frac{\partial I_x}{\partial x^t_{nx}} = -\left(\frac{\partial A}{\partial x^t_{nx}} B_x + A\frac{\partial B_x}{\partial x^t_{nx}}\right), \frac{\partial I_y}{\partial x^t_{nx}} = -\left(\frac{\partial A}{\partial x^t_{nx}} B_y + A\frac{\partial B_y}{\partial x^t_{nx}}\right), \frac{\partial I_z}{\partial x^t_{nx}} = -\left(\frac{\partial A}{\partial x^t_{nx}} B_z + A\frac{\partial B_z}{\partial x^t_{nx}}\right)$$

$$\frac{\partial I_x}{\partial x^t_{ny}} = -\left(\frac{\partial A}{\partial x^t_{ny}} B_x + A\frac{\partial B_x}{\partial x^t_{ny}}\right), \frac{\partial I_y}{\partial x^t_{ny}} = -\left(\frac{\partial A}{\partial x^t_{ny}} B_y + A\frac{\partial B_y}{\partial x^t_{ny}}\right), \frac{\partial I_z}{\partial x^t_{ny}} = -\left(\frac{\partial A}{\partial x^t_{ny}} B_z + A\frac{\partial B_z}{\partial x^t_{ny}}\right)$$

$$\frac{\partial I_x}{\partial x^t_{nz}} = -\left(\frac{\partial A}{\partial x^t_{nz}} B_x + A\frac{\partial B_x}{\partial x^t_{nz}}\right), \frac{\partial I_y}{\partial x^t_{nz}} = -\left(\frac{\partial A}{\partial x^t_{nz}} B_y + A\frac{\partial B_y}{\partial x^t_{nz}}\right), \frac{\partial I_z}{\partial x^t_{nz}} = -\left(\frac{\partial A}{\partial x^t_{nz}} B_z + A\frac{\partial B_z}{\partial x^t_{nz}}\right)$$

(5.17)

$$\begin{aligned}
\frac{\partial A}{\partial x_{px}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{px}^{t}} \\
\frac{\partial A}{\partial x_{py}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{py}^{t}} \\
\frac{\partial A}{\partial x_{pz}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{pz}^{t}}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial A}{\partial x_{nx}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{nx}^{t}} \\
\frac{\partial A}{\partial x_{ny}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{ny}^{t}} \\
\frac{\partial A}{\partial x_{nz}^{t}} &= \left(\frac{1}{1+\dfrac{b_1}{K_1 \Delta t}} \right) \left(\frac{K_\infty(\varepsilon^t)}{L^0} + b_1 \left(1 + \frac{K_\infty(\varepsilon^t)}{K_1} \right) \left(\frac{1}{L^0 \Delta t} \right) \right) \frac{\partial L^t}{\partial x_{nz}^{t}}
\end{aligned} \tag{5.18}$$

$$\begin{gathered}
\frac{\partial B_x}{\partial x_{px}^t}=\frac{1}{L^t}-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{px}^t},\frac{\partial B_x}{\partial x_{py}^t}=-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{py}^t},\frac{\partial B_x}{\partial x_{pz}^t}=-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{pz}^t}\\
\frac{\partial B_x}{\partial x_{nx}^t}=-\frac{1}{L^t}-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nx}^t},\frac{\partial B_x}{\partial x_{ny}^t}=-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{ny}^t},\frac{\partial B_x}{\partial x_{nz}^t}=-\frac{\left(x_{px}^t-x_{nx}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nz}^t}\\
\\
\frac{\partial B_y}{\partial x_{px}^t}=-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{px}^t},\frac{\partial B_y}{\partial x_{py}^t}=\frac{1}{L^t}-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{py}^t},\frac{\partial B_y}{\partial x_{pz}^t}=-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{pz}^t}\\
\frac{\partial B_y}{\partial x_{nx}^t}=-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nx}^t},\frac{\partial B_y}{\partial x_{ny}^t}=-\frac{1}{L^t}-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{ny}^t},\frac{\partial B_y}{\partial x_{nz}^t}=-\frac{\left(x_{py}^t-x_{ny}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nz}^t}\\
\\
\frac{\partial B_z}{\partial x_{px}^t}=-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{px}^t},\frac{\partial B_z}{\partial x_{py}^t}=-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{py}^t},\frac{\partial B_z}{\partial x_{pz}^t}=\frac{1}{L^t}-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{pz}^t}\\
\frac{\partial B_z}{\partial x_{nx}^t}=-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nx}^t},\frac{\partial B_z}{\partial x_{ny}^t}=-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{ny}^t},\frac{\partial B_z}{\partial x_{nz}^t}=-\frac{1}{L^t}-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{\left(L^t\right)^2}\frac{\partial L^t}{\partial x_{nz}^t}
\end{gathered}\tag{5.19}$$

$$\begin{gathered}
\frac{\partial L^t}{\partial x_{px}^t}=\frac{\left(x_{px}^t-x_{nx}^t\right)}{L^t},\frac{\partial L^t}{\partial x_{py}^t}=\frac{\left(x_{py}^t-x_{ny}^t\right)}{L^t},\frac{\partial L^t}{\partial x_{pz}^t}=\frac{\left(x_{pz}^t-x_{nz}^t\right)}{L^t}\\
\frac{\partial L^t}{\partial x_{nx}^t}=-\frac{\left(x_{px}^t-x_{nx}^t\right)}{L^t},\frac{\partial L^t}{\partial x_{ny}^t}=-\frac{\left(x_{py}^t-x_{ny}^t\right)}{L^t},\frac{\partial L^t}{\partial x_{nz}^t}=-\frac{\left(x_{pz}^t-x_{nz}^t\right)}{L^t}
\end{gathered}\tag{5.20}$$

To solve the nonlinear equation given in Equation 5.15, the Newton's Method can be used, but this method does not guarantee convergence unless the initial guess is sufficiently close to the correct solution. On the other hand, the Newton's Method with Line Search combines the rapid local convergence of the Newton's method with a globally convergent strategy that guarantees some progress towards the correct solution at each iteration [73]. The computational complexity of the Newton's Method with Line Search does not allow real-time performance even for a particle system having rather small number of particles Since the optimization computations are performed off-line, this is not a concern.

The numerical solution of the SDN for the cubic mesh is developed in C++ using the Newton's Method with Line Search algorithm supplied by Numerical Recipes [73]. Our C++ implementation is validated by comparing our solution with the numerical solution returned by ANSYS finite element package. It is observed that the results are identical.

The derivation of the numerical solution of an SDN composed of Voight Models or Maxwell Solids with N=2 is very similar to the given derivation for Maxwell Solid with N=1. The force function given in Equation 5.4 should be changed according to the characteristic constitutive equation of the element used in SDN.

5.3 Estimation of the Nonlinear Strain-dependent Stiffness Coefficient and the Viscoelastic Material Coefficients of the Maxwell Solids in a SDN

This section presents a set of novel optimization algorithms developed for estimating the nonlinear strain-dependent stiffness coefficient, and the viscoelastic material coefficients of a SDN composed of Maxwell Solids with N=1.

As a proof of concept, a 27-node (3x3x3) 3D cubic mesh having an edge-length of 24 mm is formed (see Figure 5.2). The particles of the mesh are connected to each other based on Moore-neighborhood. Each line in Figure 5.2 corresponds to a Maxwell solid with N = 1 between two neighboring particles. In Moore-Neighborhood, a particle can have up to 26 neighbors (see the blue particle in Figure 5.2). We observed that the SDN model of the tissue phantom shows a more realistic deformation behavior when Moore-

Neighborhood is used. Since the mesh has diagonal elements in addition to the elements along the principal axes in Moore-neighborhood, the bulging effect (i.e. bending outward) can be simulated successfully when the model is compressed from the top just like the bulging effect observed when the silicon phantom is compressed using the robotic indenter (see Figure 4.3).

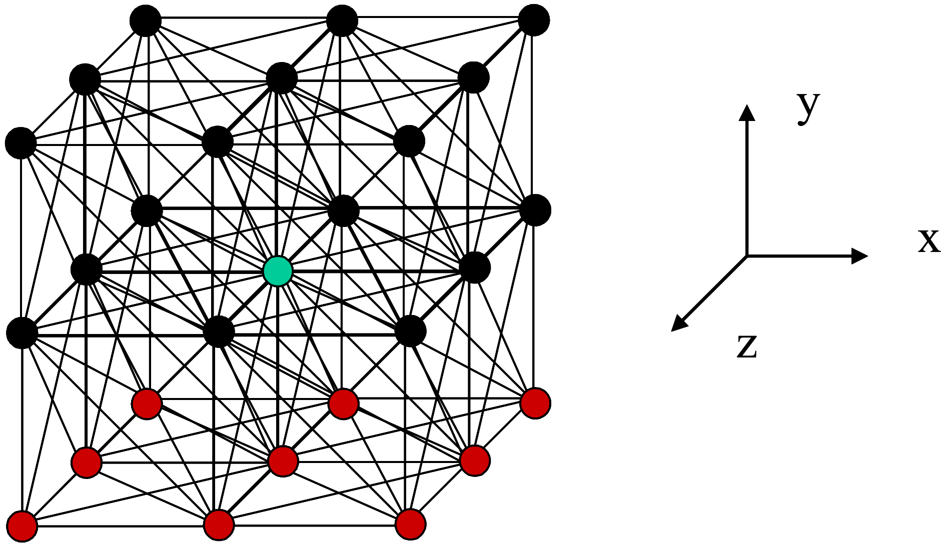


Figure 5. 2: A 27-node 3D cubic mesh is constructed based on Moore-neighborhood. The blue colored particle has 26 neighbors. For the simulations, the red colored particles are fixed to the ground while the remaining ones are allowed to move.

A set of optimization algorithms is developed to estimate the material coefficients of the elements in the 27-node mesh and also in a 125-node cubic mesh.

5.3.1 An Optimization Algorithm for Estimating the Nonlinear Strain-dependent Behavior of the Elements in a Spring-Damper Network (SDN)

The optimization algorithm explained in this section estimates the $K_\infty(\varepsilon)$ coefficient, such that the response of the cubic mesh is sufficiently close to that of the lumped model under the same static loading.

For the implementation, the particles on the bottom surface of the cubic mesh are fixed in all dofs (see Figure 5.2). The particles on the top surface of the mesh are also fixed in x- and z-dofs, but allowed to move in the y direction during the static loading. The cubic mesh is compressed to 9.7 mm depth ($\varepsilon = 0.4$) along the –y direction at a rate of 0.1 mm/sec (10^{-4} mm/millisecond) as the way the silicone sample is compressed by the robotic indenter. Note that it takes 97000 compression steps to reach 9.7 mm depth. The time step used for the numerical solution of the SDN is set to 1 millisecond and the force response of the mesh to static loading is calculated at each time step. The force response of the cubic mesh along the +y direction is the sum of the response forces along the +y direction of all the particles on the top surface.

During the static loading, the influence of the time-dependent viscoelastic effects on the force response is negligible since the rate of loading is very slow. Hence, the influence of the Maxwell Model used in the generalized Maxwell Solid on the force response is insignificant and it does not matter what type of viscoelastic model is used during static loading for the estimation of stiffness coefficient $K_\infty(\varepsilon)$. For computational efficiency, the cubic mesh is constructed from nonlinear Voight Models instead of nonlinear viscoelastic Maxwell Solids. The Voight Model reaches to the steady-state immediately after a constant initial strain is applied whereas the Maxwell Solid of any kind require some time to reach the steady-state (Section 3.1.2.3). At each time step of the static loading, the mesh is compressed by 10^{-4} mm and the load is applied long enough to eliminate the influence of viscoelastic effects. Although there is no difference between the magnitudes of steady-state force values of the Voight Model and the Maxwell Solid to a unit strain, (note that the steady-state response depends on the value of $K_{\infty}(\varepsilon)$ only), the amount of time needed to complete the static loading simulation is much shorter when a mesh composed of Voight Models is used.

During the optimization, we seek for the best solution of strain-dependent elastic modulus and set the value of b in the Voight Model to a small value such that the numerical solution of SDN does not become unstable under the static loading. The optimization algorithm seeks to determine the optimum coefficients of the polynomial $K_{\infty}(\varepsilon)_{ELEMENT}$ such that the force response of the cubic mesh, F_{SDN}, is sufficiently close to the force response of the lumped model, F_{LUMPED}, under the same static loading (see Equation 4.3). The algorithm first evaluates $F_{LUMPED}(\varepsilon)$ at 97000 different strain values varying from zero strain to 0.4, and records the data set as $F_{REFERENCE}$. It also takes a copy of $F_{REFERENCE}$ into F_{GUESS}. It then sets the $K_{\infty}(\varepsilon)_{ELEMENT}$ to the $K_{\infty}(\varepsilon)_{LUMPED}$ as an initial guess for the optimizations. Then, the cubic mesh is compressed to 9.7 mm depth ($\varepsilon = 0.4$) at a rate of 10^{-4} mm/millisecond (recall that it takes 97000 compression steps to reach 9.7 mm depth). At each time step, the force response of the mesh is calculated using the particles on the top surface of the mesh. After reaching the desired compression depth, the force response of the cubic mesh for 97000 steps is saved to F_{SDN}, and compared with the desired force response, $F_{REFERENCE}$, of the lumped model (Figure 5.3).

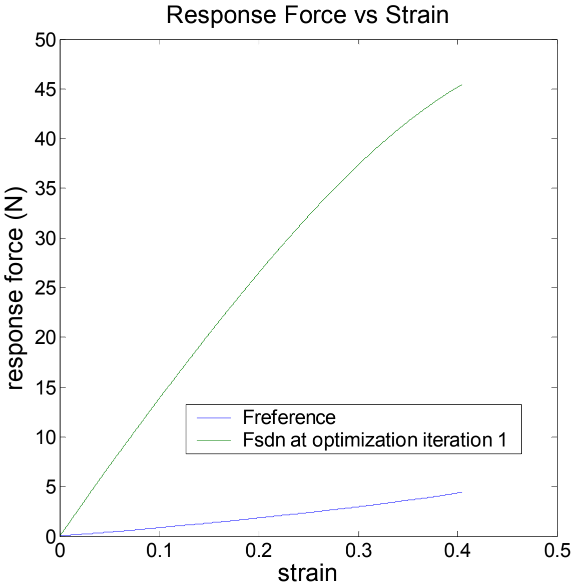


Figure 5. 3: Force responses of the lumped model ($F_{REFERENCE}$) and that of cubic mesh after the first optimization iteration

To compute a polynomial function guess for the stiffness of the elements, $K_{\infty}(\varepsilon)_{ELEMENT}$, we first divide $F_{REFERENCE}$ (Figure 5.3) to F_{SDN} (Figure 5.3) term-by-term (for 97000 data points) to obtain a ratio, what we named as "multiplication factor" (see Figure 5.4) and then multiply F_{GUESS} by this multiplication factor (see Figure 5.5).

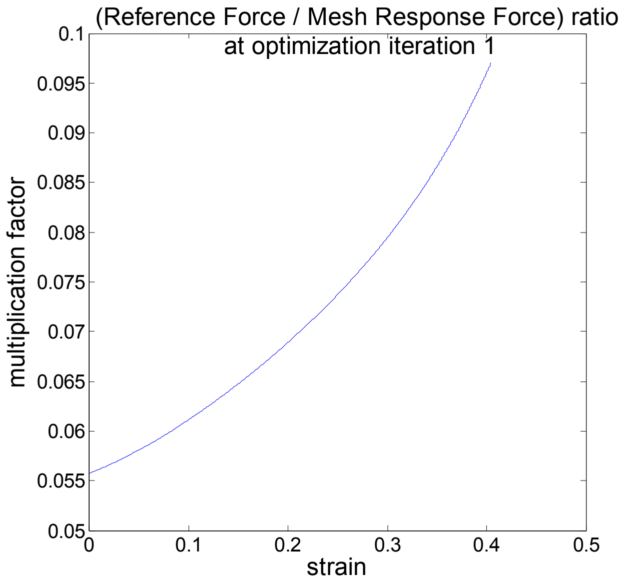


Figure 5. 4: The ratio of reference force to the mesh response force at 97000 strain points

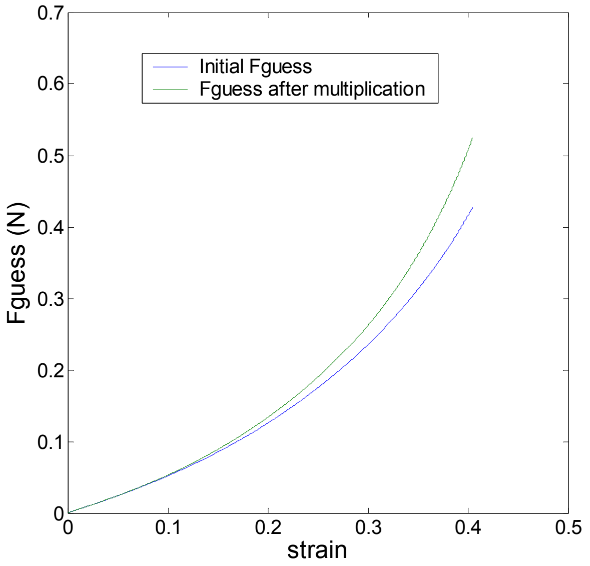


Figure 5. 5: F_{GUESS} after multiplied by the multiplication factor

Finally, the algorithm fits a 5th order polynomial to the updated F_{GUESS} to arrive at a new polynomial function similar to the one given in Equation 4.3. If the resulting polynomial is divided by ε, a new polynomial guess for $K_\infty(\varepsilon)_{ELEMENT}$ is obtained. This procedure is repeated until the current residual error ($F_{REFERENCE} - F_{SDN}$) is more than the minimum residual calculated so far. The polynomial function with the minimum residual is the optimum polynomial function for the strain-dependent element stiffness. In Figure 5.6 we provide the pseudo-code of the optimization algorithm.

```
                              Initialization
$K_\infty(\varepsilon)_{ELEMENT} = K_\infty(\varepsilon)_{LUMPED}$
Form $F_{REFERENCE}$ data set from $F_{LUMPED}(\varepsilon)$
$F_{GUESS} = F_{REFERENCE}$
CurrentResidual = $\infty$

                                 Iteration
DO
        MinimumResidual = CurrentResidual
        $K_\infty(\varepsilon)_{OPTIMUM} = K_\infty(\varepsilon)_{ELEMENT}$
        Form $F_{SDN}$ data set via applying static loading to mesh
        CurrentResidual = norm($F_{REFERENCE} - F_{SDN}$, 2)
        MultiplicationFactor = $F_{REFERENCE}$ ./ $F_{SDN}$ (term-by-term division)
        $F_{GUESS} = F_{GUESS}$ .* MultiplicationFactor (term-by-term multiplication)
        PolynomialFit = polyfit($F_{GUESS}$, 5)
        $K_\infty(\varepsilon)_{ELEMENT}$ = PolynomialFit / $\varepsilon$
WHILE (CurrentResidual < MinimumResidual)
```

Figure 5. 6: Pseudo-code for estimating the optimum nonlinear element function, $K_\infty(\varepsilon)$

When the algorithm is executed for the 27-node cubic SDN mesh, it converges to the optimum $K_\infty(\varepsilon)$ function in 10 iterations. Table 5.1 lists the guessed polynomial functions and the resulting residuals after each iteration.

Iteration #	$K_\infty(\varepsilon)_{OPTIMUM}$ & $K_\infty(\varepsilon)_{ELEMENT}$	Residual
1	$K_\infty(\varepsilon)_{OPTIMUM} = 146.89\varepsilon^4 - 131.36\varepsilon^3 + 47.47\varepsilon^2 - 0.70\varepsilon + 8.18$ $K_\infty(\varepsilon)_{ELEMENT} = 32.82\varepsilon^4 - 23.54\varepsilon^3 + 8.17\varepsilon^2 - 0.18\varepsilon + 0.47$	8243.89
2	$K_\infty(\varepsilon)_{OPTIMUM} = 32.82\varepsilon^4 - 23.54\varepsilon^3 + 8.17\varepsilon^2 - 0.18\varepsilon + 0.47$ $K_\infty(\varepsilon)_{ELEMENT} = 67.91\varepsilon^4 - 46.94\varepsilon^3 + 14.48\varepsilon^2 - 0.68\varepsilon + 0.49$	91.22
3	$K_\infty(\varepsilon)_{OPTIMUM} = 67.91\varepsilon^4 - 46.94\varepsilon^3 + 14.48\varepsilon^2 - 0.68\varepsilon + 0.49$ $K_\infty(\varepsilon)_{ELEMENT} = 99.48\varepsilon^4 - 68.62\varepsilon^3 + 19.91\varepsilon^2 - 1.16\varepsilon + 0.51$	35.79
4	$K_\infty(\varepsilon)_{OPTIMUM} = 99.48\varepsilon^4 - 68.62\varepsilon^3 + 19.91\varepsilon^2 - 1.16\varepsilon + 0.51$ $K_\infty(\varepsilon)_{ELEMENT} = 124.53\varepsilon^4 - 86.13\varepsilon^3 + 24.15\varepsilon^2 - 1.53\varepsilon + 0.52$	16.17
5	$K_\infty(\varepsilon)_{OPTIMUM} = 124.53\varepsilon^4 - 86.13\varepsilon^3 + 24.15\varepsilon^2 - 1.53\varepsilon + 0.52$ $K_\infty(\varepsilon)_{ELEMENT} = 143.74\varepsilon^4 - 99.72\varepsilon^3 + 27.42\varepsilon^2 - 1.82\varepsilon + 0.53$	8.55
6	$K_\infty(\varepsilon)_{OPTIMUM} = 143.74\varepsilon^4 - 99.72\varepsilon^3 + 27.42\varepsilon^2 - 1.82\varepsilon + 0.53$ $K_\infty(\varepsilon)_{ELEMENT} = 158.33\varepsilon^4 - 110.13\varepsilon^3 + 29.91\varepsilon^2 - 2.04\varepsilon + 0.53$	5.45
7	$K_\infty(\varepsilon)_{OPTIMUM} = 158.33\varepsilon^4 - 110.13\varepsilon^3 + 29.91\varepsilon^2 - 2.04\varepsilon + 0.53$ $K_\infty(\varepsilon)_{ELEMENT} = 169.34\varepsilon^4 - 118.04\varepsilon^3 + 31.81\varepsilon^2 - 2.21\varepsilon + 0.54$	4.31
8	$K_\infty(\varepsilon)_{OPTIMUM} = 169.34\varepsilon^4 - 118.04\varepsilon^3 + 31.81\varepsilon^2 - 2.21\varepsilon + 0.54$ $K_\infty(\varepsilon)_{ELEMENT} = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	3.96
9	$K_\infty(\varepsilon)_{OPTIMUM} = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$ $K_\infty(\varepsilon)_{ELEMENT} = 183.84\varepsilon^4 - 128.54\varepsilon^3 + 34.34\varepsilon^2 - 2.44\varepsilon + 0.55$	3.92
10	$K_\infty(\varepsilon)_{OPTIMUM} = 183.84\varepsilon^4 - 128.54\varepsilon^3 + 34.34\varepsilon^2 - 2.44\varepsilon + 0.55$	3.97

	$K_\infty(\varepsilon)_{ELEMENT} = 188.49\varepsilon^4 - 131.92\varepsilon^3 + 35.15\varepsilon^2 - 2.51\varepsilon + 0.55$	

Table 5. 1: Convergence of $K_\infty(\varepsilon)$

As can be seen from Table 5.1, the optimum stiffness function for the elements is

$$K_\infty(\varepsilon)_{OPTIMUM} = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54 \tag{5.21}$$

Figures 5.7 through 5.10 present the convergence behavior of F_{SDN} to $F_{REFERENCE}$, and the behavior of the estimated $K_\infty(\varepsilon)_{ELEMENT}$. Figure 5.11 shows the nonlinear response of the $K_\infty(\varepsilon)_{ELEMENT}$ as function of strain.

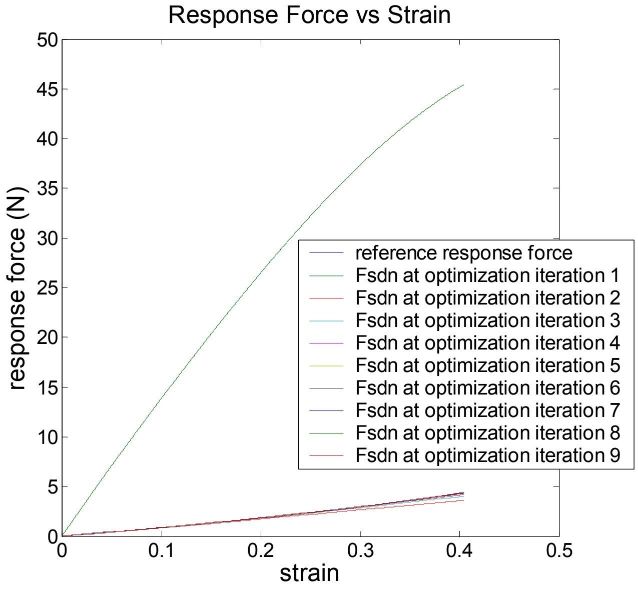


Figure 5. 7: Convergence of the force response obtained from the cubic mesh, F_{SDN}, to the force response of the lump model

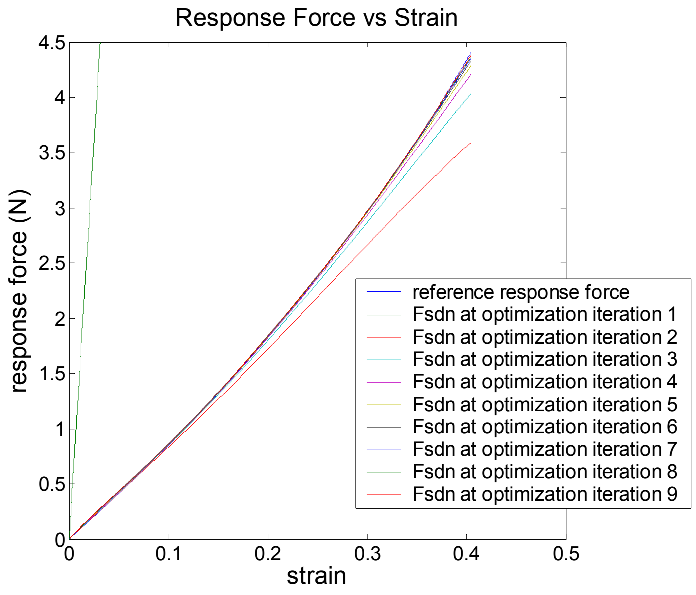


Figure 5. 8: Convergence of the force response obtained from the cubic mesh, F_{SDN}, to the force response of the lump model (zoomed into 0-4.5 N region)

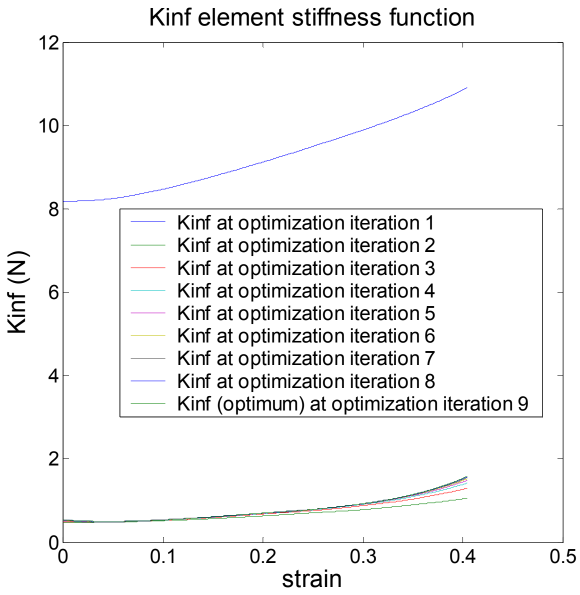


Figure 5. 9: Convergence of the element stiffness function, $K\infty(\varepsilon)_{ELEMENT}$

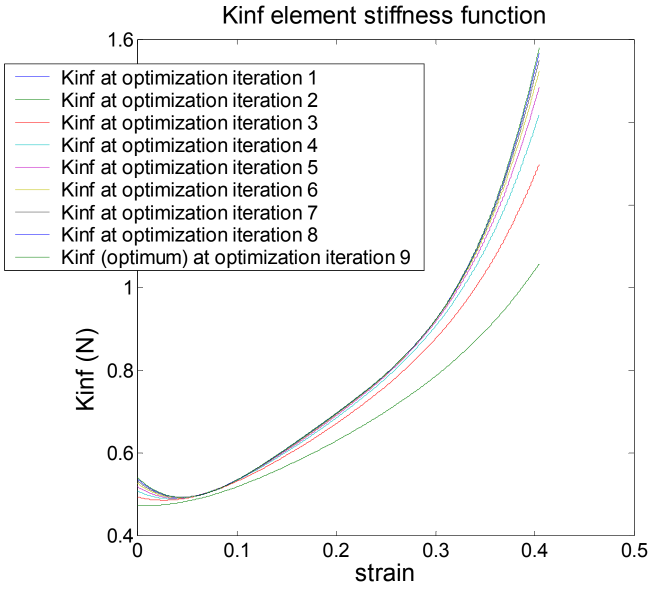


Figure 5. 10: Convergence of the element stiffness function, K∞(ε)$_{ELEMENT}$ (zoomed into 0.4-1.6 N region)

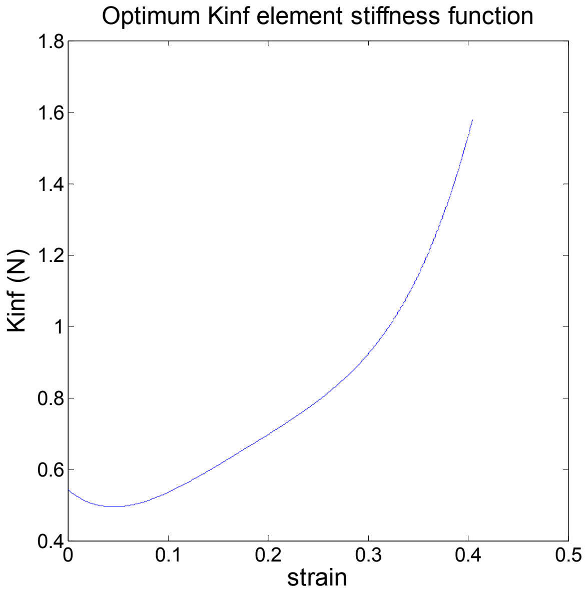


Figure 5. 11: Optimum K$_{\infty}$(ε)$_{ELEMENT}$ function

The optimization algorithm was also tested with a refined 125-node (5x5x5) cubic mesh having the same dimensions. It took 7 iterations to converge to an optimum polynomial function $K_\infty(\varepsilon)$ with a residual of 4.76.

$$K_\infty\left(\varepsilon\right)_{OPTIMUM} = 68.69\varepsilon^4 - 50.59\varepsilon^3 + 13.53\varepsilon^2 - 1.0\varepsilon + 0.19 \tag{5.22}$$

Figure 5.12 presents the optimum element stiffness functions for the 27- and 125-node meshes.

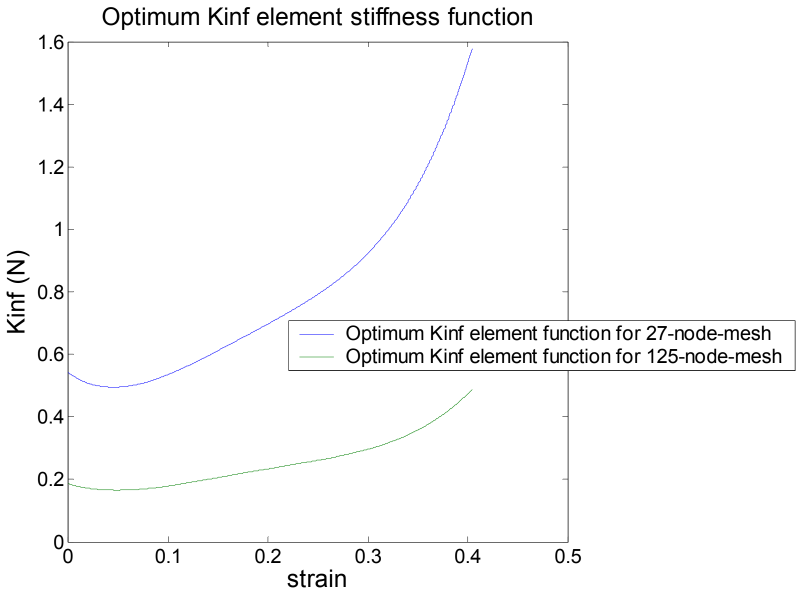


Figure 5. 12: Optimum $K_\infty(\varepsilon)_{\text{ELEMENT}}$ for the 27- and 125-node meshes

Note that the $K_\infty(\varepsilon)_{\text{ELEMENT}}$ function calculated through this optimization algorithm is in units of Newton and must be divided by the “characteristic” length of any given mesh (i.e. the rest length of the shortest element in the mesh) to calculate the stiffness function of the elements for that particular mesh. For example, for a cubic mesh having an edge length of 0.024 m, the characteristic length is 0.012 m in a 27-node cube mesh and 0.006 m in a 125-node mesh. Figure 5.13 presents the absolute stiffness function of the individual elements in 27-node and 125-node meshes.

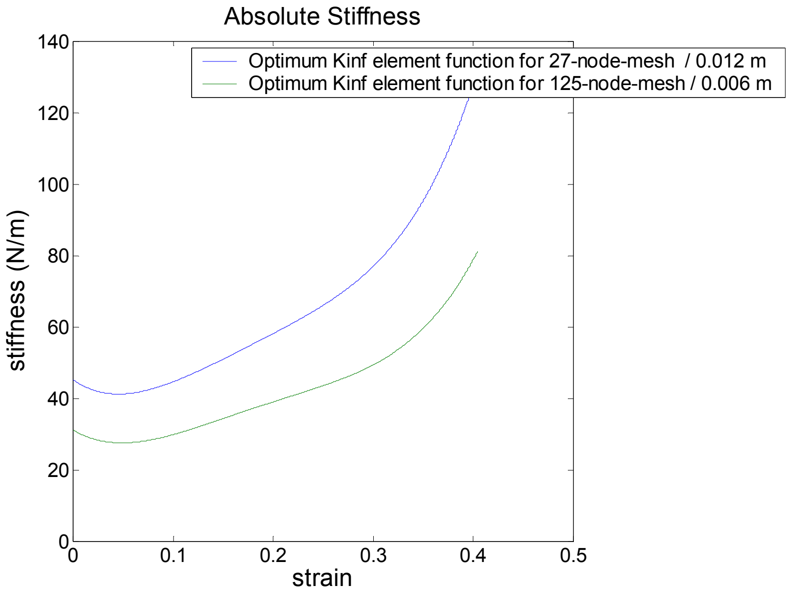


Figure 5. 13: Absolute stiffness function of the individual elements in the 27- and 125-node meshes

5.3.2 An Optimization Algorithm for Estimating the Time-dependent Viscoelastic Behavior of the Elements in a SDN

The optimization algorithm explained in this section estimates the optimum value of the viscoelastic material coefficients K_1 and b_1, such that the force response of the cubic mesh is sufficiently close to that of the lumped nonlinear viscoelastic model in ramp & hold experiments.

The time-dependent stress response of the Maxwell Solid with N=1 to a ramp & hold strain input was given in Equation 3.50. Rearranging this equation for a nonlinear viscoelastic Maxwell Solid having a strain-dependent nonlinear E∞(ε) function, we have

$$\begin{aligned} \sigma\left(t\right) &= A\left[z(t) - z(t-c)u(t-c)\right] \\ z\left(t\right) &= \mathrm{E}_{\infty}(\varepsilon)t + \eta_1\left(1 - \exp\left(-t/\tau_1\right)\right) \end{aligned} \tag{5.23}$$

Since the element does not have an area at the tip, the stress in Equation 5.23 can be replaced with the force function as

$$\begin{aligned} F\left(t\right) &= A\left[z(t) - z(t-c)u(t-c)\right] \\ z\left(t\right) &= K_{\infty}(\varepsilon)t + b_1\left(1 - \exp\left(-t/\tau_1\right)\right) \\ \tau_1 &= \frac{b_1}{K_1} \end{aligned} \tag{5.24}$$

The nonlinear and viscoelastic material properties of the lumped Maxwell Solid model with N=1 were already derived in Chapter 4. To examine the force-relaxation response of the lumped model to an arbitrary ramp & hold input, we calculated the response of the lump model to a ramp & hold input of ε = 0.25 (6 mm) applied for 50 seconds by inserting these properties into Equation 5.24 (see Figure 5.14).

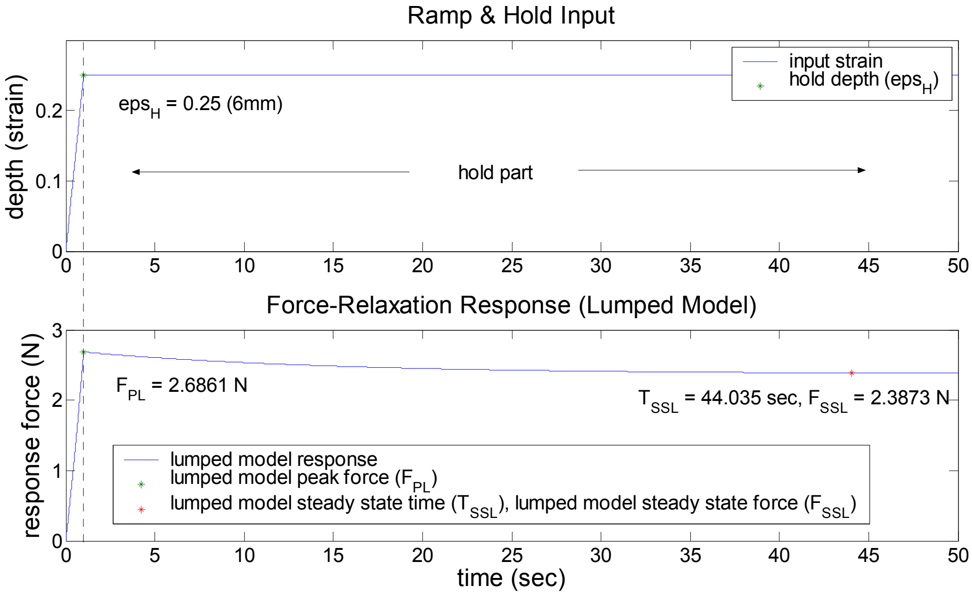


Figure 5. 14: Force-relaxation behavior of the experimentally characterized lumped model in response to a ramp & hold input of ε = 0.25 (6 mm)

To come up with an optimization scheme for the estimation of K$_1$ and b$_1$, we inspect the hold part ($t \geq c$) of Equation 5.24, which can be written as

$$F\left(t\right) = K_\infty(\varepsilon_H)\varepsilon_H - Ab_1\left(1-\exp\left(c\frac{K_1}{b_1}\right)\right)\exp\left(-t\frac{K_1}{b_1}\right) \tag{5.25}$$

where ε$_H$ is the constant strain during the ramp part of the experiment, $A = \dfrac{\varepsilon_H}{c}$ is the slope of the ramp, and c = 1 second is the ramp duration.

A careful examination of Figure 5.14 and Equation 5.25 reveal 3 pieces of important information which are used to develop an optimization algorithm to estimate the material coefficients of the elements in a SDN. F$_{PL}$ is the peak force of the lumped model at t = 1 sec. T_{SSL} is the time when the force response of the model reaches to steady-state. It is assumed that the force response reaches to steady-state when the slope of the curve is less than 10^{-3}. And finally, F_{SSL} is the steady-state force at T_{SSL}. We attempt to match the peak force, the steady-state time, and the steady-state force of the cubic mesh (i.e., F_{PM}, T_{SSM}, F_{SSM}) to those of the lump model. As a result, the optimization scheme for estimating the optimum values of K_1 and b_1 is composed of three phases: the first phase is for matching F_{PM} to F_{PL}, the second phase is for matching T_{SSM} to T_{SSL}, and the third phase, which performs a fine-tuning on $K_\infty(\varepsilon)$, is necessary for matching F_{SSM} to F_{SSL}.

During the initialization part of the optimization algorithm, Equation 5.25 is executed with the material coefficients of the lumped model to calculate F_{PL}, T_{SSL}, and F_{SSL} in response to the ramp & hold strain input. The initial values for $K_\infty(\varepsilon)$, K_1, and b_1 of the elements in the mesh model are set to the values estimated in Chapter 4 for the lumped model. Then, the mesh is simulated under the given ramp & hold input to calculate the values of F_{PM}, T_{SSM}, and F_{SSM} (Figure 5.15). Finally, the optimization iterations are initialized to reduce the relative error for the optimization variables (i.e., F_{PE}, T_{SSE}, and F_{SSE}) to less than 1%.

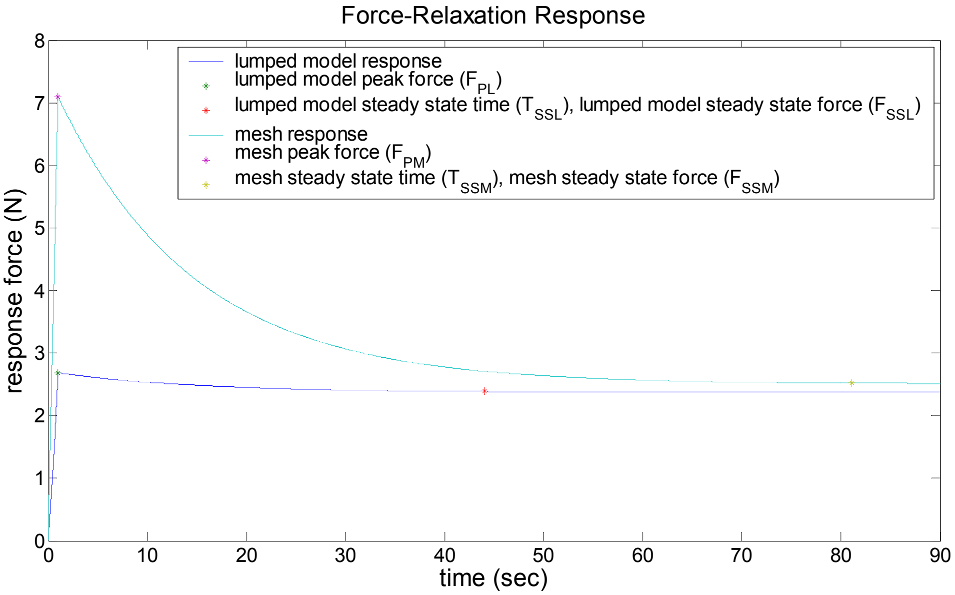


Figure 5. 15: Force-relaxation behavior of lumped Maxwell Solid with N=1 and SDN with initial K_1 and b_1 values in response to a ramp & hold input of ε = 0.25 (6 mm)

In each iteration of the first phase, the algorithm runs the mesh model with the current values of K_1 and b_1 for 1 second and apply the ramp input to compute F_{PM} and the relative error in peak response. If the relative error is more than 1%, it then multiplies both K_1 and b_1 (so that the ratio of $\frac{K_1}{b_1}$ is constant) by $\frac{F_{PL}}{F_{PM}}$ ratio. The first phase is iterated until either F_{PM} is close enough to F_{PL} so that the relative error in peak response is less than 1% or the relative error starts becoming larger than the minimum relative error calculated so far (see Figure 5.16 and Table 5.2). At the end of the first phase, for example, the F_{PM} is calculated as 2.711750 N for the cubic mesh of 27 nodes.

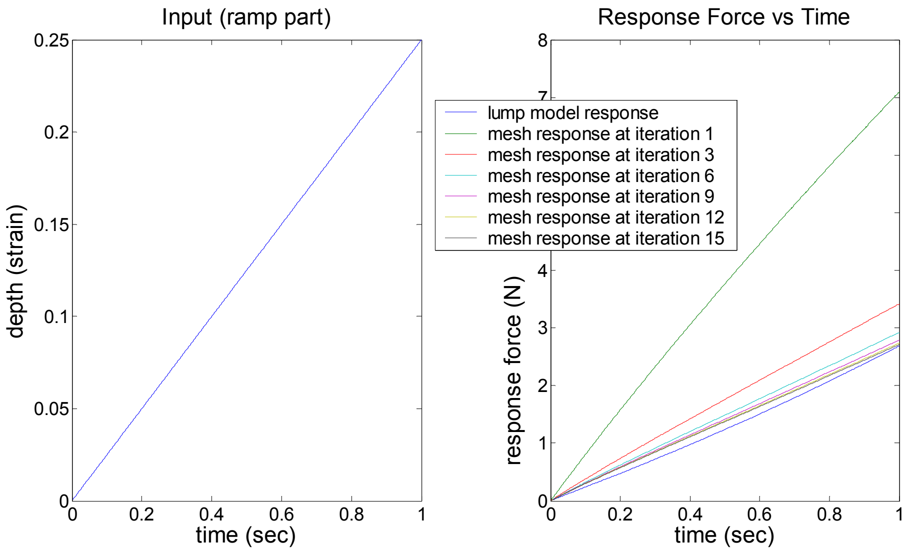


Figure 5. 16: Convergence of F_{PM} to F_{PL}. Note that the peak response (at time = 1 sec) obtained from cubic mesh gets closer to that of the lumped model after each iteration.

Iteration #	**$K_{1\ OPTIMUM}$ (N) – $b_{1\ OPTIMUM}$ (Ns)**	**F_{PE} (%)**
1	$K_{1\ OPTIMUM}$ = 1.296299, $b_{1\ OPTIMUM}$ = 17.873062	164.06
2	$K_{1\ OPTIMUM}$ = 0.490900, $b_{1\ OPTIMUM}$ = 6.768420	50.42
3	$K_{1\ OPTIMUM}$ = 0.326346, $b_{1\ OPTIMUM}$ = 4.499580	27.20
4	$K_{1\ OPTIMUM}$ = 0.256555, $b_{1\ OPTIMUM}$ = 3.537327	17.35
5	$K_{1\ OPTIMUM}$ = 0.218616, $b_{1\ OPTIMUM}$ = 3.014230	12
6	$K_{1\ OPTIMUM}$ = 0.195193, $b_{1\ OPTIMUM}$ = 2.691272	8.69
7	$K_{1\ OPTIMUM}$ = 0.179579, $b_{1\ OPTIMUM}$ = 2.475994	6.49
8	$K_{1\ OPTIMUM}$ = 0.168633, $b_{1\ OPTIMUM}$ = 2.325071	4.94
9	$K_{1\ OPTIMUM}$ = 0.160685, $b_{1\ OPTIMUM}$ = 2.215486	3.82
10	$K_{1\ OPTIMUM}$ = 0.154766, $b_{1\ OPTIMUM}$ = 2.133873	2.98
11	$K_{1\ OPTIMUM}$ = 0.150274, $b_{1\ OPTIMUM}$ = 2.071937	2.35
12	$K_{1\ OPTIMUM}$ = 0.146816, $b_{1\ OPTIMUM}$ = 2.024260	1.86
13	$K_{1\ OPTIMUM}$ = 0.144124, $b_{1\ OPTIMUM}$ = 1.987154	1.48
14	$K_{1\ OPTIMUM}$ = 0.142012, $b_{1\ OPTIMUM}$ = 1.958029	1.18
15	$K_{1\ OPTIMUM}$ = 0.140343, $b_{1\ OPTIMUM}$ = 1.935015	0.95

Table 5. 2: Convergence behavior of K_1 and b_1

In each iteration of the second phase, the algorithm runs the mesh model with the current values of K_1 and b_1 for the ramp & hold input until the force response of the cubic mesh reaches to steady-state. Then, the steady-state time is recorded as T_{SSM} and compared to T_{SSL}. If the relative error is more than 1%, K_1 is kept constant and b_1 is multiplied by $\frac{T_{SSL}}{T_{SSM}}$ ratio. The second phase is iterated until either T_{SSM} is close enough to T_{SSL} so that the relative error is less than 1% or it is larger than the minimum relative error calculated so far (see Table 5.3 and Figure 5.17). At the end of the second phase, for example, T_{SSM} is calculated as 43.717 sec for the cubic mesh of 27 nodes. However, as expected, it is observed that the change in b_1 slightly influenced F_{PM}, which is updated to 2.714767 N.

Iteration #	**$b_{1\ \text{OPTIMUM}}$ (Ns)**	**T_{SSE} (%)**
1	1.935015	12.17
2	2.203140	4.7
3	2.312092	1.83
4	2.355420	0.72

Table 5. 3: Convergence behavior of b_1

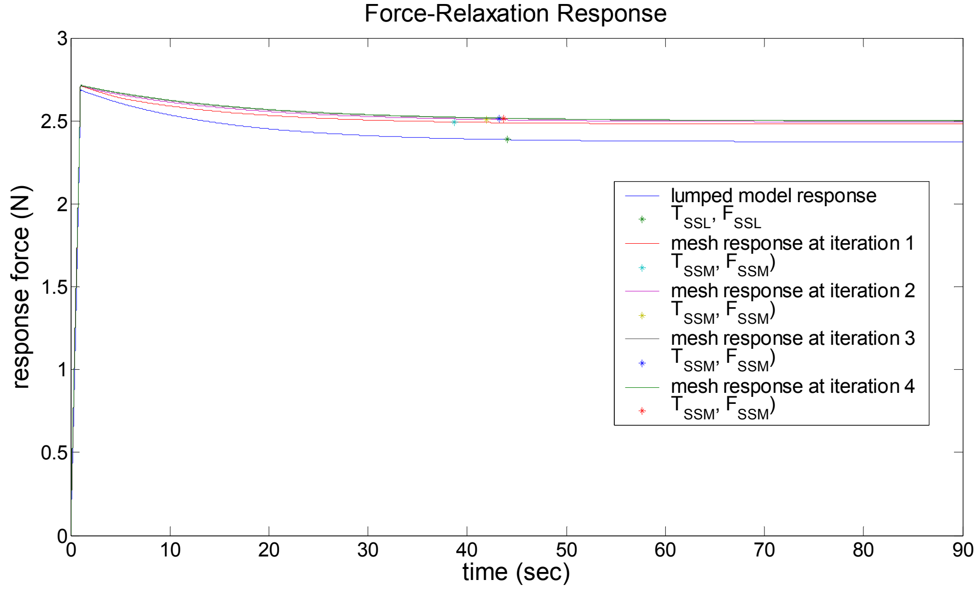


Figure 5. 17: Convergence of T_{SSM} to T_{SSL}

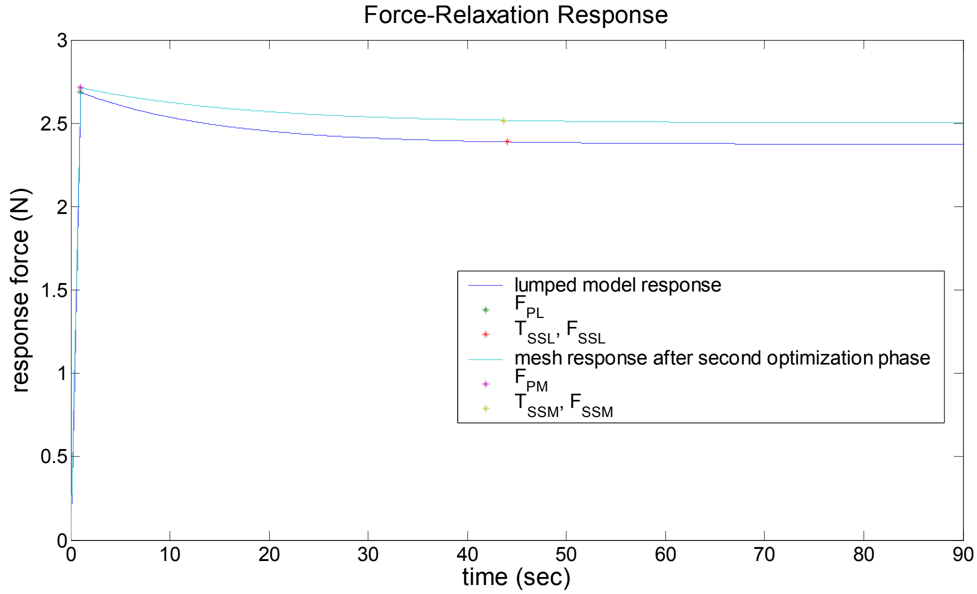


Figure 5. 18: Comparison of the force-relaxation response of the lumped model and that of the mesh with optimized K_1-b_1 values at the end of second phase

Figure 5.18 compares the force-relaxation behavior of the lumped model to that of the mesh after the second phase. It is observed that the peak forces and the steady-state times of the mesh and the lumped models are very close to each other. However, the force values at steady-state are slightly off. The relative error in steady-state force is 5.4% for this particular case. The reason behind this difference was explained in Section 4.3.3. Hence, in the third phase, we fine-tune the $K_\infty(\varepsilon)$ function so that F_{SSM} matches to F_{SSL}. The algorithm multiplies the $K_\infty(\varepsilon)$ with a coefficient until the steady-state error is less than 1 % (refer to the pseudo-code in Figure 5.21 for the calculation of this coefficient at each iteration). Table 5.4 shows the drop in steady-state force error after each iteration in the third phase, and Figure 5.19 compares the force-relaxation response of the mesh to the one obtained from the lumped model after the third phase.

Iteration #	**$K_\infty(\varepsilon)_{OPTIMUM}$ (N)**	**F_{SSE} (%)**
1	$K_\infty(\varepsilon)_{OPTIMUM} = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	5.4
2	$K_\infty(\varepsilon)_{OPTIMUM} = 167.98\varepsilon^4 - 117.29\varepsilon^3 + 31.45\varepsilon^2 - 2.21\varepsilon + 0.51$	0.2

Table 5. 4: Convergence of $K_\infty(\varepsilon)$

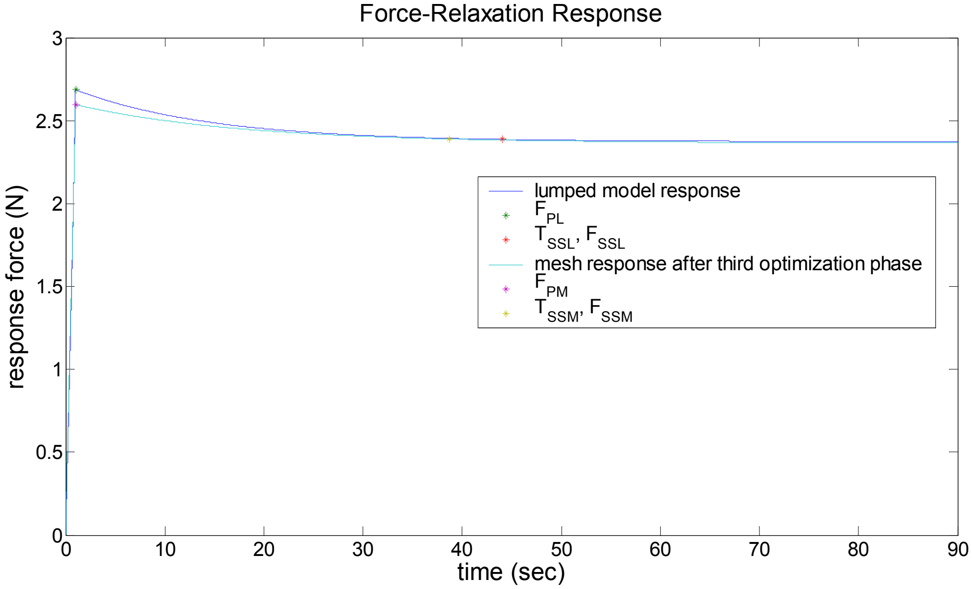


Figure 5. 19: Comparison of the force-relaxation response of the mesh with the one obtained from the lumped model after the third phase

As it is observed from Figure 5.19, fine-tuning on $K_\infty(\varepsilon)$ in phase 3 resulted in a change in the values of F_{PM} and F_{SSM}. The relative error in peak force and steady-state time are now 3.3% and 2.2%, respectively. To remedy this problem, all three phases of the proposed optimization algorithm are repeated until all the relative errors are smaller than 1%. For instance, for the case of 6 mm ramp & hold input, the algorithm estimated the optimum material coefficients in 3 global iterations. The optimum values are given in Table 5.5 and Figure 5.20 compares the force-relaxation response of the mesh to the one obtained from the lumped model.

Optimized values for 6mm ramp & hold input	**F_{PE} (%)**	**T_{SSE} (%)**	**F_{SSE} (%)**
$K_1 = 0.159$ N $b_1 = 2.251$ Ns $K_\infty(\varepsilon)_{OPTIMUM} = 167.98\varepsilon^4 - 117.29\varepsilon^3 + 31.45\varepsilon^2 - 2.21\varepsilon + 0.51$ N	0.85	0.36	0.91

Table 5. 5: Optimum material coefficients for 6 mm ramp & hold input

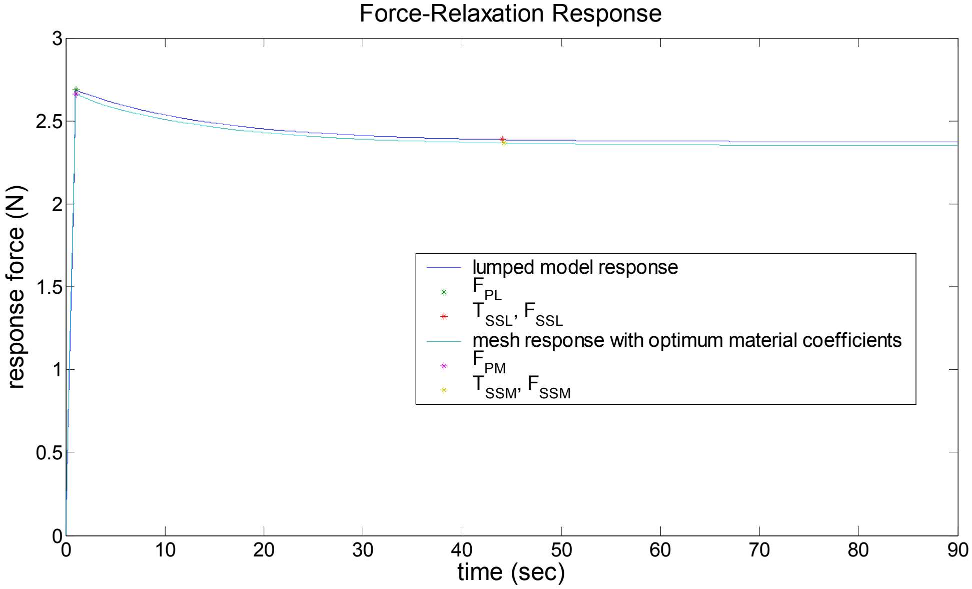


Figure 5. 20: Comparison of the force-relaxation response of the mesh to that of the lumped model after several global iterations.

As can be seen from Figure 5.20, the mesh response with the optimum coefficients is very close to the response of the lumped model for the same ramp & hold strain input (6 mm). The errors in Table 5.5 can be further reduced if an error bound smaller than 1% is used. However, having a tighter error bound would result in more iterations and increase the computation time. Figure 5.21 summarizes the optimization algorithm in the form of a pseudo-code.

```
                          Initialization
Initialize ErrorBound
Initialize MaximumNumberOfIterations
Compute $F_{PL}$, $T_{SSL}$, $F_{SSL}$ via applying ramp & hold input to Equation 5.24
$K_\infty(\varepsilon) = K_\infty(\varepsilon)_{OPTIMUM}$
$K_1 = K_{1\ INITIAL}$
$b_1 = b_{1\ INITIAL}$
Compute $F_{PM}$, $T_{SSM}$, $F_{SSM}$ via applying ramp & hold input to the mesh
$F_{PE} = abs(F_{PM} - F_{PL}) / F_{PL}$
$T_{SSE} = abs(T_{SSM} - T_{SSL}) / T_{SSL}$
$F_{SSE} = abs(F_{SSM} – F_{SSL}) / F_{SSL}$

                          Iteration
WHILE ($F_{PE}$ > ErrorBound OR $T_{SSE}$ > ErrorBound OR $F_{SSE}$ > ErrorBound)

    PHASE 1

    Minimum$F_{PE} = \infty$

    DO
        Compute $F_{PM}$ via applying ramp input to the mesh
        $F_{PE} = abs(F_{PM} - F_{PL}) / F_{PL}$

        IF ($F_{PE}$ < Minimum$F_{PE}$)
            $K_{1\ OPTIMUM} = K_1$
            $b_{1\ OPTIMUM} = b_1$
            Minimum$F_{PE} = F_{PE}$
        ELSE
            break
        END
        IF ($F_{PE}$ > ErrorBound)
            $K_1 = K_1 \times (F_{PL} / F_{PM})$
            $b_1 = b_1 \times (F_{PL} / F_{PM})$
        END
    WHILE ($F_{PE}$ > ErrorBound)

    PHASE 2

    $MinimumT_{SSE} = \infty$

    DO
        Compute $F_{PM}$, $T_{SSM}$ and $F_{SSM}$ via applying ramp & hold input to the mesh
        $T_{SSE} = abs(T_{SSM} - T_{SSL}) / T_{SSL}$

        IF ($T_{SSE} < MinimumT_{SSE}$)
            $b_{1\ OPTIMUM} = b_1$
            $MinimumT_{SSE} = T_{SSE}$
        ELSE
            break
        END

        IF ($T_{SSE}$ > ErrorBound)
            $b_1 = b_1 \times (T_{SSL} / T_{SSM})$
        END
    WHILE ($T_{SSE}$ > ErrorBound)

    Phase 3

    $MinimumF_{SSE} = \infty$
    NoIter = 0
```

```
        DO
                NoIter = NoIter + 1
                Compute $F_{SSM}$ via applying ramp & hold input to the mesh
                $F_{SSE}$ = abs($F_{SSM}$ – $F_{SSL}$) / $F_{SSL}$

                IF ($F_{SSE}$ > ErrorBound)
                        IF ($F_{SSE}$ < Minimum$F_{SSE}$)
                                Minimum$F_{SSE}$ = $F_{SSE}$
                                MinimumError$F_{SSM}$ = $F_{SSM}$
                                MinimumError$K_\infty(\varepsilon)$ = $K_\infty(\varepsilon)$
                                Difference = $F_{SSE}$
                        ELSE
                                Difference = Difference / 2
                        END

                        IF (MinimumError$F_{SSM}$ > $F_{SSL}$)
                                MultiplicationFactor = (1 - Difference)
                        ELSE
                                MultiplicationFactor = (1 + Difference)
                        END

                        $K_\infty(\varepsilon)$ = MultiplicationFactor x MinimumError$K_\infty(\varepsilon)$
                END
        WHILE ( ($F_{SSE}$ > ErrorBound)  AND (NoIter < MaximumNumberOfIterations) )
END
```

Figure 5. 21: Pseudo-code for estimating the optimum viscoelastic and nonlinear material coefficients of the SDN

The optimization algorithm explained above can handle different initial guesses as long as the guesses don't make SDN unstable. When the optimization algorithm is given different sets of initial guesses as shown in Table 5.6, the estimated material coefficients do not change much.

Initial guesses	**Estimated coefficients**
$K_1 = 1.296$, $b_1 = 17.873$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.159$, $b_1 = 2.251$ $K_\infty(\varepsilon) = 167.98\varepsilon^4 - 117.29\varepsilon^3 + 31.45\varepsilon^2 - 2.21\varepsilon + 0.51$
$K_1 = 12.962$, $b_1 = 17.873$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.165$, $b_1 = 2.241$ $K_\infty(\varepsilon) = 170\varepsilon^4 - 118.71\varepsilon^3 + 31.83\varepsilon^2 - 2.24\varepsilon + 0.52$
$K_1 = 0.1296$, $b_1 = 17.873$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.166$, $b_1 = 2.246$ $K_\infty(\varepsilon) = 169.34\varepsilon^4 - 118.25\varepsilon^3 + 31.70\varepsilon^2 - 2.23\varepsilon + 0.52$
$K_1 = 1.296$, $b_1 = 178.73$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.166$, $b_1 = 2.240$ $K_\infty(\varepsilon) = 169.36\varepsilon^4 - 118.26\varepsilon^3 + 31.70\varepsilon^2 - 2.23\varepsilon + 0.52$
$K_1 = 1.296$, $b_1 = 1.7873$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.165$, $b_1 = 2.241$ $K_\infty(\varepsilon) = 169.98\varepsilon^4 - 118.69\varepsilon^3 + 31.82\varepsilon^2 - 2.24\varepsilon + 0.52$
$K_1 = 12.962$, $b_1 = 1.787$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.165$, $b_1 = 2.238$ $K_\infty(\varepsilon) = 169.81\varepsilon^4 - 118.58\varepsilon^3 + 31.79\varepsilon^2 - 2.24\varepsilon + 0.52$
$K_1 = 0.1296$, $b_1 = 1.7873$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.158$, $b_1 = 2.263$ $K_\infty(\varepsilon) = 169.42\varepsilon^4 - 118.30\varepsilon^3 + 31.71\varepsilon^2 - 2.23\varepsilon + 0.52$
$K_1 = 12.96$, $b_1 = 178.73$ $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$	$K_1 = 0.158$, $b_1 = 2.252$ $K_\infty(\varepsilon) = 167.92\varepsilon^4 - 117.25\varepsilon^3 + 31.43\varepsilon^2 - 2.21\varepsilon + 0.51$

Table 5. 6: The optimum material coefficients estimated for different initial guess sets for 6mm ramp & hold input in a 27-node mesh

On the other hand, when the same optimization algorithm is executed with different ramp & hold inputs, the estimated material properties change with the input strain. For the 27-node SDN, the optimization algorithm was executed with different ramp & hold inputs for the initial guesses of K1 = 1.296, b1 = 17.873, and $K_\infty(\varepsilon) = 177.63\varepsilon^4 - 124.03\varepsilon^3 + 33.25\varepsilon^2 - 2.34\varepsilon + 0.54$. Table 5.7 tabulates the optimum material coefficients estimated for the different inputs (also see Figures 5.22, 5.23 and 5.24).

Ramp & Hold Input (mm)	**Optimized values for 6mm ramp & hold input**
2	$K_1 = 0.061$, $b_1 = 0.908$ $K_\infty(\varepsilon) = 172.3\varepsilon^4 - 120.31\varepsilon^3 + 32.25\varepsilon^2 - 2.27\varepsilon + 0.53$
2.5	$K_1 = 0.07$, $b_1 = 1.029$ $K_\infty(\varepsilon) = 173.342\varepsilon^4 - 121.04\varepsilon^3 + 32.45\varepsilon^2 - 2.28\varepsilon + 0.53$
3	$K_1 = 0.082$, $b_1 = 1.199$ $K_\infty(\varepsilon) = 173.20\varepsilon^4 - 120.94\varepsilon^3 + 32.42\varepsilon^2 - 2.28\varepsilon + 0.53$
3.5	$K_1 = 0.095$, $b_1 = 1.376$ $K_\infty(\varepsilon) = 172.54\varepsilon^4 - 120.48\varepsilon^3 + 32.3\varepsilon^2 - 2.28\varepsilon + 0.53$
4	$K_1 = 0.107$, $b_1 = 1.548$ $K_\infty(\varepsilon) = 171.67\varepsilon^4 - 119.87\varepsilon^3 + 32.14\varepsilon^2 - 2.26\varepsilon + 0.52$
4.5	$K_1 = 0.12$, $b_1 = 1.73$ $K_\infty(\varepsilon) = 170.77\varepsilon^4 - 119.24\varepsilon^3 + 31.97\varepsilon^2 - 2.25\varepsilon + 0.52$
5	$K_1 = 0.131$, $b_1 = 1.907$ $K_\infty(\varepsilon) = 169.9\varepsilon^4 - 118.63\varepsilon^3 + 31.8\varepsilon^2 - 2.24\varepsilon + 0.52$
5.5	$K_1 = 0.145$, $b_1 = 2.065$ $K_\infty(\varepsilon) = 169\varepsilon^4 - 118.01\varepsilon^3 + 31.64\varepsilon^2 - 2.23\varepsilon + 0.52$

6	$K_1 = 0.159$, $b_1 = 2.251$ $K_\infty(\varepsilon) = 167.98\varepsilon^4 - 117.29\varepsilon^3 + 31.45\varepsilon^2 - 2.21\varepsilon + 0.51$
6.5	$K_1 = 0.168$, $b_1 = 2.463$ $K_\infty(\varepsilon) = 169.47\varepsilon^4 - 118.33\varepsilon^3 + 31.72\varepsilon^2 - 2.23\varepsilon + 0.52$
7	$K_1 = 0.18$, $b_1 = 2.706$ $K_\infty(\varepsilon) = 169.27\varepsilon^4 - 118.2\varepsilon^3 + 31.68\varepsilon^2 - 2.23\varepsilon + 0.52$
7.5	$K_1 = 0.198$, $b_1 = 2.944$ $K_\infty(\varepsilon) = 169.1\varepsilon^4 - 118.08\varepsilon^3 + 31.65\varepsilon^2 - 2.23\varepsilon + 0.52$
8	$K_1 = 0.203$, $b_1 = 3.339$ $K_\infty(\varepsilon) = 166.6\varepsilon^4 - 116.33\varepsilon^3 + 31.19\varepsilon^2 - 2.19\varepsilon + 0.51$
8.5	$K_1 = 0.219$, $b_1 = 3.499$ $K_\infty(\varepsilon) = 167.76\varepsilon^4 - 117.14\varepsilon^3 + 31.4\varepsilon^2 - 2.21\varepsilon + 0.51$
9	$K_1 = 0.223$, $b_1 = 3.591$ $K_\infty(\varepsilon) = 167.74\varepsilon^4 - 117.13\varepsilon^3 + 31.4\varepsilon^2 - 2.21\varepsilon + 0.51$

Table 5. 7: The optimum material coefficients estimated for different ramp & hold inputs in a 27-node mesh

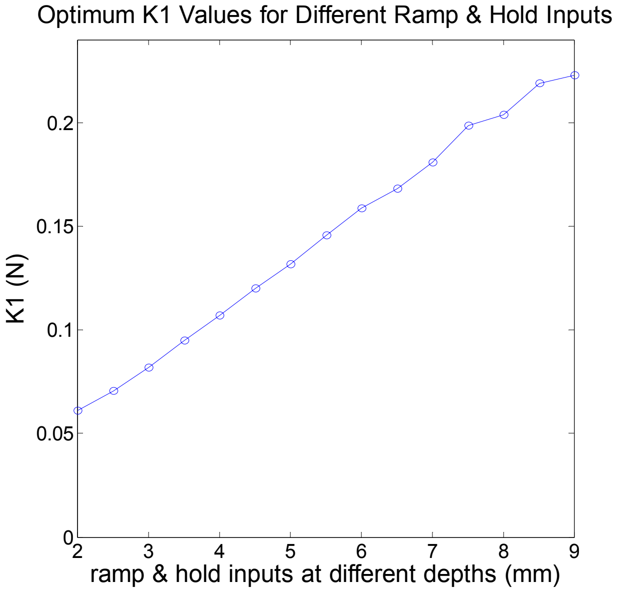


Figure 5. 22: The optimum K_1 values estimated for different ramp & hold inputs

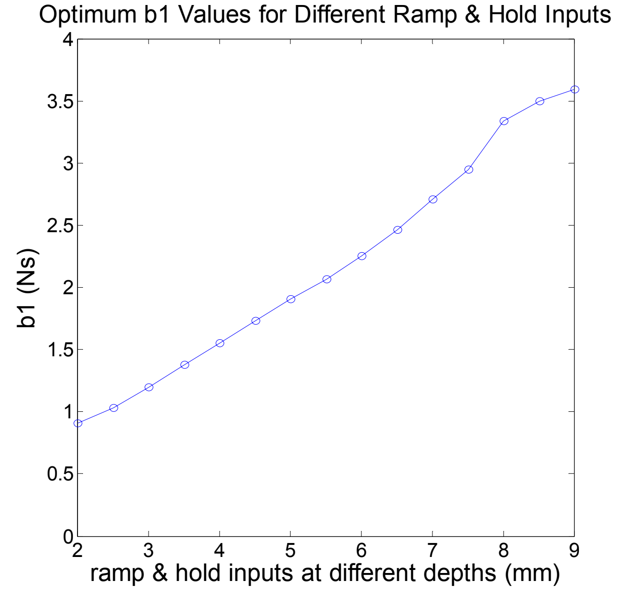


Figure 5. 23: The optimum b_1 values estimated for different ramp & hold inputs

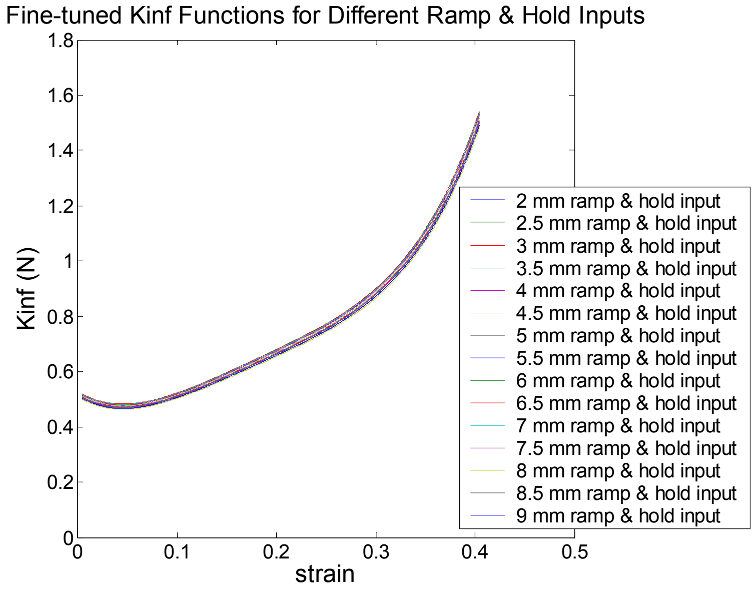


Figure 5. 24: The Optimum $K_\infty(\varepsilon)$ functions estimated for different ramp & hold inputs

As can be seen from Table 5.7, the estimated material coefficients of the elements in the mesh are different for different ramp & hold inputs (see Figure 5.25) This is consistent since Figures 5.22 and 5.23 suggest that K_1 and b_1 coefficients are actually strain-dependent functions just like K_∞.

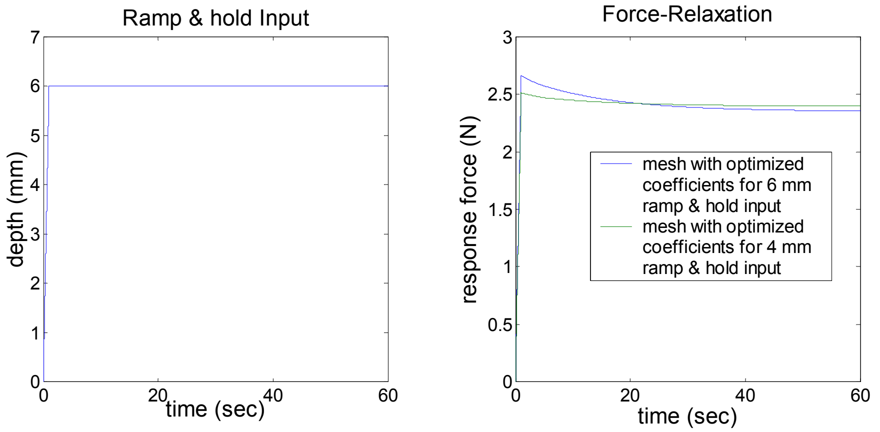


Figure 5. 25: Force-relaxation responses of the mesh is simulated using the coefficients optimized for 6 mm and 4 mm ramp & hold inputs.

For the 125-node SDN, the optimization algorithm was executed with different ramp & hold inputs for the initial guesses of K1 = 1.296, b1 = 17.873, and $K_\infty(\varepsilon)_{OPTIMUM} = 68.69\varepsilon^4 - 50.59\varepsilon^3 + 13.53\varepsilon^2 - 1.0\varepsilon + 0.19$. Table 5.8 tabulates the optimum material coefficients estimated for different inputs. (also see Figures 5.26, 5.27, 5.28, and 5.29)

Ramp & Hold Input (mm)	**Optimized values for 6mm ramp & hold input**
2	$K_1 = 0.018$, $b_1 = 0.266$ $K_\infty(\varepsilon) = 66.87\varepsilon^4 - 49.24\varepsilon^3 + 13.17\varepsilon^2 - 0.97\varepsilon + 0.18$
2.5	$K_1 = 0.021$, $b_1 = 0.3$ $K_\infty(\varepsilon) = 67.61\varepsilon^4 - 49.79\varepsilon^3 + 13.32\varepsilon^2 - 0.98\varepsilon + 0.18$
3	$K_1 = 0.025$, $b_1 = 0.351$ $K_\infty(\varepsilon) = 67.8\varepsilon^4 - 49.93\varepsilon^3 + 13.36\varepsilon^2 - 0.98\varepsilon + 0.18$
3.5	$K_1 = 0.029$, $b_1 = 0.405$ $K_\infty(\varepsilon) = 67.74\varepsilon^4 - 49.88\varepsilon^3 + 13.35\varepsilon^2 - 0.98\varepsilon + 0.18$

4	$K_1 = 0.033$, $b_1 = 0.46$ $K_\infty(\varepsilon) = 67.58\varepsilon^4 - 49.77\varepsilon^3 + 13.31\varepsilon^2 - 0.98\varepsilon + 0.18$
4.5	$K_1 = 0.036$, $b_1 = 0.511$ $K_\infty(\varepsilon) = 67.45\varepsilon^4 - 49.67\varepsilon^3 + 13.28\varepsilon^2 - 0.98\varepsilon + 0.18$
5	$K_1 = 0.04$, $b_1 = 0.561$ $K_\infty(\varepsilon) = 67.38\varepsilon^4 - 49.62\varepsilon^3 + 13.27\varepsilon^2 - 0.98\varepsilon + 0.18$
5.5	$K_1 = 0.043$, $b_1 = 0.61$ $K_\infty(\varepsilon) = 67.38\varepsilon^4 - 49.62\varepsilon^3 + 13.27\varepsilon^2 - 0.98\varepsilon + 0.18$
6	$K_1 = 0.046$, $b_1 = 0.655$ $K_\infty(\varepsilon) = 67.42\varepsilon^4 - 49.65\varepsilon^3 + 13.28\varepsilon^2 - 0.98\varepsilon + 0.18$
6.5	$K_1 = 0.05$, $b_1 = 0.704$ $K_\infty(\varepsilon) = 67.45\varepsilon^4 - 49.67\varepsilon^3 + 13.29\varepsilon^2 - 0.98\varepsilon + 0.18$
7	$K_1 = 0.053$, $b_1 = 0.753$ $K_\infty(\varepsilon) = 67.38\varepsilon^4 - 49.62\varepsilon^3 + 13.27\varepsilon^2 - 0.98\varepsilon + 0.18$
7.5	$K_1 = 0.057$, $b_1 = 0.811$ $K_\infty(\varepsilon) = 67.14\varepsilon^4 - 49.45\varepsilon^3 + 13.23\varepsilon^2 - 0.97\varepsilon + 0.18$
8	$K_1 = 0.063$, $b_1 = 0.88$ $K_\infty(\varepsilon) = 66.31\varepsilon^4 - 48.83\varepsilon^3 + 13.06\varepsilon^2 - 0.96\varepsilon + 0.18$
8.5	$K_1 = 0.07$, $b_1 = 0.948$ $K_\infty(\varepsilon) = 64.85\varepsilon^4 - 47.75\varepsilon^3 + 12.77\varepsilon^2 - 0.94\varepsilon + 0.18$
9	$K_1 = 0.078$, $b_1 = 1.01$ $K_\infty(\varepsilon) = 63.68\varepsilon^4 - 46.89\varepsilon^3 + 12.54\varepsilon^2 - 0.92\varepsilon + 0.17$

Table 5. 8: The optimum material coefficients estimated for different ramp & hold inputs in a 125-node mesh

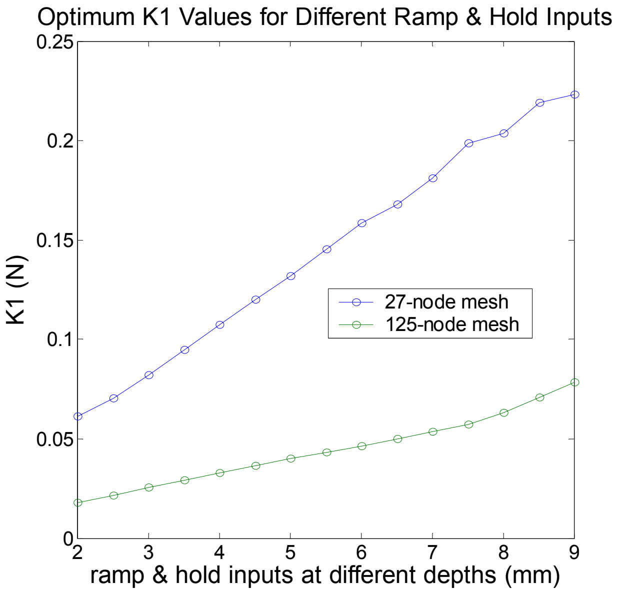


Figure 5. 26: The optimum K_1 values estimated for different ramp & hold inputs

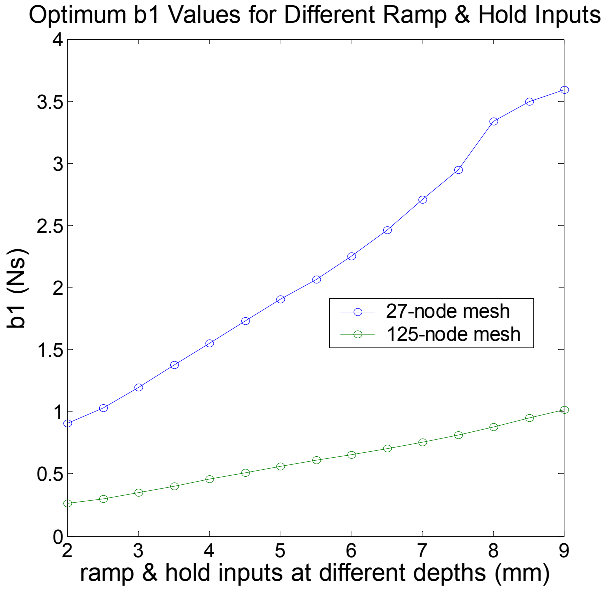


Figure 5. 27: The optimum b_1 values estimated for different ramp & hold inputs

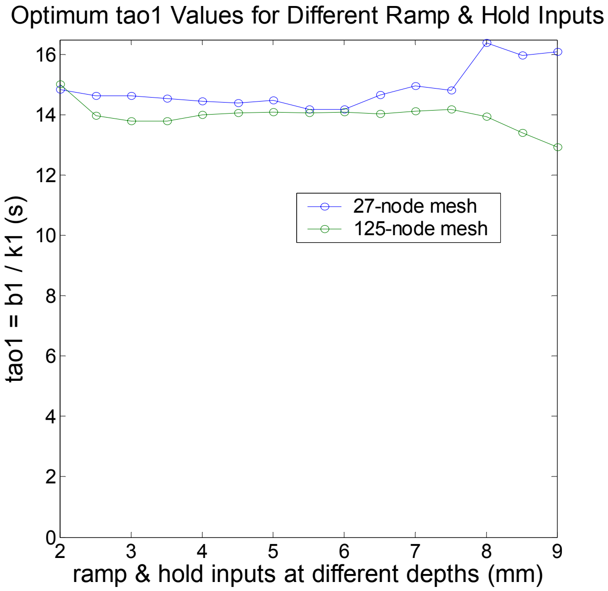


Figure 5. 28: Ratio of b_1 to K_1 (i.e., tao_1), for different ramp & hold inputs

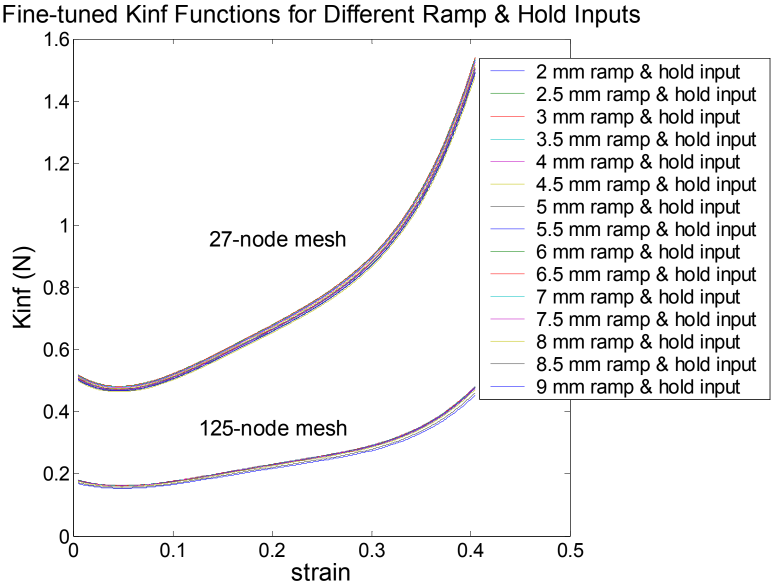


Figure 5. 29: The optimum $K_\infty(\varepsilon)$ functions estimated for different ramp & hold inputs

Chapter 6

CONCLUSIONS and FUTURE WORK

6.1 Conclusions

Soft tissues exhibit complex, nonlinear, anisotropic, nonhomogeneous behavior. Because of this nonhomogeneity, soft tissues have both coordinate- and direction-dependent properties. Time- and rate-dependent behavior caused by viscoelasticity is also common. Development of realistic organ-force models is an integral part of the surgical simulators that are used for training surgeons in real-like surgery settings. A realistic organ-force model should be able to present the complex nonlinear (i.e., strain-dependent) viscoelastic (i.e., time- and rate-dependent) mechanical properties of the real tissues.

In this thesis, we developed and presented parameter estimation techniques for a particle system that is able to mimic the complex nonlinear viscoelastic behavior of an experimentally-measured and characterized tissue-like phantom. The earlier studies on parameter estimation of particle-based systems are limited to estimation of the linear and nonlinear spring coefficients, mass distribution and topology identification. To our knowledge, none of the previous studies present a solution to the problem of material property estimation of a nonlinear and viscoelastic particle-based system based on the experimental data.

We first measured the strain-dependent nonlinear response and time-dependent viscoelastic response of a tissue-like silicon phantom using a robotic indenter fitted with a force sensor. We modeled the phantom using a lumped nonlinear viscoelastic Maxwell

Solid with N=1 and N=2, and characterized the material properties of the lumped model via curve-fitting. Then, we constructed a 3D massless SDN based on Moore Neighborhood. The elements of the network connecting the particles are nonlinear viscoelastic Maxwell Solids with N=1. The system is validated using ANSYS. We developed a set of novel optimization algorithms for estimating the material coefficients of the elements in the mesh such that the response of the simulated mesh is identical to the lumped model. The optimization algorithm first estimates the nonlinear strain-dependent elastic response of the network elements via static-loading experiments and then, estimates the viscoelastic material properties K_1 and b_1 using the ramp & hold experiments through simulation..

6.2 Future Work

It is discussed in Section 4.3.3 that Maxwell Solid with N=2 is a better approximation for the experimentally-measured relaxation force response. It is possible to construct a 3D mesh composed of Maxwell Solids with N=2, however, it is not straight-forward to come up with an optimization algorithm for the estimation of the parameters of such a mesh. The force-relaxation function of a single nonlinear Maxwell Solid with N=2 is given as:

$$F(t) = \mathrm{K}_{\infty}(Ac)Ac - Ab_1 \exp\left(-t\frac{K_1}{b_1}\right)\left(1-\exp\left(c\frac{K_1}{b_1}\right)\right) - Ab_2 \exp\left(-t\frac{K_2}{b_2}\right)\left(1-\exp\left(c\frac{K_2}{b_2}\right)\right) \quad (6.1)$$

Finding an optimization scheme for Equation 6.1 is not trivial because the force-relaxation behavior of each element in the mesh is affected by K_1, K_2, b_1, and b_2. It is difficult to come up with intelligent guesses for these parameters based on a methodology which minimizes the residual between the reference and the simulated responses. The first idea that comes to my mind is to apply simulated-annealing which will assign random guesses for those parameters within some range and then try to find the guesses that result in the minimum residual. Obviously, this approach does not return the optimum solution, In fact, the solution may stuck at local minima.

In this thesis, the optimization results are presented for 27- and 125-node-meshes. In the future, we are planning to test the optimization algorithm with a much finer mesh to investigate with the effect of mesh resolution on the results. Also, we used voxels (i.e. cubes) for the geometric model of the mesh, but the goal is to run the same optimization algorithm with different size and type of geometric mesh elements In this way, experimentally measured nonlinear and viscoelastic tissue properties can be integrated into a reference FE model in arbitrary shape to estimate the material properties of the corresponding 3D particle model.

Currently, the SDN is time-discretized and solved via Newton's Method with Line search due to the nonlinear set of equations in the system. The computational complexity of the algorithm does not allow us to run the model in real-time. While this is not the most critical issue for the off-line optimization computations, it is a concern for the surgical simulations. We are planning to develop a faster algorithm to solve the nonlinear set of equations for achieving real-time performance.

We did not consider the mass of the object during the formulation of the particle-based model in Chapter 5. Mass of the object introduces inertial effects and should be considered for more realistic modeling. However, adding mass resulted in oscillatory motion in our initial attempts, which did not allow us to apply a logical optimization scheme for parameter estimation. We are planning to work on this problem in the future.

BIBLIOGRAPHY

[1] C. Basdogan, **M. Sedef,** Matthias Harders, Stefan Wesarg, “VR-Based Simulators for Training in Minimally Invasive Surgery,” *IEEE Computer Graphics and Applications,* vol. 27, no. 2, 2007, pp. 54-66.

[2] E. Samur, **M. Sedef,** C. Basdogan, L. Avtan, O. Duzgun, “A Robotic Indenter for Minimally Invasive Measurement and Characterization of Soft Tissue Behavior,” *Medical Image Analysis*, vol. 11, no.4, 2007, pp. 361-373.

[3] **M. Sedef,** E. Samur, and C. Basdogan, “Real-Time Finite Element Simulation of Linear Viscoelastic Soft Tissue Behavior Based on Experimental Data,” *IEEE Computer Graphics and Applications,* vol. 26, no. 6, 2006, pp. 58-68.

[4] **M. Sedef,** E. Samur, and C. Basdogan, “Visual and Haptic Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data,” *IEEE 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems,* March 25-26, 2006, Washington D.C., USA

[5] E. Samur, **M. Sedef,** C. Basdogan, L. Avtan, O. Duzgun, “A Robotic Indenter for Minimally Invasive Characterization of Soft Tissues,” *Proceedings of the 19th International Conference on Computer Assisted Radiology and Surgery*, vol. 1281, pp. 713-718, June 2005, Berlin.

[6] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, M.A. Srinivasan, “Haptics in Minimally Invasive Surgical Simulation and Training,” *IEEE Computer Graphics and Applications*, vol. 24, no.2, 2004, pp. 56-64.

[7] A. Liu, F. Tendick, K. Cleary, C. Kaufmann, “A Survey of Surgical Simulation: Applications, Technology, and Education,” *Presence*, vol. 12, no. 6, 2003, pp. 599-614.

[8] S. Srinivasan, D.P. Mital, and S. Haque, “A Quantitative Analysis of the Effectiveness of Laparoscopy and Endoscopy Virtual Reality Simulators,” *Computers & Electrical Engineering*, vol. 32, no. 4, 2006, pp. 283-298.

[9] “Simulators for Training: Assessment, Validation and Acceptance Strategies,” Executive Summary of the Simulator for Training Workshop, *Medicine Meets Virtual Reality*, 2003, www.medicalsim.org/virgil/Exec_Summary.pdf.

[10] K.D. Reinig, C.G. Rush, H.L. Pelster, V.M. Spitzer, J.A. Heath, “Real-Time Visually and Haptically Accurate Surgical Simulation,” *Studies in Health Technology and Informatics*, vol. 29, 1996, pp. 542-545.

[11] R. Paget, M. Harders, and G. Szekely, “A Framework for Coherent Texturing in Surgical Simulators,” *Proc. 13th Pacific Conf. Computer Graphics and Applications,* 2005, pp. 112-114; ftp://ftp.vision.ee.ethz.ch/publications/proceedings/eth_biwi_00354.pdf.

[12] M. Kauer, “Inverse Finite Element Characterization of Soft Tissues with Aspiration Experiments,” doctoral dissertation, Inst. of Mechanical Systems, Swiss Fed. Inst. of Technology (ETH), Zurich, 2001.

[13] M.P. Ottensmeyer, “Minimally Invasive Instrument for In Vivo Measurement of Solid Organ Mechanical Impedance,” doctoral dissertation, Dept. of Mechanical Eng., Massachusetts Inst. of Technology, 2001.

[14] J. Kim, “Virtual Environments for Medical Training: Graphical and Haptic Simulation of Tool-Tissue Interactions,” doctoral dissertation, Dept. of Mechanical Eng., Massachusetts Inst. of Technology, 2003.

[15] M.P. Ottensmeyer, A.E. Kerdok, R.D. Howe, S.L. Dawson, “The effects of testing environment on the viscoelastic properties of soft tissues,” *Proc. of International Symposium on Medical Simulation*, 2004, pp. 9-18.

[16] F.J. Carter, T.G. Frank, P.J. Davies, D. McLean, A. Cuschieir, "Measurements and modeling of the compliance of human and porcine organs," *Medical Image Analysis,* vol. 5, 2001, pp. 231-236.

[17] B.K. Tay, N. Stylopoulos, S. De, D.W. Rattner, M.A. Srinivisan, "Measurement of in-vivo force response of intra-abdominal soft tissues for surgical simulation," *Medicine Meets Virtual Reality*, 2002, pp. 514-519.

[18] J.D. Brown, J. Rosen, Y.S. Kim, L. Chang, M.N. Sinanan, B. Hannaford, "In-vivo and in-situ compressive properties of porcine abdominal soft tissues," *Medicine Meets Virtual Reality*, vol. 94 2003, pp. 26-32.

[19] I. Brouwer, J. Ustin, L. Bentley, A. Sherman, N. Dhruv, F. Tendick, "Measuring in vivo animal soft tissue properties for haptic modeling in surgical simulation," *Medicine Meets Virtual Reality*, 2001, pp. 69-71.

[20] L. Gao, K.J. Parker, R.M. Lerner, S.F. Levinson, "Imaging of the elastic properties of tissue: a review," *Ultrasound Med. Biol*. vol. 22, 1996, pp. 959–977.

[21] Z. Yong-Ping, A.F. Mak, "An ultrasound indentation system for biomechanical properties assessment of soft tissues in-vivo," *IEEE Transactions on Biomedical Engineering*, vol. 43, 1996, pp. 912-918.

[22] A. Manduca, T.E. Oliphant, M.A. Dresner, J.L. Mahowald, S.A. Kruse, E. Amromin, J.P. Felmlee, J.F. Greenleaf, R.L. Ehman, "Magnetic resonance elastography: Non-invasive mapping of tissue elasticity," *Medical Image Analysis*, vol. 5, 2001, pp. 237–254.

[23] L. Han, J.A. Noble, M. Burcher, "A novel ultrasound indentation system for measuring biomechanical properties of in vivo soft tissue," *Ultrasound in Med. & Biol.*, vol. 29, 2003, pp. 813-823.

[24] U. Kuhnapfel, H.K. Cakmak, and H. Maab, "Endoscopic Surgery Training Using Virtual Reality and Deformable Tissue Simulation," *Computers & Graphics,* vol. 24, no. 5, Oct. 2000, pp. 671-682.

[25] M. Bro-Nielsen and S. Cotin, “Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation,” *Computer Graphics Forum*, vol. 15, no. 3, 1996, pp. 57-66.

[26] C. Basdogan, C. Ho, and M.A. Srinivasan, “Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration,” *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 3, 2001, pp. 267-285.

[27] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, S. Weghorst, “Real-Time Finite Element Modeling for Surgery Simulation: An Application to Virtual Suturing,” *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 3, 2004, pp. 314-325.

[28] D. James and D.K. Pai, “A Unified Treatment of Elastostatic and Rigid Contact Simulation for Real Time Haptics,” *Haptics-e, Electronic J. Haptics Research*, vol. 2, no. 1, 2001, http://www.haptics-e.org/Vol_02/he-v2n1.pdf.

[29] I.F. Costa and R. Balaniuk, “LEM—An Approach for Real Time Physically Based Soft Tissue Simulation,” *Proc. IEEE Int’l Conf. Robotics and Automation* (ICRA 01), IEEE Press, 2001, vol. 3, pp. 2337-2343.

[30] S. Cotin, H. Delingette, and N. Ayache, “A Hybrid Elastic Model Allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation,” *Visual Computer*, vol. 16, no. 8, 2000, pp. 437-452.

[31] G. Picinbono, H. Delingette, and N. Ayache, “Non-linear and Anisotropic Elastic Soft Tissue Models for Medical Simulation,” *Proc. IEEE Int’l Conf. Robotics and Automation* (ICRA 02), IEEE Press, 2002, vol. 2, pp. 1370-1375.

[32] J. Brown, K. Montgomery, J.C. Latombe, M. Stephanides, “A Microsurgery Simulation System,” *Proc. 4th Int’l Conf. Medical Image Computing and Computer-Assisted Intervention* (Miccai 01), Springer, 2001, pp.137-144.

[33] S. De, Y.J. Lim, M. Manivannan, M.A. Srinivasan, "Physically Realistic Virtual Surgery Using the Point-Associated Finite Field (PAFF) Approach," *Presence: Teleoperators & Virtual Environments*, vol. 15, no. 3, 2006, pp. 294-308.

[34] D. Bielser, P. Glardon, M. Teschner, M. Gross, "A State Machine for Real-Time Cutting of Tetrahedral Meshes," *Graphical Models*, vol. 66, no. 6, 2004, pp. 398-417.

[35] N. Molino, Z. Bao, and R. Fedkiw, "A Virtual Node Algorithm for Changing Mesh Topology during Simulation," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 385-392.

[36] C. Basdogan, C Ho, and M. Srinivasan, "Simulation of Tissue Cutting and Bleeding for Laparoscopic Surgery Using Auxiliary Surfaces," *Medicine Meets Virtual Reality*, 1999, pp. 38-44

[37] H.W. Nienhuys and A.F. van der Stappen, "A Surgery Simulation Supporting Cuts and Finite Element Deformation," *Proc. 4th Int'l Conf. Medical Image Computing and Computer-Assisted Intervention* (Miccai 01), Springer, 2001, pp. 153-160.

[38] D. Steinemann, M. harders, M. Gross, G. Szekely, "Hybrid Cutting of Deformable Solids," *Proc. IEEE Conf. Virtual Reality* (VR 06), IEEE CS Press, 2006, pp. 35-42.

[39] T. Chanthasopeephan, J.P. Desai, and A.C.W. Lau, "Deformation Resistance in Soft Tissue Cutting: A Parametric Study," *Proc. 12th Int'l Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems* (Haptics 04), IEEE CS Press, 2004, pp. 323-330.

[40] M. Mahvash and V. Hayward, "Haptic Rendering of Cutting: A Fracture Mechanics Approach," *Haptics-e, Electronic J. Haptics Research*, vol. 2, no. 1, 2001, http://www.haptics-e.org/Vol_02/he-v2n3.pdf.

[41] J. Zatonyi, R. Paget, G. Szekely, M. Grassi, M. Bajka, "Real-Time Synthesis of Bleeding for Virtual Hysteroscopy," *Medical Image Analysis*, vol. 9, no. 3, 2005, pp. 255-266.

[42] M. Chabanas and E. Promayon, "Physical Model Language: Towards a Unified Representation for Continuous and Discrete Models," *Proc. Int'l Symp. Medical Simulation* (ISMS 04), LNCS 3078, Springer, 2004, pp. 256-266.

[43] M.C. Cavusoglu, T. Goktekin, and F. Tendick. “GiPSi: A Framework for Open Source/Open Architecture Software Development for Organ Level Surgical Simulation,” *IEEE Trans. Information Technology in Biomedicine*, vol. 10, no. 2, 2006, pp. 312-321.

[44] P. Leskovsky, M. Harders, and G. Szekely, “A Web-Based Repository of Surgical Simulator Projects,” *Proc. Medicine Meets Virtual Reality,* IOS Press, 2006, pp. 311-315.

[45] F.J. Carter, M.P. Schijven, R. Aggarwal, T. Grantcharov, N.K. Francis, G.B. Hanna, J.J. Jakimovics, “Consensus Guidelines for Validation of Virtual Reality Surgical Simulators,” *Surgical Endoscopy*, vol. 19, no. 12, 2005, pp. 1523-1532.

[46] N. Stylopoulos, S. Cotin, S. Dawson, M. Ottensmeyer, P. Neumann, R. Bardsley, M. Russel, P. Jackson, D. Rattner, “CELTS: A Clinically-Based Computer Enhanced Laparoscopic Training System,” *Studies in Health Technology and Informatics*, vol. 94, 2003, pp. 336-342.

[47] Y.C. Fung, Biomechanics: Mechanical properties of living tissues. 2nd ed. Springer-Verlag, New York, 1993.

[48] C.A. Felippa, “Nonlinear Finite Element Methods,” Department of Aerospace Engineering Sciences and Center for Space Structures and Controls, 2001.

[49] R. Lakes, Viscoelastic Solids. CRC Press, 1998.

[50] A. Ya. Malkin, Viscoelasticity

[51] P. Macioce, “Viscoelastic Damping 101”.
http://www.roushind.com/news_downloads/white_papers/SV_Damping101.pdf

[52] S. Shaw, J.R. Whiteman, “Robust Adaptive Finite Element Schemes for Viscoelastic Solid Deformation: An Investigative Study”, Final report for the US Army’s European Research Office Seed Project, 1999.

[53] W.N. Findley, J.S. Lai, K. Onaran, Creep and Relaxation of Nonlinear Viscoelastic Materials, Dover, New York, 1976.

[54] D. Roylance, Engineering Viscoelasticity, MIT, 2001

[55] K.M. Pryse, A. Nekouzadeh, G.M. Genin, E.L. Elson, G.I. Zahalak, “Incremental Mechanics of Collagen Gels: New Experiments and a New Viscoelastic Model,” *Annals of Biomedical Engineering*, vol. 31, 2003, pp. 1287-1296.

[56] M.C. Cavusoglu, D. Feygin, F. Tendick, “A critical study of the mechanical and electrical properties of the PHANToM haptic interface and improvements for high-performance control,” *Presence: Teleoperators and Virtual Environments*, vol. 11, 2002, no. 6, pp. 555-568.

[57] J. Rosen, B. Hannaford, M.P. MacFarlane, M.N. Sinanan, “Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-experimental performance evaluation**,”** *IEEE Transactions on Biomedical Engineering*, vol. 46, 1999, pp. 1212-1221.

[58] T. Hu, J.P. Desai, “Characterization of Soft-Tissue Material Properties: Large Deformation Analysis,” *Proceedings of International Symposium on Medical Simulation* (Lecture Notes in Computer Science LNCS 3078), 2004, pp. 28-37, Cambridge, MA, June 17-18.

[59] D. Valtorta, E. Mazza, “Dynamic measurement of soft tissue viscoelastic properties with a torsional resonator device,” *Medical Image Analysis*, vol. 9, 2005, no. 5, pp. 481-490.

[60] F. Tendick, M. Downes, T. Goktekin, M.C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, M. Hegarty, L.W. Way, “A virtual environment testbed for training laparoscopic surgical skills,” *Presence*, vol. 9, 2000, no. 3, pp. 236-255.

[61] M. Bro-Nielsen, D. Helfrick, B. Glass, X. Zeng, M. Phil, H. Connacher, “VR simulation of abdominal trauma surgery,” *Medicine Meets Virtual Reality: Arts Science and Technology*, pp. 117-123

[62] K. Montgomery, L. Heinrichs, C. Bruyns, S. Wildermuth, C. Hasser, S. Ozenne, D. Bailey, “Surgical simulator for hysteroscopy: A case study of visualization in surgical training,” *IEEE Visualization,* 2001.

[63] D. Bielser, M. H. Gross, “Open surgery simulation,” *Medicine Meets Virtual Reality*, 2002.

[64] E. Keeve, S. Girod, R. Kikinis, B. Girod, “Deformable modeling of facial tissue for craniofacial surgery simulation,” *Computer Aided Surgery*, vol. 3, 1998, pp.228-238.

[65] National Instruments, Analog input card, NI PCI-6034
http://sine.ni.com/nips/cds/view/p/lang/en/nid/11916

[66] http://www.smooth-on.com/

[67] O. Etzmuss, J. Gross, and W. Strasser, “Deriving a Particle System from Continuum Mechanics for the Animation of Deformable Objects,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 9, 2003, no.4, pp. 538 – 550.

[68] G. Bianchi, M. Harders, and G. Székely “Mesh Topology Identification for Mass-Spring Models,” *Medical Image Computing and Computer-Assisted Intervention,* 2003, (MICCAI 03).

[69] G. Bianchi, B. Solentharel, G. Székely, M. Harders, “Simultaneous Topology and Stiffness Identification for Mass-Spring Models based on FEM Reference Deformations,” *Medical Image Computing and Computer-Assisted Intervention,* 2004 *(MICCAI 04).*

[70] Z. Pezzementi, D. Ursu, S. Misra, A.M. Okamura, “Modeling Realistic Tool-Tissue Interactions with Haptic Feedback: A Learning-Based Method,” *16th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems,* 2008.

[71] D. d'Aulignac, M.C. Cavusoglu and C. Laugier, “Modeling the Dynamics of the Human Thigh for a Realistic Echographic Simulator with Force Feedback,” *Medical Image Computing and Computer-Assisted Intervention,* 1999, (MICCAI 99).

[72] B.A. Lloyd, G. Székely, and M. Harders, “Identification of Spring Parameters for Deformable Object Simulation,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, 2007, no. 5, pp. 1081 – 1094.

[73] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing. 2nd ed. Cambridge University Press, Cambridge, 1992.