

TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR ANGEWANDTE MECHANIK

Real Time Haptic Simulation of Deformable Bodies

Dipl.-Ing. Univ. Chen Zhao

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. Gunther Reinhart

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. habil. Heinz Ulbrich
2. Univ.-Prof. Dr.-Ing. Dr.-Ing. habil. Alois Knoll

Die Dissertation wurde am 16.09.2009 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen am 08.02.2010 angenommen.

For My Family

Acknowledgement

This dissertation presents my research work done in the institute of Applied Mechanics at the Technical University of Munich (Technische Universität München), as a research assistant. I would like to thank my Ph.D. supervisor and the examination board. At the same time, I appreciate my friends and colleagues, who have helped me by discussions of different topics and review of my dissertation, as well as a number of people who provided comments, support and advice.

I would like to gratefully and sincerely thank my family and relatives, especially my parents and my grandparents. They support me continuously in every situation without any conditions. With their instruction and guide, I could always be heading in the correct direction. I would also like to express my gratitude to my wife for her supporting. Without them, everything is impossible. Finally, I have to give my respect to my hometown. Her land and people have reared me. Furthermore in times of difficulties, I can always draw courage and power from her culture and her history to overcome them.

Zürich, 5 March 2010

Zhao, Chen

Contents

1. Introduction	1
1.1. Haptics	1
1.2. Contact Model and Haptic Simulation	4
1.3. Objectives and Organizations	7
2. Geometric Modeling	10
2.1. Geometric Measurement	10
2.2. Surface Reconstruction	11
2.2.1. Voronoi Diagram	13
2.2.2. Delaunay Triangulation	14
2.2.3. Power Crust	15
2.3. Mesh Generation	15
2.4. Contact Detection	16
3. Modeling of Solid Material	20
3.1. Constitutive Equation	20
3.1.1. Linear Elastic Material	21
3.1.2. Linear Viscous Material	22
3.1.3. Linear Viscoelastic Material	22
3.2. Finite Element Method	25
3.2.1. Basic Concept	25
3.2.2. Boundary Conditions	27
3.3. Linear Elastic Model	28
3.4. Linear Viscoelastic Model	30
3.5. Inhomogeneous Model	32
4. Contact Force Evaluation	36
4.1. Finite Element Evaluation	36
4.1.1. Conjugate Gradient Method	36
4.1.2. Practical Implementation	38
4.2. Haptic Rendering	40
4.3. Analytical Contact Force Model	40
4.4. Data Fusion Algorithms	42
4.4.1. Extended Kalman Filter	44
4.4.2. Example	46
5. Model Identification	48
5.1. Problem Statement	48
5.2. Model Learning	48
5.2.1. Location Sample Estimation	49
5.2.2. Preprocessing	51
5.2.3. Location Sample Classification	52

5.2.4. Material Region Estimation	55
5.3. Model Parameter Optimization	57
5.3.1. Optimization Problem	58
5.3.2. Gauss-Newton Method	58
5.3.3. Global Optimization Algorithm	61
5.4. Implementation of Model Identification	62
5.4.1. Linear Elastic Model Identification	63
5.4.2. Viscoelastic Model Identification	63
5.4.3. Inhomogeneous Model Identification	64
5.4.4. Analytical Contact Force Model	66
5.4.5. Extended Kalman Filter	68
5.5. Adaptive Model Identification	69
5.6. Model Verification and Update	71
6. Experiments and Simulations	73
6.1. Experimental System	73
6.1.1. Experimental Robot	73
6.1.2. Robot Kinematics	75
6.1.3. Communication	79
6.2. Virtual Reality	79
6.3. Polyurethane	86
6.3.1. Linear Model	86
6.3.2. Viscoelastic Model	87
6.4. Inhomogeneous Material	88
6.4.1. Modeling	89
6.4.2. Results	90
6.5. Telerobotic Operation	91
6.5.1. Problem Statement	91
6.5.2. Telepresence Scenario	92
6.5.3. Modeling	96
6.5.4. Results	97
6.6. Animal Tissue	99
6.6.1. Problem Statement	99
6.6.2. Experimental Setting	99
6.6.3. Modeling	99
6.6.4. Results	102
7. Conclusion	105
A. Appendix	107
A.1. QR-Decomposition	107
A.2. Robotic End Effector	108
A.3. Material Value in Section 6.2	108
Bibliography	111

1. Introduction

Babies can distinguish some textures right from birth.

1.1. Haptics

Touch sense is one of the most important ways, which give human being the senses of its environment. By some means, it is the oldest sense of a person, the first sense of baby and the last remained sense before the death. In medicine, the human senses are defined as the traditional five sensations. The four among them are defined as special senses, i.e. the vision, audition, gustation and olfaction. Whereas, the touch sensation is known as tactile. It is also noted as the somatosensory system, including verity modalities of pressure, vibration, pain, kinesthesia and temperature [4]. Each modality includes the receptors, the central nervous system and the pathways, Fig. 1.1. The receptors of touch sensation are mechanoreceptors, which are sensitive to mechanical signals, such as pressure and vibration, and can convert them to nervous signals. They are located in human skin, muscles, joints and many other organs. However they are not uniformly distributed in the whole body. The hands contain a large number of these receptors and contribute more to this sensation system than any other parts of human body. These effects are illustrated by using the homunculus [33]. The signals from the mechanoreceptors are transmitted through the pathway to the central nervous system. The information from the complete somatosensory system are integrated and processed in the central nervous system. By this means, one person can obtain the global image of his whole body [86]. What it can do, is much more than normal considerations, for instance a health person can walk and keep balance without vision, this is based on the internal contact information of muscles, skeletons and tendons. Within the Collaborative Research Centre SFB453 “High-Fidelity Telepresence and Teleaction” supported by the German Research Foundation (DFG), the technical representation of sensations in a remote or inaccessible environment is investigated. Specially, the subproject M7 concentrates on the presentation of haptic information.

In this thesis, the contact situations are different manual operations, i.e. contacts between hands and objects. Therefore, the touch sensation of skin should be introduced briefly as biological background. It can be divided into two kinds for human being, i.e. the glabrous skin and hairy skin. The touch sensation of glabrous skin have attracted more interest in the touch sensation, whereas the cells in hairs have also mechanoreceptors, and those in the cochlea are the most sensitive receptors, which can convert air vibrations into audio signals. This is already investigated as another theme [14]. The nerve endings of the first kind have been detailed in some works [86, 45], and the two major mechanoreceptors in skin are the Meissner’s

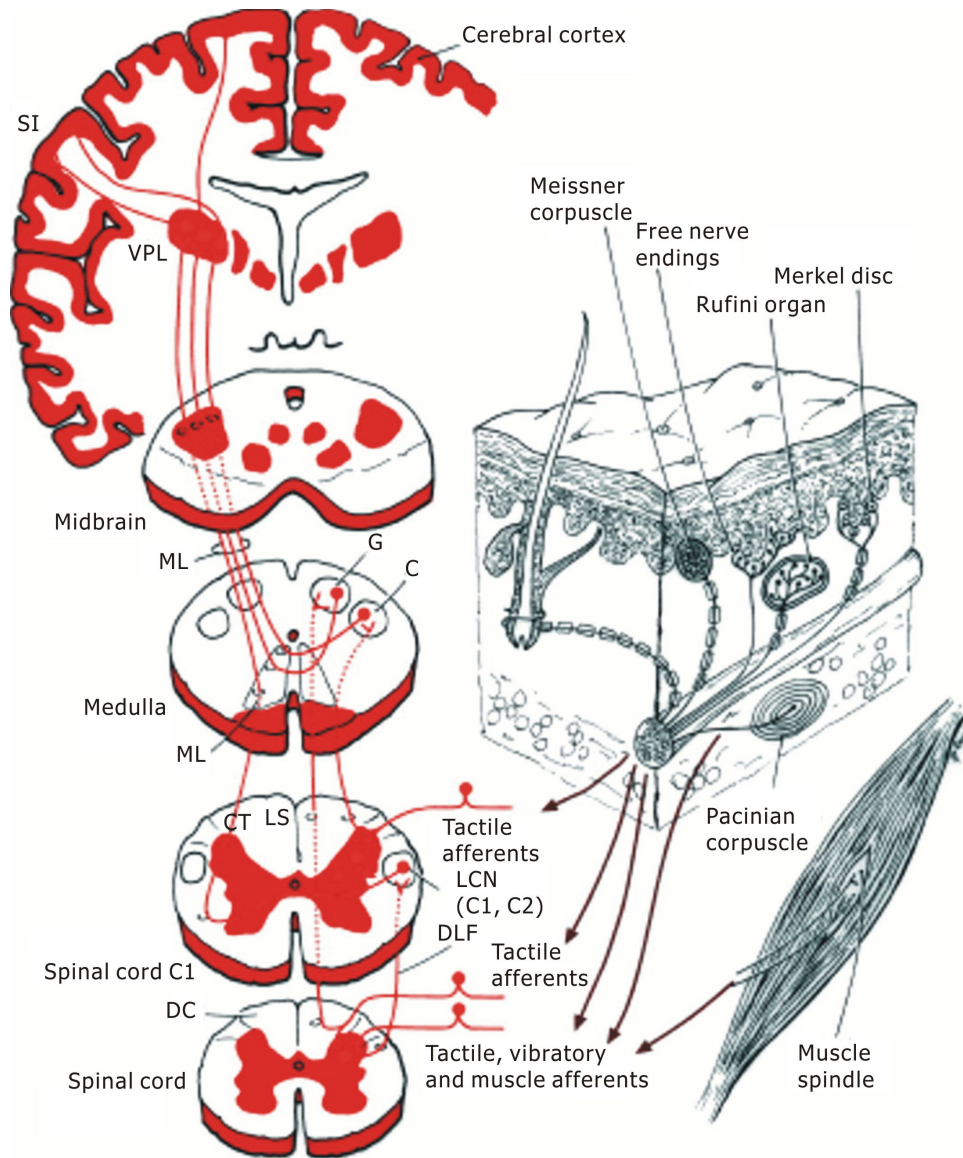


Figure 1.1.: The Somatosensory system for contact and position sensation, including the receptors and the central pathway and the central nervous system. The Meissner's corpuscle and the Pacinian corpuscle are important mechanical receptors, [45].

corpuscles for light and low frequency signals [21] and the Pacinian corpuscles for deep pressure and high frequency mechanical vibrations, Fig. 1.2. It should be noted that during human movement, the information from human receptors and actuators, for instance muscles, can not be separated completely. By this means, the touch sensation provides humans the necessary information in almost all areas, whenever actions are going to be done, from handling tools to manipulating modern instruments, from household chores to high level operations and accurate surgery.

In robotics and computer science, the touch sense is generally called haptics, which comes from the according Greek word. With the development of mechanical engineering, mechatronics and computer science, the haptic technology is rising as a new research area in recent years. It can present the haptic sensation to users

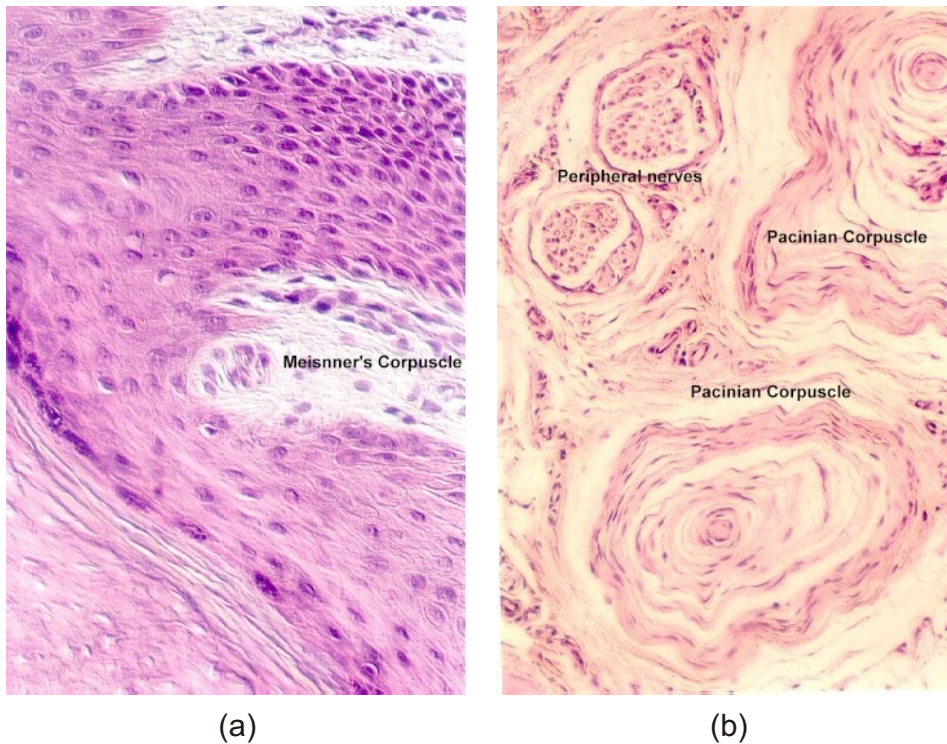


Figure 1.2.: Haptic receptors of skin, the Meissner's corpuscle (a) and the Pacinian corpuscle (b) [24]

in virtual and real environments, which may be remote, dangerous or unaccessible, for instance blue water explorers and telerobots. By this means, these systems can be regarded as an extension of the human haptic sensation, illustrated in Fig. 1.3. From the mechanical aspect, the haptic information is in the form of contact force in connection with the corresponding motion characteristics, such as boundary displacements and velocities.

Hence, in robotics and virtual reality, haptic information is also important to users for robotic-environment interactions and sensing of a virtual environment [96]. However in many situations, accurate contact information is unavailable. This may lead to difficulties and inconvenience during operations in these applications. Then the haptic simulations have been developed to solve these problems.

As shown above, the human sensation system is extremely complicated, for instance, two concurrent contacts, which are as close together as $2[mm]$, on the finger can already be felt separately. Hence, it should be noticed that the modern science and technology at present can not yet rebuilt the human sensation system completely. Furthermore, there are always different distortion sources during the measurement, representation and transmission of the haptic information. Therefore, the goal of the haptic technology is to take the most important information into consideration, and to provide the possibly accurate high-fidelity haptic information in another environment to users.

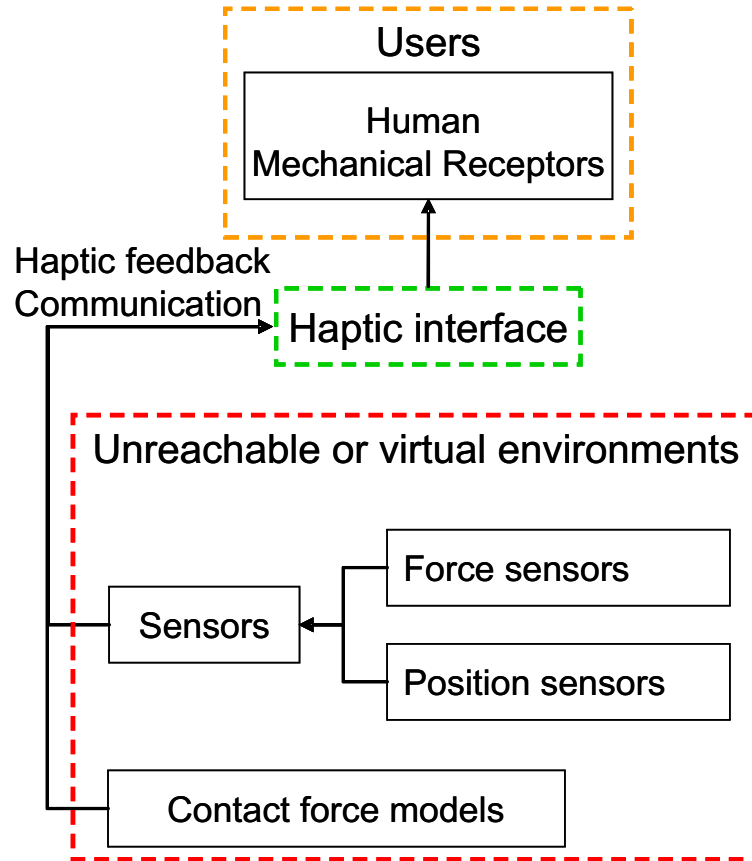


Figure 1.3.: Haptic technology. Three major components are involved, the real or virtual environment information, the haptic interface including the associated transmission of force feedback, and the human operator. The haptic information in a remote environment can be presented to the user through the haptic interface. This diagram illustrates mainly the information flow for the haptic sensation. In the opposite direction, human operations can be performed in the environment using some actuators.

1.2. Contact Model and Haptic Simulation

In many situations, the measurement of contact force may be unavailable in real-time, for instance it may be delayed during communication. In virtual reality, haptic information must be simulated. In this cases, the contact force models can be used for the simulation of the contact reaction force, i.e. the above mentioned haptic simulations are based on the models.

The contact force models of deformable solids can simulate the materials' internal mechanical states, the stress-strain states under boundary conditions, such as boundary tractions and displacements. Eventually, the contact reaction forces and the displacements on the boundary are determined by numerical simulations. These force models are investigated in many areas, for instance surgical robots, virtual reality, aerospace technology and computer graphics. The major requirements of the contact force models are accuracy and computational expense.

Research in this area is based on diverse modeling methods. Analytical models are potentially promising approaches [35]. They are easy for implementation, economical

in computing time, and they can provide accurate simulations in many applications. However, these methods are not suitable for the modeling of continuous materials, because the behavior of these materials is analytically unsolvable. Thus, for the modeling of general solid objects, numerical approaches are necessary. The mass-spring model [69] has been implemented in some works, however the finite volume method (FVM) [114], the boundary element method (BEM) [61], and the finite element method (FEM) can provide better performance and flexibility for modeling of objects with complex material behavior and geometry. Generally, the FEM is more accurate than the FVM. For solid deformable bodies, a number of research using the FEM has shown its capability for high level contact force simulations in varying situations, [125, 74, 31].

From application aspects, virtual reality is a notable area of these force models, from computer games to professional simulators. The haptic models give user the possibility to feel the “virtual” environment “really and truly”. Another important application area is robotics. The performance of telerobots can be enhanced significantly by integration of force feedback. However commutation time delay is an important challenge to applying the force measurements as force feedback. To overcome this difficulty, telepresence systems with contact force models can be used, to estimate the contact situation and to compensate for the time delay [124, 27, 101]. These telerobotic systems can assist human being in many areas, such as the onboard operations of satellites and other space vehicles, blue water explorations and bomb defusing. Another high interest research area of his interest is medical robotics technology, as a new kind of instruments [12], including surgery robots and medical training simulators. In connection with these equipments, new treatments can be developed, such as telemedicine. It is reasonable to expect, that quite a number of patients may benefit to recover quickly by using these new technologies in the future.

For material behavior, the linear elastic material is implemented in a number of research projects, because of its simplicity of implementation and low complexity of computation. Furthermore, most materials can be modeled as linear elastic material in some range. Hyperelastic, also known as the Green elastic or rubber-like, materials also attract some researchers, as a kind of typical and idealized models [71]. With the rapid development of computer science, more and more interest has been shown to the modeling of viscoelastic materials, for their time dependent stress-strain relations, [52, 62, 22], since almost all natural solid materials simultaneously have the elastic and viscous behavior, more or less. Especially, the development of the above mentioned medical technology accelerates the research of accurate haptic models of animal tissues in the recent years [103, 102], and the viscoelastic material is the most suitable theoretical model for these objects [43]. For continuum mechanics and the analytical description of diverse materials, the work of Haupt [52] may be referred to, for more detailed information.

In practice, the contact force models should always simulate some expected objects. This condition is one of the dominant requirements of the force models, especially for advanced applications. This ability can be measured as the comparability of the haptic simulations and the according reference data. Hence it is also a measure for the accuracy of the models. For this reason, model identification and optimization are necessary. In many works, the optimization is formulated as a least squares

problem. To solve this problem, curve fitting and other optimization algorithms are investigated. Generally these solving methods are based on the Gauss-Newton method. However the original Gauss-Newton method can not guarantee the global minimum of the objective functions. This is especially important for the optimizations with a number local minima. There are different algorithms and ideas to obtain the global minimum [58], for instance implicit filtering [46]. In this way, the Gauss-Newton method can be modified for global optimization.

In order to model objects, which are completely unknown in the beginning, a model learning procedure is normally necessary, using initial data. Artificial neural networks (ANN), which are a series of biomimetic methods of brains, are widely applied for data mining and model learning [98]. Many mathematical approaches can be implemented in the form of an ANN. Some of them are based on traditional statistic methods, such as the principal component analysis (PCA) and some classification methods. ANN can also be employed for the parameter estimation of finite element models [90, 51].

The computational expense is another important requirement for contact force models, and it is frequently the main challenge to applying the finite element (FE) models in real-time applications, especially for complex nonlinear FE models. Some methods have been investigated to solve this problem, for instance the multi-mesh structure [6], precalculation and prerecording [103]. However the computational efficiency of multi-mesh FE models is also limited because of the FEM, and the multi-mesh structure may influence the performance and the accuracy of force simulations, since the numerical evaluation depends on the model mesh as well. And the prerecording may restrict the flexibility and robustness of the method in many applications, because the objects have to predefined.

In this thesis, to deliver real time haptic rendering, the FEM based haptic model is integrated with an analytical contact force model (ACFM), which is less accurate, but available with high output rate, such as more than $1000[Hz]$. The information from both models is fused by using an extended Kalman filter (EKF). Two main purposes can be achieved with this implementation. One is smoothed haptic simulations without strong step changes, and the other is the real time reaction by taking boundary displacement information into account during the computation of the FE models.

The general procedure of contact force modeling and haptic simulations using the finite element method begins with geometric modeling. The first steps is to create the geometric (volume) models of the objects with mesh structure. Subsequently, the constitutive equations should be implemented in the volume models. Then the force models have not only the volume definition but also the mechanical properties. This means they can describe deformable bodies. Model identification and model learning are executed to adjust the model behavior to specific object. Eventually, the models are prepared for haptic simulations, and during the simulations, the models can be verified continuously, Fig. 1.4.

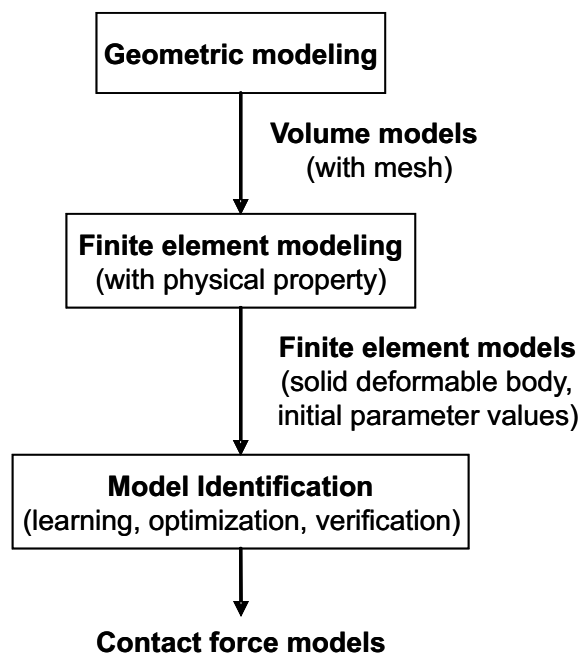


Figure 1.4.: The modeling procedure of the contact reaction force model.

1.3. Objectives and Organizations

The thesis focuses on the modeling of the mechanical behavior of deformable solid objects using the finite element method. The goal is to develop the modeling methods for real time haptic simulations with the consideration of the dominant conditions, accuracy and computational expense. For real time haptic simulations, the output rate is critical, especially for high level applications. The basic criterion is that the haptic rendering should be smooth enough without significant effects on the user. For this case, the computing time for force evaluation should not be greater than $20 \sim 30[ms]$. Further, in order to take full advantage of the haptic devices, an output rate of $1000[Hz]$ is expected. And for the accuracy, as mentioned above, the model identification is required.

The performance of the contact force models depends on the physical properties of the materials, the contact conditions and the complexity of implementations. The fundamental thought here is to develop a set of modeling methods as modules, including geometric modeling, material modeling, model identification and varying approaches to reduce the computing time. For material modeling, just the basic and applicable material characteristics have been taken into account. In order to model a concrete object efficiently, the modeling modules should be combined and integrated appropriately. Therefore, besides the different modeling methods, an adaptive method is also developed to select the most suitable material model and modeling methods for a given object.

There are diverse theoretical models and constitutive equations for the description of material features. Since the modeling approaches and the haptic simulations in this thesis are primarily considered to be applied in telepresence and medical technology, the selection of implemented materials' properties is corresponding to

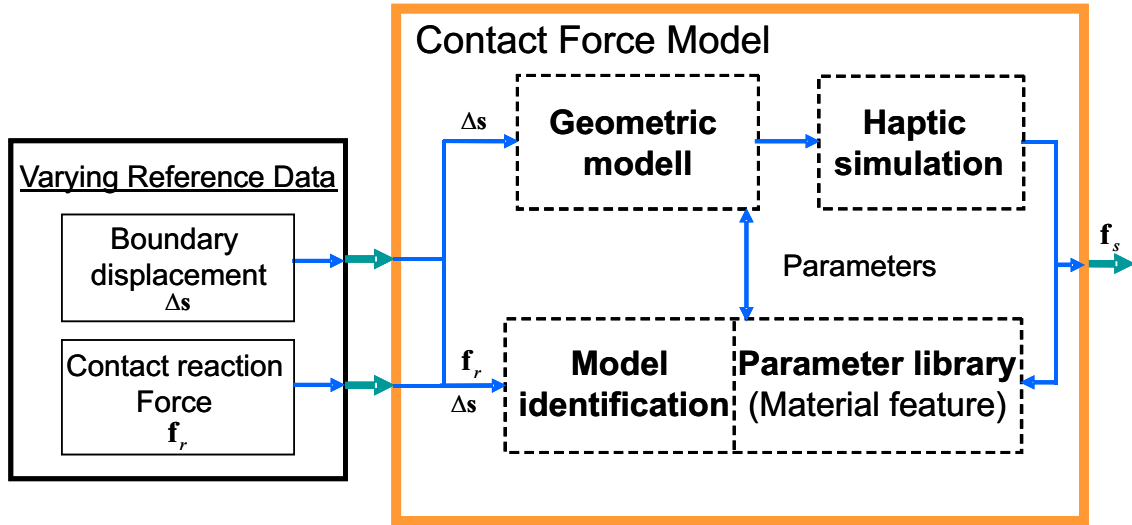


Figure 1.5.: Contact Force Model. The structure and the major components in the contact force models are illustrated, such as the geometric model, the material parameters, the model identification and the haptic simulation. They are detailed in the following chapters.

this aim. However, the modeling method can also be used for varying situations in other areas.

Furthermore, as a modular system, there is an important advantage: it is possible to extend the system with new functionalities just by importing and integrating some corresponding modules and defining the necessary settings, without modifications of the complete system.

This work is organized according to the procedure to solve a universal contact force modeling problem. Generally, the geometric models are the necessary requirements for the force models, and it is assumed that there is not any available information in the beginning. Hence, as the first step, the geometric modeling is detailed in Chapter 2, including the measurement methods of surfaces, the surface reconstruction using a set of discrete points, mesh generation with volume elements and the contact detection, as well as the determination of the boundary displacements. Chapter 3 focuses on the implementation of the material features, i.e. constitutive equations, into the geometric models. This means the models are not only volumes anymore, but virtual objects with physical properties. For different materials, there are different constitutive equations, and some of them are selected and are implemented. Chapter 4 presents the contact force evaluation for the finite element force models. In order to obtain real time simulations, the modeling methods for the analytical models are also elaborated, and the model based data fusion method is applied. By this means, the computational latency of the FEM force models can be compensated. The model identification is detailed in the fifth chapter, including the model learning, model parameter optimization, the identification for different models and the model verification. Furthermore, the adaptive modeling method is detailed as well. In Chapter 6, the modeling methods and the haptic simulation are tested and illustrated with experiments, using different objects with varying mechanical properties. The

experimental system and application examples are detailed. Finally, the work is concluded and summarized in the last chapter.

2. Geometric Modeling

There can be no contact force models and haptic simulations without the available geometric models of objects. Therefore, the geometric modeling of the objects is one of the necessary preparations, and should be the first step of modeling of deformable bodies.

Geometric modeling is the procedure to create numerical models which can describe the geometric properties of objects, including area, volume and contour. For the contact force model, mesh generation and geometric interactions, i.e. contact detection, also provide necessary geometric information for the following numerical implementation and contact force evaluation. Thus, they are also considered as parts of geometric modeling in this chapter.

In some cases, the geometric models may be imported, or there is information of the objects available. Some well known examples are the CAD models, which are widely used in diverse engineering areas, and the computer tomography (CT) scanners are standard equipments in medicine at present. However for more general situations, it is assumed in this chapter, that there is no geometric information of the objects available at the beginning. Hence the procedure starts with geometric measurements, and eventually, the models with geometric mesh should be created, Fig. 2.1.

2.1. Geometric Measurement

Without imported models, the geometric modeling has to start with the surface measurement, from which the original information of the geometry can be obtained. There are many possibilities for this task, such as the image processing using image sequence [66, 67] and varying scanning methods [124]. The expected data format of the result is a scatter plot of sample points, which are suitable for the following modeling procedure.

The image processing is based on the photos of objects from different viewpoints and aspects. With the optic flow, the corresponding pixels of an identical 3D position in different pictures can be found. In connection with the geometric relation of the viewpoints, the 3D position can be subsequently determined. Analogously, a scatter plot with a number of 3D points can be created.

The scanning procedures can be with and without contact, for example mechanical and optical scanning, respectively. They can measure the sample positions on the surface directly in the sensor coordinate system, and then using some methods, for instance the robot forward kinematics, the positions in inertial coordinate system can be determined. The computer tomography (CT) measurements are also scanning, however they measure several sections of the objects, this means not only the exterior geometry but also the interior structure are measured. For this case, sample points

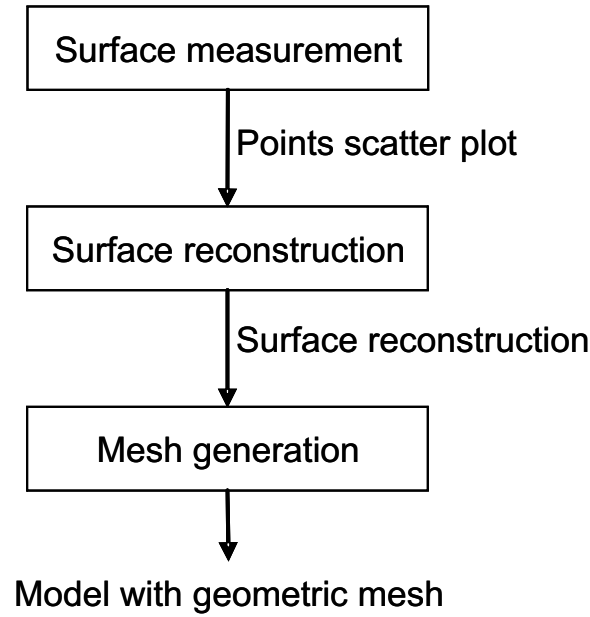


Figure 2.1.: Procedure of the Geometric Modeling. It is assumed that there is not any geometric information at the beginning. If the geometry is known by other means, the corresponding process may be shortened.

on the interior and exterior surface can be drawn simply from the sections without image processing.

Both methods have their advantages for different applications, laser scanning is more accurate than the stereo images, whereas the image processing is efficient for large and geometrically complex objects. The accuracy of the surface has strong influence on the contact force simulation in connection with deformation on the boundary. Since the objects, which are considered in this thesis, are generally not longer than 50[cm], the geometry measurements here are obtained with the laser scanning. And the CT scanners are widely used in medicine to examine human internal tissues.

Furthermore, it should be noted that, for latter modeling of closed surfaces, the sample points on all sides of an object are necessary. And yet, the measurements of the bottom side of the objects are often problematic, and it may be nearly impossible in many situations, such as the experimental setting in Chapter 6. To solve this problem, the bottom surface is considered as a plane, and can be created by projecting the upper surface on the base plane. It should be taken into account that some redundant nodes and triangles have to be eliminated, and some additional triangles must be inserted. By this means, the sample points on all sides can be obtained.

2.2. Surface Reconstruction

The surface reconstruction is the process to establish the numerical model, which is a closed surface, of an object surface, based on sample points. The surface is presented in the form of triangles. Hence, this reconstruction is also known as triangulation. The Delaunay method is popular for this purpose [16]. It can maximize the minimum

Table 2.1.: *Stl* file format (ASCII).

```

facet normal  $n_x$   $n_y$   $n_z$ 
  outer loop
    vertex  $v_{1x}$   $v_{1y}$   $v_{1z}$ 
    vertex  $v_{2x}$   $v_{2y}$   $v_{2z}$ 
    vertex  $v_{3x}$   $v_{3y}$   $v_{3z}$ 
  endloop
endfacet

```

Table 2.2.: *Ply* file format (ASCII).

```

ply
... (definition:)
... (- format, elements, properties and so on)
... (- comments ...)
end_header
 $x_1$   $y_1$   $z_1$ 
... (position of the vertices, the vertex  $v_i$  has the index  $i$ )
 $x_n$   $y_n$   $z_n$ 
(the first polygon)
... (definition of polygons using the index of the vertices)
(the last polygon)

```

angle in all triangles, however the default results of the Delaunay algorithm are convex surfaces, and concave surfaces can be obtained only by setting up the algorithms properly, i.e. the constrained Delaunay triangulation [63], for example with defined limit of the minimum angle [106]. For the Delaunay triangulation algorithms, there are open source programs available, for instance the Visualization Toolkit (VTK) [65].

The results of triangulations are closed surfaces, which are suitable for the mesh generation. They can be described in the arrays of triangulations and vertices. However, in order to implement the modeling methods as different modules, flexible data interfaces are always valuable. In computer graphics, some geometry file formats are defined for this purpose, and they are applied in many open source and commercial software and diverse applications [19], for instance the *stl* (stereo-lithography) files and the *ply* (polygon file format) files [20]. In the ASCII *stl* format, a file starts with the string “solid (surface name)” and is terminated with the string “endsolid (surface name)”. Between them, there is the description of the triangles. Each triangle is defined as a block, Tab. 2.1, in which the normal vector and the position of the three triangle vertices are defined. The format of the *ply* files has two parts, the header block and the data block. The header defines the format, element and other properties of the *ply* geometry, and the data block describes the position of the vertices and the polygons. This format is illustrated in Tab. 2.2. The *stl* files are easy for checking, because of the clear definition of the triangles, while the *ply* format is compact. For the same geometry, the *ply* file is normally much shorter than the *stl* file.

Besides the ASCII formats, the two geometry file formats have the corresponding binary formats as well. In this implementation, both kinds of geometry files are applied with the ASCII format, because they are intuitive for users to read. For the visualization of the reconstructed surfaces, the software Blender [18] and Matlab [77] are used.

Furthermore, there exist other file formats and programming languages to define geometry, such as the VRML (virtual reality modeling language) [76], which is much more powerful than the *stl* and *ply*. It can provide animation, lighting and many other functions to establish a virtual world. However it is too complex for this simple task of describing triangulations. Hence, the VRML is suitable for the visualization of varying simulations [97, 56].

The triangulation algorithms are detailed as follows. The Voronoi diagram is the basis of both the Delaunay algorithm and the power crust, and is introduced at first.

2.2.1. Voronoi Diagram

The Voronoi diagram [10] is the geometric structure of a domain, based on a set of sample positions. It is the subdivision of the domain according to specific criteria. The Voronoi diagram has been used in many areas, for instance robotics, physic and geography. In different applications, the criteria may be different, such as distance, cost and time. In the surface reconstruction, it is connected with the triangulation methods, and the subdivision is according to distances between the sample points.

Assuming n available sample points, also known as vertices $\{p_i : i = 1, \dots, n\}$, Fig. 2.2(a), which are not collinear, the Voronoi diagram of them is the subdivision with n cells $\{c_i^v : i = 1, \dots, n\}$. This means each cell contains and only contains one vertex inside of it, as the central vertex. The vertex in connection with the corresponding cell are noted with the identical index i . For an arbitrary position \tilde{p} in the cell c_i^v , the vertex p_i is always the nearest vertex, that is

$$d_{i\tilde{p}} < d_{j\tilde{p}} \quad \text{with} \quad i \neq j, \quad (2.1)$$

where d is the distance between two positions. And the cell c_i^v is defined by the bisectors of line segments $p_i p_j$, with $i \neq j$,

$$c_i^v = \cap_{i \neq j} D_{ij}^h, \quad (2.2)$$

where D_{ij}^h is the half domain which contains the vertex p_i , defined by the bisector of $p_i p_j$. This means the cells are convex polygons and the boundary of the cells are the segments of the bisectors, Fig. 2.2(b). For n collinear points, the Voronoi diagram is defined directly by $n - 1$ parallel lines.

It should be noted that not all vertices and bisectors are necessary to determine a cell c_i^v in general, if there are a number of sample points. I.e. it is not necessary to consider all vertices for one cell at the same time. For the computing of the Voronoi diagram, the Fortune's algorithm [40] is optimal in terms of computational expense. The basic idea of it is to scan the expected domain, and create the Voronoi diagram step by step. In the two dimensional case, i.e. a plane, the scanning is using a sweep

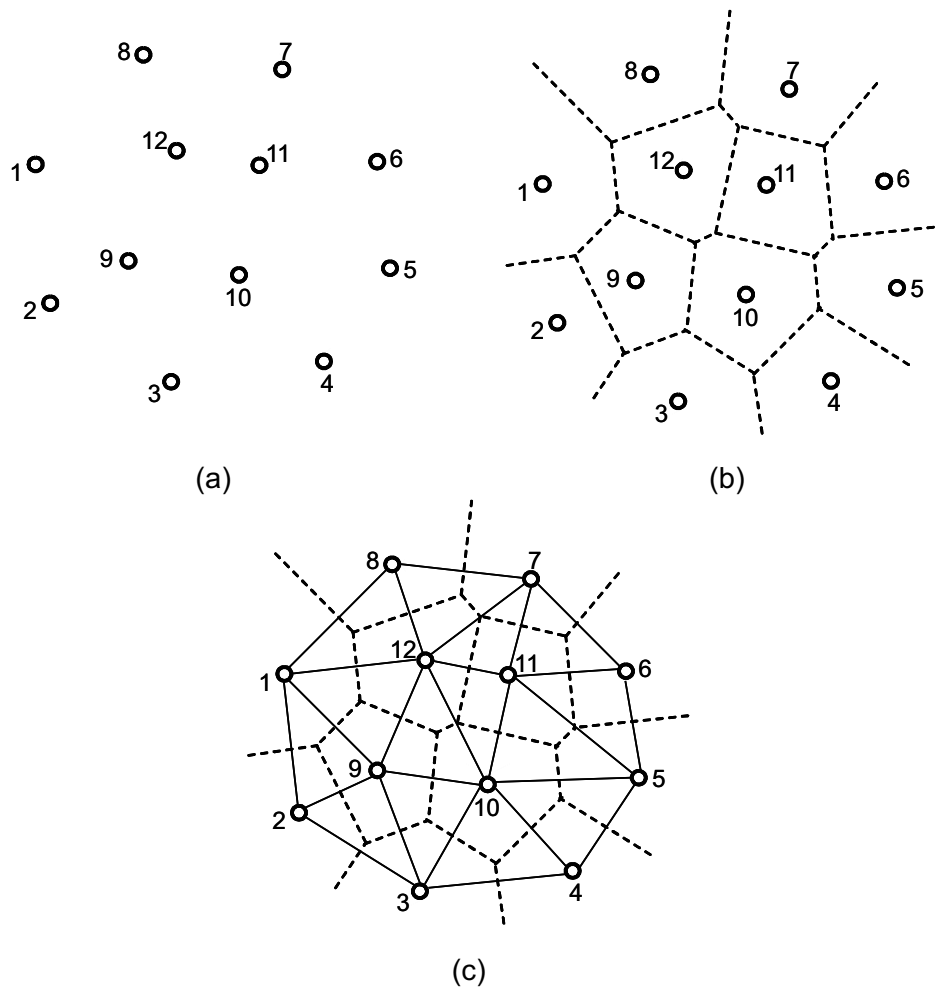


Figure 2.2.: Voronoi diagram and Delaunay triangulation. This is a two dimensional example using 12 points, including the original discrete sample points (a), the Voronoi diagram of these points (b) and the Delaunay triangulation of them (c)

line, from one side to the opposite side over the plane. Applying the consideration into three dimensional situations, the similar sweep algorithms can also be applied and the Voronoi diagram of 3D vertices can be created for the surface reconstruction.

2.2.2. Delaunay Triangulation

The original Delaunay triangulation is based on the subdivision, derived by the Voronoi diagram [16, 41]. If the cells c_i^v and c_j^v have a common edge, the corresponding vertices p_i and p_j should be connected with a straight-line. Then, the embedding of the vertices $\{p_i\}$ can be created, Fig. 2.2(c), and this network is called the Delaunay graph of the vertices.

In comparison with the Voronoi diagram, it can be concluded that they both are the same to each other in graphic theory. The bounded areas in the Delaunay graph are always triangles, i.e. the Delaunay triangulation. It is the optimal triangulation

of the given vertices by means of maximization of the minimum angle of all angles in the triangles during the triangulation. The circumcircle of each triangle is an empty circle, without any other vertices.

Since the original Delaunay triangulation is a triangulation of the convex hull of the points, the constrained Delaunay triangulation [23] is applied to make the triangulation according to the boundary defined by the sample points accurately. The similar thought is applied for mesh generation in the next section too, as the constrained Delaunay tetrahedralization.

2.2.3. Power Crust

The power crust is also an algorithm for surface reconstruction, based on the power diagram [9], which is a kind of generalized Voronoi diagram with weight, and the medial axis transform (MAT) [2]. It is also suitable for concave surfaces. The open source program *power crust* is available for this algorithm.

Both the Delaunay triangulation and the power crust have been implemented and applied in this thesis. In practice, the power crust is easier to use for complex surface than the original Delaunay triangulation. However, it should be noticed that the reconstructions of contours with sharp edges are often problematic, and the triangulation may provide better results, if the sample points distributed nearly uniformly over the object surface.

2.3. Mesh Generation

The domains, which are defined by the closed surfaces, can be regarded as volume objects. In order to determine the stress-strain relations of them numerically in the latter chapters, a geometric model has to be discretely divided into a number of sub-domains, which are also called the volume elements, for the implementation of the finite element method (FEM).

As the last step for the geometric modeling, the mesh generation is performed according to the Delaunay algorithm [107, 82], i.e. the volume elements are well-shaped tetrahedra, and the quality mesh can be generated. This means the shape of each tetrahedron is bounded by specific limits, such as the smallest angle and the aspect ratio. It is suggested to apply the radius-edge ratio as the measure of the element shape [83]. The radius-edge ratio of a element is defined as the ratio of the radius of the circumscribed sphere r_{cs} to the minimum edge l_{min} of the tetrahedron,

$$r_{tet} = \frac{r_{cs}}{l_{min}}. \quad (2.3)$$

Well shaped tetrahedra have small values. By this means, thin and flat tetrahedra can be avoid, Fig 2.3(a) and (b). These geometric conditions of tetrahedra are necessary for the accuracy and the convergence of the contact force evaluation using the finite element method.

Similar to the Delaunay triangulation, the Delaunay tetrahedralizations are always convex. However, general volumes, which are defined by the close surfaces, are not

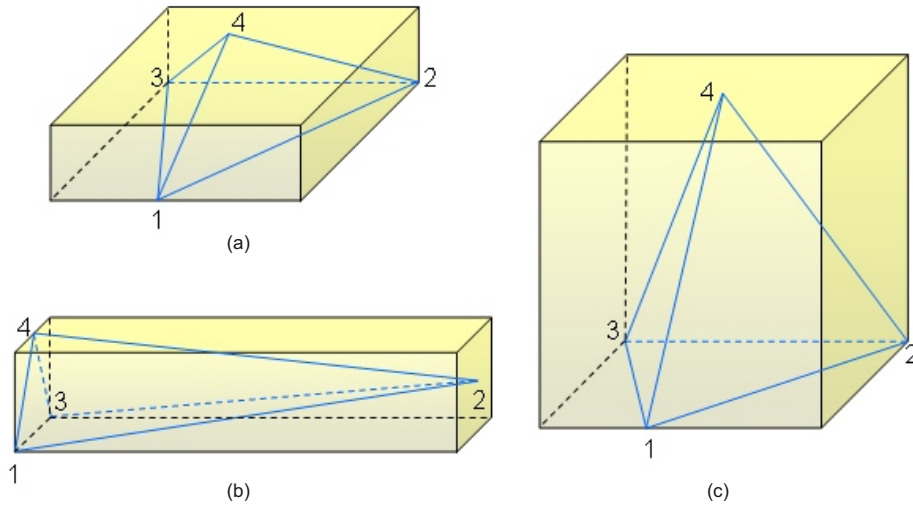


Figure 2.3.: Tetrahedra with different shapes. The flat (a) and thin (b) tetrahedra have relative large radius-edge ratios and should be avoided. Whereas the tetrahedra with good shape (c) have small values and are more suitable for the evaluation using the finite element method.

always in this case. During the mesh generation, the domain boundary must be respected. Different methods are investigated to solve this problem, such as the incorporation of special tetrahedra [54]. The conforming Delaunay algorithm can be applied for concave domains [85], however it may generate a number of small scale elements and the associated additional vertices. In this implementation, the constrained Delaunay tetrahedralization (CDT) [108, 109] is applied. Using the CDT, not all but the most important advantages of the original Delaunay algorithm can be maintained, and the CDT inserts merely few vertices in comparison with the other methods.

In this section, the mesh generation starts with the closed surface, which is defined in the form of piecewise linear complex (PLC). It is a set of fundamental geometric elements, such as vertices, segments and facets. The geometry files *stl* and *ply* are used as the container of the PLC. The constrained Delaunay tetrahedralization is implemented using the linear tetrahedra for the quality mesh generation to fill the domain of the closed surface. The open source program *TetGen* [111, 110] is used for this task.

2.4. Contact Detection

In the previous sections, the geometric modeling of objects is introduced. However, in multi-body systems, not only the geometry but also the position, orientation and interaction of the bodies are important. In virtual reality, the multi-body simulation can present the relation and motion of the bodies [42]. Different objects can be modeled as rigid or deformable bodies, according to the application conditions. As a kind of universal interactions, contacts should be taken into account in this systems,

especially for haptic simulations.

The contacts depend on varying physical effects, for instance deformation, velocity, sliding and friction [11], as well as the contact situations. In this section, the following conditions are presumed for the tool-object contacts.

- The objects are much softer than the operation tools.
- In comparison with the contact surface of the objects, the sectional area of the tools is very small.
- The velocity of the tools with respect to the objects is low during the contacts.
- The objects are static in the inertial system, and there are always some fixed constraints at the boundary. This means the objects can not translate and rotate freely as complete bodies.

Hence the contacts are regarded as point-surface interactions. The tools can be considered as rigid bodies, and just the objects are modeled as deformable. Due to the boundary constraints, the objects always have deformations during the contacts. Moreover, since the interactions are always in low speed, the contacts are quasi-static. Furthermore, sliding and friction are not considered in this contact force model.

A contact detection algorithm is implemented for this point-surface contact problem [124]. At first, the tool and the object are defined in the same inertial coordinate system. Then, for the given position and velocity of the tool, the three nearest nodes (\mathbf{p}_{c1} , \mathbf{p}_{c2} and \mathbf{p}_{c3}) from the tool (t) are found on the object surface, based on the distance between the tool and nodes. Defining the tool position as $[x_t \ y_t \ z_t]$, the distances can be calculated as follows,

$$d_{ti} = \sqrt{(x_t - x_{ni})^2 + (y_t - y_{ni})^2 + (z_t - z_{ni})^2}. \quad (2.4)$$

Subsequently, a plane (o) can be determined by the three nodes, written in the general form,

$$ax + by + cz = d. \quad (2.5)$$

The coefficients in the plane equation can be determined by solving the linear system

$$\begin{aligned} & \begin{bmatrix} a & b & c \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}_{c1} \\ \mathbf{p}_{c2} \\ \mathbf{p}_{c3} \end{bmatrix} = \begin{bmatrix} d \\ d \\ d \end{bmatrix} \\ & \rightarrow \begin{cases} ax_1 + by_1 + cz_1 = d \\ ax_2 + by_2 + cz_2 = d \\ ax_3 + by_3 + cz_3 = d. \end{cases} \end{aligned} \quad (2.6)$$

The vector $\mathbf{v}_o = [a \ b \ c]$ is the normal vector of the plane. Since there are always two normal vectors, \mathbf{n}_{o1} and \mathbf{n}_{o2} , for a plane in opposite directions, the velocity

vector (\mathbf{v}_t) of the tool has to be considered. Defining the angles between the normal vectors and the velocity vector as θ_1 and θ_2 :

$$\theta_i = \arccos \left(\frac{\mathbf{n}_{oi} \cdot \mathbf{v}_t}{|\mathbf{n}_{oi}| |\mathbf{v}_t|} \right). \quad (2.7)$$

Then the normal vector with greater angle should be selected. The border case $\theta_1 = \theta_2 = 90^\circ$ can be neglected, because this case should not happen in this contact situation.

Then the signed point-surface distance between the tool (t) and the plane (o) can be calculated,

$$d_{to} = \frac{ax_T + by_T + cz_T + d}{\sqrt{a^2 + b^2 + c^2}}. \quad (2.8)$$

According to the definition of the normal vector, a positive d_{to} indicates that the tool and the object are divided, and a negative distance means that contacts appear. Then the contact position is defined at the nearest node, and without sliding, the contact node is constant during a single contact. Using a location index, the searching procedure may be accelerated significantly, that means searching is only done in a small region in the near from the tool. This contact detection is implemented in Algorithm 2.1.

Algorithm 2.1: Contact detection

```

Define the current position of the tool,  $p_t$ 
if the node index is available then
    Searching in the index
    Determinate the three nearest nodes
else
    Traversal all nodes one by one,
    Calculate the distance  $d_{ti}$ 
    Select the three nearest nodes
end if
Determine the plain  $P$ 
Determine the normal vector
Calculate the distance  $D$ 
Judging the contact
if there is contact then
    Determine the boundary displacement
else
    There is neither any deformation, nor contact force
end if

```

For other operation situations, there are also approaches for the tool-object haptic problem, taking sliding into account [73]. Furthermore, for general contact problems, the contact situations may be complex and different. In these cases, different algorithms of contact detection have been investigated, such as the bounding volume hierarchies (BVH), which has been proved to be one of the most efficient methods for

the contact detection [116, 122]. Some stochastic methods are also developed, which are especially suitable for real time and interaction applications. The spatial division has also be applied in this area [115]. There are professional multi-body simulation software packages, which are suitable for the collision detection, for instance the program *Bullet*.

3. Modeling of Solid Material

Solid materials can be modeled numerically and analytically. The analytical models are more accurate and simple in implementation and computation. However, there is not any analytical way for the modeling of universal objects in continuum mechanics, thus the numerical methods are used here. The material characteristics are implemented using the finite element method (FEM). First of all, some typical material features should be introduced.

3.1. Constitutive Equation

In continuum mechanics, the material behavior is generally described by the stress-strain relations in the form constitutive equations, which are mathematical models of real materials. There are many diverse materials, however some typical mechanical characteristics are contained by varying materials. In order to present them accurately in mathematic formulations, different constitutive equations are developed. Each of them can describe some principal material behavior, respectively. To apply these theoretical models in practice, the aim is to determine the most suitable constitutive equations for some given real materials. In this thesis, the modeling tasks are concentrated on soft solid materials. Elasticity is implemented as the most basic material feature, and viscoelasticity is also considered, since nearly all real materials behave in this way more or less, especially the animal tissues, which have been highly interesting objects for haptic simulations in medical and biological areas in recent years. Furthermore, inhomogeneous materials are common in real world, therefore it is also valuable to develop the modeling methods for them.

The constitutive equations describe the relations of stress-strain processes. In the following derivations, stress and strain are formulated in the tensor form [72], i.e. the stress tensor,

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix}, \quad (3.1)$$

and the strain tensor

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \epsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \epsilon_z \end{bmatrix}. \quad (3.2)$$

Based on the principle of conservation of angular momentum, the stress tensor is

symmetric with the following property,

$$\tau_{xy} = \tau_{yx} \quad \tau_{yz} = \tau_{zy} \quad \tau_{zx} = \tau_{xz}. \quad (3.3)$$

In the strain tensor, the similar relations are also available for the shear entries,

$$\gamma_{xy} = \gamma_{yx} \quad \gamma_{yz} = \gamma_{zy} \quad \gamma_{zx} = \gamma_{xz}. \quad (3.4)$$

Then both tensors can be defined by 6 different entries, respectively. Then the constitutive equation can be written in the general form

$$\boldsymbol{\sigma} = \mathbf{F}_c(\boldsymbol{\epsilon}). \quad (3.5)$$

The general function \mathbf{F}_c is defined for varying material models, respectively. In this chapter, the thermo effects of materials during deformations are omitted.

3.1.1. Linear Elastic Material

Linear elasticity is based on the geometric linearization, and it is fundamental to the classic theory of the solid mechanics [118]. The well known simplified Hooke's law

$$\sigma = E\epsilon \quad (3.6)$$

is the simplest situation of this material under one dimensional load. The complete constitutive equation of linear elastic materials, the Hooke's law, in tensor form is as follows,

$$\boldsymbol{\sigma} = \frac{E}{1+\nu} \left[\boldsymbol{\epsilon} + \frac{\nu}{1-2\nu} \cdot \text{tr } \boldsymbol{\epsilon} \cdot \mathbf{I} \right], \quad (3.7)$$

where E is the Young's modulus, and ν is the Poisson's ratio. For real materials, ν is always in the range ($0 \sim 0.5$), and the boundary value 0.5 corresponds to incompressible materials, which are idealized cases. There are also other coefficient settings for the linear elastic materials, for instance the bulk modulus K and the shear modulus G . These parameters can be converted into each other,

$$K = \frac{E}{3(1-2\nu)} \quad (3.8)$$

and

$$G = \frac{E}{2(1+\nu)}. \quad (3.9)$$

Thus, at the same time, just two of them are necessary to describe the material behavior completely. Hence, in the following derivation, the Young's modulus and the Poisson's ratio are used, as well as for the corresponding modeling and model identification. It should be noted that this material model is merely available for small deformations, since in this area, both the material feature and the strain state

can be approximated as linear relations [49].

3.1.2. Linear Viscous Material

The perfect fluid is modeled as the linear viscous material, which neglects a general effect of fluid, i.e. the internal friction. It is also known as the Newtonian fluid, the idealization of fluid material [119]. The constitutive equation of it is

$$\boldsymbol{\sigma} = -p(\rho) \cdot \mathbf{I} + \eta \cdot \nabla \mathbf{v} + \lambda \cdot \text{tr} \nabla \mathbf{v} \cdot \mathbf{I}, \quad (3.10)$$

in which η is the viscosity and ρ is the density of the fluid, \mathbf{v} is the velocity of the material points.

Since the central point is the modeling of solid materials, the introduction of the linear viscous fluid is for the following derivation of the viscoelastic constitutive equation in the next section. It is sufficient merely to take the viscous effects into consideration, while the other effects are described by the other terms in the viscoelastic constitutive equation. For this reason, further simplifications can be assumed. If the fluid is incompressible, the third term can be omitted, i.e. $\text{tr} \nabla \mathbf{v} = 0$. And for the fluid with constant flow velocity, the diagonal entries vanish. Sometimes the effects of material density to the internal-strain state can also be neglected, then only the shear stress is remained. Eventually, the linear viscous constitutive equation is simplified as follows, in Cartesian coordinate system,

$$\boldsymbol{\sigma} = \eta \dot{\boldsymbol{\epsilon}} = \eta \begin{bmatrix} 0 & \frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} & \frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} & 0 & \frac{\partial v_z}{\partial z} + \frac{\partial v_z}{\partial y} \\ \frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} & \frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z} & 0 \end{bmatrix}, \quad (3.11)$$

where $\dot{\boldsymbol{\epsilon}}$ is the rate of strain tensor. In the one dimension case, for instance in direction x , it can be formulated as

$$\tau_{xy} = \eta \frac{\partial v_x}{\partial y} = \eta \dot{\gamma}_{xy}. \quad (3.12)$$

The analogous relations of this reduced constitutive equation are applied in the derivation in next section. This simplified formulation is suitable, because the other material feature can be presented by the elastic components in the viscoelastic models.

3.1.3. Linear Viscoelastic Material

As mentioned above, the viscoelastic materials are nonlinear materials and have the features of pure elastic and viscous materials at the same time. For the modeling of these materials in this implementation, the linear viscoelastic behaviors [89] is assumed, analogous to linear elastic and the Newton fluid, that is the Boltzmann superposition is valid. Furthermore, it is also presumed that the material in the considered region is isotropic and homogeneous. For anisotropic materials, the approach is similar, but the material parameters are dependent on the directions.

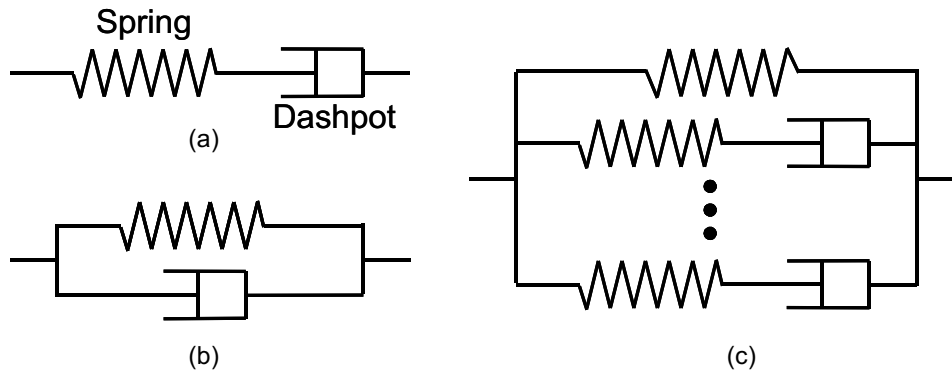


Figure 3.1.: Spring-dashpot models for viscoelastic materials. The basic elements are springs and dashpots for the elastic and viscous components, respectively. Different models can be established by diverse combinations of the elements, such as the Maxwell model (a), the Kelvin model (b) and the generalized model (c) with $2n - 1$ elements.

For inhomogeneous materials, the modeling method is introduced in the following sections.

The constitutive equation of linear viscoelastic materials can be defined in the relaxation form or the creep form. In latter derivations in this section, the relaxation formulation is used, which is also known as the stress process. It is the stress response corresponding to a series of given strain states. The external effects of this process is the traction or reaction force response to the given boundary displacement history.

Based on the Boltzmann superposition, the linear viscoelastic constitutive relation can be formulated as follows,

$$\sigma = \int_0^t \mathbf{I}(t - \tau) \dot{\epsilon}(\tau) d\tau, \quad (3.13)$$

where $\dot{\epsilon}$ is the time derivative of the strain tensor, and \mathbf{I} is the relaxation function, which depends on the strain, the strain rate and the time scale. Hence (3.13) can explain the dependence of the stress-strain relation on the loading history. However, the details of the effects are hidden in the relaxation functions. In order to determine them, the spring-dashpot models are used [26], and the elastic and the viscous contributions are described by the springs and dashpots, respectively, Fig. 3.1. For the one dimensional case in the figures, the constitutive equations of both linear materials, (3.7) and (3.10), can be simplified into the following stress-strain relations, respectively, as mentioned above,

$$\begin{aligned} \sigma_e &= E \cdot \epsilon_e \\ \sigma_v &= \eta \cdot \dot{\epsilon}_v. \end{aligned} \quad (3.14)$$

There are three different rheological models, using springs and dashpots, i.e. the Maxwell model, the Kelvin model and the generalized model. The last one consists of varying Maxwell elements and a spring, which are parallel to each other. Fig. 3.1.

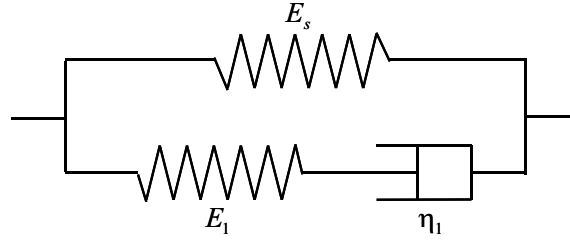


Figure 3.2.: The simplest generalized model with three parameters, a spring E_s and a Maxwell element E_1 and η_1 .

The boundary condition of the Maxwell model is

$$\lim_{t \rightarrow \infty} \sigma = 0, \quad (3.15)$$

and for the Kelvin model, it is

$$\lim_{t \rightarrow \infty} \epsilon = 0. \quad (3.16)$$

The generalized model can provide varying behavior by using n diverse Maxwell elements, and the following relations are always available,

$$\begin{aligned} \sigma_s + \sum_{i=1}^n \sigma_i &= \sigma \\ \epsilon_s &= \epsilon_{ei} + \epsilon_{vi} = \epsilon, \end{aligned} \quad (3.17)$$

where s is the index of the static term described by the single spring.

As an example, considering the simplest generalized model with only one spring (E_s) and a single Maxwell element (E_1 and η_1), Fig.3.2. Since, $\sigma_s + \sigma_1 = \sigma$ and $\epsilon_{1e} + \epsilon_{1v} = \epsilon_s = \epsilon$, the linear differential equation of this spring-dashpot system can be formulated as follows,

$$\dot{\sigma} + \frac{E_1}{\eta_1} \sigma = \frac{E_s E_1}{\eta_1} \epsilon + (E_s + E_1) \dot{\epsilon}. \quad (3.18)$$

It can be solved with the relaxations boundary condition, $\sigma(0) = 0$, and then the solution, including the homogeneous and particular parts, is

$$\sigma(t) = \int_0^t \left[E_s + E_1 e^{-\frac{E_1}{\eta_1}(t-\tau)} \right] \dot{\epsilon}(\tau) d\tau. \quad (3.19)$$

This equation is the constitutive relation of this model, with the relaxation equation

$$\Gamma = E_s + E_1 e^{-\frac{E_1}{\eta_1}(t-\tau)}. \quad (3.20)$$

Analogously, applying the relation (3.17) for a generalized model with N Maxwell elements, the linear viscoelastic constitutive equation in the general form can be

derived,

$$\sigma(t) = \int_0^t \left[E_s + \sum_{i=1}^N E_i e^{-\frac{E_i}{\eta_i}(t-\tau)} \right] \dot{\epsilon}(\tau) d\tau, \quad (3.21)$$

and the relaxation function is defined accordingly in connection with a Prony series, as follows,

$$\Gamma(t) = E_s + \sum_{i=1}^N E_i e^{-\frac{t}{\tau_i}}, \quad (3.22)$$

where E_s is the elastic modulus of the static component, E_i and τ_i are the relaxation amplitude and the relaxation time of the i -th Maxwell element, respectively. The viscosities are relevant for in the relaxation times,

$$\tau_i = \frac{\eta_i}{E_i}. \quad (3.23)$$

Since (3.21) just describes the stress-strain relation under 1-dimensional load, however for a general 3-dimensional deformable body, the influence of the strains in other directions on the stress should be taken into account as well. Thus, analogous to (3.7), the Poisson's ratio ν is involved. Then in connection with E_s , the static linear elasticity can be implemented in the viscoelastic model. The material parameters of a linear viscoelastic model can be defined as

$$\mathbf{x} = \begin{bmatrix} E_s & \nu & E_1 & \tau_1 & \cdots & E_N & \tau_N \end{bmatrix}^T. \quad (3.24)$$

3.2. Finite Element Method

In order to implement the above shown material feature in the contact force models and to use them to solve practical problems, the finite element method (FEM) [127, 126] is applied. The FEM can solve problems in many different areas, for instance thermodynamics and electro magnetic fields. In this thesis, it is implemented to solve continuum mechanic problems. In this chapter, it is assumed that the geometric models with mesh are available, and the goal is just to apply the material feature into them.

3.2.1. Basic Concept

The finite element method (FEM) is a series of numerical methods to solve a differential equation system,

$$\mathfrak{D}(\mathbf{u}) = \mathbf{0}, \quad (3.25)$$

with certain boundary conditions,

$$\mathfrak{B}(\mathbf{u}) = \mathbf{0}. \quad (3.26)$$

Then it is reasonable to transform this problem into an equivalent integration form as follows,

$$\int_V \mathbf{v}_d^T \mathfrak{D}(\mathbf{u}) dV + \int_S \mathbf{v}_b^T \mathfrak{B}(\mathbf{u}) dS = 0, \quad (3.27)$$

in which V and S are the effective volume of the differential equations and the effective area of the boundary conditions. The vectors \mathbf{v}_d and \mathbf{v}_b are two vectors of arbitrary functions. The FEM can try to solve \mathbf{u} in the following form as approximations,

$$\mathbf{u} \approx \sum_{a=1}^n \mathbf{N}_a \tilde{\mathbf{u}}_a = \mathbf{N} \tilde{\mathbf{u}}, \quad (3.28)$$

where \mathbf{N}_a are shape functions. Applying the Galerkin method, the approximation of the arbitrary functions in \mathbf{v} can be defined in a finite discrete form,

$$\mathbf{v} = \sum_{b=1}^n \mathbf{N}_b \delta \mathbf{u}_b. \quad (3.29)$$

In (3.28-29), a and b are the indices of nodes. Then, if the differential equations are linear, the integration (3.27) can be rewritten into the form,

$$\mathbf{A}_1 \mathbf{u} + \mathbf{A}_2 \dot{\mathbf{u}} + \mathbf{A}_3 \ddot{\mathbf{u}} + \cdots + \mathbf{f} = 0. \quad (3.30)$$

In this thesis, the contact problem can be modeled as the following case,

$$\mathbf{K} \mathbf{u} + \mathbf{f} = 0. \quad (3.31)$$

This formulation is suitable to solve solid mechanics problems under the contact conditions defined in Chapter 2, with the stiffness matrix \mathbf{K} , the force vector \mathbf{f} and the displacement vector \mathbf{u} .

The matrices and vectors of one FE model are the sum of all the corresponding element and nodal terms, and noted as $\overset{m}{\mathcal{A}}(\cdot)$. That is, the model stiffness is the assembly of all element stiffness matrices according to nodes,

$$\mathbf{K} = \overset{m}{\mathcal{A}}_{e=1}(\mathbf{K}^e). \quad (3.32)$$

Analogously, for the nodal displacement

$$\mathbf{u} = \overset{m}{\mathcal{A}}_{e=1}(\mathbf{u}^e), \quad (3.33)$$

and the force vectors

$$\mathbf{f} = \overset{m}{\mathcal{A}}_{e=1}(\mathbf{f}^e), \quad (3.34)$$

the similar summary relations are available. The nodal displacement and nodal force

vectors of a single element can be defined node-wise, as follows,

$$\mathbf{u}^e = \begin{bmatrix} u_{x1} & u_{y1} & u_{z1} & \cdots & u_{xl} & u_{yl} & u_{zl} \end{bmatrix}^T, \quad (3.35)$$

and

$$\mathbf{f}^e = \begin{bmatrix} f_{x1} & f_{y1} & f_{z1} & \cdots & f_{xl} & f_{yl} & f_{zl} \end{bmatrix}^T, \quad (3.36)$$

where l is the number of the nodes of each element.

On the opposite side, the nodal terms can be determined from the model vectors and matrices, as unassemble. They are marked directly as the inverse of the assembly, $\mathcal{A}_{i=1}^n(\cdot)$, such as the element nodal displacement,

$$\mathbf{u}^e = \mathcal{A}_{e=1}^m(\mathbf{u}). \quad (3.37)$$

Furthermore, it should be noticed that the model stiffness matrices are sparse matrices, and there are mathematic and numerical methods to calculate them efficiently [48, 117]. By this means, the calculations, especially the matrix multiplications, can be accelerated significantly.

In the FEM, the strain inside of a tetrahedron can be determined based on the nodal displacement, in the vector form, and for a single node, it yields,

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \quad (3.38)$$

where u , v and w are displacements, and the stress is also defined in the vector form accordingly,

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z & \tau_{xy} & \tau_{yz} & \tau_{xz} \end{bmatrix}^T. \quad (3.39)$$

3.2.2. Boundary Conditions

Some interactions of the test objects and the environment are modeled as boundary conditions. They can be divided into two kinds, the boundary displacement and the boundary traction. The displacement boundary conditions at some positions should be transformed to the corresponding nodes, and the nodal displacements of them are specified using the given boundary displacement vector \mathbf{u}^0 ,

$$\mathbf{u} = \mathbf{u}^0. \quad (3.40)$$

The boundary conditions with known boundary tractions \mathbf{f}^0 are formulated as

$$\mathbf{f} = \mathbf{f}^0 = \mathbf{G}^T \boldsymbol{\sigma} \quad (3.41)$$

for accordant nodes, and the matrix \mathbf{G} is the transformation matrix from stress to nodal force, which contains the direction cosines as follows,

$$\mathbf{G} = \begin{bmatrix} n_x & 0 & 0 \\ 0 & n_y & 0 \\ 0 & 0 & n_z \\ n_y & n_x & 0 \\ 0 & n_x & n_y \\ n_z & 0 & n_x \end{bmatrix}, \quad (3.42)$$

where n_x , n_y and n_z are the direction cosines.

3.3. Linear Elastic Model

The linear finite element model means not only the constitutive equation but also the geometry and the elements are linear. That is, as basic FE models, they are complete linear. In this implementation, the linear tetrahedron, which contains 4 nodes and 6 edges, is used as the volume element, Fig. 2.3.

As mentioned above, the elastic constitutive equation (3.7) can be implemented in the vector form in these models as

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f}, \quad (3.43)$$

where \mathbf{u} and \mathbf{f} are the nodal displacement and nodal force vectors, respectively. The matrix \mathbf{K} is the model stiffness matrix, which is the sum of all element stiffness matrices $\{\mathbf{K}^e\}$. A single element stiffness matrix is defined as follows,

$$\mathbf{K}^e = V^e \mathbf{B}^T \mathbf{C}^e \mathbf{B}, \quad (3.44)$$

in which V^e is the volume of the tetrahedron. If the coordinate of a node is noted as $\tilde{\mathbf{u}}_i = [x_i \ y_i \ z_i]$, it yields for the tetrahedron defined by the nodes $\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \tilde{\mathbf{u}}_3$ and $\tilde{\mathbf{u}}_4$,

$$V^e = \frac{1}{6} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}. \quad (3.45)$$

The matrix \mathbf{B} contains the shape functions. It can describe the relation between the strain and the corresponding nodal displacements,

$$\boldsymbol{\epsilon} \approx \mathbf{B} \tilde{\mathbf{u}} \xrightarrow{\text{linearization}} \boldsymbol{\epsilon} = \mathbf{B} \tilde{\mathbf{u}}. \quad (3.46)$$

Matrix \mathbf{B} is a constant matrix for linear tetrahedra here. The matrix \mathbf{C}^e is the elastic matrix of the elements and defines the material characteristics of them. It is identical for the same materials,

$$\mathbf{C}^e = \frac{E}{d} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2 \end{bmatrix}, \quad (3.47)$$

in which E is the Young's modulus, ν is the Poisson's ratio and the scalar coefficient d is defined as $d = (1+\nu)(1-2\nu)$.

For a single tetrahedron, the nodal displacement and force vectors are noted as,

$$\mathbf{u}^e = [x_1 \ y_1 \ z_1 \ x_2 \ y_2 \ \cdots \ y_4 \ z_4]^T, \quad (3.48)$$

and

$$\mathbf{f}^e = [f_{x1} \ f_{y1} \ f_{z1} \ f_{x2} \ f_{y2} \ \cdots \ f_{y4} \ f_{z4}]^T. \quad (3.49)$$

These are in the component wise formulation. They can also be written node wise, and the implementation is analogous. Then the coordinate of an arbitrary position $[x \ y \ z]$ in the tetrahedron can be defined using the volume coordinate $[\xi_1 \ \xi_2 \ \xi_3 \ \xi_4]$,

$$\begin{aligned} 1 &= \xi_1 + \xi_2 + \xi_3 + \xi_4 \\ x &= \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 + \xi_4 x_4 \\ y &= \xi_1 y_1 + \xi_2 y_2 + \xi_3 y_3 + \xi_4 y_4 \\ z &= \xi_1 z_1 + \xi_2 z_2 + \xi_3 z_3 + \xi_4 z_4. \end{aligned} \quad (3.50)$$

The shape functions \mathbf{N}_a can be determined subsequently. Then the matrix \mathbf{B} is as follows,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 \end{bmatrix}.$$

Assembling them of all tetrahedra into (3.43) using the relations (3.32-34), a complete linear elastic FE model can be established. And the evaluation of the contact reaction force is directly according to (3.43), using the numerical methods in the next chapter.

3.4. Linear Viscoelastic Model

The linear viscoelastic contact force models are also implemented using finite element method. They are nonlinear models because of the constitutive equation. However, the above shown viscoelastic constitutive equation (3.21) is dependent on the time scale, and cannot be applied directly in the FEM. Since the numerical methods are based on the discrete model, the relation (3.21) have to be transformed into the discrete formulation, in the first step.

The goal is the discrete relation with recursive form, which is suitable for the FE implementation. Then the current states are merely dependent on previous states, generally the states at the last time step, and only a few storage spaces are necessary. For a single Maxwell element, the following relation can be split from the complete constitutive equation (3.21),

$$\sigma^i = \int_0^t E_i e^{-\frac{E_i}{\eta_i}(t-\tau)} \dot{\epsilon} d\tau. \quad (3.51)$$

After the time discretization, it yields for the point of time $k+1$

$$\sigma_{k+1}^i = E_i \int_0^{k+1} e^{-\frac{E_i}{\eta_i}(k-\tau)} \dot{\epsilon} d\tau, \quad (3.52)$$

and the discrete time form can be derived as follows, for a given time interval Δt ,

$$\begin{aligned} \sigma_{k+1}^i &= e^{-\frac{E_i}{\eta_i} \Delta t} E_i \int_0^k e^{-\frac{E_i}{\eta_i}(k-\tau)} \dot{\epsilon} d\tau + E_i \int_k^{k+1} e^{-\frac{E_i}{\eta_i}(k+1-\tau)} \dot{\epsilon} d\tau \\ &= e^{-\frac{E_i}{\eta_i} \Delta t} \sigma_k^i + E_i \int_k^{k+1} e^{-\frac{E_i}{\eta_i}(k+1-\tau)} \dot{\epsilon} d\tau \\ &= e^{-\frac{E_i}{\eta_i} \Delta t} \sigma_k^i + E_i \int_k^{k+1} e^{-\frac{E_i}{\eta_i}(k+1-\tau)} d\tau \frac{\Delta \epsilon_{k+1}}{\Delta t} \\ &= e^{-\frac{E_i}{\eta_i} \Delta t} \sigma_k^i + \eta_i \left(1 - e^{-\frac{E_i}{\eta_i} \Delta t}\right) \frac{\Delta \epsilon_{k+1}}{\Delta t}. \end{aligned} \quad (3.53)$$

With (3.23), it can be rewritten as,

$$\sigma_{k+1}^i = e^{-\frac{\Delta t}{\tau_i}} \sigma_k^i + E_i c_i^t \Delta \epsilon_{k+1}, \quad (3.54)$$

where the coefficient c_i^t is defined as

$$c_i^t = \frac{\tau_i}{\Delta t} \left(1 - e^{-\frac{\Delta t}{\tau_i}}\right). \quad (3.55)$$

Substituting (3.54) into the stress relation of generalized models (3.17) yields the

discrete time constitutive equation,

$$\sigma_{k+1} = E_s \epsilon_{k+1} + \sum_{i=1}^N \left(e^{-\frac{\Delta t}{\tau_i}} \sigma_i^k + E_i c_i^t \Delta \epsilon_{k+1} \right), \quad (3.56)$$

which is suitable for the FE implementation.

The second step is the implementation of the discrete constitutive relation. To apply the one dimensional constitutive equation (3.56) in practical deformable models and to solve the stress-strain relations numerically, the finite element method in this section is also implemented using linear tetrahedra. Considering the equation of the linear elastic models,

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f}, \quad (3.57)$$

in order to formulate the finite element relation of linear viscoelastic materials analogous to that, (3.56) can be transformed as

$$\left(E_s + \sum_{i=1}^N E_i c_i^t \right) \epsilon_{k+1} = \sigma_{k+1} - \sum_{i=1}^N e^{-\frac{\Delta t}{\tau_i}} \sigma_k^i + \sum_{i=1}^N E_i c_i^t \epsilon_k. \quad (3.58)$$

For the static term, analogous to linear elastic models, the Poisson's ratio has been taken into (3.58), applying it for each tetrahedron in a given FE model, the relation between \mathbf{u} and \mathbf{f} is available in the following form,

$$\mathbf{K}_v \cdot \mathbf{u}_{k+1} = \mathbf{f}_{k+1}^e - \mathbf{f}_{k+1}^h = \tilde{\mathbf{f}}_{k+1}, \quad (3.59)$$

where \mathbf{K}_v is the viscoelastic stiffness matrix, $\tilde{\mathbf{f}}_{k+1}$ is the pseudo nodal force vector, including the external force vector \mathbf{f}_{k+1}^e and the stress history term \mathbf{f}_{k+1}^h . This equation can explain that the contact reaction force at some point of time is dependent not only on the current displacement on the boundary but also on the previous loading history, which is described by the stress history terms. In detail, the viscoelastic stiffness matrix of the FE model, analogously, is the sum of all element stiffness matrices $\{\mathbf{K}_v^e\}$, which are defined as,

$$\mathbf{K}_v^e = \mathbf{B}^T \mathbf{C}^e \mathbf{B} V^e \left(1 + \sum_{i=1}^N \frac{E_i}{E_s} c_i^t \right). \quad (3.60)$$

They can be determined by using the following tetrahedral parameters, V^e is the volume of a single tetrahedron, the matrix \mathbf{B} contains the shape functions, and is still a constant matrix, as that of the linear models. The matrix \mathbf{C}^e is the elastic matrix, which defines the effects of the static elastic material parameters, and it is identical to (3.47). The stress history term of each tetrahedron is illustrated in the form of the element history force,

$$\mathbf{f}_{k+1}^{eh} = \mathbf{B}^T V^e \sum_{i=1}^N e^{-\frac{\Delta t}{\tau_i}} \mathbf{f}_k^i - \mathbf{B}^T \mathbf{C}^e \mathbf{B} V^e \sum_{i=1}^N \frac{E_i}{E_s} c_i^t \mathbf{u}_k, \quad (3.61)$$

where i is the index of the Maxwell elements and the element nodal force component \mathbf{f}_k^i of the i -th Maxwell element can be calculated recursively as follows,

$$\mathbf{f}_{k+1}^i = e^{-\frac{\Delta t}{\tau_i}} \mathbf{f}_k^i + \frac{E_i}{E_s} c_i^t \mathbf{C}^e \mathbf{B} \Delta \mathbf{u}_{k+1}. \quad (3.62)$$

Then the model history force \mathbf{f}^h in (3.59) is the sum of \mathbf{f}_{k+1}^{eh} of all elements,

$$\mathbf{f}_{k+1}^h = \mathcal{A}_{b=1}^n(\mathbf{f}_{k+1,b}^{eh}). \quad (3.63)$$

By this means the FE models with linear viscoelastic behavior are created. Compared with the linear models, the contact force evaluation of these models is computationally more expensive, because the matrices and vectors in (3.60-63) have to be calculated. The evaluation is detailed in Chapter 4.

3.5. Inhomogeneous Model

In continuum mechanics, inhomogeneity characterizes the physical property, if the behavior of a object is variable for different regions under mechanical loads. This difference of the force-displacement relations is the external reflection of the internal variety of the stress-strain relations in the heterogeneous regions. This means the theoretical models, constitutive equations, of the material are not identical for the whole body. There are some typical examples of this material, such as composites, however, in the strict sense, most solid bodies in the real world are more or less inhomogeneous. Thus, it is an important and universal phenomena for haptic simulations. In this case, it is necessary to implement the most suitable material feature for each region, for instance different material values should be assigned for these regions. Then the inhomogeneous contact force models with different regions can present this material feature. In many works, the inhomogeneous materials have been taken into account, especially in medical applications [31, 1]. However, the modeling and the identification of the inhomogeneous models are always problematic and computationally expensive. In connection with identification of inhomogeneous FE models in Chapter 5, a general modeling method of the material inhomogeneity is introduced.

During the modeling, the implementation of a single region is the same as that of the homogeneous models in the former sections. The difference is to model varying regions and to integrate them together as a whole body. There are two alternatives, one is generating the mesh of different regions separately, and then all the components of the regions are integrated as the complete model [125]. The other is based on the identical geometric mesh structure, which is independent on the region boundaries, and each tetrahedron should be determined in which region it belongs. The first one has the advantage that the region boundary is always along the element boundary, however, the remesh is necessary, which is extremely expensive in computing time for the latter model identification. Moreover, the force evaluation may be affected by the different mesh structure as well. The simulations and experiments in Chapter 6

have shown that the second method is more efficient. It is also suitable for the model identification and real time haptic simulations. After exact model identification, the influence of the boundary derivation can be neglected. Furthermore, the effects of the mesh structure on the model accuracy can be eliminated using the second modeling method.

As an example, considering a model with n heterogeneous regions, and all the regions are linear elastic, however the material characteristics are different. According to (3.44), the element stiffness matrices are as follows,

$$\mathbf{K}_j^e = V^e \mathbf{B}^T \mathbf{C}_j^e \mathbf{B}, \quad (3.64)$$

where j is the region index. For homogeneous linear elastic materials, the matrices \mathbf{C}^e and \mathbf{B} are identical for all the tetrahedra in the model. But for inhomogeneous materials, as mentioned above, the elastic matrix \mathbf{C}_j^e is dependent on the position, i.e. the inhomogeneous regions. Hence, the material differences among these regions are described in the elastic matrices,

$$\mathbf{C}_j^e = \frac{E_j}{d_j} \begin{bmatrix} (1 - \nu_j) & \nu_j & \nu_j & 0 & 0 & 0 \\ \nu_j & (1 - \nu_j) & \nu_j & 0 & 0 & 0 \\ \nu_j & \nu_j & (1 - \nu_j) & 0 & 0 & 0 \\ 0 & 0 & 0 & (1 - 2\nu_j)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1 - 2\nu_j)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1 - 2\nu_j)/2 \end{bmatrix}, \quad (3.65)$$

with

$$d_j = (1 + \nu_j)(1 - 2\nu_j). \quad (3.66)$$

Eventually, there is the collection $\{\mathbf{C}_i^e\}$ available for a given inhomogeneous material.

If it is assumed that the region boundaries are estimated, which will be shown in Chapter 5, then the goal is to assign the correct \mathbf{C}_j^e and \mathbf{K}_j^e to each tetrahedron, i.e. to determine which elements belong to which regions. Since the region boundaries are generally modeled as polyhedrons in computational graphics, and the boundary of each region is closed, it can be imaged for one tetrahedron, that a ray starts from the center of the tetrahedron and extends in an arbitrary direction, then several line-surface intersections between the ray and the closed region boundary can be determined. As the criterion, if the number of intersections is odd, the tetrahedron lies inside of this region, otherwise outside of it.

In many applications, the inhomogeneous regions are distributed two dimensionally, for example in the x-y plane, or sometimes the inhomogeneous objects can be modeled in this way, if there is not enough information of the material, and the central purpose is to simulate the haptic behavior of the objects. These models can be called the simplified inhomogeneous models. In these cases, the intersection problem is also simplified as line-line intersections, since only the projections of the boundaries, which are polygons in computer graphics, should be taken into account, then the

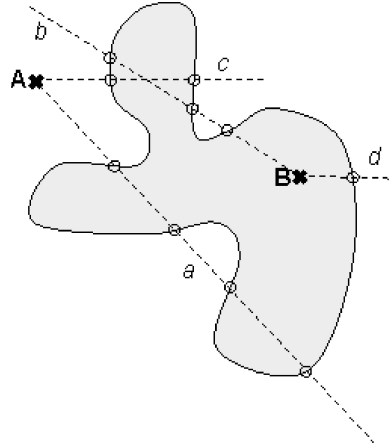


Figure 3.3.: Region determination of two given position points A and B using rays. The rays a and b are in arbitrary directions from the points, respectively, whereas c and d are horizontal.

principle remains the same, however the computational expense can be reduced. Fig. 3.3 illustrates this method in a 2D example for the points A and B with the arbitrary rays a and b . In this implementation, the horizontal rays, c and d , are used, so that the computational expense can be reduced further. This region detection approach is detailed in Algorithm 3.1.

This method has the advantage that the geometric mesh of the model is always identical, and a remesh is not necessary, even if the region boundary is modified. This way, the influence of the mesh structure on the accuracy of the models is reduced as well.

If the regions are not linear elastic, the modeling approach is the same, merely the material characteristics, which should be assigned to each element, are not E and ν in \mathbf{C}^e , but the corresponding parameters in the constitutive relations.

Algorithm 3.1: Region detection

Initialization:

- Import the geometric model
- Number of the regions, n_{reg}
- Import the region boundaries
- Whether simplified inhomogeneous models or not

for each tetrahedron **do**

tetrahedron i

Calculate the center of the i -th tetrahedron

Create the horizontal ray

for each region **do**

counter $c_{intersection} = 0$

for each primitive (polygon or segment) of this boundary **do**

if intersection of the current primitive **then**

$c_{intersection} = c_{intersection} + 1$

end if

end for

if $c_{intersection}$ is odd **then**

Inside of the region

Assign the corresponding material characteristics

Break

end if

end for

Next element

end for

Inhomogeneous FE model with n_{reg} regions

4. Contact Force Evaluation

This chapter focuses on the contact force evaluation using FE models and the methods for haptic rendering. Since FE models are computationally expensive, especially the nonlinear models, a model based data fusion method is also introduced for real time haptic simulations.

4.1. Finite Element Evaluation

The finite element models are generally large scale systems. In order to solve them, considering (3.43) and (3.59), the relations between the boundary displacement and the nodal force of the FE models in the last chapter can be formulated as the following form,

$$\mathbf{Ax} = \mathbf{b}, \quad (4.1)$$

where \mathbf{A} is a symmetric and positive definite matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$. Hence the contact force evaluation of these FE models can be regarded as the solving of this linear system. There are two well known methods for this problem, the conjugate gradient method (CG) and the Gaussian elimination.

4.1.1. Conjugate Gradient Method

In this chapter, the evaluations of the FE models are based on the conjugate gradient method [8, 88, 55]. The basic idea of the CG is the generation of a set of conjugate vectors \mathbf{p}_i , which satisfy the following relation,

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0 \quad \text{with} \quad i \neq j. \quad (4.2)$$

This means that the conjugate vectors are linearly independent of each other.

The CG algorithm is also an optimization algorithm, thus in principle, the linear system (4.1) should be rewritten as an optimization problem with the objective function of the vector \mathbf{x} ,

$$\mathbf{f}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x}. \quad (4.3)$$

This is a convex quadratic function, and the residual of the linear system, \mathbf{r} , is defined as follows, which is also equal to the gradient of the objective function. Transforming (4.1) into the following form,

$$\mathbf{Ax} - \mathbf{b} \stackrel{\text{def}}{=} \mathbf{r} = \nabla \mathbf{f}, \quad (4.4)$$

the solution of (4.1) is equivalent to the minimization of the residual iteratively, using the linear search method. Then in each iteration, \mathbf{x} can be determined according to

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_{k-1} \mathbf{p}_{k-1}, \quad (4.5)$$

where α is the one dimensional minimizer along the direction \mathbf{p} ,

$$\alpha_{k+1} = -\frac{\mathbf{r}_{k+1}^T \mathbf{p}_{k+1}}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}. \quad (4.6)$$

The algorithm converges in at most n steps, for a problem with arbitrary start values, \mathbf{x}_0 . Furthermore, the convergence of a concrete problem is dependent on the distribution of the eigenvalues of the coefficient matrix.

Applying the CG algorithm to the evaluation of FE models, the coefficient matrix corresponds to the stiffness matrix, and \mathbf{x} and \mathbf{b} are the nodal displacement and the nodal force vectors. For the solid mechanic problems, most entries in the vector \mathbf{u} are generally unknown. There are two situations of this problem, one is the calculation of the deformation by some boundary traction, and the other is the evaluation of the contact reaction force based on the known boundary displacements.

From the aspect of the model evaluation, the boundary conditions are critical factors, and they can be divided into two kinds, constraints and boundary loads. The load may be variable displacements and external forces at some boundary nodes, whereas for the constraints, the displacements of the corresponding nodes are always constant, and generally they are equal to zero. Both of the two boundary conditions are implemented in the nodal displacement and force vectors, \mathbf{u} and \mathbf{f} , respectively. Then, the CG algorithm solves the FE deformable bodies problems in three phases:

- Initialize: Given the matrix \mathbf{K} , the starting value \mathbf{u}_0 , the boundary conditions, including the traction in vector \mathbf{f} and the constraints of the model.
- Loop: Iterative solution of (4.1). The optimization should be terminated if specific conditions are satisfied, such as the convergence criterion and the maximal number of iterations. In each iteration, the following terms should be determined, including the deformation,

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{p}_k, \quad (4.7)$$

and the optimizer

$$\alpha_{k+1} = -\frac{\mathbf{r}_{k+1}^T \mathbf{p}_{k+1}}{\mathbf{p}_k^T \mathbf{K} \mathbf{p}_k}. \quad (4.8)$$

The descent direction \mathbf{p}_{k+1} is dependent on the steepest descent direction,

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k \mathbf{K} \mathbf{p}_k, \quad (4.9)$$

and the previous direction \mathbf{p}_k , that is,

$$\mathbf{p}_{k+1} = -\mathbf{r}_k + \beta_{k+1} \mathbf{p}_k. \quad (4.10)$$

The auxiliary factor β is required to ensure that \mathbf{p}_{k+1} and \mathbf{p}_k are conjugate with respect to \mathbf{K} ,

$$\beta_{k+1} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}. \quad (4.11)$$

- Output: Extract the required values from the corresponding nodes, if the convergent criterion is satisfied.

In this chapter, since the contact force is more important for the haptic simulation in comparison to the boundary displacement, the emphasis is to evaluate the nodal force using the algorithm, if the boundary deformation at some positions is given. Eventually, the contact reaction force can be extracted as the sum of the according nodal forces.

$$\mathbf{f}^{react} \xleftarrow{Extract} \mathbf{f}^{nodal} \quad (4.12)$$

As mentioned above, the stiffness matrices are sparse matrices, and the according algorithms are used for the matrix multiplications. Then the computational expense is obviously reduced. For the evaluation of the boundary displacement, the algorithm is similar, but the boundary conditions are different.

The conjugate gradient (CG) method is especially suitable for large scale problems, since it does not need to modify the coefficient matrix and it converges sometimes more quickly than other methods. It should be noticed that the convergence of the CG method depends on the distribution of the matrix \mathbf{K} , and further on the mesh of the FE model as well. Hence the models with better mesh structures are converging more quickly than those with bad mesh. The simulations have confirmed this conclusion by using the models with quality mesh, shown in Chapter 2, and normal mesh.

4.1.2. Practical Implementation

The conjugate gradient (CG) method can be implemented for the linear elastic models by substituting the vectors \mathbf{u} and \mathbf{f} directly into the algorithms, i.e.

$$\begin{cases} \mathbf{u} \rightarrow \mathbf{x} \\ \mathbf{f} \rightarrow \mathbf{b}. \end{cases} \quad (4.13)$$

Then the boundary displacements and the contact reaction forces can be determined.

However, for the viscoelastic models, the CG algorithms can not be used directly with external force \mathbf{f}_{k+1}^e , but by using the pseudo nodal force $\tilde{\mathbf{f}}_{k+1}$. Furthermore, it should be noted that the evaluation of the viscoelastic models is according to each point of time,

$$\begin{cases} \mathbf{u}_{k+1} \rightarrow \mathbf{x} \\ \tilde{\mathbf{f}}_{k+1} \rightarrow \mathbf{b}. \end{cases} \quad (4.14)$$

This way, $\tilde{\mathbf{f}}_{k+1}$ can be determined. Since the history terms depend only on the states of the previous time step, it can be calculated directly, using (3.61-62). Subsequently, the external force \mathbf{f}_{k+1}^e can be determined recursively, (3.59), and the contact reaction force at the corresponding nodes can be extracted from it.

In principle, the more the parallel Maxwell elements are implemented, the more accurate and detailed effects can be described, however the computational requirements also increase significantly with it. Furthermore, some time constants may have little importance in practice. Hence as the trade-off, referring to the simulation results, the generalized models involving one spring and two Maxwell elements are often applied in many implementations, so that the three most important characteristics can be taken into account: the static elastic component, the short-term and the long-term relaxations. This means $N = 2$ in (3.56), and the corresponding model parameters are

$$\chi = [E_s \quad \nu \quad E_1 \quad \tau_1 \quad E_2 \quad \tau_2]^T. \quad (4.15)$$

In some applications, it may be necessary to include more Maxwell elements, however the algorithm remains still analogously. Then for viscoelastic models, the practical implementation of (3.59-63) for a time sequence of force evaluation is illustrated in Algorithm 4.1.

Algorithm 4.1: Force evaluation of linear viscoelastic FE models

Initialization:

- \mathbf{u}_0 and \mathbf{f}_0^{eh}
- Define Δt
- Time scale $t = 0$

loop

Determine $\tilde{\mathbf{f}}_{k+1}$ and \mathbf{u}_{k+1} , using the CG method.

Calculate \mathbf{f}_{k+1}^{eh}

Determine \mathbf{f}_{k+1}^h , using (3.63)

Determine \mathbf{f}_{k+1}^e :

- $\mathbf{f}_{k+1}^c - \mathbf{f}_{k+1}^h = \tilde{\mathbf{f}}_{k+1}$

Extract the contact force from the corresponding nodes

Calculate the contact reaction force

$t = t + \Delta t$

if $t > t_{max}$ **then**

Break

end if

if Quit **then**

Break

end if

end loop

For inhomogeneous models, the evaluations of elements with different material features are according to the associated methods.

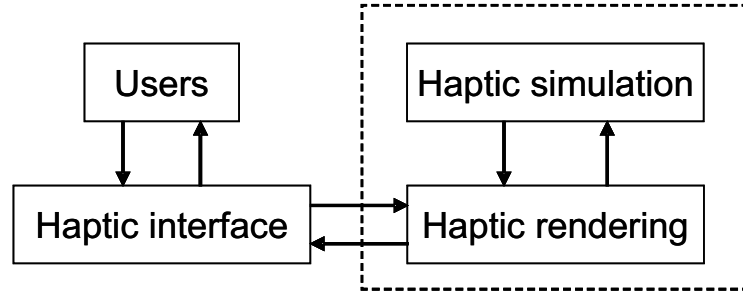


Figure 4.1.: Haptic rendering. The users send operations and receive force feedback through some haptic devices as interface. And the contact force model provide the haptic simulations and haptic rendering.

4.2. Haptic Rendering

The haptic rendering [100] gives human being the chance to apply the haptic ability in other real or virtual environments. Some works present the haptic simulation of human hands in virtual reality [70, 112]. In this section, the central point is to represent the haptic information based on contact models and haptic simulations, Fig. 4.1.

The haptic simulations are presented to users by using haptic devices [53], as the user interface, which have been employed in a lot of research [27, 103]. Most haptic devices can provide contact force at $1000[Hz]$, however the implementation of the above mentioned finite element models with normal PCs is computationally too expensive for the real time haptic rendering using these haptic devices. Several quantitative examples are detailed in Chapter 6. In the simulations, time delays at about $50[ms]$ may already lead to significant roughness in feeling and affect the operations. Furthermore, if the computational expense depends on the model mesh structure, the output rates of different models may be different as well. That is an additional difficulty for users to learn through experience and training, such as by using surgical simulators.

To solve this problem, data fusion methods for real time haptic simulations are developed in the following sections.

4.3. Analytical Contact Force Model

The analytical solution of the contact reaction force has also been investigated in some papers [35]. The most important advantage of it is the simplicity in implementation and the computational expense. Furthermore, the analytical solutions are more accurate than the numerical ones. There are two kinds of the analytical models, one is based on the system dynamic equation

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} + \mathbf{f} = 0, \quad (4.16)$$

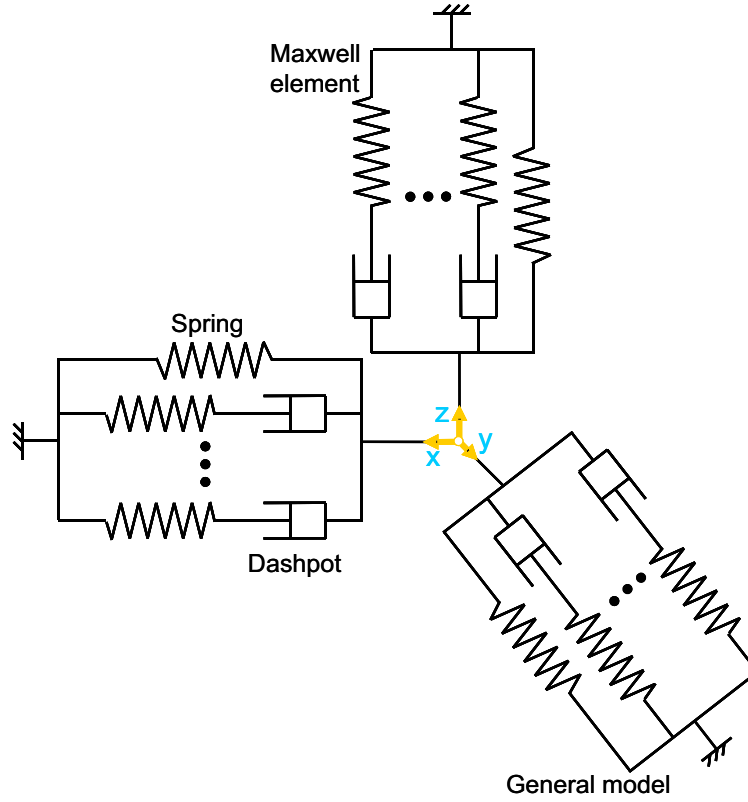


Figure 4.2.: Analytical contact force model of viscoelastic materials. This diagram show a component at a single position, consisting of the generalized models in each coordinate direction.

using idealized models, for instance springs. These models can merely describe limited material behavior. Therefore, they can just be applied for simple systems. The other is using the constitutive equations. However, they are solvable only for some special cases, for instance the plane stress state and the plane strain state.

In this section, a new modeling method of analytical contact force models (ACFM) is developed, based on the spring-dashpot elements. The concrete implementation of these models depends on the according material features. For the linear elastic materials merely the spring elements are necessary, whereas for the viscoelastic materials the generalized Maxwell elements are used, analogously to Section 3.1.3. The following descriptions are according to the viscoelastic analytical models, that of linear elastic materials is analogous.

The basic structure of the ACFM is that in each coordinate direction, a generalized model is applied, respectively, Fig. 4.2. However, the differential equations describe not the stress-strain states but the force-displacement relation. Thus, the parameters are not the Young's modulus and the viscosity, but the spring constant k and the mechanical damping coefficient d . These models are less accurate, however they are computationally efficient for real time haptic simulations, and the output rate may be over $1000[Hz]$ with most PCs in recent years.

Moreover, in order to improve the accuracy of the ACFMs, the analytical force models in Fig. 4.2 are defined as sub-models in connection with their positions, this

means the analytical models are defined discrete at several positions. For an object, the ACFM can be divided into two parts, the known part and the unknown part. The known part is a set of sub-models at the positions, where the corresponding information is available. The number of these sub-models are finite. Whereas the unknown part consists of infinite sub-models, at arbitrary positions. The features of the unknown components can be determined based on the known sub-models. Since the structure of each sub-model is relatively simple and it is always the same for all sub-models, the ACFMs are noted using the parameters of it. For a single sub-model, the parameters are defined as a function of positions,

$$\chi_i = f(x_i, y_i, z_i), \quad (4.17)$$

where χ_i is the vector of the model parameters. Then the known part of the complete ACFM is formulated as the collection of χ_i , which are components at available positions with definitive characteristics,

$$\{\chi_i, x_i, y_i, z_i : i = 1, \dots, m\}. \quad (4.18)$$

During the force estimations, a template component, an unknown sub-model, is created at the desired position, and the parameters of it are determined by using interpolation based on the known components. Some known sub-models are selected as samples and it must be noticed, that the samples in the similar direction of the current position should be neglected. Then the local parameters of the template component are determined as the weighted mean of the parameters of the selected samples,

$$\chi_{temp} = \sum_{i=1}^n \frac{\bar{d}_i}{\sum_{j=1}^n \bar{d}_j} \chi_i, \quad (4.19)$$

where \bar{d} is the distance between the required position and the known components. It is assumed that there are enough available components, and n of them are involved in the calculation. By this means, although the sub-models are merely material models, without geometric information, the effects of the geometry can be taken into consideration by calculating the weighted average of the components at different positions.

The analytical method can be applied for linear and nonlinear models. However, it should be mentioned that the linear models are generally economical in computation, and the expense can be limited to fit the requirements of real time simulation by some implementation techniques, such as using selected numerical methods and appropriate algorithm settings in Section 4.1. Hence the ACFMs are especially suitable for the nonlinear, viscoelastic models.

4.4. Data Fusion Algorithms

To obtain real time haptic simulations using the two modeling methods, some difficulties have to be overcome. For the finite element models, the computational

expense is one of the most dominant challenges. The analytical contact force models (ACFM) are computationally efficient, however the accuracy of the ACFM can not satisfy the requirement of high level haptic simulations. Hence, suitable data fusion approaches should be developed, and the basic idea is to fuse the accurate, low frequency haptic simulations derived by the FE models with the high frequency information derived the analytical contact force models using the data fusion method. As a time sequence problem, this fusion procedure can be considered as a Bayesian estimation problem. That is to estimate the current probability of system states using all available measurements [17].

The Kalman filter (KF), introduced by Kalman [64], is a model based optimal estimation method for the Bayesian problems. It is suitable for linear systems. In order to apply the filtering algorithm to nonlinear systems, several KF based approaches have been developed, for instance the extended Kalman filter (EKF) and the sigma-point Kalman filters (SPKF) [80]. The fundamental thought of these approaches is the linearization of the nonlinear system at a system state. In all of these estimation methods, a Gaussian distribution is assumed. A more capable approach for the Bayesian estimations is the sequential Monte Carlo method (SMC) [32, 5], also known as the particle filtering (PF). As a simulation based method, it is suitable for arbitrary posterior distributions of the system states, also for non-Gaussian and nonlinear systems.

The system models can be written in the following general form, i.e. the system equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}), \quad (4.20)$$

and the observation equation

$$\mathbf{z} = \mathbf{g}(\mathbf{x}, \mathbf{v}). \quad (4.21)$$

Both functions \mathbf{f} and \mathbf{g} can be nonlinear, the vectors \mathbf{w} and \mathbf{v} are the system and measurement noises, respectively.

The goal of the data fusion in haptic rendering is to provide haptic simulations with high frequency, such as $1000[Hz]$. The original data come from the two models, one is relatively accurate with low output rate, the other is suitable for high output rate. In this case, the noises \mathbf{w} and \mathbf{v} can be better determined, as the weights of the information in the fusion procedure. Therefore the non-Gaussian effects are not dominant, and the computational expense of the SMC has little meaning. Furthermore, the nonlinear components in an analytical model can be modeled as a set of piecewise linear models, hence the extended Kalman filter is suitable for this application.

It should be noted that the “measurement” in the following sections in this chapter means the measurement in the state-space models not the factual measurement in experiments.

4.4.1. Extended Kalman Filter

The extended Kalman filter (EKF) is a widely used model estimation and data fusion algorithm [78, 123]. There is a variety of algorithms based on the EKF. The typical applications of it are for example motion estimation systems, navigation systems and inertial measurement system (IMS). Here, the EKF is applied in a new area as an engine for haptic rendering, i.e. to provide force feedback.

As a sub-optimal observer, the EKF estimates the system states based on linearized system models, which are derived by the linearization and time discretization of the general system model (4.20) and (4.21), with the system equation,

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{G}_{k+1} \mathbf{u}_{k+1} + \mathbf{w}_{k+1}, \quad (4.22)$$

and the observation equation

$$\mathbf{z}_{k+1} = \mathbf{H}_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1}, \quad (4.23)$$

where \mathbf{x} is the state vector, \mathbf{u} is the input vector and \mathbf{z} is the output vector. The matrices Φ , \mathbf{G} and \mathbf{H} are the transition matrix, input matrix and output matrix, respectively. Since the system equations of the ACFMs are linear, analogous to (3.56), they can be implemented directly in these equations. However the transition and input matrices are nonlinear functions of positions,

$$\begin{aligned} \Phi &= \Phi(x, y, z) \\ \mathbf{G} &= \mathbf{G}(x, y, z) \end{aligned} \quad (4.24)$$

The vectors are defined as follows: the system state vector,

$$\mathbf{x} = \mathbf{f}_{ACFM} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T, \quad (4.25)$$

the input vector

$$\mathbf{u} = \begin{bmatrix} \Delta s_x & \Delta s_y & \Delta s_z & \Delta \dot{s}_x & \Delta \dot{s}_y & \Delta \dot{s}_z \end{bmatrix}^T, \quad (4.26)$$

and the output vector

$$\mathbf{z} = \mathbf{f}_{FEM} = \begin{bmatrix} f_{xm} & f_{ym} & f_{zm} \end{bmatrix}^T. \quad (4.27)$$

where Δs is the boundary displacement.

The extended Kalman filter estimates system states in two steps, the first is the prediction based on previous measurements $\{\mathbf{z}_{1:k}\}$,

$$\mathbf{x}_{k+1|k} = \Phi_k \mathbf{x}_{k|k} + \mathbf{G}_k \mathbf{u}_k. \quad (4.28)$$

The predictions of the measurements

$$\mathbf{z}_{k+1}^p = \mathbf{H}_k \mathbf{x}_{k+1|k} \quad (4.29)$$

and the covariance matrix

$$\mathbf{P}_{k+1|k} = \Phi_k \mathbf{P}_{k|k} \Phi_k^T + \mathbf{Q}_k^n \quad (4.30)$$

are also determined with available information. The second step is the state update using the current measurements \mathbf{z}_{k+1} , the simulation from the FE model,

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{z}_{k+1}^p). \quad (4.31)$$

The covariance matrix is also updated,

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} + \mathbf{K}_{k+1} \mathbf{S}_{k+1} \mathbf{K}_{k+1}^T, \quad (4.32)$$

with the Kalman gain matrix

$$\mathbf{K}_{k+1} = \frac{\mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T}{\mathbf{S}_{k+1}}, \quad (4.33)$$

and the auxiliary matrix

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}^n. \quad (4.34)$$

Applying the model (4.22) and (4.23) to the EKF, the data fusion algorithm for haptic simulations is illustrated in Algorithm 4.2. The EKF predicts the

Algorithm 4.2: Extended Kalman filter

Initialization.

loop

Step k

Prediction using (4.28-30)

Update using (4.31-34)

$k = k + 1$

if Quit **then**

Break

end if

end loop

contact reaction force using the system equation (4.28), which is according to the analytical models, and updates the force using available information from the finite element models, as the system measurements, (4.31). With this means, multifarious information is fused. The dominant parameters of performance of the EKF are the noise densities in the diagonal matrices \mathbf{Q}^n and \mathbf{R}^n ,

$$\begin{aligned} \mathbf{Q}^n &= \text{diag} [q_{ii}] = \text{diag} [\sigma_{fx} \quad \sigma_{fy} \quad \sigma_{fz}] \\ \mathbf{R}^n &= \text{diag} [r_{ii}] = \text{diag} [\sigma_{fxm} \quad \sigma_{fym} \quad \sigma_{fzm}]. \end{aligned} \quad (4.35)$$

The diagonal entries q_{ii} and r_{ii} are the noise densities of the corresponding noises w_i , v_i in the system and measurement noise vectors, respectively. If there are enough

available samples, they can be calculated as follows,

$$\sigma_i^2 = \frac{1}{n-1} \sum_j^n (x_{ij} - \bar{x}_i). \quad (4.36)$$

4.4.2. Example

The modeling methods of the ACFM and the data fusion approaches are introduced in the last sections. In order to explain them intuitively, they are here illustrated in an example. Considering the simplest analytical model of the viscoelastic materials with only one Maxwell element and a spring in each coordinate direction, the relation of the force f and the displacement Δs in one dimension is described in the following equation, analogous to (3.18),

$$\begin{aligned} \dot{f} + \frac{k_1}{d_1} f &= \frac{k_s k_1}{k_1} \Delta s + (k_s + k_1) \dot{\Delta s} \\ \Rightarrow \dot{f} &= -\frac{k_1}{d_1} f + \frac{k_s k_1}{k_1} \Delta s + (k_s + k_1) \dot{\Delta s}. \end{aligned} \quad (4.37)$$

Substituting the time derivative of contact force $\dot{f} = \frac{f_{k+1} - f_k}{\Delta t}$ into this equation, and applying it in three coordinate directions, it yields

$$\begin{bmatrix} f_{x,k+1} \\ f_{y,k+1} \\ f_{z,k+1} \end{bmatrix} = \begin{bmatrix} -\frac{k_{x1}}{d_{x1}} f_{x,k} + \frac{k_{xs} k_{x1}}{k_{x1}} \Delta s_{x,k} + (k_{xs} + k_{x1}) \Delta \dot{s}_{x,k} + f_{x,k} \\ -\frac{k_{y1}}{d_{y1}} f_{y,k} + \frac{k_{ys} k_{y1}}{k_{y1}} \Delta s_{y,k} + (k_{ys} + k_{y1}) \Delta \dot{s}_{y,k} + f_{y,k} \\ -\frac{k_{z1}}{d_{z1}} f_{z,k} + \frac{k_{zs} k_{z1}}{k_{z1}} \Delta s_{z,k} + (k_{zs} + k_{z1}) \Delta \dot{s}_{z,k} + f_{z,k} \end{bmatrix}. \quad (4.38)$$

Subsequently, the matrices can be determined corresponding to the system equation (4.38), including the transition matrix,

$$\Phi = \begin{bmatrix} 1 - \frac{k_{x1}}{d_{x1}} \Delta t & 0 & 0 \\ 0 & 1 - \frac{k_{y1}}{d_{y1}} \Delta t & 0 \\ 0 & 0 & 1 - \frac{k_{z1}}{d_{z1}} \Delta t \end{bmatrix}, \quad (4.39)$$

the input matrix

$$\mathbf{G} = \begin{bmatrix} g_x^1 & 0 & 0 & g_x^2 & 0 & 0 \\ 0 & g_y^1 & 0 & 0 & g_y^2 & 0 \\ 0 & 0 & g_z^1 & 0 & 0 & g_z^2 \end{bmatrix} \quad (4.40)$$

and the output matrix

$$\mathbf{H} = \mathbf{I}^{3 \times 3}, \quad (4.41)$$

where the factors are defined as

$$\begin{aligned} g_i^1 &= \frac{k_{i1} k_{is}}{d_{i1}} \Delta t \\ g_i^2 &= (k_{i1} + k_{is}) \Delta t. \end{aligned} \quad (4.42)$$

As mentioned above, since the ACFM is less accurate, the parameters

$$\chi_{ACFM} = [k_{xs} \quad k_{x1} \quad d_{x1} \quad k_{ys} \quad k_{y1} \quad d_{y1} \quad k_{zs} \quad k_{z1} \quad d_{z1}]^T \quad (4.43)$$

are defined as a function of positions,

$$\chi_{ACFM} = f(x, y, z), \quad (4.44)$$

to improve the accuracy. The noise density matrices are identical as (3.45).

Furthermore, the numerically determined contact reaction forces are dependent on the time interval $\Delta t = t_{k+1} - t_k$ as well, hence it is reasonable to define Δt as a constant for accurate haptic simulations.

5. Model Identification

The force models must represent the behavior of corresponding real or virtual objects. The geometry of the objects can be measured or imported as shown in Chapter 2, however it is difficult to measure the material parameters directly. Therefore, the material characteristics of the force models should be identified.

5.1. Problem Statement

The models with the geometric mesh and some predefined values of the model parameters can already simulate the contact force in principle. These models can be used, for instance in virtual reality, if the model parameters are known. However in many situations, the model parameters are at least partially unknown or inaccurate, hence they should be determined based on available reference data, which are generally measurements during experiments. The model parameters in the force models involve mainly the geometric parameters and the material parameters. The geometric parameters specify the model geometry, as well as the mesh structure. On the one hand, some of them can be measured directly, and the accuracy of the geometric measurements is normally sufficient for haptic simulations, for instance the object surface. On the other hand, some geometric parameters can be determined according to application conditions. In many applications, there is a trade-off among these conditions, such as the mesh density for the computational expense and the model accuracy. The measurement of material characteristics is more problematic. Therefore, the model identification here is concentrated on the material parameters, which describe the material feature and behavior, as shown in Chapter 3. Thus, it is assumed in this chapter that the geometric models are already established, but the material features, the constitutive relations, are not implemented in them.

This model identification can be divided into two steps. The first one is the determination of which theoretical model is suitable for a given object and which parameters should be implemented. The second step is assigning values to the material parameters so that the models can be adjusted to simulate the given objects. These two steps are known as the model learning and the model parameter optimization, which are detailed in the following sections.

5.2. Model Learning

The model learning can be regarded as a data mining procedure [104], i.e. extracting the most valuable information from diverse data. There are different methods for this task, such as varying classification methods. For the contact force models, the direct information of materials is initially unavailable, hence it is assumed in the beginning

that the model is homogeneous and linear elastic. If the material is inhomogeneous, significant differences will be detected among the samples from different sample positions in the subsequent classification. This means if the material is linear elastic, the model can be optimized directly, otherwise if the material is inhomogeneous or nonlinear, the primary estimations of the heterogeneous regions and the theoretical models are necessary. This unsupervised model learning is especially important for the inhomogeneous models, because it is much more difficult in comparison to other nonlinear effects, such as the viscoelasticity.

At first, the measurements from a single sample position are used for the estimation of the local material parameters using the Gauss-Newton method, detailed in Section 5.3.2, under the assumption of homogeneity, and the parameters are defined as the location sample \mathbf{x}^l . In the second step, the location samples should be classified to estimate the inhomogeneous regions. However, the correlation of the information in \mathbf{x}^l may be complex, and the computational expense of the classification increases significantly with the size of the location sample vectors. Hence preprocessing methods [15] are valuable to extract the feature and to reduce the dimension. The principal component analysis (PCA) is a well known approach for this purpose [95, 120]. Using the PCA, the location samples are transformed into the corresponding principal component vectors. The accuracy and efficiency of the classification can be improve obviously.

The classification of the location samples, or the principal component vectors, is implemented using the hierarchical cluster analysis [75]. If the distances of different layers are beyond a threshold, the inhomogeneous regions in the material should be recognized, and the samples should be divided into several groups, which correspond to the different regions, respectively. Otherwise the linear elastic assumption should be accepted in the subsequent modeling.

For the inhomogeneous models, after the identification of the inhomogeneous regions, the boundaries between them are estimated using virtual scan points and the quadratic discriminant analysis (QDA) [113]. Subsequently, the average material parameters of the location samples in the same region are defined as the corresponding values of it, then the primary model is established, Fig. 5.1. The applied methods and the data processing are detailed in the following sections.

5.2.1. Location Sample Estimation

The original data of a test object for the model identification are a set of contact reaction force with the associated boundary displacements at different positions. They are defined as a collection of the original samples $\{\mathbf{x}_i^o : i = 1, \dots, m\}$

$$\begin{aligned} \mathbf{x}_i^o &= [\mathbf{f}_s^T \quad \mathbf{p}_s^T \quad \Delta \mathbf{p}^T]^T \\ &= [f_x \quad f_y \quad f_z \quad x \quad y \quad z \quad \Delta x \quad \Delta y \quad \Delta z]^T, \end{aligned} \quad (5.1)$$

in which m is the number of the samples. The vectors \mathbf{f}_s , \mathbf{p}_s and $\Delta \mathbf{p}$ are the force measurement, the sample position and the boundary displacement.

The aim is to obtain suitable direction information of the material, i.e. the

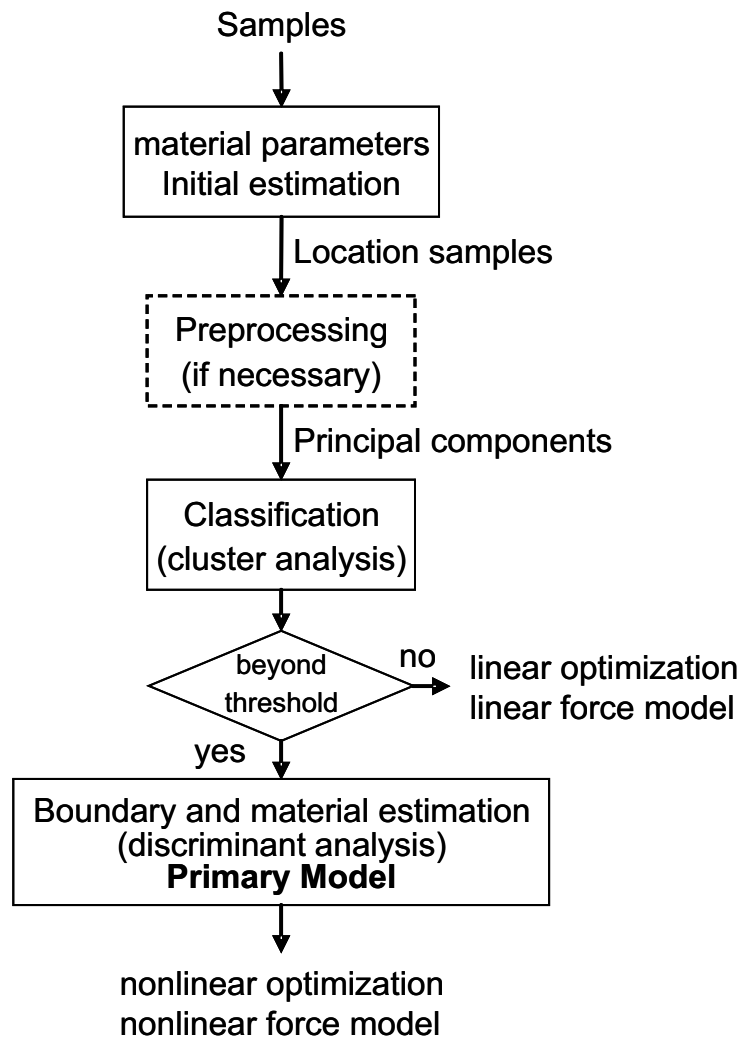


Figure 5.1.: Unsupervised model learning. The preprocessing using the PCA is an optional function, which can improve the performance of the model learning.

constitutive relation and the corresponding material parameters. Since the material behavior is unknown at the beginning, the basic material model, elasticity, is assumed. That means, the object is modeled as homogeneous material with the linear elastic constitutive equation, and the material parameters are assigned with default values. This is the initial estimation of the test object.

As a simple constitutive equation, it is naturally not sufficient in many situations. However, just with this initial estimation, the following detailed modeling of different materials can be carried out. In order to include the material diversity within the object as much as possible, the original samples are sorted according to their sample positions, and the samples at the same sample position are used to create the

location samples, which are defined as follows,

$$\begin{aligned}\mathbf{x}_i^l &= \begin{bmatrix} (\mathbf{p}_i^s)^T & (\boldsymbol{\chi}_i^l)^T \end{bmatrix}^T \\ &= \begin{bmatrix} x & y & z & E & \nu \end{bmatrix}^T,\end{aligned}\tag{5.2}$$

where $\boldsymbol{\chi}_i^l$ is the material parameters of the i -th location sample. For linear elastic materials, the material characteristics are the Young's modulus (E) and the Poisson's ratio (ν). The estimation of the two parameters is based on the comparison of the simulation and the original samples. It is modeled as a least squares problem, and can be solved using the Gauss-Newton method, Section 5.3. This initial estimation of the location samples is illustrated in Algorithm 5.1.

Algorithm 5.1: Initial estimation of the location samples

```

Import samples  $\{\mathbf{x}_i^o\}$ 
Sorting them according to sample positions.
Import the geometric models with the mesh structure.
Set the default values  $E_0$  and  $\nu_0$ .
for  $i = 0$  to  $m$  do
    Assign the material parameter:
     $E = E_0$  and  $\nu = \nu_0$ 
    Parameter identification using Gauss-Newton Method, Section 5.3.
    Defining the location sample  $\mathbf{x}_i^l$ 
     $i = i + 1$ 
end for
The set of location samples:  $\{\mathbf{x}_i^l\}$ 

```

Then the location samples are the first direct estimations of the material.

5.2.2. Preprocessing

For the subsequent model identification and analysis, the location samples should be prepared and processed. However the location samples are varied and sometimes they are correlated complexly with each other, so that they may not represent the characteristics of the test objects accurately, and the differences may be not distinguished enough. Thus, the efficiency and accuracy of the classification can be affected significantly. For this reason, it is not suitable to apply them directly for the model learning in some cases. To reduce these effects, preprocessing can be used as an optional procedure. That means the preprocessing is not necessary for all model identifications. But it can improve the quality of the input data as well as the following data processing by extracting the most important information and difference from the input data.

In this application, the principal component analysis method (PCA) is implemented [59]. The aim is to reduce the dimensionality of the input data by extracting the most important feature into the first few entries of the principal component vectors. The central point of the PCA is a set of linear combinations, known as the

principal component transformation of input variables, which is an orthogonal transformation. This means, the new variables after the transformation are uncorrelated with each other. For an original n -row sample vector \mathbf{x}_i , the linear transformation is

$${}^i\mathbf{q} = \mathbf{U} \cdot \mathbf{x}_i, \quad (5.3)$$

where the output vector \mathbf{q} is the principal component vector, and the entries in it are principal components of \mathbf{x}_i ,

$${}^i\mathbf{q} = [{}^iq_1 \quad \cdots \quad {}^iq_n]^T. \quad (5.4)$$

The transformation matrix \mathbf{U} consists of the eigenvectors of the covariance matrix of the input vectors $\{\mathbf{x}_i\}$. Hence the output principal components are orthogonal to each other, and the variance of the outputs is reduced in comparison with that of the inputs. The size of the input vector and the principal component vector is the same, however the variance of the input data is concentrated in the first few principal components. Hence the dimension of the data can be reduced, and the principal component vectors can be reduced in the following form, for example with just two entries,

$${}^i\mathbf{q} \rightarrow {}^i\mathbf{q}^r = [{}^iq_1 \quad {}^iq_2]^T. \quad (5.5)$$

The PCA can also be implemented in a linear neural network.

Applying \mathbf{x}^l as the input data, the corresponding reduced principal component vectors \mathbf{q}^{rl} can be used as an alternative to the location samples for the classification in the next section.

5.2.3. Location Sample Classification

After the primary estimation of the location samples, the first direct material information is available. However, the initial assumption is not sufficient in many cases, and the material features of the whole object should be estimated subsequently in more detailed ways. For this reason, the location samples are further investigated. A basic question is whether there are significant differences among them and how distinguishable the differences are. The physical meaning of the significant differences may be diverse material regions.

The mathematic formulation of this problem is the unsupervised classification as follows. Defining a collection of n samples, $\{\mathbf{x}_i : i = 1, \dots, n\}$, as the input data, which may be the location samples or the corresponding principal components after the principal component analysis.

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_i^l & \text{without PCA} \\ \mathbf{q}^l \text{ or } \mathbf{q}^{rl} & \text{with PCA} \end{cases} \quad (5.6)$$

Then the goal is to divide the samples into g homogeneous classes, and g is the unknown number of the classes. The cluster analysis is suitable for this task [75],

and the classes are also called clusters.

The cluster analysis, as a classification method, has been used in many applications [94, 3]. In this chapter, the hierarchical clustering is implemented to sort the location samples into different groups. This procedure is based on the distances between the samples. There are diverse methods for the distance calculation, for instance the Euclidean distance, the Manhattan distance and the Mahalanobis distance. Here the normal Euclidean distances are calculated,

$$d_{ij} = \|\mathbf{x}_i^l - \mathbf{x}_j^l\|_2. \quad (5.7)$$

Then a cluster tree, or dendrogram, which involves all the location samples, can be created based on these distances. In order to verify the cluster tree, the cophenetic correlation coefficient, \hat{c}_{coph} , is used as a measure to characterize the correlation between the normal distances of the location samples and the cophenetic distances in the cluster tree,

$$\hat{c}_{coph} = \frac{\sum_{i < j} (d_{ij} - \bar{d}) \cdot (l_{ij} - \bar{l})}{\sqrt{\sum_{i < j} (d_{ij} - \bar{d})^2 \cdot \sum_{i < j} (l_{ij} - \bar{l})^2}}, \quad (5.8)$$

where d_{ij} is the normal distance between the i -th and j -th samples, and \bar{d} is the average distance, whereas l_{ij} and \bar{l} are the similar terms of cophenetic distances, respectively. Since the cophenetic distances l_{ij} are the distances between different layers, they can be calculated using different values, i.e. the positions of the layers can be defined as the middle, the furthest or the nearest values of them. This difference may affect the cluster tree and the cophenetic correlation coefficient. In order to take all samples into account uniformly, the middle values are calculated in this implementation. As a correlation coefficient, \hat{c}_{coph} is in the range $\hat{c}_{coph} \in [0 \ 1]$, and the higher the value, the more original information is represented in the cluster tree.

Since the cluster tree links all the location samples in different hierarchies, the classification of the samples is according to the division of the cluster tree into different partitions, whereas over fitting should also be avoided. The partitions correspond to the inhomogeneous regions in the model. Therefore a division criterion should be defined suitably. There are two alternatives, one is the number of the groups, if it is known or presumed. According to the given number of the groups, the cluster tree is cut at some proper height. The other is the natural division, which is based on the difference among the entries in the cluster tree. This division can be concluded as: if significant differences exist, suitable groups should be recognized. To evaluate the differences, the measure is defined as the ratio of the distances of the neighbor layers,

$$r_{ca}^i = \frac{d^i}{\frac{1}{n} \sum_{l=1}^n d^{i-l}}, \quad (5.9)$$

where d^i is the distance of the i -th layer, and n is the number of layers, which are

taken into account. If some threshold thr_{ca} is exceeded,

$$r_{ca}^i > thr_{ca}, \quad (5.10)$$

it is satisfactory to create a new group with the associated samples in the cluster tree. Checking all the layers over the cluster tree, then the samples can be classified completely. The selection of the criterion depends on the application conditions.

Algorithm 5.2: Hierarchical cluster analysis

```

Import the input samples,  $\mathbf{x}^l$  or  $\mathbf{q}^{rl}$ 
Calculate the distances between the samples  $\{d_{ij}\}$ .
Determine the mean,  $\bar{d}$ 
loop
    Define the settings of the cophenetic distances
    Calculate the cophenetic distances of the samples  $\{l_{ij}\}$ .
    Determine the mean,  $\bar{l}$ 
    Determine the dendrogram to illustrate the samples.
    Calculate the cophenetic correlation coefficient  $\hat{c}_{coph}$ 
    if  $\hat{c}_{coph}$  below the threshold  $\hat{c}_{coph} < t\hat{c}_{coph}$  then
        The dendrogram can present the input data properly
        Break
    end if
end loop
if the number of groups  $n_{g1}$  is defined then
    Classification according to  $n_{g1}$ 
    Assign the samples into  $n_{g1}$  groups
     $n_g = n_{g1}$ 
else if threshold  $thr_{ca}$  is defined then
    Natural classification
    Calculate  $r_{ca}^i$  for each layer
    Classification according to  $thr_{ca}$ 
    Assign the samples into  $n_{g2}$  groups
     $n_g = n_{g2}$ 
end if
Output  $n_g$  and the classified location samples

```

After the classification, Algorithm 5.2, n_g groups are created, and each sample is assigned into some group unambiguously. This means n_g heterogeneous regions should be modeled in the contact force model. Then the corresponding location samples can be extended as,

$$\mathbf{x}_i^l = [x \quad y \quad z \quad E \quad \nu \quad i_g]^T, \quad (5.11)$$

where i_g is the group index of each location sample. Since the material parameters, especially the Young's modulus, have the dominant influence on the principal components and the cluster analysis, this classification is mainly according to the material attributes. This way, the connection between the sample positions on the object and

the material feature can be established, and the number of the partitions after the division of the cluster tree is the estimation of the inhomogeneous regions in the material. Then the values of the material parameters in each region are defined as the averages of the location samples in this region. Assuming m location samples are located in a region i , it yields,

$$\chi_i^r = \frac{1}{m} \sum_{j=1}^m \chi_j^l, \quad (5.12)$$

where χ_i^r and χ_j^l are the according material parameters of the region and of the j -th location sample, respectively.

Furthermore, it should be noted that some other effects can also lead to significant difference among location samples, besides the inhomogeneity, for instance different boundary constraints at the bottom or the geometry of the bottom, which are problematic for laser scanning, image processing and other normal measurement technologies. However, if there is not enough information, and the concrete situations of the object are not clear, these effects can not be model accurately. Because the aim is to provide model based haptic simulations, these effects can be modeled as inhomogeneity of material as well, and it can be assumed that the inhomogeneous models may be suitable for the simulation of these objects.

5.2.4. Material Region Estimation

If the material is inhomogeneous, different material models and parameter values should be assigned in the heterogeneous regions. In addition, it is also necessary to determine the regions' boundaries. A classification algorithm is developed for this purpose, using the discriminant analysis (DA).

The discriminant analysis is a statistic classification method, which is based on the Bayesian decision. It is widely applied in varying areas [68, 84]. For a known division of sample space $\{k_i : i = 1, 2, \dots\}$, the DA establishes a classification rule and applies it to allocate an arbitrary given sample into a division with the largest likelihood. Hence this classification is different from the classification using the cluster analysis, because it is supervised with learning data. Varying approaches are investigated based on the DA, and they are called according to the discriminant functions, such as the linear discriminant analysis (LDA) and the quadratic discriminant analysis (QDA). In these methods, a multivariate Gaussian distributed conditional probability is assumed.

In this section, the quadratic discriminant analysis (QDA) is implemented, which is better than the linear discriminant analysis (LDA) for data fitting. A multi-class, N , classification problem can be formulated as the posterior probability $p(k|\mathbf{x})$. The mathematic meaning of it is the likelihood of the class k , for the given particular sample \mathbf{x} , then this classification problem can be solved by

$$c(\tilde{\mathbf{z}}) = \arg \max_k p(k|\tilde{\mathbf{z}}). \quad (5.13)$$

If the prior probability of the class k is defined as $p(k)$, and $f(\mathbf{x}|k)$ is the conditional

probability density of sample \mathbf{x} , given the class k . With the Bayes theorem, the posterior probability can be determined as,

$$p(k|\mathbf{x}) = \frac{p(\mathbf{x}|k) \cdot p(k)}{\sum_{n=1}^N p(\mathbf{x}|i) \cdot p(i)}. \quad (5.14)$$

Substituting (5.14) into (5.13), and considering the maximum a posteriori (MAP) hypotheses, the solution of the classification problem can be rewritten,

$$c(\tilde{\mathbf{z}}) = \arg \max_k p(\mathbf{x}|k) \cdot p(k). \quad (5.15)$$

As mentioned above, the conditional density is presumed as

$$f(\mathbf{x}|k) = \frac{1}{(2\pi)^{N/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}, \quad (5.16)$$

then substituting (5.16) into (5.15) leads to the quadratic discriminant function (5.17) of the class k , given \mathbf{x} ,

$$g_k(\mathbf{x}) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \log p(k). \quad (5.17)$$

Eventually, the classification rule can be concluded as follows, with the discriminant function,

$$c(\tilde{\mathbf{z}}) = \arg \max_k g_k(\tilde{\mathbf{z}}). \quad (5.18)$$

Applying the quadratic discriminant analysis for the boundary estimation of heterogeneous regions, the basic thought is to check a large number of arbitrary points in the domain, and to determine to which region belong these points respectively. This is also a classification problem, however unlike the location samples, the material feature of an arbitrary point is naturally unknown, therefore the classification can not be according to the material parameters in the accurate way but to the position of the points, and here it is assumed that the regions are continuous. Therefore, it is just an estimation of the region boundaries. The selection of the arbitrary points is regarded as a virtual scanning process, and the points are called scan points. The parameters of them are their positions.

$$\mathbf{x}_i^{vc} = \begin{bmatrix} x & y & z \end{bmatrix}^T \quad (5.19)$$

In this implementation, the scan points are created uniformly in the domain.

Since the QDA is a supervised learning method, the classified location samples, derived by the hierarchical cluster analysis, are the training data of the discriminant functions. However, different from the former classification according to the material feature, in the training, the important relation is the membership of the location samples in groups and the corresponding positions. After the training, the virtual scan points are classified according to their positions. Then the points at the boundary of different regions can be used to reconstruct the boundary surface. By this

means, the boundaries between varying material regions can be primarily estimated, Algorithm 5.3. These identification methods are verified by the experiments in Chapter 6.

Algorithm 5.3: Region boundary estimation

```

Import the classified location samples,  $\mathbf{x}^l$ 
Training of the QDA using the location samples
- Calculated the probabilities
- Determine the discriminant function  $g_k(\mathbf{x})$ 
Define the domain of the object
Create  $n$  scan points in the domain
for  $i = 0$  to  $n$  do
  Classification of  $\mathbf{x}_i^{vc}$ , using the QDA
  Sort  $\mathbf{x}_i^{vc}$ 
end for
Determine the boundary points
Create the regions' boundary
  
```

Sequentially, a FE model can be recreated using all available classification information, including the estimation of the material parameters and the regions' boundaries. This model is a primary inhomogeneous model, although the model parameters are not accurate enough, it has the basic behavior of the inhomogeneous test object.

5.3. Model Parameter Optimization

Generally, the model learning and classification can just determine the model parameters in some range, but not accurately. For high level haptic simulations, the model parameters have to be adjusted using reference data, i.e. the contact reaction force \mathbf{f}_r , so that the simulations \mathbf{f}_s can be a good match with the reference. The deviation of simulations can be defined as the residual sum of squares (rss),

$$_{rss}r(\chi) = \|\mathbf{f}_r - \mathbf{f}_s(\chi)\|_2. \quad (5.20)$$

In this formulation, the numbers of the entries (n) of both force vectors must be the same, i.e. $\mathbf{f} \in \mathbb{R}^{n \times 1}$

It is certainly understandable, the more available reference data, the more accurately the model parameters can be optimized, because the random errors can be averaged by large quantity of samples. However the computational expense increases with the data size. Thus a trade-off is sometimes necessary. There is another condition: in order to optimize the model parameters, the number of reference data must be greater than or at least equal to the number of the to be optimized parameters,

$$n_{ref} \geq n_{parameters}. \quad (5.21)$$

The border case is enough for the optimization in principle, however the effects of random errors can disturb the optimization completely. Thus, in this implementation,

the following condition is satisfied, to reduce the stochastic effects,

$$n_{ref} \geq c_{opt} \cdot n_{parameters} \quad \text{with} \quad c_{opt} > 1, \quad (5.22)$$

where c_{opt} is a scalar coefficient. In the experiments in Chapter 6, it is normally defined $c_{opt} = 10$.

5.3.1. Optimization Problem

Generally, the material parameters are determined experimentally by using special equipment, such as one dimensional tension and torsion tests. But in many cases these tests are impossibility, hence the parameters should be identified and optimized using simple reference data, which can be already obtained even in common experimental conditions. For haptics, the most general contact data of deformable bodies are the measurements of the contact force and the boundary displacements at some positions, this information is available in almost all tests and operations.

With the comparison of the force measurements \mathbf{f}_r and the simulations \mathbf{f}_s , derived by the force models, the parameter optimization should be formulated as the minimization of the error function (5.20), subject to constraints. This problem can be stated as a least squares problem [88, 13] by minimization of the objective function f^{obj} on the model parameters,

$$\begin{aligned} \min_{\boldsymbol{\chi}} f^{obj}(\boldsymbol{\chi}) \\ f^{obj}(\boldsymbol{\chi}) = \frac{1}{2} \sum_{i=1}^n (r_i^{obj})^2 = \frac{1}{2} \sum_{i=1}^n (f_{ri} - f_{si}(\boldsymbol{\chi}))^2, \end{aligned} \quad (5.23)$$

where the vector $\boldsymbol{\chi}$ consists of the material parameters implemented in the models, r_i^{obj} is an entry of the residual vector \mathbf{r}^{obj} .

Since the model parameters have the corresponding physical meanings, they are constrained in some ranges. Hence this parameter optimization is a constrained optimization problem, and the constraints are presented as a number of equations and inequalities,

$$\begin{cases} f_i^c(\boldsymbol{\chi}) = 0 \\ f_j^c(\boldsymbol{\chi}) \geq 0, \end{cases} \quad (5.24)$$

where $\{f_i^c : i = 1 \dots n\}$ is the collection of functions which describe the constraints in the optimization. Thus, the constraints are regarded as equality constraints and inequality constraints separately. The concrete definitions of them are presented in Section 5.4 for each kind of model.

5.3.2. Gauss-Newton Method

This nonlinear least squares problem of the material parameter vector $\boldsymbol{\chi}$ can be solved using the Gauss-Newton method, which is an iterative algorithm. The parameters will be correctly optimized, if convergence criterion is satisfied.

In each iteration, the objective function (5.23) is linearized locally. Thus, the algorithm seeks the steepest decent direction δ_k as a line search method, by means of solving the following equation,

$$\mathbf{J}_k^T \mathbf{J}_k \delta_k = -\mathbf{J}_k^T (\mathbf{f}_r - \mathbf{f}_s(\chi_k)),$$

where the Jacobian matrix \mathbf{J}_k is the first-order partial derivative of \mathbf{r}^{obj} . The entries in \mathbf{J} are determined at each point of time,

$$j_{ij} = \frac{\partial r_i^{obj}}{\partial \chi_j}. \quad (5.25)$$

They can be calculated numerically or analytically. The numerical solutions can be obtained, whenever the corresponding data are ready, whereas the accurate analytical solutions are available, only if f^{obj} is also analytical. However, this is not always the case. For the finite element contact force models, the Jacobian matrices should be determined using the numerical method,

$$j_{ij} = \frac{\partial r_i^{obj}}{\partial \chi_j} = \frac{r_i^{obj}(\chi_j + \Delta\chi_j) - r_i^{obj}(\chi_j)}{\Delta\chi_j}. \quad (5.26)$$

In order to simplify the calculation, the QR-decomposition of the Jacobian matrix is implemented as follows,

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{Q}_{1k} & \mathbf{Q}_{2k} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_{1k} \cdot \mathbf{R}_k. \quad (5.27)$$

The algorithm of it has been detailed in Appendix A.1. By this means, the search direction can be determined,

$$\delta_k = -\mathbf{R}_k^{-1} \cdot \mathbf{Q}_{1k}^T (\mathbf{f}_r - \mathbf{f}_s(\chi_k)). \quad (5.28)$$

Then the parameters are modified along this direction for the next iteration, with step length λ_k ,

$$\chi_{k+1} = \chi_k + \lambda_k \cdot \delta_k. \quad (5.29)$$

The effects of the constraints are different for the equality constraints and the inequality constraints. For the first kind, the descent direction must be along the constraint equations. Hence the derivative of the constraints, ∇f^c , has dominant effects in the optimization. The Lagrangian function is defined,

$$f^{\mathcal{L}}(\chi, \lambda^{\mathcal{L}}) = f^{obj}(\chi) - \sum_{i=1}^m \lambda_i^{\mathcal{L}} f_i^c(\chi), \quad (5.30)$$

with the Lagrange multiplier $\lambda_i^{\mathcal{L}}$. The derivative of it is as follows,

$$\nabla_{\chi} f^{\mathcal{L}}(\chi, \lambda^{\mathcal{L}}) = \nabla f^{obj}(\chi) - \sum_{i=1}^m \lambda_i^{\mathcal{L}} \nabla f_i^c(\chi). \quad (5.31)$$

Then the optimization is equivalent to seek a χ to satisfy the following condition,

$$\nabla_{\chi} f^{\mathcal{L}}(\chi, \lambda^{\mathcal{L}}) = 0. \quad (5.32)$$

For the second one, the inequality constraints affect the parameter optimization only if the current step or the next step is in contact with the boundary of the constraints. Otherwise, the optimizations can be considered as unconstrained problems. When some constraints are active, the Lagrangian function can also be defined. Hence, this can also be specified as the definition of factor $\lambda_i^{\mathcal{L}}$ for each constraint,

$$\begin{cases} \lambda_i^{\mathcal{L}} = 0 & i\text{-th constraint inactive} \\ \lambda_i^{\mathcal{L}} > 0 & i\text{-th constraint active.} \end{cases} \quad (5.33)$$

It can be formulated as a complementarity condition at some solution point,

$$\lambda_i^{\mathcal{L}} f_i^c(\chi) = 0.$$

The factors δ and λ in (5.28-29) are the original step length and the corrective coefficient, which can ensure the convergence according to the relevant conditions. It can be determined by using the backtracking method, Algorithm 5.4.

Algorithm 5.4: Backtracking

```

Set the initial step length
Define the factor  $c$ 
while the step length condition is not fulfilled do
    Modify the step length factor
    Calculate the objective function
     $\lambda = c \cdot \lambda$ 
    if minimal step length:  $\lambda < \lambda_{min}$  then
        Break
    end if
end while

```

The simplest step length condition is

$$r_{ss} r_{k+1} < r_{ss} r_k. \quad (5.34)$$

However, this criterion may be ineffective and can lead to errors for some objective functions. The Wolfe condition

$$f^{obj}(\chi_{k+1}) \leq f^{obj}(\chi_k) + c_k \lambda_k \nabla f_k^{obj,T} \delta_k \quad (5.35)$$

is better than the first one. With it, the efficient convergence can be ensured. For

the least squares problem, both criteria can be applied, because its objective function (5.23) has always non-negative values.

In the practical implementation in this thesis, constraints are taken into account by calculating the projection of search direction according to the constraints. Eventually, the complete algorithm of the Gauss-Newton method is illustrated in Algorithm 5.5. The optimization conditions include the maximal number of iterations, λ_{min} , and the convergence condition. To define a general convergence criterion, which is independent on the size of the force vectors, the deviation defined in (5.20) is replaced by root mean square (rms),

$$rmsr(\chi) = \frac{1}{\sqrt{n}} \|\mathbf{f}_r - \mathbf{f}_s(\chi)\|_2, \quad (5.36)$$

where n is the size of the force vectors. Then the convergence criterion is

$$rmsr_{k+1} < rmsr_{max}. \quad (5.37)$$

Algorithm 5.5: Gauss-Newton Method

```

Initialization
while iterative number  $i < i_{max}$  do
    Calculate the derivative of the objective function,  $\mathbf{J}$ 
    Determine the descent direction of the next step  $\delta_k$ 
    Calculate the initial step length
    Checking the constraints
    if Active then
        Calculate the projection of  $\delta_k$ 
    end if
    Determine the step length using the backtracking
    if Convergence criterion then
        Break
    end if
     $i \leftarrow i + 1$ 
     $rmsr_k \leftarrow rmsr_{k+1}$ 
end while

```

5.3.3. Global Optimization Algorithm

With respect to the model parameters, the value space of the objective function (5.23) may contain numerous local minima. In order to determine the global minimum of the objective function, the global optimizations are necessary to overleap the local minima. Global optimizations are always problematic, and different techniques and methods are investigated [57, 58]. Furthermore, these problems are always computationally expensive.

As a solution, the implicit filtering can be selected to seek the minimum [47], and the open source program IFFCO [25] (implicit filtering for constrained optimization),

which is implemented using Fortran and Matlab, has been used by some applications [87]. For the parameter optimization in this chapter, the similar thought is applied, and the Gauss-Newton method is enhanced as the global Gauss-Newton method, with varying settings. Since the force models are often employed in real time systems, and the computing expense should be reduced as low as possible, the algorithm is implemented using C++, and the software library *lapack* [30] has been integrated to accelerate the calculation of large scale matrices.

The basic idea is to execute the Gauss-Newton method repeatedly with different starting values and coefficients, for instance factors λ and c . This global Gauss-Newton algorithm is illustrated in Algorithm 5.6.

Algorithm 5.6: Global Optimization

```

Initialization:
- Determine the possible value space of the parameters
- Determine a set of values in this range as the candidates of the start values,  $n$  entries.
while  $j < n$  do
    Initial setting of the optimization
    Select the start value
    Run the Gauss-Newton method, Algorithm 5.5
    if Convergence then
        Storage the result
        Storage the associated settings and start values
        next
    else
        next
    end if
     $j \leftarrow j + 1$ 
end while
Select the minimum of  $r_{ss}T$  or  $r_{ms}T$ 
Check the global convergence condition
if Convergence condition (global) then
    Assign the corresponding parameter values as the global minimum setting
end if

```

5.4. Implementation of Model Identification

Applying the learning and optimization methods for the force models, the procedures may be different in varying applications. For the above introduced force models, they are described as follows. In this chapter, the model identification methods are implemented as modules, and can be combined in different ways. Furthermore, an adaptive algorithm is developed for unknown materials as general situations.

5.4.1. Linear Elastic Model Identification

As the simplest model in this thesis, the linear isotropic elastic model contains merely two parameters, i.e. E and ν , and the constraints are

$$\begin{cases} E > 0 \\ 0 < \nu < 0.5. \end{cases} \quad (5.38)$$

The model parameter vectors of these linear elastic models are defined as,

$$\boldsymbol{\chi}^{le} = [E \quad \nu], \quad (5.39)$$

the model identification is according to (5.28) and (5.29).

In some applications, it is only necessary that the Young's modulus is identified, whereas the Poisson's ratio is assigned as a constant. Then, the identified E can partially compensate for the influence of the deviation of the Poisson's ratio. This setting has its advantage in computational expense, i.e. only a few steps are required for the optimization. Hence, it is especially suitable for the real time model identification, such as in telepresence, and the accuracy of the simulation is still sufficient with small deformations [124]. The algorithm is illustrated in Algorithm 5.7.

Algorithm 5.7: Identification of linear elastic models

Initialization:

- Initial settings

- $E = E_0$

if optimization: ν **then**

- $\nu = \nu_0$

- Define the constraints

end if

Run the Gauss-Newton method, Algorithm 5.5 or the global optimization algorithm, Algorithm 5.6

if Convergence **then**

Assign the identified parameter values to the model

end if

Since these models are linear, the optimizations of them are relatively economical in computing time. Normally, if a suitable convergence criterion is defined, the algorithm is always convergent in a few iterations, especially if the identification of the Poisson's rate is not involved. The global optimization algorithm is not necessary in general.

5.4.2. Viscoelastic Model Identification

The viscoelastic models are nonlinear, thus in the optimization, more effects have to be taken into account in comparison with the optimization of linear models. Therefore, the global optimization algorithm is more robust and should be selected for these models. Moreover, the settings and constraints of the parameters are more complex as well.

During the optimization, the parameter vector (3.24) or in the special case (4.15) are substituted into Algorithm 5.6. And the complete optimization algorithm of the viscoelastic models is shown as follows, Algorithm 5.8.

Algorithm 5.8: Optimization of linear viscoelastic models

Initialization:
 - Initial settings,
 - $E_S = E_0$ and $\nu = \nu_0$
 - Define the number of the terms in the Prony series (n)
 - Assign the relaxation parameters:
 - $E_1 \dots E_n$ and $\tau_1 \dots \tau_n$
 - Define the constraints
 Run the global optimization algorithm, Algorithm 5.6
 Check the results of the global optimization
if convergence **then**
 Assign the identified parameter values to the model
end if

The constraints of the model parameters are in the following form,

$$\begin{cases} E > 0 \\ 0 < \nu < 0.5 \\ E_i > 0 \\ T_{i-} < \tau_i < T_{i+}. \end{cases} \quad (5.40)$$

Normally, the material parameters are constrained according to their physical meaning, however the relaxation time coefficients are defined subjectively in different time intervals. This setting is according not only to the physical meaning, but also to the expected performance of the model. As mentioned, only finite entries in the Prony series can be implemented in practice, generally less than 5 terms. On the one hand, the computing time increases obviously with this number, and on the other hand, some terms have little practical importance, for instance the relaxation amplitudes may be relatively small and the relaxation times may be extremely long. In order to take the most important and valuable information into consideration with few entries, the time coefficients are defined corresponding to some periods, $\{[T_{i-} \ T_{i+}] : i = 1, \dots, n\}$. It should also be avoid that different relaxation times with respect to overlapping time intervals, as redundancy. Another advantage of this constraint is that each time factor has its own scope, and the performance of the force models can be better adjusted. The experiments and simulations have verified the advantages of this setting. Furthermore, the effects of reference data are also important, that is that periods with few samples are meaningless.

5.4.3. Inhomogeneous Model Identification

The major differences between the optimization of homogeneous and inhomogeneous models lie in the parameters to be optimized. The model parameters of the inho-

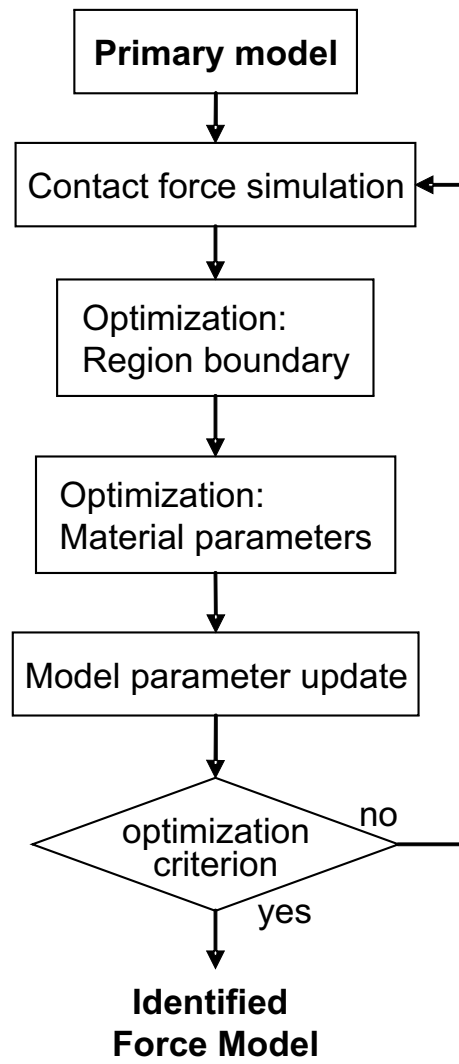


Figure 5.2.: Model parameter optimization of inhomogeneous force models. The optimization begins with the primary models, this means the heterogeneous regions are already identified and the boundaries are already estimated. In each iteration, the optimization of the material materials and the boundaries are separated.

homogeneous models consist of not only the material parameters in the corresponding constitutive equations, but also the parameters, which describe the heterogeneous regions, i.e. the regions' boundary.

As shown in Fig. 5.1, the identification for the inhomogeneous models begins with the unsupervised learning, hence the model parameters are defined as follows in the primary models,

$$\chi = [\chi_m \quad \chi_b], \quad (5.41)$$

where the vectors χ_m and χ_b are for the material parameters and regions' boundaries, in the form of boundary points, respectively.

Substituting χ into the optimization algorithms, the parameters can be optimized

iteratively. It should be noted that the effects of the model parameters on the components of contact force in different directions, f_x , f_y and f_z , are different. Hence the following adjustments may improve the performance of the parameter optimizations. The optimizations of the material parameters and the region boundary are separated in each iteration, and different measurements are used separately for the different parameters as well, i.e. the force components, which are vertical to the contact surface, should have more weight, about $3 \sim 10$ times, in the material parameter optimization. Whereas the parallel components involve more information of the distribution of the inhomogeneous regions. The simulations have confirmed these adjustments.

If it is known that there are m heterogeneous regions and n boundary points, which have been determined in the primary model, and all the regions are linear elastic, then the model parameter vector is defined as follows,

$$\chi = [E_1 \quad \nu_1 \quad \cdots \quad E_m \quad \nu_m \quad x_1 \quad y_1 \quad z_1 \quad \cdots \quad x_n \quad y_n \quad z_n]. \quad (5.42)$$

This parameter optimization procedure of inhomogeneous models is illustrated in Fig. 5.2.

According to the model parameters (5.41), there are two kinds of parameter constraints. They are for the material parameters and region boundary, respectively. The first kind is dependent on the material characteristics of each region, and is similar as those of the linear models or the viscoelastic models. In the second one, the boundaries are constrained in some range, $[\Delta x \quad \Delta y \quad \Delta z]$, surrounding the estimation in the primary models $[x^p \quad y^p \quad z^p]$. Furthermore, it is certainly clear that the boundaries must lie within the object. Then these constraints are formulated as follows,

$$\begin{cases} (x_i^p - \Delta x < x_i < x_i^p + \Delta x) \cap (x \in V_d) \\ (y_i^p - \Delta y < y_i < y_i^p + \Delta y) \cap (y \in V_d) \\ (z_i^p - \Delta z < z_i < z_i^p + \Delta z) \cap (z \in V_d), \end{cases} \quad (5.43)$$

where V_d is the domain of the object. For the simplified inhomogeneous models in Section 3.5, it is merely necessary to optimize the coordinates in one direction.

5.4.4. Analytical Contact Force Model

For the analytical contact force model (ACFM) parameters, the formulation of the least squares problem and the according optimization algorithms are also available. This means they can be optimized analogously as the FEM parameters. However for analytical models, there is the important advantage, that the derivative matrix can be calculated analytically. By this means, one of the most computationally expensive parts of the model identification can be reduced significantly.

Considering the objective function of the least squares problem, the derivatives of the objective function \mathbf{f}^{obj} are equivalent to those of the force simulation \mathbf{f}_s , because

of the following relation:

$$\begin{aligned}\nabla \mathbf{r}^{obj} &= -\nabla \mathbf{f}_s(\boldsymbol{\chi}) \\ \Rightarrow \nabla r_i^{obj} &= -\nabla f_{si}(\boldsymbol{\chi})\end{aligned}\quad (5.44)$$

Hence, the matrix \mathbf{J} can be determined analytically, by calculating the partial derivatives of \mathbf{f}^{obj} with respect to the parameters $\boldsymbol{\chi}$.

$$\mathbf{J}_k = \frac{\partial \mathbf{r}_k^{obj}}{\partial \boldsymbol{\chi}} = -\frac{\partial \mathbf{f}_{s,k}(\boldsymbol{\chi})}{\partial \boldsymbol{\chi}} = \left\{ -\frac{\partial f_{si,k}}{\partial \chi_j} : i = x, y, z; j = 1, 2, \dots \right\} \quad (5.45)$$

It can be concluded, that the optimization of ACFM is different from that of the FE model by the calculation of the Jacobian matrix.

As an example, the introduced simple model with one Maxwell element in Section 4.4.2 is further investigated in this section. According to (4.38)-(4.43), the Jacobian matrix of the objective function at time step $k + 1$ is written in the following form,

$$\mathbf{J}_k = \begin{bmatrix} \frac{\partial f_{sx,k}}{\partial k_{xs}} & \frac{\partial f_{sx,k}}{\partial k_{x1}} & \frac{\partial f_{sx,k}}{\partial d_{x1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial f_{sy,k}}{\partial k_{ys}} & \frac{\partial f_{sy,k}}{\partial k_{y1}} & \frac{\partial f_{sy,k}}{\partial d_{y1}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial f_{sz,k}}{\partial k_{zs}} & \frac{\partial f_{sz,k}}{\partial k_{z1}} & \frac{\partial f_{sz,k}}{\partial d_{z1}} \end{bmatrix}. \quad (5.46)$$

As mentioned in Section 4.3, the different directions are decoupled in the analytical models. Thus, the derivative of the components in a single direction is

$$\frac{\partial f_{si,k}}{\partial \chi_i} = [j_{i1} \quad j_{i2} \quad j_{i3}] \cdot \Delta t, \quad (5.47)$$

with

$$\begin{aligned}j_{i1} &= \frac{k_{i1}}{d_{i1}} \Delta s_i + \Delta \dot{s}_i \\ j_{i2} &= -\frac{f_{si,k-1}}{d_{i1}} + \frac{k_{is}}{d_{i1}} \Delta s_i + \Delta \dot{s}_i \\ j_{i3} &= \frac{k_{i1}}{d_{i1}^2} f_{si,k-1} - \frac{k_{is} k_{i1}}{d_{i1}^2} \Delta s_i\end{aligned}\quad (5.48)$$

where i is the direction index, and the parameters of the same direction is defined as

$$\boldsymbol{\chi}_i = [k_{is} \quad k_{i1} \quad d_{i1}]^T. \quad (5.49)$$

The parameters in the analytical models are also constrained as,

$$\begin{cases} k_{is} > 0 \\ k_{ij} > 0 \\ D_{ij-} < d_{ij} < D_{ij+}, \end{cases} \quad (5.50)$$

where i is the index of the coordinate directions, and j is the index of the Maxwell

elements in ACFM. For coefficients k , the constraints are according to their physical meaning. Whereas the constraints of the damping d are analogous to those of the relaxation times of the viscoelastic models. The principle is also to limit the effects of each Maxwell element in its range, and to better adjust the performance of the ACFM.

Since the geometric information of the object is not involved in the analytical models, as mentioned above, the parameters of the ACFM are defined as a function of positions to improve the accuracy. The sub-models are created at all sample positions, where the reference data are available. Thus, the parameter optimization of them also depends on the position. After that, a set of parameters in connection with the associated positions is calculated, (4.18). When there is contact at some position, a temporary sub-model is going to be created at the current position. Defining $n = 3$, then the three nearest known samples are found, if there are enough entries. Subsequently, the parameters of the temporary component are determined as the weighted mean of the parameters of the selected samples, using (4.19),

$$\mathbf{x}_{temp} = \sum_{i=1}^3 \frac{\bar{d}_i}{\bar{d}_1 + \bar{d}_2 + \bar{d}_3} \mathbf{x}_i. \quad (5.51)$$

Then the contact force at this position can be simulated using the temporary sub-model.

5.4.5. Extended Kalman Filter

The parameters in the extended Kalman filter algorithm have strong effects on the performance of the data fusion. Hence they should also be determined appropriately. The important parameters in EKF are the system and measurement noise densities in the matrices \mathbf{Q}^n and \mathbf{R}^n , which determine the influence of different data in the fusion process. They can also be optimized using an optimization algorithm, just as the FE model parameters. However in the case of haptic rendering, all the information is from simulations, hence there is not any stochastic noises in the system. Thus, these noise densities can be determined merely to adjust the simulation results. Since the information from the FE model is much more authentic than that derived by the ACFM, the FE simulations are regarded as the reference in the data fusion. This means the high frequency data from the ACFM are adjusted by the low frequency accurate data from the FE model. Therefore the matrices \mathbf{Q}^n and \mathbf{R}^n can be determined definitively based on experience.

However, it has to be mentioned that during the time of the force evaluation by the FE model, the boundary displacement and the displacement rate may have already changed. For these cases, the simulation derived by the analytical model should be considered to have more weight than normal, especially for contacts with high boundary displacement rate. Thus the noise densities are defined as linear functions of the boundary displacement rate, for the system noise

$$\sigma_{fi} = c_i \cdot \Delta \dot{s}_i \cdot \tilde{\sigma}_{fi} + \sigma_{fi}^0, \quad (5.52)$$

and for the measurement noise

$$\sigma_{fim} = -c_{im} \cdot \Delta \dot{s}_i \cdot \tilde{\sigma}_{fim} + \sigma_{fim}^0, \quad (5.53)$$

where $\tilde{\sigma}_{fi}$ and σ_{fi}^0 are the boundary displacement rate dependent and independent parts of the system noise, respectively. Analogously, $\tilde{\sigma}_{fim}$ and σ_{fim}^0 are the according terms of the measurement noise. The nonnegative factors c_i and c_{im} are the gains for both noises, which describe the effects of the displacement rate. For a given application, the maximum boundary displacement rate is normally known. The expected performances of the data fusion at static conditions and with the maximal boundary displacement rate are also defined. Then the noise densities at these states,

$$\begin{cases} \sigma_{fi,min} = \sigma_{fi}^0 \\ \sigma_{fim,min} = \sigma_{fim}^0 \end{cases} \quad \Delta \dot{s}_i = 0$$

$$\begin{cases} \sigma_{fi,max} = \tilde{\sigma}_{fi} + \sigma_{fi}^0 \\ \sigma_{fim,max} = \tilde{\sigma}_{fim} + \sigma_{fim}^0 \end{cases} \quad \Delta \dot{s}_i = \Delta \dot{s}_{max}, \quad (5.54)$$

can be determined based on simulations and experiments. With the following settings and definitions,

$$\begin{aligned} c_i &= c_{im} \\ \tilde{\sigma}_{fi} &= \tilde{\sigma}_{fim}, \end{aligned} \quad (5.55)$$

the coefficients in (5.52) and (5.53) can be determined. Then applying the linear relations with an arbitrary displacement rate in the range,

$$\Delta \dot{s}_i \in [0 \quad \Delta \dot{s}_{max}], \quad (5.56)$$

the corresponding noise densities can be calculated directly.

5.5. Adaptive Model Identification

The basic model identification methods and the implementation of different models are presented above. The content of this section is to derive a way to model completely unknown materials. The introduced modeling approaches are regarded as modules in a library. An adaptive model identification algorithm is developed, in order to select the most suitable items to model an object based on reference data.

At the beginning, the geometry should be measured or imported, and the mesh is generated. Then the reference data are sorted according to the contact positions. Furthermore, the data at the same contact positions with the same or similar boundary displacements should be noted specially, if they are available.

Then the modeling procedure begins with criteria. In this chapter, the necessary

criteria are as follows:

$$\begin{cases} {}_{th}r_{ca} & \text{for the identification of heterogeneous regions} \\ \Delta \mathbf{s}_0 & \text{for the same or similar boundary displacement} \\ {}_{th}r_{ft} & \text{for the change rate of contact force in respect to time scale.} \end{cases} \quad (5.57)$$

The last two are applied for the identification of viscoelastic effects. In comparing with the ${}_{th}r_{ca}$ and the judgment of inhomogeneous materials, which are detailed as model unsupervised learning in the former sections, the identification of the viscoelasticity is much more simple. The criterion

$$\Delta \mathbf{s}_\epsilon = [\Delta \mathbf{x}_\epsilon \quad \Delta \mathbf{y}_\epsilon \quad \Delta \mathbf{z}_\epsilon] \quad (5.58)$$

defines a very small range. The boundary displacement in this range

$$\Delta \mathbf{s} \in [\Delta \bar{\mathbf{s}} - \Delta \mathbf{s}_\epsilon \quad \Delta \bar{\mathbf{s}} + \Delta \mathbf{s}_\epsilon] \quad \text{with} \quad \Delta \mathbf{s}_\epsilon \rightarrow \mathbf{0} \quad (5.59)$$

can be regarded as similar, in which $\Delta \bar{\mathbf{s}}$ is some average boundary displacement. The ratio r_{ft} measures the rate of change of contact forces at different points in time during one relaxation process, i.e.

$$r_{ft} = \frac{\Delta f}{\Delta t}. \quad (5.60)$$

If it is beyond a threshold, the viscoelasticity should be modeled in the force models,

$$r_{ft} > {}_{th}r_{ft}. \quad (5.61)$$

However, it should be noted if the number of these samples is not sufficient, (5.20), the viscoelastic model still cannot be established.

The identification of the FE models is illustrated in Fig. 5.3. If real time simulations are expected, and the analytical models are needed, they can be created and identified according to the FE models directly. No additional judgments and criteria are necessary for the ACFM.

In addition, some measures may improve the contact force modeling further. Firstly, since the parameter optimization depends on the reference data and the start values, the material parameters of different models can be stored in a model library, according to the model characteristics, such as the material, the element shape and the average element size. Then the model identification with the selected initial values can be accelerated significantly. Secondly, the quality of the reference data has important effects on the model identification and the accuracy of the haptic simulations. Since the samples may be drawn during the manually controlled operations, sometimes they are not accurate enough, and in some experiments, the samples may be not sufficient for the optimization of model parameters. Especially, during the modeling of the region boundary, the identification depends closely on the sample positions. For instance, the lack of samples in some areas may affect the boundary identification in these areas obviously. For this reason, some sampling

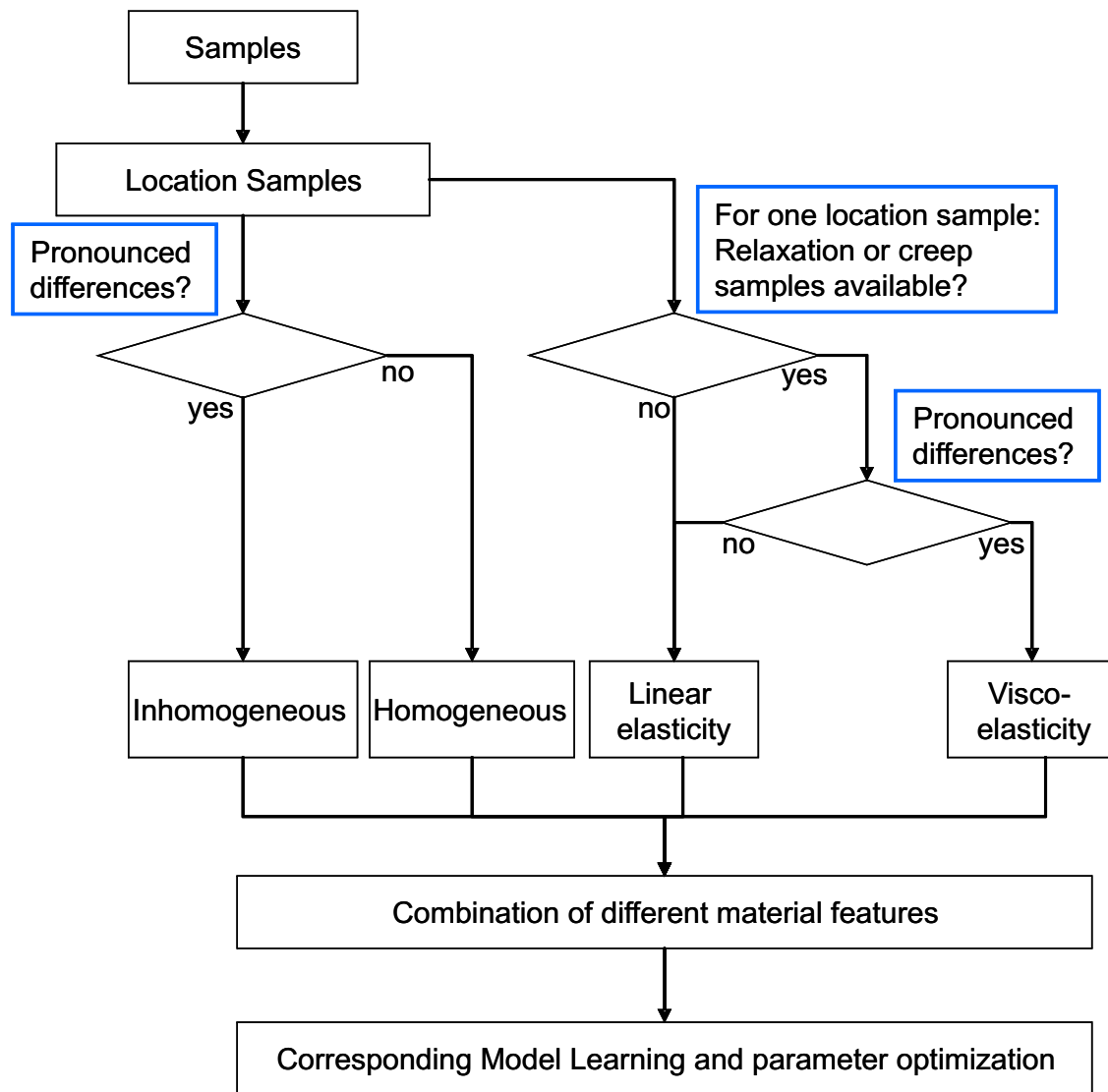


Figure 5.3.: Adaptive model identification. The detailed model learning and optimization methods are developed in the former sections.

methods and data mining algorithms could provide better performance of the force models.

5.6. Model Verification and Update

After the model identification, the contact force models can provide haptic simulations in diverse applications, however the simulations may sometimes not satisfy the accuracy requirements. There are two major reasons for these cases. One is that the operation environment and contact situations have been modified during operations. In many telerobotic applications, such as satellite repair on orbit, the position and orientation of objects may have changed after the model identification. The other is,

since the model identification is based on measurements, as reference data, it may contain varying deviations. To reduce or eliminate the effects of these disturbances, large quantities of data are used, (5.22). However, the effects of some strong stochastic errors can still not be neglected completely. In order to recognize the modifications and errors, the model should be continuously monitored and verified, if possible. The necessary information is the reference data during the operation, for instance delayed measurements. They should be compared with simulations, and if significant deviations are detected and the corresponding reference data are available, the model parameters should be identified again using the introduced modeling methods, completely or partially, depending on the conditions. Then the contact force model can be updated, Algorithm 5.9.

Algorithm 5.9: Model verification and model update

```

Starting the simulations
loop
  if reference data available then
    Simulation with model verification
    Read the reference data
    Sort of the simulation and the reference data
    Comparing  $\mathbf{f}_s$  and  $\mathbf{f}_r$ 
    if significant deviation,  $rmsT > rmsT_{th}$  then
      Determine for geometric update, material update or both
      collection reference data
      Model Identification
      Rebuild the model or assign parameter with new values.
    end if
  else
    Simulation without model verification
  end if
  if stop simulation then
    Break
  end if
end loop
Quit
  
```

According to the model parameters, the modifications can be geometric or material, including the geometry, the position and the material feature of objects. For instance, if the contact forces vanish at some position where they should exist, there is a high likelihood that the position of the object has changed. When the new geometric measurements are obtained, the geometric model can be rebuilt.

6. Experiments and Simulations

In this chapter, the force modeling methods and haptic simulations are verified and tested. Two major parts are involved. Firstly, the experimental system is detailed. Secondly, different experiments are carried out with diverse objects. They can be divided into three kinds. Virtual reality tests have shown the basic functions of the modeling methods and model based haptic simulations. Then individual experiments focus on the modeling of materials using different fundamental theoretical models, including linear elastic, viscoelastic and inhomogeneous models. Eventually, the last experiments are demonstrations of the haptic simulations in applications with complete modeling procedures. As examples, they illustrate the practical implementations of the real time haptic simulations in medical technology and telepresence.

6.1. Experimental System

6.1.1. Experimental Robot

The experiments are carried out with an experimental robotic system, which can execute manipulation, measurement, modeling and control tasks. The system contains a 6 degree-of-freedom robot, haptic user interface, control computers, communication units and support devices, such as power supply unit and breakout box. It can be configured for varying applications, such as the laser scanner, the haptic experiment table and the telepresence system. In this experiment, the robot is mainly used as an operator for haptic tasks. Based on the measurements during the manipulations, the performance of the force modeling and real time haptic simulations are investigated. First of all, the experimental robot and its subsystems are detailed.

The contact operations are executed by the 6 degree-of-freedom robot, which consists of 5 PowerCube modules and a robotic end effector, which is illustrated in Fig. 6.1. A force/torque sensor is mounted on the robotic end effector to measure the contact force during the contact operations. There is also a laser distance sensor, for the measurement of geometry, as mentioned in Chapter 2. Its structure and the montage of the sensors ensure that the central line of the bar is parallel with the laser rays, and coincides with the rotation axis of the last joint. This condition simplifies the calculation of the position of the laser measurements, Appendix A.2.

The multi-axis analogue output force/torque sensor (ATI Mini 40) [7] has the measurement range $\pm 40[N]$ for the sensor x and y directions and $\pm 120[N]$ for the z axis for force measurements. For torque measurements, they are $\pm 2[Nm]$ for all axes. It should be noticed that, in addition to the random noises during the measurements, the force measurements contain system deviations, including sensor drift and orientation errors of the sensors' sensitive axes. However the most important

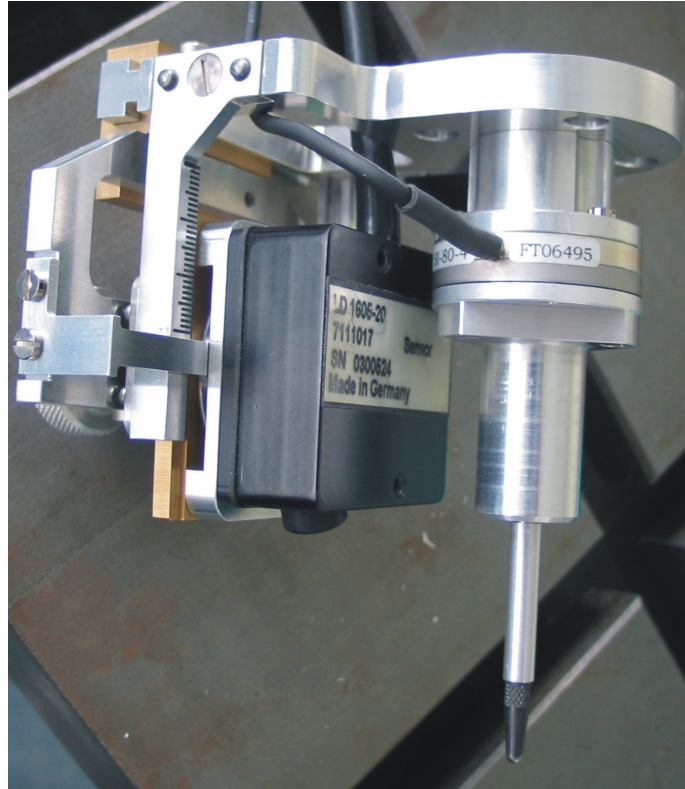


Figure 6.1.: The robotic end effector.

one is the mounting error of the force sensor. During the assembling of the sensor on the last robot joint with a flange, there may be significant unknown deviation between the robot coordinate system and the sensor system. Fortunately, it can be determined with an initialization procedure. This means the force sensor should be calibrated, especially before the first use. In the calibration procedure, the robot moves the sensor to several programed positions, then the transformation matrix from the sensor system to the robot system can be determined. The laser sensor (LD1605-20, Micro-epsilon) has the measurement range $55 \sim 75[mm]$ [81], and in connection with the forward kinematics of the robot, the positions of the sample points can be transformed to the inertial coordinate system. Both sensors are illustrated in the Fig. 6.2. Furthermore, since the characteristics of some materials depend on temperature, for instance the viscoelastic objects, the temperature measurements of the object, using the DS1820 thermometer from the Dallas Semiconductor [29], during the experiments are considered as additional information to the models. The sensor has an accurate measurement range from $-10^{\circ}C$ to $+85^{\circ}C$, and it is illustrated in Fig. 6.3 with the sensor adapter (DS9097).

Each of the first four actuators of the modular robot has one rotational degree of freedom, whereas the last one has two degrees of freedom, as well as a flange to fix the robotic end effector. The motion control of the robot is detailed in Algorithm 6.1. As a robot, the modules can be mounted as diverse configurations [34], the configuration for the following experiments is illustrated in Fig. 6.4.

The haptic device in the experiments is the Phantom Desktop from “Sens-

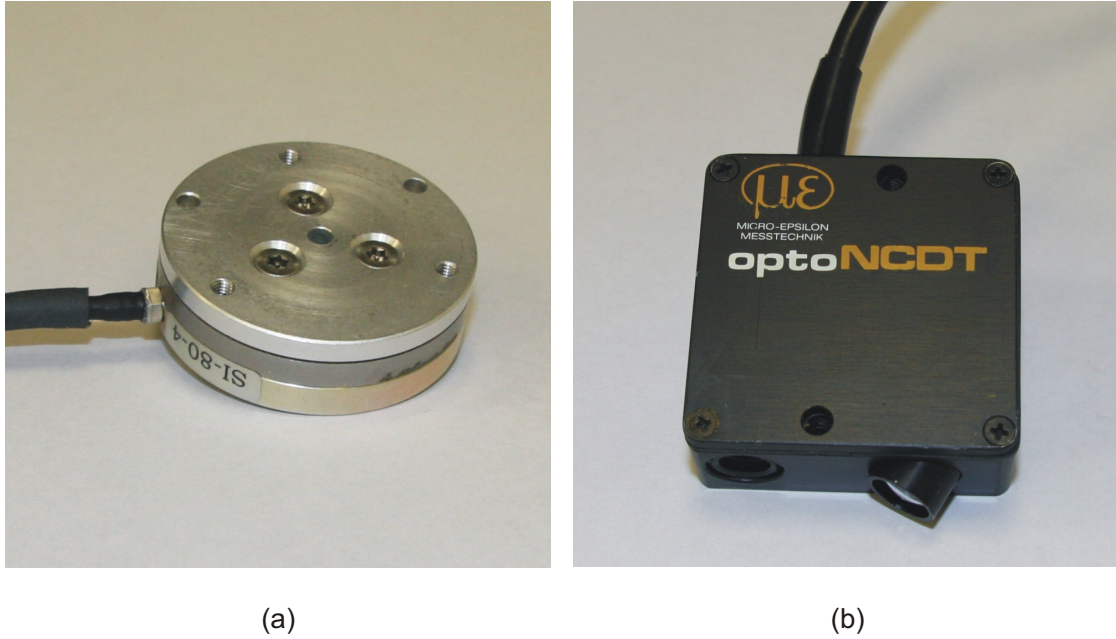


Figure 6.2.: The force/torque sensor (a) and the laser distance sensor (b).

Able” [105], which can reproduce the force signal in 3 dimensions in the range of $0 \sim 7.9[N]$. As the haptic interface, on one hand, it can convert the motion signal of human operators to control the robot. On the other hand, it can present the force feedback to the user. The Phantom, in connection with the control system of the robot, is illustrated in Fig. 6.5.

The contact situation is also an important condition in haptics, however in the experiments, the haptic scenarios are different. Generally the bar touches some objects, and the measurements are drawn during the operations as the reference data for the model parameter identification. The actual contact situations are described in every experiment.

Furthermore, in order to illustrate the position and orientation unambiguously in the following experiments, the inertial coordinate system (I) is used. It is identical with the robotic coordinate system (R), which is defined at the basis of the robot. The origin of it is defined at the center of the robot foot, the z axis points upward vertically, the x axis directs forward and the y axis is according to the right hand rule, Fig. 6.4.

6.1.2. Robot Kinematics

As a kind of mechanical structures, the kinematics and dynamics of robots are investigated in detail [39]. For this experimental robot, the position and the orientation of the robot tool are defined as

$$\mathbf{p}_e = [x_e \ y_e \ z_e \ \phi_e \ \theta_e \ \varphi_e], \quad (6.1)$$

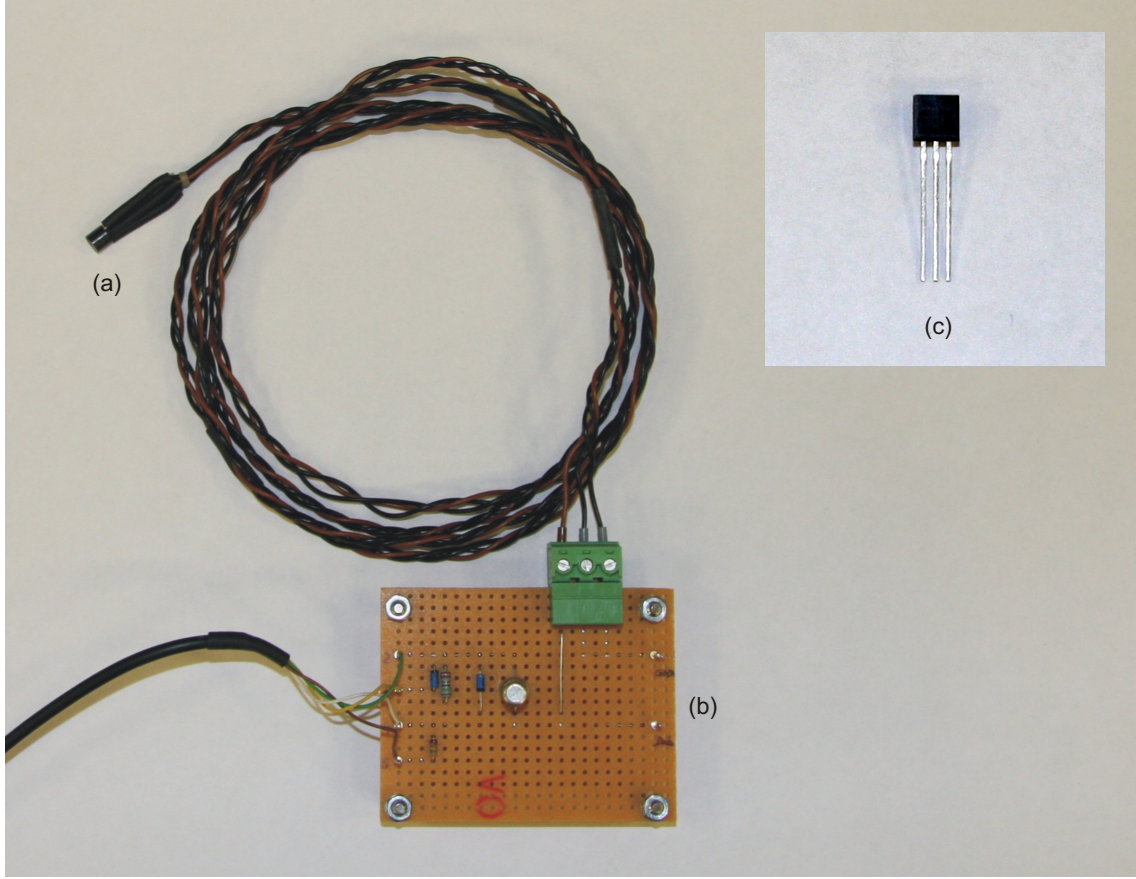


Figure 6.3.: The Temperature sensor, including sensor (a) and the DS9097 adapter (b), and (c) shows the detail of the temperature sensor.

in which the orientation is presented in the form of Euler angles. These states can be determined based on the rotation angles of all robot joints,

$$\alpha_{rj} = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6]^T. \quad (6.2)$$

This process is known as the robotic forward kinematics [60]. The transformation matrix of a robot link (i) can be calculated recursively through each link before it one by one,

$$\mathbf{A}_{iI} = \mathbf{A}_{i(i-1)} \cdot \mathbf{A}_{(i-1)I}, \quad (6.3)$$

as well as the position vector,

$${}^i\mathbf{r}_{Ib,i} = \mathbf{A}_{i(i-1)} \cdot {}^{(i-1)}\mathbf{r}_{Ib,(i-1)} + \mathbf{A}_{i(i-1)I} \mathbf{r}_{bj,(i-1)} + {}^i\mathbf{r}_{jb,i}, \quad (6.4)$$

where the index j and b are for the joint and the barycenter of the robot links. As a serial robot, \mathbf{r}_{bj} and \mathbf{r}_{jb} are the vectors from the barycenter to the next joint and from the last joint to the barycenter, respectively. Then the position in the inertial

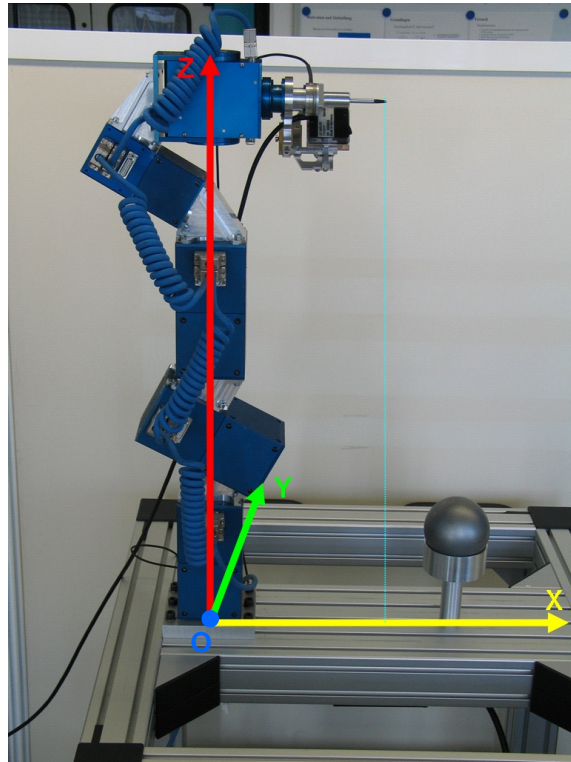


Figure 6.4.: The experimental robot and the robot coordinate system. This state is defined as the zero-position of the robot, i.e. all the joints are at 0° . The robot coordinate system is an inertial system, defined at the robot bases.



Figure 6.5.: Phantom Desktop. The computer screen shows the interface of the robot motion control system.

Algorithm 6.1: Robot control using CAN-bus

```

Initialization
- CAN-bus communication
- force/torque sensor
- laser sensor
if calibration then
  - Force/torque sensor calibration
end if
Search the modules
Define the motion mode
Move to the home position
Move to the zero position
loop
  (for each module)
  Set velocity
  Set acceleration
  Set position
  Move ...
  Checking status
  if quit then
    Break
  end if
end loop

```

coordinate system (I) is

$${}_I\mathbf{r}_{Ib,i} = \mathbf{A}_{Ii} \cdot {}_i\mathbf{r}_{Ib,i}. \quad (6.5)$$

By this means, the position and orientation of the robotic end effector can be calculated as the last link of the robot. Since the I system is identical to the robot system (R), it yields for the robot foot as the first link,

$$\mathbf{A}_{1I} = I^{3 \times 3}. \quad (6.6)$$

On the opposite side, for a given operation, the position and the orientation of the robotic end effector can be determined directly. The unknown terms are the rotation angles of the joints. They have to be determined using the inverse kinematics,

$$\{{}_I\mathbf{r}_{e,I}\mathbf{A}_e\} \xrightarrow{\text{inverse kinematics}} \boldsymbol{\alpha}_{rj}. \quad (6.7)$$

For a robot, if three rotation axes share an identical intersection point, the inverse kinematics has analytical solutions [91], hence the applied configuration in this experiment has an important advantage, i.e. its inverse kinematics can be solved analytically.

It should be noticed that the solutions of an inverse kinematics problem are not unique in general. However, only one of them defines the best suitable motion of the robot to reach the goal position. There are different criteria to select the

best solution as optimization problems, for example velocity, acceleration or energy expenditure [34].

6.1.3. Communication

In order to use the robotic system in diverse experiments, the robot, the haptic interface and the control computers are set with divided configurations. The connections among them are based on general communication technologies, for which two principles are taken into consideration. For the direct control, the robot is connected to the control computer with CAN-bus. For remote operations, internet connections are established.

The communication between the robot and the control computer is ensured with a controller area network (CAN), using the PCI133 CAN-bus card [36]. As a communication standard, the CAN-bus protocol allows high-speed communication between varying devices without any network host. The information is transmitted in the CAN data format as packages from the sender to the receiver, which are generally sensors, actuators and micro-controllers. The priority of the messages is given according to the message identifier. In this robotic system, communication is necessary among the electronic motors, encoders and the control computer. The information is the motion commands from the controller and the acknowledgment and statuses of the sensors and actuators. In the set-up, programmable interface is also used [37, 38].

The communication between the computers is based on the internet technology. The most popular protocols of internet communications are the transmission control protocol (TCP) [93] and the user diagram protocol (UDP) [92]. The TCP provides reliable and error free communications by using “data stream”, acknowledgment, retransmission and other mechanisms, while the sent data using UDP are in the form of datagrams, and the data transmission is unreliable. However the TCP network is heavily loaded by looking after the communication and correcting errors. Hence, the unreliable UDP communication is more efficient than the TCP. Therefore, real time data are generally transmitted using UPD. A well known example is the real time video and audio in the internet. In this chapter, the UDP is used for the transmission of real time haptic information using the program ‘SFBComm’ from the SFB 453. This implementation is especially suitable for the telepresence and telerobot applications.

6.2. Virtual Reality

In this section, the experiments using contact force models of virtual objects are investigated. The models with different constitutive equations and material feature are used with the same geometric model, which is a cuboid, $120[mm] \times 210[mm] \times 60[mm]$. Then the results of these models are analyzed and compared. Furthermore, the simulations are presented using the Phantom as a typical virtual reality example.

There are two aspects in the haptic simulation, one is the computational time. To keep the comparability, an identical computer with a $3.0GHz$ processor (CPU)

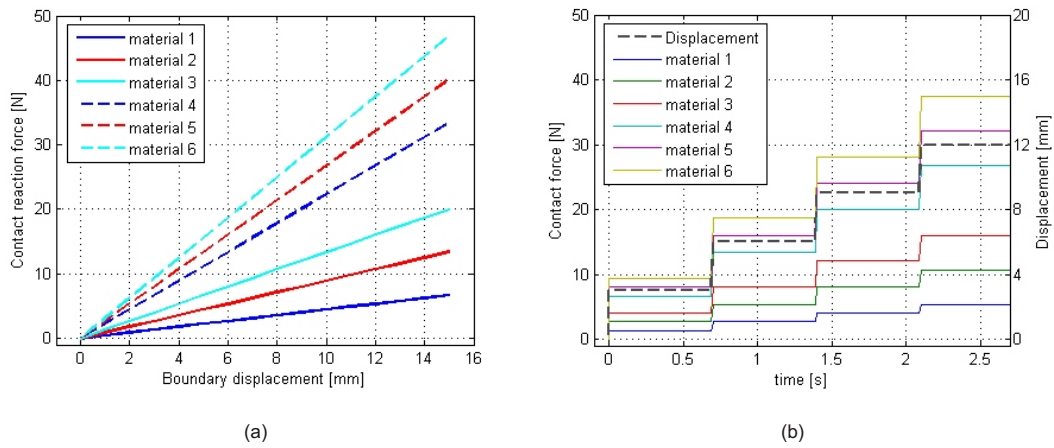


Figure 6.6.: The relation of boundary displacement and according contact reaction force, including the contact force response to a cyclic process with constant absolute value of the boundary displacement rate (a), and the curves of contact force and boundary displacement in connection with time scale (b).

is used for the different models. The other is the accuracy. In the simulations, the material behavior and the stress-strain relations are verified. However in this section, the models are based on virtual deformable bodies, hence the reference data and model identifications are not available for these models. The quantitative accuracy analyses of the haptic simulations are detailed in the following experiments.

The linear elastic and viscoelastic constitutive equations are implemented in two models, respectively. As the most important mechanical behavior, the stress-strain relations are presented by the relations of the boundary displacement and the according contact reaction force. For Hooke's materials, they are lines, Fig. 6.6(a). The time processes of the contact force are illustrated in Fig. 6.6(b), with respect to the boundary displacements. It is obvious that they have linear relations, and are independent on the former states.

In viscoelastic materials, the stress-strain relations, as well as the force-displacement relations, show hysteresis, Fig. 6.7(a). The curves of some typical relaxation processes are shown in Fig. 6.7(b). Both diagrams have merely shown the basic features of viscoelastic materials. In order to analyze the detailed effects of different material parameters, the following tests are carried out, and the effects of the motion functions, the boundary displacement rates and the relaxation times are investigated separately, Fig. 6.8. As the conclusion, the theoretical feature of the viscoelasticity is modeled and verified by these simulations, including the dependence of the internal stress on the loading history and time scale. The time processes of these experiments are illustrated in Fig. 6.9, 6.10, 6.11, 6.12 and 6.13, as reference, and the concrete settings of these virtual viscoelastic materials are given in Appendix A.3.

These models validate the fundamental functionality of the modeling of material features, the methods can be used for other implementations latter. However, as another basic performance, the computational expense also has to be surveyed explicitly. For this reason, both models and an inhomogeneous model with linear elastic materials are considered. Furthermore, the models with different mesh

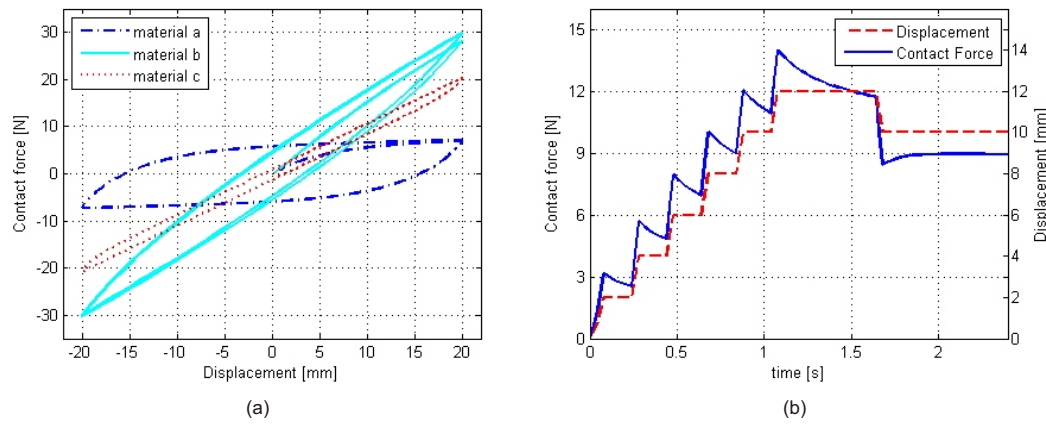


Figure 6.7.: The relation of boundary displacement and according contact reaction force with different material parameters. (a) illustrates the contact force response to a cyclic process with constant absolute value of the boundary displacement rate, *material a*: $E_s = 2000[N/m^2]$, $E_1 = 20000[N/m^2]$, $E_2 = 8000[N/m^2]$ and the relaxation times are 0.6 and 0.1 second; *material b*: the elastic modulus are 20000, 10000 and 8000[N/m²] for E_s , E_1 , and E_2 , τ_1 and τ_2 are the same as those of material a; *material c*: the elastic modulus are 20000, 10000 and 6000[N/m²], respectively, and the time constants are 0.1 and 0.2[s]. And the displacement rates are 0.01[m/s] for material a and c, and 0.05[m/s] for the third material. The Poisson's ratio is 0.3 for all materials. The curves of contact force and boundary displacement in connection with time scale are shown in (b).

structures are also taken into account, i.e. the normal mesh and the quality mesh. The computing time is compared in Tab. 6.1. For the creation of the models, the linear model is efficient, with about 95[ms], and the inhomogeneous model is the most expensive one due to the region determination of the tetrahedra, about 395[ms]. The creation of the viscoelastic model is a little more expensive than the linear model, because there are some additional vectors and matrices for the model and each element with respect to the effect of loading history, it takes about 120[ms]. In detail, the computation expense of this procedure includes two major parts, the geometric mesh, circa 90[ms], which is identical for all models, and the assignment of the stiffness matrix. The second part is significantly different for the diverse models, the inhomogeneous model takes about 300[ms], inclusive of the region determination of the tetrahedra, whereas for the other models, it is not so heavily loaded. These creation procedures are merely necessary in the initialization phase. In the simulation phase, the computational expenses of the contact force evaluations are identical for the linear and inhomogeneous models. It is about 15[ms] at each step. Whereas with the viscoelastic model, it takes about 30[ms], because of the calculation of the stress history terms. As special cases, if the boundary displacement is the same as that at the last evaluation, the conjugate gradient (CG) algorithm can take the previous values, and outputs it without iteration. Hence the force evaluation takes just a few milliseconds.

As introduced in Chapter 2, the quality mesh has significant advantages for FE models in connection with the accuracy of FEM and the convergence of the CG

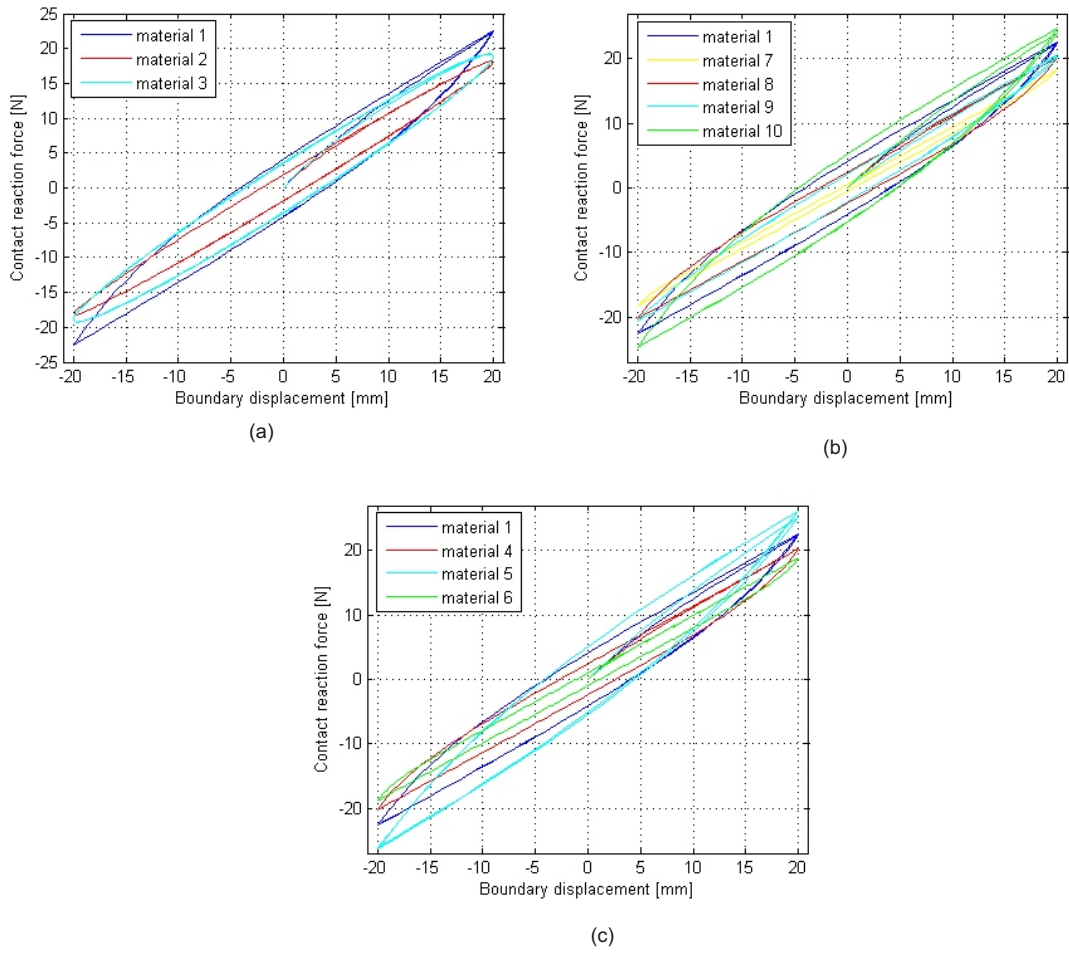


Figure 6.8.: The relation of boundary displacement and contact force with different motion functions (a), different boundary displacement rates (b) and different relaxation time coefficients (c).

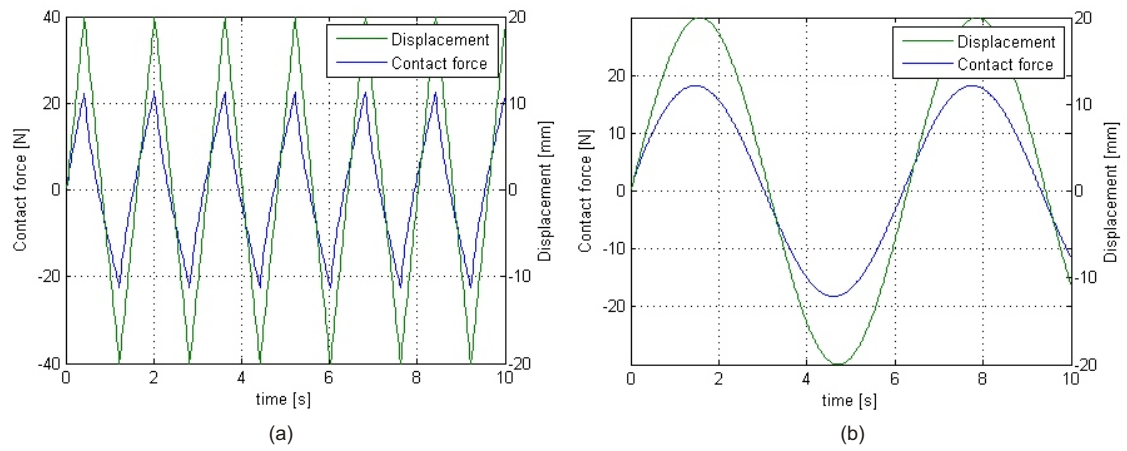


Figure 6.9.: Experiment setting 1 (a) and 2 (b).

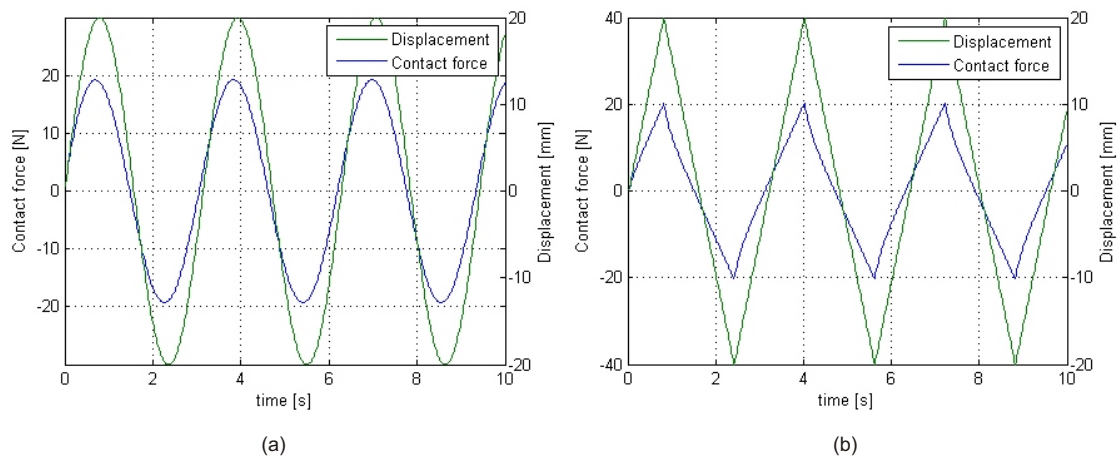


Figure 6.10.: Experiment setting 3 (a) and 4 (b).

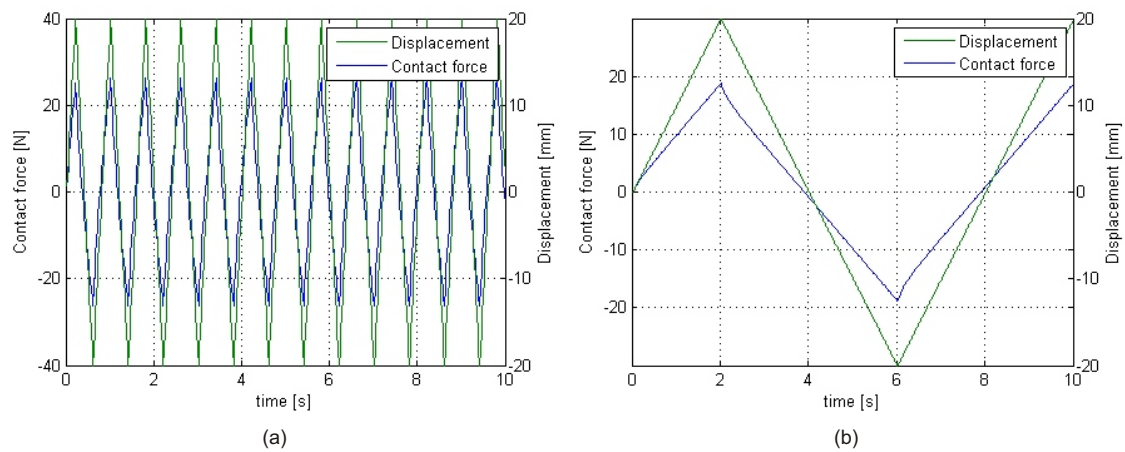


Figure 6.11.: Experiment setting 5 (a) and 6 (b).

Table 6.1.: Comparison the computational expense of the models, in [ms]. All the models are using the quality mesh.

force model	model creation	force evaluation [ms]
linear model	95	6 or 15
viscoelastic model	120	10 or 30
inhomogeneous model	395	6 or 15

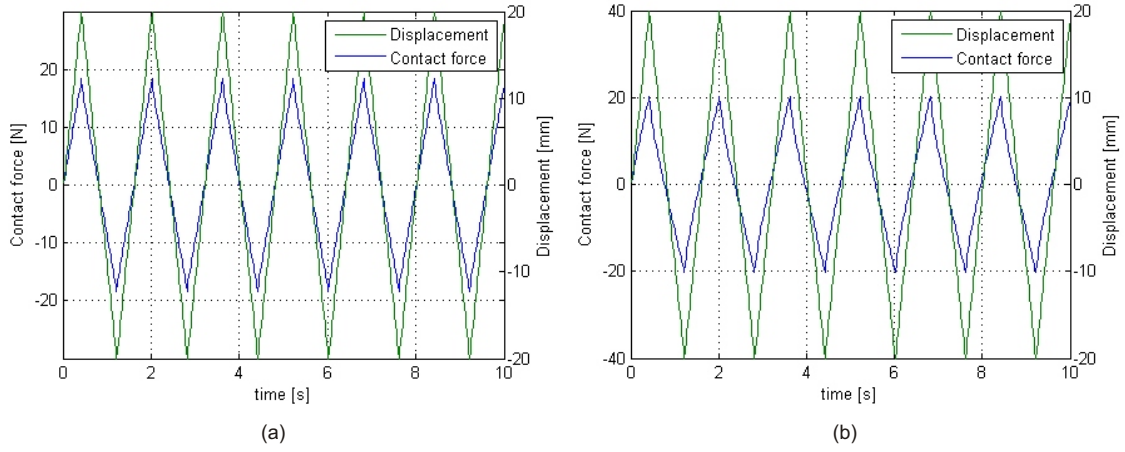


Figure 6.12.: Experiment setting 7 (a) and 8 (b).

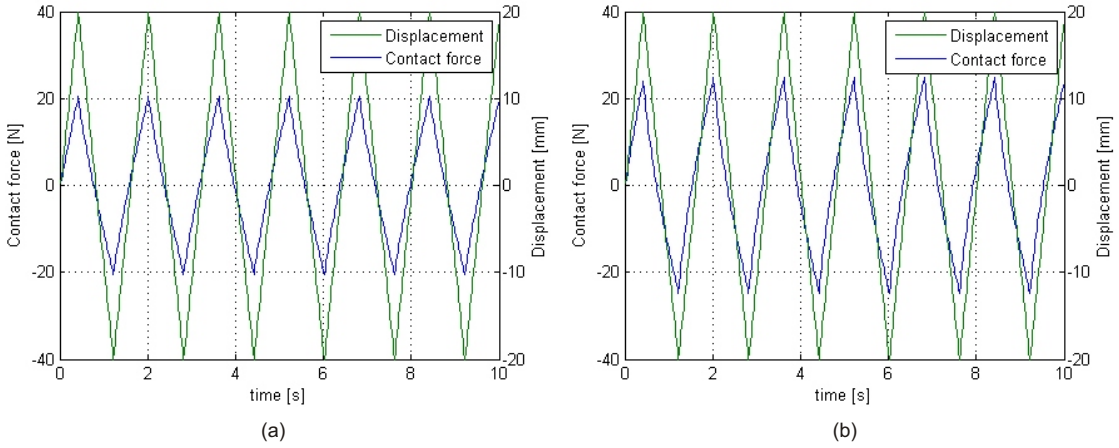


Figure 6.13.: Experiment setting 9 (a) and 10 (b).

algorithm. Generally, the quality mesh may insert some nodes and create more elements than the normal mesh. That means the model initialization may become slightly longer. However, without thin and flat elements, the CG algorithm will converge more quickly. By this means, the computing time for force evaluations can be reduced to about $\frac{1}{3} \sim \frac{1}{4}$ of that of the normal mesh in this experiment. In Tab. 6.2, the computing times of different models with both the mesh methods, also including the models from the latter sections, are summarized and compared. It should be noticed that the heart model with normal mesh has accuracy problem, Section 6.6. Hence, the following conclusion can be drawn: the quality mesh is much more accurate and economical for numerical solution processes in comparison with the normal mesh. In this Chapter, without special note the models are always with quality mesh.

Table 6.2.: Comparison of the model with the normal mesh and the quality mesh, including different geometries and material features. Some of them are from the following experiments.

geometry	quality mesh			normal mesh		
	number of notes	number of tetrahedra	force evaluation [ms]	number of notes	number of tetrahedra	force evaluation [ms]
ball	120	285	5 or 13	118	280	5 or 100
cuboid (linear elastic)	230	568	6 or 15	226	548	6 or 110
cuboid (viscoelastic)	230	568	10 or 30	226	548	10 or 130
heart	535	1763	20 or 90	157	414	10 or 110

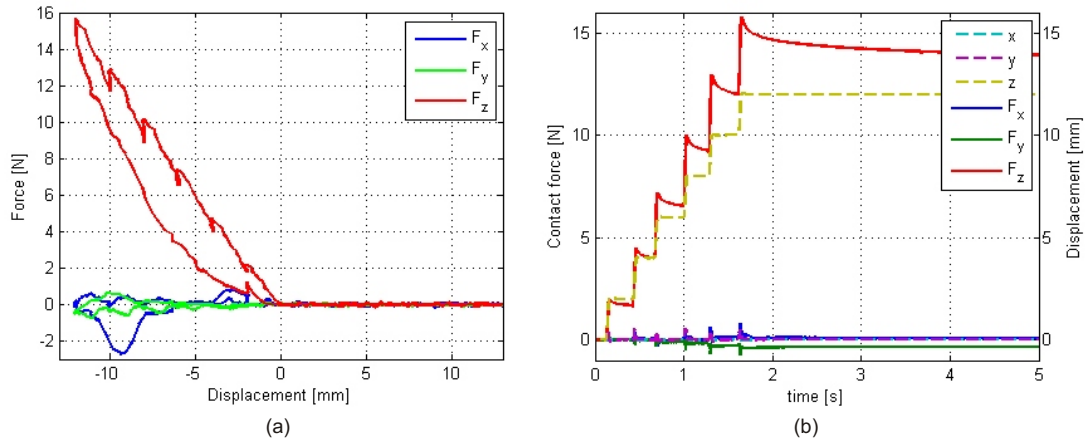


Figure 6.14.: Measurements during contact with the polyurethane cuboid, including the relation of the contact force and boundary displacement (a) and the time process (b).

6.3. Polyurethane

Polyurethanes (PU) are a special kind of polymers. They have a notable advantage that the physical characteristics of them with different micro structures can be adjusted in a very extensive range, including the mechanical properties. Thus they are widely used in industrial and research areas.

In this experiment, a homogeneous cuboid is tested, made of polyurethane, Sylomer from Getzner [44]. Then it is modeled as linear elastic and viscoelastic material respectively, without and with respect to the effects of the loading history. The identical measurements are provided as reference data. Furthermore, the geometric measurements and the mesh, with 230 nodes and 568 tetrahedra, are also the same for both models, to keep the comparability of them. The goal is to verify both modeling methods, and to compare the performance of the linear elastic and viscoelastic models.

The contact situation is: a bar presses the cuboid with different depths and holds the positions for some seconds. For the identification of the models, the cuboid is tested at five different positions on the surface, one of them is illustrated in Fig. 6.14 as an example. Then samples are selected from the measurements as reference data. As mentioned above, they involve the contact reaction force and associated boundary displacements.

6.3.1. Linear Model

As the first model, the cuboid is modeled as linear elastic material. The linear tetrahedra and linear constitutive equations in Chapter 3 are implemented. The initial values of the material are assigned to the model parameters, i.e.

$$E = 20000[N/m^2] \quad \text{and} \quad \nu = 0.3. \quad (6.8)$$

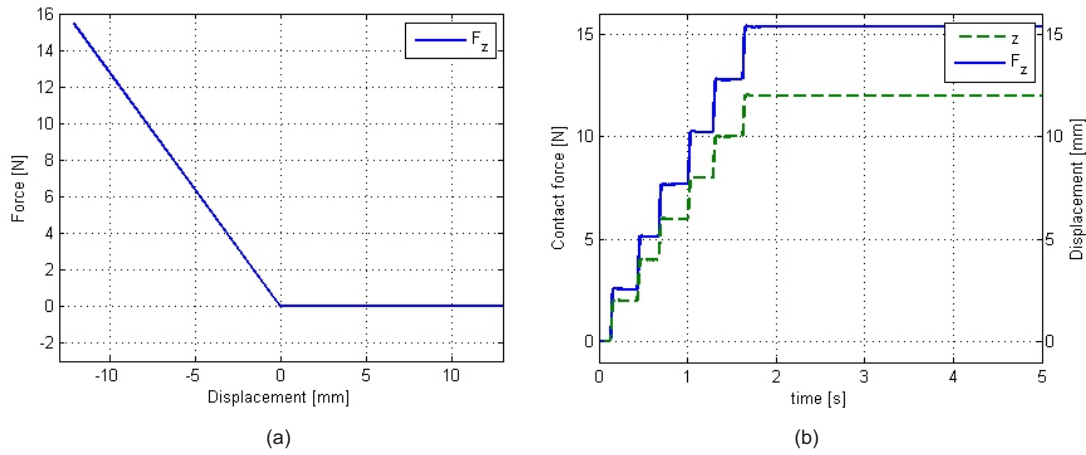


Figure 6.15.: Simulation of the polyurethane cuboid using the linear elastic model, including the relation of the contact force and boundary displacement (a) and the time process (b).

The force evaluation is directly according to the conjugate gradient algorithm, and the computation time is about 15[ms]. In this case, the analytical models are not implemented, since the force feedback with more than 60[Hz] is already acceptable using the Phantom in this application.

As shown in Fig. 6.14, the x and y components of contact force are in the vicinity of zero, and the force components in the z direction are much greater than the other two. Hence the simulation of the contact reaction force in the z direction is investigated in detail.

Since the to be optimized parameters are just the Young's modulus and the Poisson's ratio, it is not necessary to use a large number of samples. The time stamp has no effects in the linear models. For this reason, 20 ~ 30 samples are sufficient for the identification of this linear model at each sample position, i.e. location samples. Substituting the samples into Algorithm 5.7, the optimizations converge after several iterations, and then five location samples are available.

Since there is no significant derivation among the material parameters of the location samples, the cuboid can be regarded as homogeneous, and the unsupervised model learning and the subsequent inhomogeneous modeling procedure are not necessary. This conclusion is a good match with the fact of this cuboid. Eventually, assigning the average material values to the model, the corresponding simulations are illustrated in Fig. 6.15.

6.3.2. Viscoelastic Model

However the deviations between the simulations and the reference are obvious in the relaxation periods. Considering the adaptive model identification, the criteria for the viscoelastic models are checked. Subsequently, the contact forces at the same sample position with similar boundary displacement are compared, and significant deviations in the form of r_{ft} are detected. For this reason, the viscoelastic model is

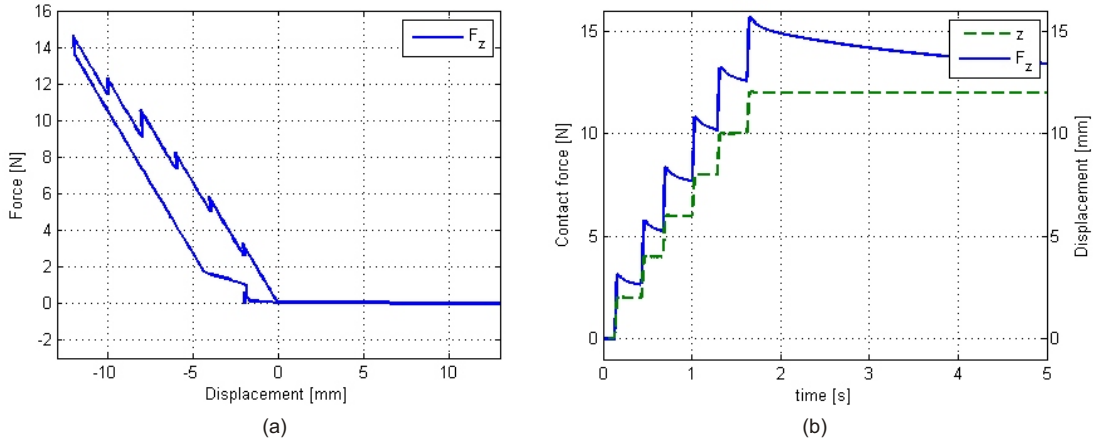


Figure 6.16.: Simulation of the polyurethane cuboid using the viscoelastic model, including the relation of the contact force and boundary displacement (a) and the time process (b).

implemented as an attempt, to investigate and simulate the polyurethane cuboid with more details.

The modeling in this section is based on the same measurements and using the identical geometric model. The difference is that the selected samples are recorded with the time stamps, since the loading history should be taken into consideration in viscoelastic materials. Being aware that the material parameters here are more than those of the linear model, more samples should be drawn. The first term of the Prony series is implemented in this model, thus the model parameters are defined as

$$\chi = [E_s \quad \nu \quad E_1 \quad \tau_1]^T. \quad (6.9)$$

The sample time are $30[ms]$.

After the model identification and model parameter optimization, the viscoelastic model provides better simulations in comparison with the linear model, Fig. 6.16.

It can be recognized that the curves in Fig. 6.16 are different from those in Fig. 6.14. The reasons lie in the fact that only one Maxwell element is implemented. In comparing with the experiments of animal tissue in Section 6.6, in which the simulations can fit the measurements better with more Maxwell elements, the effects of the Maxwell elements can be explained.

6.4. Inhomogeneous Material

The experiment with inhomogeneous material is carried out also by the robotic system, and the simulations are presented by using the Phantom. The test object is an inhomogeneous cuboid, $120[mm] \times 210[mm] \times 60[mm]$, Fig. 6.17. It consists of two heterogeneous regions, made of different kinds of polyurethane foam with different material values. Both regions are distributed over the horizontal x - y plane, the boundary between them is a curve, Fig. 6.18(b).

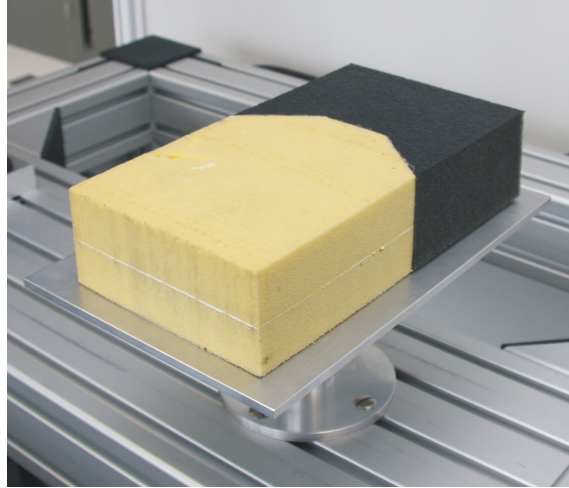


Figure 6.17.: The inhomogeneous object, lying on the test table, the dark material is softer than the light one.

The central goal of this experiment is to model the inhomogeneity in materials, with the different regions being modeled as linear elastic materials. The detailed modeling of polyurethanes has been illustrated in Section 6.3.2.

6.4.1. Modeling

As an object with simple geometry, which is similar to the former experiment, it is also measured with laser scanning. The FE model consists of 230 nodes and 568 tetrahedra, with the stiffness matrix $\mathbf{K} \in \mathbb{R}^{690 \times 690}$.

During the operation, the bar contacts the upper surface at 20 sample positions, and 500 samples are drawn from the measurements as the reference. A force model is established just using the samples, and then the model has been carefully compared with the test object.

At the beginning, 20 location samples with the material parameter

$$\{E_i, \nu_i : i = 1, 2, \dots, 20\} \quad (6.10)$$

are directly created using the initial estimation and the Gauss-Newton method. Then they are sorted by the hierarchical cluster analysis. The cluster tree of them, according to the material parameters, is illustrated in Fig. 6.18(a).

The cophenetic correlation coefficient is about 0.92, this means the cluster tree has represented the original samples properly. Then the location samples can be further classified based on this cluster tree. Subsequently, the proportions of the distances of adjacent layers are calculated, and it is reasonable to divide the location samples into two groups, which correspond to two heterogeneous regions. There are respectively seven and thirteen location samples in the soft and the hard regions, Fig. 6.18(b). Then the average values of the material parameters are assigned to the according regions.

In the following step, these classified location samples are used as the training

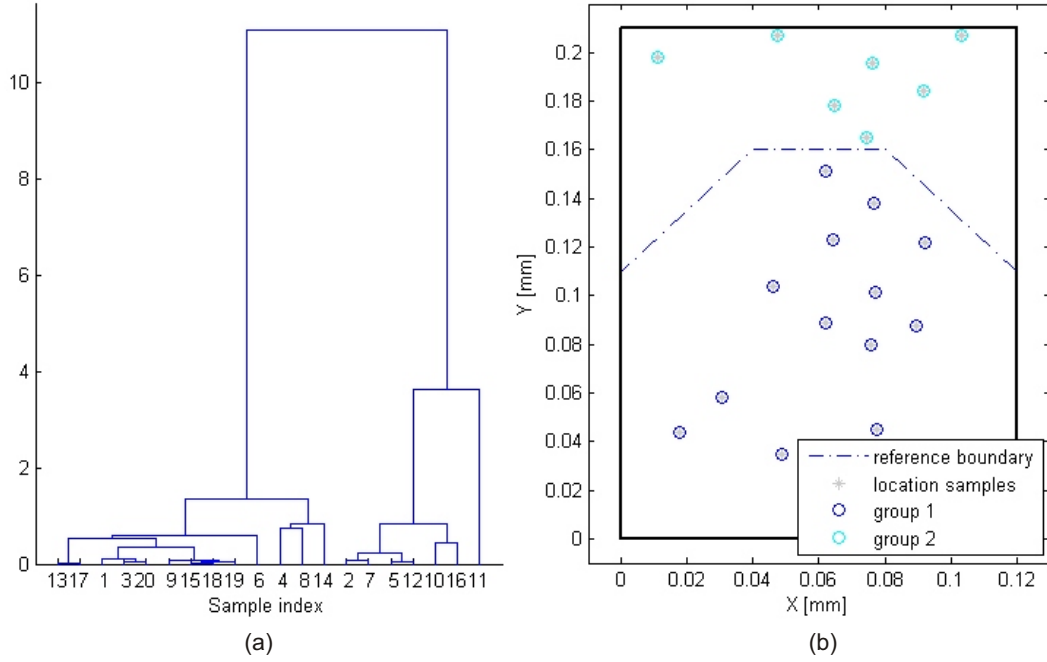


Figure 6.18.: Classification of the location samples. The location samples are sorted in the cluster tree (a) and then classified into two groups (b). The boundary is also illustrated as reference.

data to establish the discriminant equation. Then twenty thousand scan points are generated and classified, according to their positions, by the quadratic discriminant analysis (QDA). By this means, the region boundary is estimated, Fig. 6.19. Eight points are selected uniformly from the boundary between the two regions as the key points of it. Thus the primary model is created.

Subsequently, the model parameters can be optimized in detail using the global Gauss-Newton algorithm, and the optimized region boundary is illustrated in Fig. 6.20. It should be noticed during the parameter optimization that the positions of the boundary points are constrained in a small range surrounding the estimations derived by the QDA, according to (5.43).

6.4.2. Results

As shown in Fig. 6.20, all the location samples are classified correctly, and the identified boundary has similar tendency with the actual one. The left part of the boundary has more deviations than right part, because, in this experiment, the sample positions are selected randomly in the complete area domain, and few reference data are located in the right zone, especially in the near from the estimated boundary.

As the result, the haptic simulations are also represented to users as the force feedback, and the root mean square (rms) value of the simulations is about $0.1\text{--}0.3[N]$, depending on the sample positions. Generally, the errors of the hard region are much

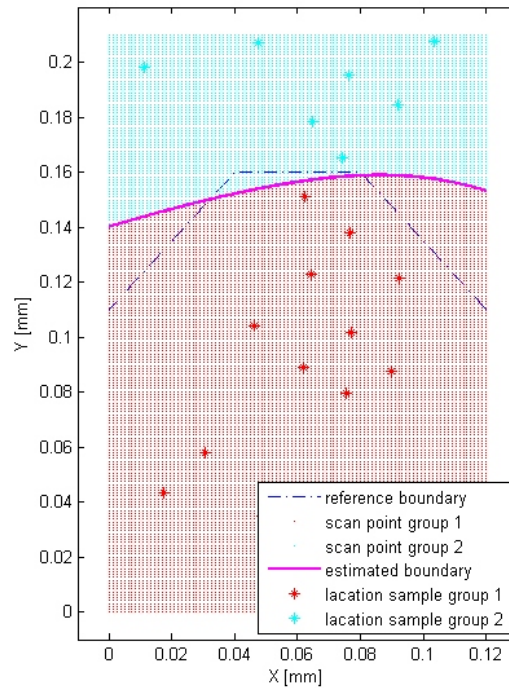


Figure 6.19.: Estimation of the region boundary using the quadratic discriminant analysis. The actual boundary is shown as the reference.

smaller than those of the soft region.

The similar modeling procedure can also be done using some commercial software, such as Ansys, however the programming interface of data exchange and contact detection using Ansys are sometimes problematic. Furthermore, the model mesh is dependent on the region boundary, it may even lead to convergence problems during force evaluations. Thus, the finite element analysis using Ansys is not satisfied for real time haptic simulations.

6.5. Telerobotic Operation

6.5.1. Problem Statement

In this section an application example is introduced. Telepresence systems give users the possibility to obtain the feeling in a remote or unreachable environment, including visual and auditory senses. Latter, the haptic information is also involved in the telepresence technology as another important component. These systems can be widely applied in many areas, for instance for aerospace and blue water explorers, robotic surgery and defusing robot in dangerous situations. A typical telepresence system consists of three components: an operator, a teleoperator and the communication between both sides, Fig. 6.21(a). Generally, in haptic workspace, the teleoperators contain diverse actuators and sensors for the manipulations of the test

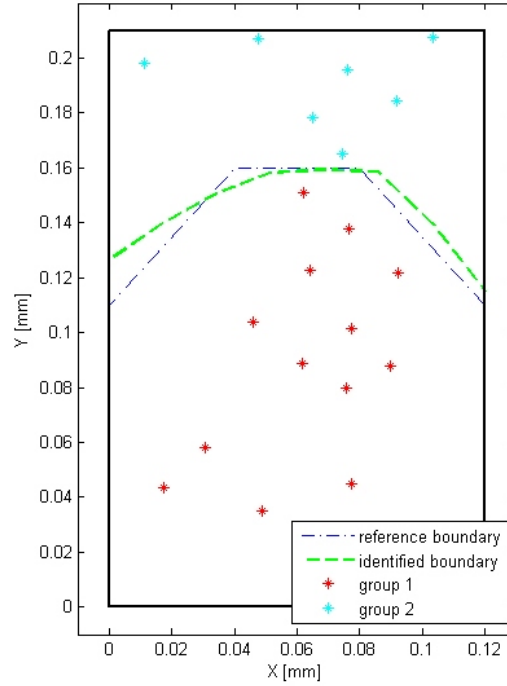


Figure 6.20.: Comparison of the identified boundary and the reference boundary.

objects, as well as the corresponding measurements. The human operators exchange the haptic information, including the displacement and the contact force, with the computer by using a haptic devices, the haptic human-machine interfaces [53], such as the Phantom. One of the most important challenges of these telepresence systems is the time delay due to communication, which leads to the inconsistency of the boundary displacement and the force feedback. It may be so difficult to control the teleoperator manually, that the complete system is instable [124, 27], if notable time delay exists, for example more than 50[ms]. In order to compensate for this time delay, the force models can be used at the operator side to provide force predictions, and then the delayed measurements can be replaced by the real time simulation from the models, Fig. 6.21(b).

As shown in the former chapters, the mechanical behavior of materials are described in the force models with the constitutive equations using the finite element method (FEM). Two major requirements of the force predictions in telepresence are consistent with those of the contact force models. The computational expense of the force evaluations has to satisfy real time force simulations, and the haptic simulations should be comparable with the corresponding reference, by means of model identifications.

6.5.2. Telepresence Scenario

The teleoperation scenario is modeled as a robot tool in contact with a deformable object. Concretely in this experiment, the experimental robot is configured as the

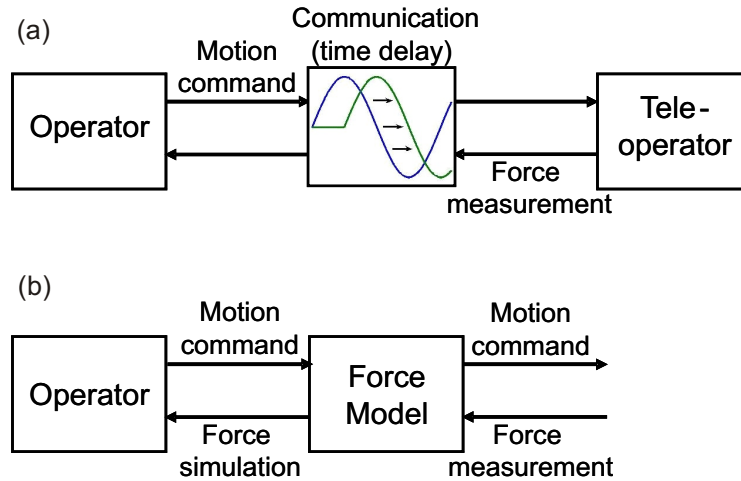


Figure 6.21.: A telepresence system (a), in which the force feedback to the users is the delayed force measurements. For the force prediction at the operator side (b), the measurements are used as the reference data for the learning and verification of the force model.

telerobot, i.e. the teleoperator, and the Phantom is used as the haptic interface of the human operator. The operation is the effector pressing a deformable sphere with a bar, illustrated in Fig. 6.22.

In order to establish the communication and keep the cooperation of the operator and the teleoperator, the system has to be set and configured properly. The implementations for the teleoperator and the operator are illustrated in Algorithm 6.2 and 6.3. In them, the vector \mathbf{s} defines the state of operator. It can help the system to select the actual function during the experiments. The meaning of each entry in the vector has been introduced in the algorithms. Furthermore, $*i \rightarrow$ and $*i \leftarrow$ denote the sending and receiving of communication signals in the i -th channel.

The goal is to provide model based haptic simulation as contact force prediction to compensate for the communication time delay. However in this experiment, the teleoperator is close to the operator, and the UDP communication between the them takes only few milliseconds. This time delay is relative short and has little disturbance for users during the operations. In order to illustrate the force prediction clearly, an additional time delay is implemented. The force feedback from the teleoperator and the motion commands sent by the operator are buffered in two stacks, respectively, which are illustrated in Fig. 6.23, the time delay is defined as Δt_{delay} . The data are pushed into the stacks at the current point of time t_c^{push} with their time stamp. Then after the time interval Δt_{delay} , they should be outputted from the stacks. Thus, the time delayed is simulated by means of using the data from the time t_c^{push} at the time $t_c^{pop} = t_c^{push} + \Delta t_{delay}$. In the actual setting, the simulated time delay is defined as 300[ms], which is comparable with that of the communication between earth stations and satellites.

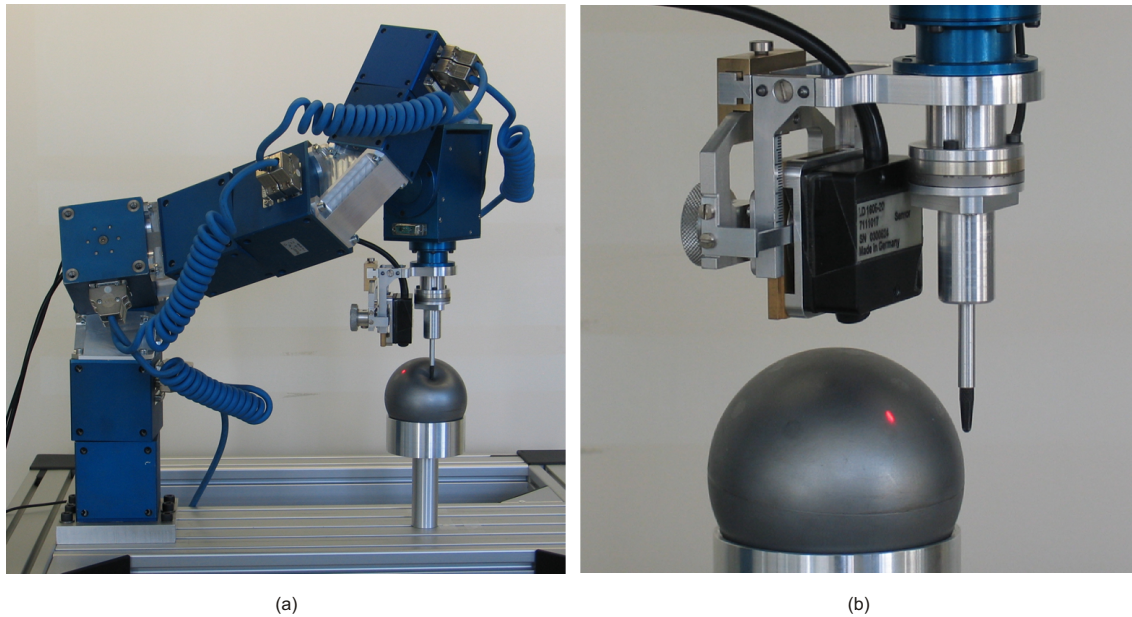


Figure 6.22.: Operation during the telepresence experiment. Teleoperator in contact with the test object (a) and the laser scanning procedure (b).

Algorithm 6.2: Teleoperator

Initialization:

- program
- UPD Communication
- Robot, Algorithm 6.1

Start "telerobotic mode"

loop

 Read the force/torque Sensor

 Receive motion command from the operator (*1 \leftarrow)

 Send contact force to the operator (*2 \rightarrow)

 Check the desired position

if accessible position **then**

 Move to the next desired position

else

 Hold the current position

end if

if quit **then**

 Exit "telerobotic mode"

end if

end loop

Quit

Algorithm 6.3: Operator

```

Initialization:
- program
- Phantom
- UPD Communication
loop
  Get input:  $c_{input}$ 
  Define state:  $s$ 
  Get the Phantom position
  Send motion command to the teleoperator (*1  $\rightarrow$ )
  Receive contact force from the teleoperator (*2  $\leftarrow$ )
  if  $s_1 = 'forceprediction'$  then
    Using the haptic simulation as force feedback
  else
    Using the received force measurements as force feedback
  end if
  if  $s_2 = 'timedelay'$  then
    Simulated time delay is effective, Fig. 6.23
  end if
  if  $s_3 = 'learning'$  then
    if sufficient samples:  $n_{sample} \geq n_{threshold}$  then
      Model identification
    else
      Draw samples
      The sample counter:  $n_{sample} = n_{sample} + 1$ 
    end if
  end if
  if  $c_{input} = 'quit'$  then
    Break
  end if
end loop
Quit

```

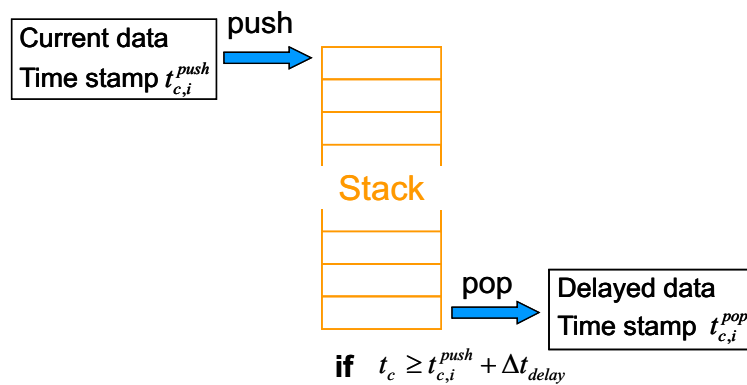


Figure 6.23.: Simulated time delay. The index i specifies a data entry, t_c denotes the current time. Data can be pushed into the stack at any time, but to pop them out from the stack, the corresponding condition have to be satisfied.

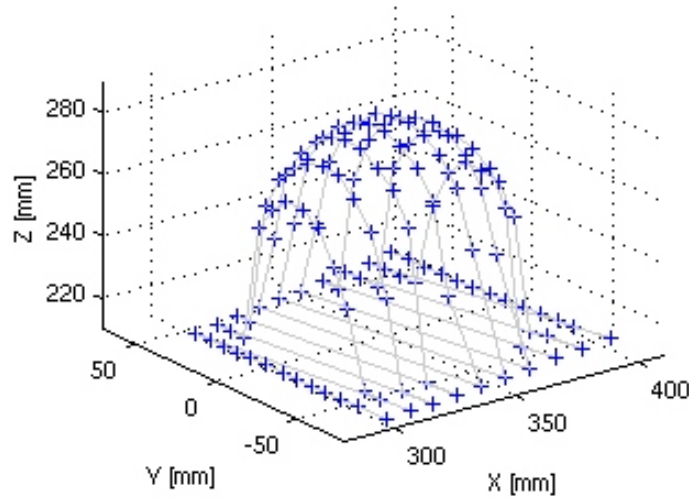


Figure 6.24.: The Geometric measurement of the Sphere. The sample points are drawn during the laser scanning, the grey curves illustrate the scanning path and the approximate contour of the sphere.

6.5.3. Modeling

The force modeling methods are used in the experimental telepresence system as an application example. At the beginning, the geometry of the sphere is measured by the laser scanning. The measured area is $130 \times 100[mm]$, and the 3D samples at 130 positions are drawn in this area. Fig. 6.24 shows the positions of these points, and with the virtual grey curve, the geometry of the object can be identified. Among them, 72 measurements (nodes) lie on the upper hemisphere. After the selection of the suitable samples, the closed surface of the object, including the projections, is created by using the surface reconstruction. Fig. 6.25 illustrates the outlook of the geometric model. It consists of the upper hemisphere, side surface and bottom, while the useless samples are eliminated. Based on the geometric mesh, the FE model with 120 nodes and 285 tetrahedra is established, which has the similar aspect as the closed surface. Then the linear elastic constitutive equation can be applied in the finite element model with arbitrary default material values. The computational expense to solve a batch of contact force needs approximately $30[ms]$, by a $3.0GHz$ computer.

During contact experiments, the robot drives the effector to contact the object on the upper surface, and the reaction forces are measured at the same time. Subsequently 20 chosen measurements are used to optimize the Young's modulus. The measurements during motion without contact should be neglected. There is a significant reduction of the residual sum of squares (rss) at the first few iterations, then the convergence condition is satisfied. Eventually, the FE model of the sphere with suitable material values is created.



Figure 6.25.: The Geometric model of the Sphere. After the surface reconstruction using these points, the closed surface is created, and the final geometric model with mesh has the same aspect as the closed surface.

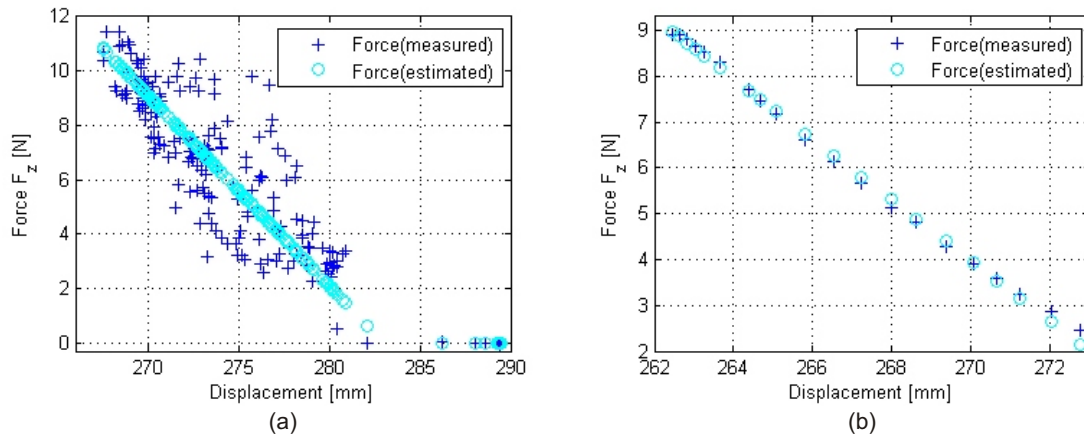


Figure 6.26.: Comparison of the measurements with the simulations derived by the contact force model. The force-displacement relation during manual motion is illustrated in (a), (b) is the detailed comparison using 20 samples.

6.5.4. Results

After the parameter identification, a series of press tests is executed. The simulated contact forces and the measurements are compared to verify the contact force model. Fig. 6.26(a) illustrates the relation between the contact forces and boundary displacements. 200 samples are drawn during several tests. Since the manual motion is carried out, there are some samples with greater random deviations in comparison with the program controlled motion. Moreover, the detailed comparison of the simulations with the corresponding reference data is also shown in Fig. 6.26(b), with 20 samples during a single test, the rms values are in the range of $0.1 \sim 0.2[N]$.

The performance of the telerobotic system and the model based force prediction

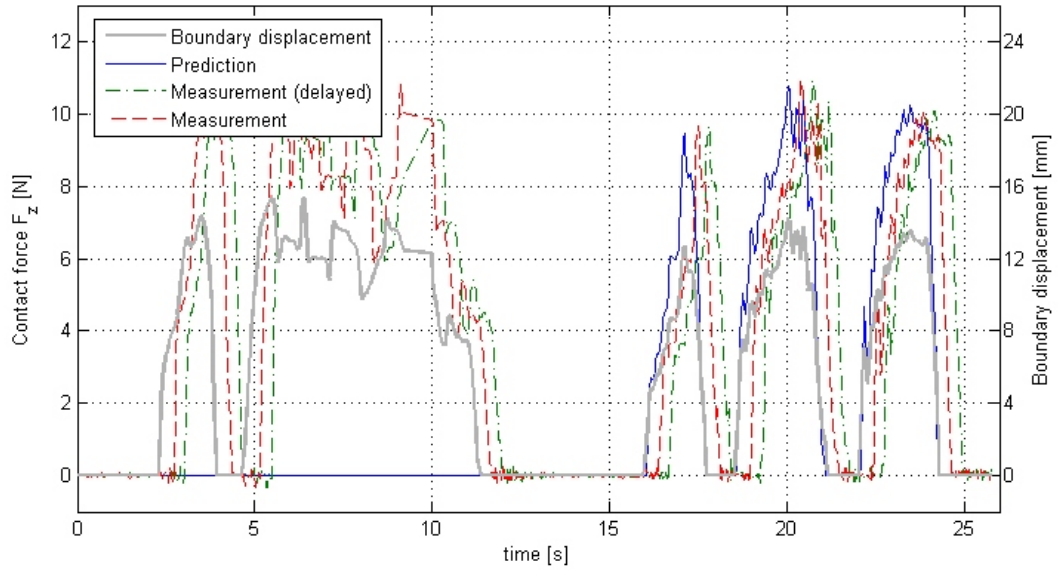


Figure 6.27.: Comparison of the force measurement and the force prediction derived by the FE model, with respect to the boundary displacement. The measurement curve is the force measurement at the teleoperator, and the delayed measurement is the measurement at the operator after the communication. Both should be the same. The model identification is at about 9[s], and takes circa 150[ms]. Before 15[s], there is not any force prediction, and the force curve is not synchronous with the boundary displacement. After it, the force prediction derived by the force model is used as the force feedback, which is always coincide with the displacement curve.

can be concluded as follows. Before the model identification, the force feedback comes from the delayed measurements, Fig. 6.21(a). It is extremely difficult for users to control the teleoperator using the Phantom, since the displacement and the force feedback are not synchronous, so that the system is instable. At the same time during the manually controlled motion and operation, the measurements, including the contact force and the associated boundary displacements, are memorized as the reference data for the model identification. Once the available data are sufficient, the model parameters are online identified during the operation. After that, the force model is used at the operator side, and it is prepared for the force prediction and haptic simulation, Fig. 6.21(b). Subsequently, the simulation derived by the identified model is applied as the force feedback to replace the force measurements, Fig. 6.27. Then the manual control of the telerobot is much easier than before, and remains always stable. The rms value in this experiment is about $0.2 \sim 0.3[N]$. If significantly disturbed samples are filtered, the rms values can be reduced obviously.

6.6. Animal Tissue

6.6.1. Problem Statement

As mentioned above, the application of haptics in medical and biological equipments is a highly interesting research field. Surgery robots have been intensively investigated in many works as new instruments [79, 99, 50]. In comparison with the traditional surgery technology, the surgery robots are more suitable for invasive operations. With these robots, it is also possible to enhance the telemedicine systems by carrying out operations in special situations or for patients in remote environments. In these cases, contact force modeling and simulation methods can be used to compensate for time delay or provide better performance, analogous to the telepresence system in the last experiment. There has also been a lot of research concentrated on some special effects, for instance the needle insertion [31]. Furthermore, these modeling and simulation methods can be used for medical simulators for trainings, to improve the skill of doctors.

In this section, the robot is used as an experimental medical instrument. An animal tissues is tested and modeled as a deformable body, using the modeling methods in former chapters, and the real time haptic simulation of it is implemented as well.

6.6.2. Experimental Setting

In this experiment, the test object is a pig heart, and the contact scenario is the bar pressing the heart at some positions, and the measurements are stored for the model identification. Then the model is imaged to be used in a telemedicine application. The telerobotic settings are similar as those in the last experiment.

As animal tissues, the hearts are inhomogeneous, anisotropic, nonlinear materials. Their behavior is dependent on the loading history as well. Furthermore, it should be noticed that the main components of hearts are the myocardium, which is a kind of muscles, and similar to the skeletal muscle [43]. The behavior of muscles can be both active and passive. There has been biomechanical research for the detailed modeling of the skeletal muscle [121] and the hearts muscle [28], using the FEM, from the biological and medical aspects. In this experiment, the heart is regarded as a passive viscoelastic deformable body, the aim is to create a model for the real time haptic simulation for the compensation for the communication time delay.

The contacts are typical relaxation tests, this means the bar presses the tissue at some position, and then it holds the position for an interval of time. During this period, the contact reaction force decreases monotonously with the time, Fig. 6.29. Based on this mechanical behavior, it is reasonable to model the tissue as viscoelastic material.

6.6.3. Modeling

The modeling of the heart starts also with the scanning procedure using the laser sensor. 130 sample points are measured, 83 of them are on the heart surface. Since

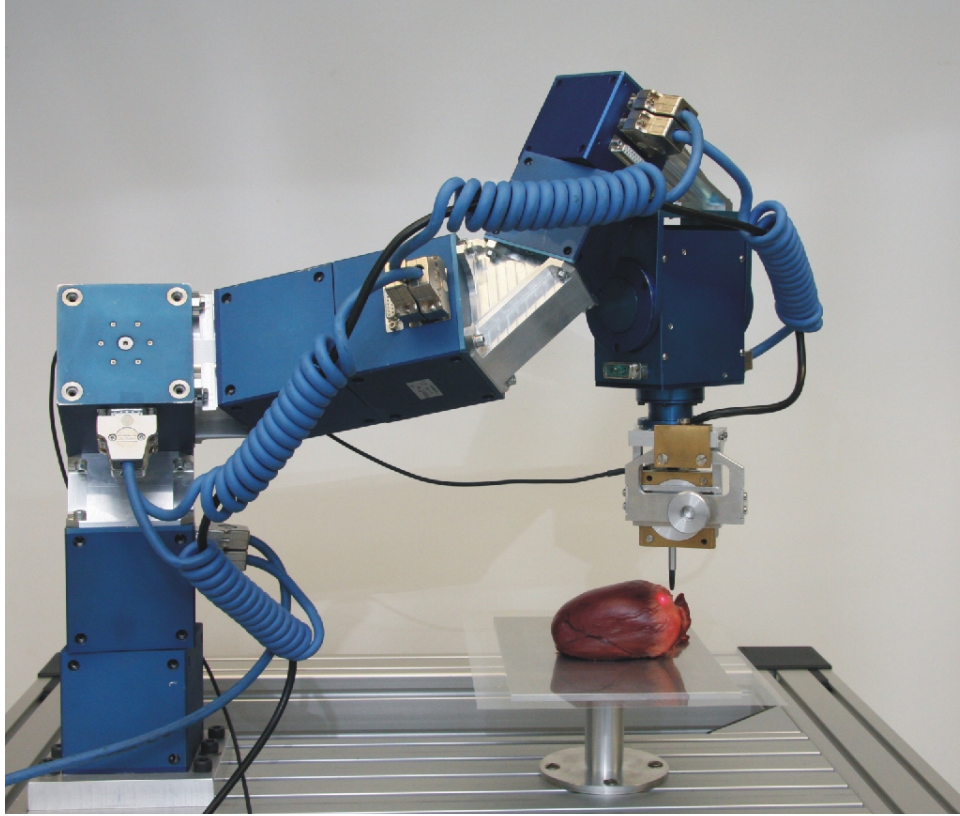


Figure 6.28.: The robot during the experiment. The light point on the heart is the current position during the scanning.

all the sample points are located on the upper or side surface, the bottom plane is also defined as the projection of the upper area. In total, there are 161 points, which define the whole surface, Fig. 6.30. Then the closed surface is established by using the surface reconstruction, and the geometric mesh structure, using Tetgen, is created to fill the volume of the closed surface, Fig. 6.31. Eventually, the complete geometric model of the heart consists of 535 nodes and 1763 tetrahedra.

At the beginning, the heart model, with two Maxwell elements, is assigned with the default parameter values. The computational time of the FE heart model is $20 \sim 90[ms]$, depending on the convergence of the force evaluations. The sample period of the haptic simulation is defined as $100[ms]$, which is sufficient for all situations. Using the measurements, the FE model parameters and the ACFM parameters are identified. Since the vertical force components are much greater than the other two horizontal components, the contact reaction force in the z direction is investigated intensively. 300 samples within 60 seconds during a single relaxation test are uniformly selected as the reference data for the FE model identification. As mentioned above, the model parameters are constrained, for the two time constants in the FE model, they are defined as

$$\begin{aligned} 0.001 < \tau_1 < 5[s] \\ 5 < \tau_2 < 60[s], \end{aligned} \tag{6.11}$$

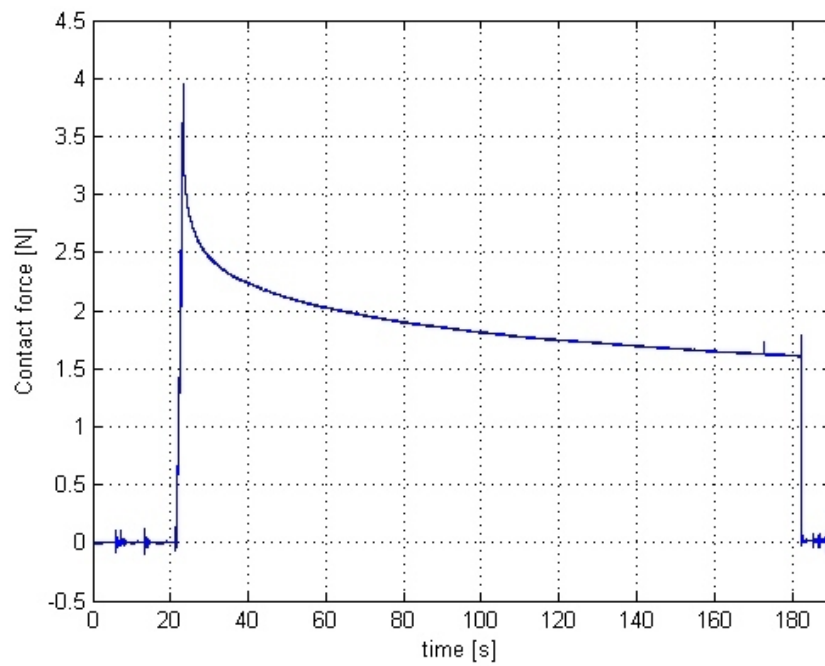


Figure 6.29.: Contact reaction force using a relaxation test. The force is monotonously decreasing with constant boundary displacement.

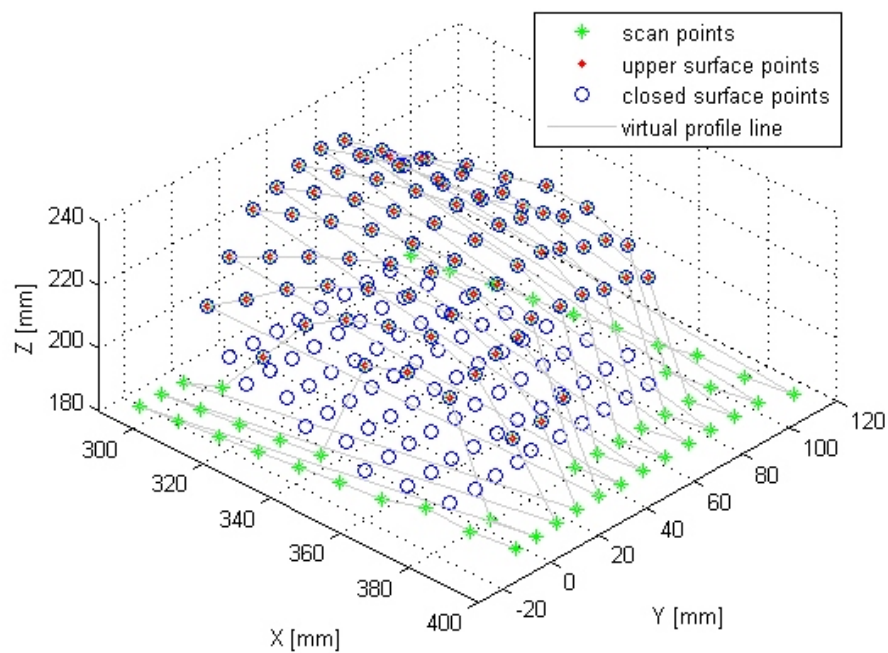


Figure 6.30.: The Geometric measurements of the heart. The grey curves illustrate the scanning path, and the approximate contour of the tissue.



Figure 6.31.: The geometric model of the heart. Its aspect is the same as that of the closed surface after the triangulation. The contour of the model in this experiment is relative rough, because the computational expense of the model is limited by the actual experimental computer. In robotic surgery using the identical modeling method, models are generally created based on the measurements from a CT. With high-end hardware, the computing power can be sufficient for the evaluation of more complex models.

for the short-term and the long-term relaxations, respectively.

During the model parameter optimization, the algorithm may not be converging with the specified start values. Among the convergence optimizations, the parameter setting with the least root mean square (rms) value is assigned as the identified values of the model parameters. The corresponding rms value is about $0.1[N]$. The optimization of the ACFM parameters is similar, using just 100 samples within 20 seconds for each sample position. The rms values are much larger than those of the FE model.

As shown in Chapter 5, the boundary displacement rate independent noise densities are defined as 0.9 for the system noise and 0.1 for the measurement noise in all directions. In this experimental setting, the boundary displacement rate lies in the range of $0 \sim 0.1[m/s]$. The basic layout is to limit the noise densities in the range $0.9 \sim 0.1$ for the system and measurement noises, then the coefficients in (5.52) and (5.53) can be determined, and the matrices \mathbf{Q}^n and \mathbf{R}^n can be calculated subsequently during the runtime. By this means, the FE simulations have enough weight in the EKF, so that the model states can be adjusted, whereas during the periods with high boundary displacement rate, the model can also respond quickly enough.

6.6.4. Results

In order to verify the identified haptic model, it is implemented in the experimental telerobotic and telepresence system for the contact force prediction to compensate for

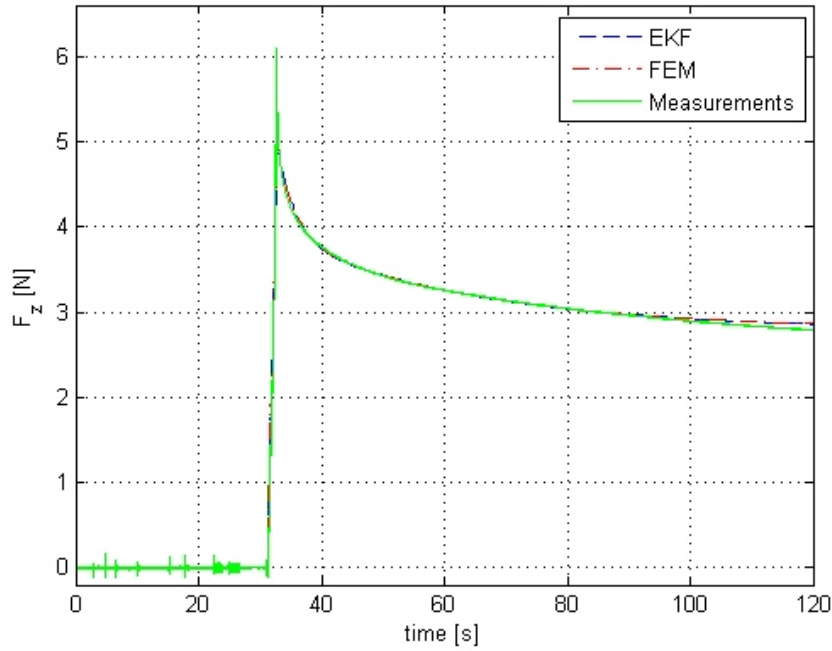


Figure 6.32.: Comparing the force measurements with the simulations derived by the finite element model and the EKF.

100[ms] time delay. This means the real time haptic simulations are used as the force feedback to replace the delayed force measurements. The experiments show that the system is always stable, and the manually controlled operations can be much easier in comparison with those without force predictions. The performance is investigated in detail by comparing the simulations with the corresponding measurements, Fig. 6.32. The deviations between them are illustrated as rms values, Fig. 6.34.

It should be noticed that the output of the haptic simulations is step by step, and in the FE simulation, the changes of the contact force may be extremely intense inside of a very short time interval, for instance 1[ms], due to the computationally expensive force evaluations, Fig. 6.33. They are a problem for high level haptic rendering. Whereas the real time simulation derived by the EKF shows its advantage in comparison with the pure FE simulation, i.e. the changes of the boundary displacements are taken into account with the 1000[Hz] sample rate, and the trajectory of the simulated contact force is smoothed. The intense changes, about 0.5 ~ 0.8[N], in the FE simulation have been eliminated as well. Furthermore, the rms error has also decreased clearly, Fig. 6.34. It is obvious that the rms values of the EKF simulations are always less than those of the pure finite element simulations.

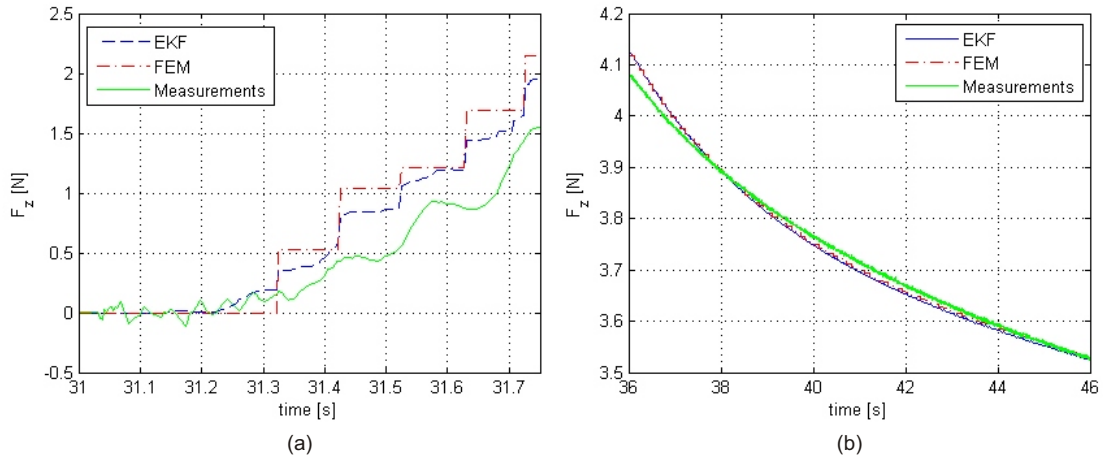


Figure 6.33.: Detailed comparison of the measurements with the simulations derived by the finite element model and the EKF. During large boundary displacement rate, the intensive changes in the FE simulations are eliminated by the EKF (a). During the relaxation procedure, the EKF provides smooth contact force simulations (b).

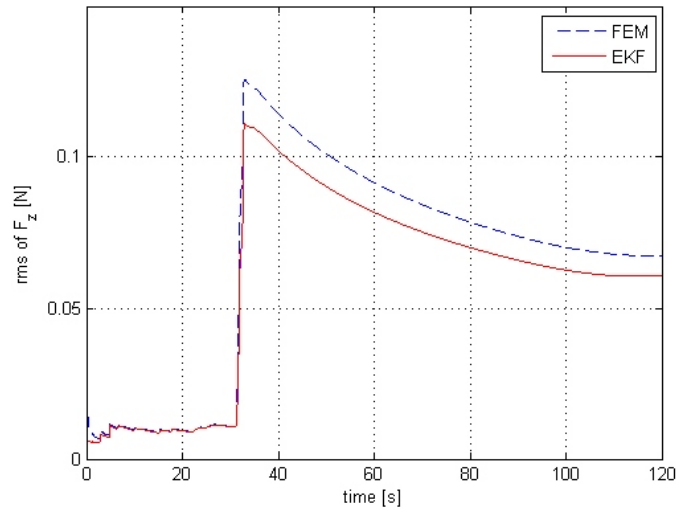


Figure 6.34.: Comparison of the rms values of the simulations derived by the pure finite element model and the EKF.

7. Conclusion

Haptic sensation is an important interaction and information between human being and the environment. In recent years, with the rapid development of computer science, it is possible to apply haptic simulations in more and more situations, for instance the aerospace, medical and biological technology. At the same time, the requirement for simulations are becoming stricter and stricter, mainly including the accuracy and the output rate. For this purpose, different force models and the appropriate modeling methods are developed and investigated. This tendency fulfills the desires of many applications to improve their performance of operations and manipulations. Connecting both sides, haptic technology has been developed as an inter-discipline. The most important application areas of it can be involved in the robotics, such as the telerobotic, haptic devices and virtual reality. Typical examples are the surgery robots. As a kind of modern medical instruments, on one hand they can provide patients with more flexibilities such as telemedicine, on the other hand, the doctors can obtain experience during virtual trainings, and the intensity of operations can be decreased. This also means that the efficiency and security of operations can be raised.

This thesis presents the methods for model based real time haptic simulations using the finite element method. They can be used in diverse applications to provide user high-fidelity haptic sensations. This modeling procedure can be divided into three parts. During the modeling and simulation, two major requirements are emphasized, the accuracy and the efficiency.

The first part concentrates on the modeling of the target objects, including the geometric modeling and the implementation of constitutive equations in the volume mesh. The modeling begins with the geometric measurements of the object surface, then the geometric model is established step by step, including surface reconstruction, geometric mesh and contact detection. Subsequently, material characteristics are given to these volume meshes. Eventually, the modeling procedure ceases with the FE models with default material values. By this means, the models of the objects are established with basic feature.

The second part is the model evaluation and model identification. The finite element evaluation is detailed for different FE models. Furthermore, for real time applications, the FE models are integrated with the analytical models with a data fusion method, in order to compensate for the computing time of the FEM.

For the case, both the model simulations and reference data are available simultaneously, the force models can be identified and optimized. Especially, an unknown object can be modeled with the most suitable material characteristics and modeling methods by using the adaptive model identification.

In the model parameter optimization, three primary error sources should be noticed: the first is the theoretical model, this deviation can be reduced by the selection of suitable theoretical models according to the objects and application conditions.

The second is the influence of varying noises, the EKF can take some stochastic information into consideration. If it is necessary, other stochastic methods may also be used. Furthermore, the reference data should be denser in the interesting areas than those of the others, to adjust the model identification and to improve the accuracy of the simulation in these areas.

The third part is the verification and application of the contact force models and the haptic simulations. For this purpose, different objects are tested and modeled in experiments and simulations, as well as some practical implementations. It is especially emphasized in this thesis to apply the modeling methods in telepresence and medical technology.

During the experiments, several effects are detected, and some experience can be obtained. Since high-fidelity real time haptic simulations are the main requirements in this area, the modeling methods have to be configured appropriately to fit the requirements. For instance, haptic simulations more than $50[Hz]$ can offer users unproblematic haptic sensation during contacts. With $1000[Hz]$ output rate, the smooth manipulations are reasonable expectation. In these cases, the analytical contact force model (ACFM) and the extended Kalman filter are necessary and capable components for the finite element models, especially for nonlinear models, however the parameters of all the methods have to be constrained and optimized correctly. Furthermore, the adjustments and settings can obviously affect and accelerate the parameter optimization.

A. Appendix

A.1. QR-Decomposition

The QR-decomposition of matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $m \geq n$, is defined as

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}, \quad (\text{A.1})$$

where

$$\begin{aligned} \mathbf{Q} &\in \mathbb{R}^{m \times m} && \text{orthogonal matrix} \\ \mathbf{R} &\in \mathbb{R}^{m \times n} && \text{upper triangular} \\ \text{rank}(\mathbf{A}) &= n && \text{full column rank} \end{aligned} \quad (\text{A.2})$$

There are several methods to execute the QR-decomposition, for instance the Householder transformation, the Gram-Schmidt method and the Givens transformation [48]. The Householder transformation is implemented in this thesis, with the Householder matrix $\mathbf{H} \in \mathbb{R}^{m \times m}$, defined as

$$\mathbf{H} = \mathbf{I} + \frac{2\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|_2^2}, \quad (\text{A.3})$$

and the Householder vector \mathbf{v} . Defining \mathbf{v} as follows,

$$\mathbf{v} = \mathbf{a} \pm \|\mathbf{a}\| \mathbf{e}_1, \quad (\text{A.4})$$

in which vector \mathbf{a} is the first column of matrix \mathbf{A} , and

$$\mathbf{e}_1 = [1 \ 0 \ \dots \ 0]^T. \quad (\text{A.5})$$

Then, the matrix \mathbf{H}_1 can be calculated according to (A.3). Multiplying \mathbf{H}_1 with \mathbf{a} , it yields

$$\mathbf{H}_1 \cdot \mathbf{a} = r_{11} \cdot \mathbf{e}_1. \quad (\text{A.6})$$

Thus, the following result can be obtained,

$$\mathbf{H}_1 \cdot \mathbf{A} = \mathbf{H}_1 \cdot \mathbf{A}_{(1)} = \begin{bmatrix} r_{11} & r_{12} \cdots r_{1n} \\ \mathbf{0} & \mathbf{A}_{(2)} \end{bmatrix}. \quad (\text{A.7})$$

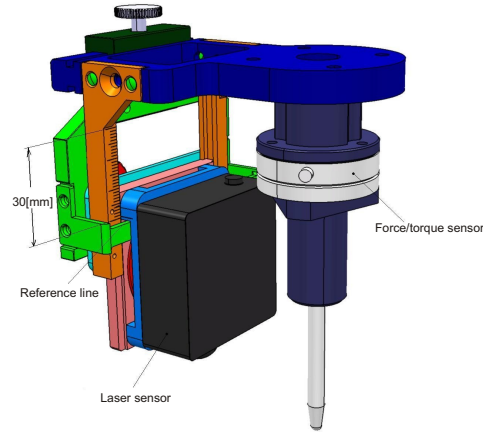


Figure A.1.: The structure of the robotic end effector with the montage of the sensors and the bar. The laser sensor is at the lowest position.

Applying the algorithm recursively, all the matrices $\{H_i : i = 1, \dots, n\}$ can be determined, and the decomposition can be obtained as follows,

$$\begin{aligned} \mathbf{Q} &= \prod_{i=1}^n H_i \\ \mathbf{R} &= \mathbf{Q}^T \mathbf{A} = \left(\prod_{i=n}^n H_i \right) \cdot \mathbf{A} \end{aligned} \tag{A.8}$$

A.2. Robotic End Effector

The robotic end effector of the experimental robot contains the force/torque sensor, the laser distance sensor and a bar for operations. The central line of the bar is parallel with the emergent laser ray, and the distance is constant. Furthermore, it is also coaxial with the force/torque sensor and the flange of the last robot link, so that the geometric relations of the sensors and the actuators can be simplified. The effector is illustrated in Fig. A.1

The position of the laser sensor is adjustable in the direction of the emergent ray. By this means, the measurement range of the laser sensor can be extended, and some complex surfaces may also be measured, Fig. A.2. The adjustable range is $0 \sim 30[mm]$.

Moreover, it should be noticed that the bar is made of aluminum. That is during contacts with soft materials, such as the soft animal tissues, it is reasonable to be regarded as a rigid body.

A.3. Material Value in Section 6.2

In Section 6.2, six and ten different experiment settings are used respectively for the linear elastic materials and the viscoelastic materials in the virtual reality experiment, to define virtual objects and contact situations. For the linear elastic models, merely

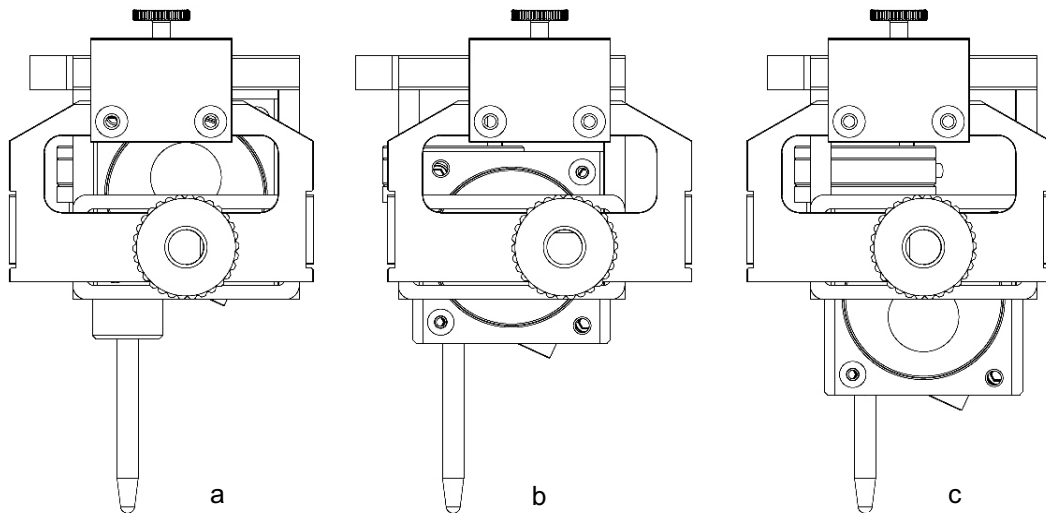


Figure A.2.: The design of the robotic end effector. The position of the laser sensor is adjustable, for instance the upper (a), middle (b) and lower positions (c).

Table A.1.: Material values of the virtual linear elastic objects.

Index	Young's modulus (E) [N/m^2]	Poisson's ratio (ν)
1	10000	0.3
2	20000	0.3
3	30000	0.3
4	50000	0.3
5	60000	0.3
6	70000	0.3

the material parameters are necessary, whereas for the viscoelastic models, the material characteristics as well as the boundary displacement are specified. The values of the parameters and settings are listed in Tab. A.1 and Tab. A.2, respectively.

Table A.2.: Material values and settings of the virtual viscoelastic objects.

Index	E_s [N/m^2]	ν	Maxwell elements	n_M	Relaxation amplitude E_i [N/m^2]	Relaxation time τ_i [s]	Boundary displacement rate \dot{m}/s
1	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.05$
2	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = 0.02 \sin(t)$
3	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = 0.02 \sin(2t)$
4	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.025$
5	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.1$
6	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.1$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.01$
7	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.01$ $\tau_2 = 0.02$	$\Delta \dot{s} = \pm 0.05$
8	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.05$ $\tau_2 = 0.1$	$\Delta \dot{s} = \pm 0.05$
9	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.01$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.05$
10	20000	0.3	2		$E_1 = 10000$ $E_2 = 6000$	$\tau_1 = 0.2$ $\tau_2 = 0.2$	$\Delta \dot{s} = \pm 0.05$

Bibliography

- [1] ABE, T. ; MAETA, K. ; HIGASHINO, H.: Elastic-Plastic Deformation of Inhomogeneous Material and Parametric Description of Deformation Behaviour. In: *JSME International Journal Series A Mechanics and Engineering* 37 (1994), Nr. 4, S. 412–420
- [2] AMENTA, N. ; CHOI, S. ; KOLLURI, R. K.: The power crust. In: *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, 2001, S. 249–266
- [3] ANGOSTO, J. M. ; ELVIRA-RENDUELES, B. ; BAYO, J. ; MORENO, J. ; VERGARA, N. ; MORENO-CLAVEL, J. ; MORENO-GRAU, S.: Wind classification through cluster analysis for the development of predictive statistical models on atmospheric pollution. In: *Advance in air pollution series* 11 (2002), December, S. 635–644
- [4] AREZZO, J. C. ; SCHAUMBURG, H. H. ; SPENCER, P. S.: Structure and Function of the Somatosensory System: A Neurotoxicological Perspective. In: *Environmental Health Perspectives* 44 (1982), S. 23–30
- [5] ARULAMPALAM, M. S. ; MASKELL, S. ; GORDON, N. ; CLAPP, T.: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. In: *IEEE Transactions on Signal Processing* 50 (2002), February, Nr. 2
- [6] ASTLEY, O. R. ; V., Hayward: Real-time Finite-elements Simulation of General Visco-elastic Materials for Haptic Presentation. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems, IROS '97*, 1997, S. 52–57
- [7] ATI INDUSTRIAL AUTOMATION (Hrsg.): *Multi-Axis Force/Torque Sensor System Installation and Operation Manual*. North Carolina USA: ATI Industrial Automation, 2005
- [8] ATKINSON, K. E.: *An Introduction to Numerical Analysis*. 2. New York : John Wiley & Sons, 1989
- [9] AURENHAMMER, F.: Power Diagrams: Properties, Algorithms and Applications. In: *SIAM Journal on Computing* 16 (1987), February, Nr. 1, S. 78–96
- [10] AURENHAMMER, F.: Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure. In: *ACM Computing Surveys* 23 (1991), Nr. 3, S. 345–405

- [11] AWREJCEWICZ, J. ; OLEJNIK, P.: Analysis of Dynamic Systems with Various Friction Laws. In: *Applied Mechanics Reviews* 58 (2005), November, Nr. 6, S. 389–411
- [12] BASDOGAN, C. ; DE, S. ; KIM, J. ; MUNIYANDI, M. ; KIM, H. ; SRINIVASAN, M. A.: Haptics in Minimally Invasive Surgical Simulation and Training. In: *IEEE Computer Graphics and Applications* (2004), April, S. 56–64
- [13] BATES, D. M. ; WATTS, D. G.: *Nonlinear Regression Analysis and Its Applications*. New York : John Wiley & Sons, 1988
- [14] BECHSTEDT, S. ; HOWARD, J.: Hearing Mechanics: A Fly in Your Ear. In: *Current Biology* 18 (2008), September, Nr. 18, S. R869–R870
- [15] BECKER, S. ; PLUMBLEY, M.: Unsupervised Neural Network Learning Procedures for Feature Extraction and Classification. In: *Journal of Applied Intelligence* 6 (1996), S. 185–203
- [16] BERG, M. ; M., Kreveld ; OVERMARS, M. ; SCHWARZKOPF, O.: *Computation Geometry: Algorithms and Applications*. 2. Berlin : Springer, 2000
- [17] BERGER, J. O.: *Statistical decision theory and Bayesian analysis*. New York : Springer, 1985 (Springer series in statistics)
- [18] BLENDER FOUNDATION (Hrsg.): *Blender Tutorial and Halp*. Blender Foundation, 2009. – <http://www.blender.org/>
- [19] BRODERICK III, C. R. ; CHEN, Q.: A Simple Interface to Computational Fluid Dynamics Programs for Building Environment Simulations. In: *Indoor Built Environ* 9 (2000), Nr. 6, S. 317–324
- [20] BURNS, M.: *The Stl Format A Tutorial on Principal Components Analysis*. 1989. – Reprinted from Section 6.5 of Automated Fabrication, Technical source: StereoLithography Onterface Specification, 3D Systems, Inc., <http://www.ennex.com/fabbers/StL.asp>
- [21] CAUNA, N. ; ROSS, L. L.: The Fine Structure of Meissner’s Touch Corpuscles of Human Finger. In: *The Journal of Biophysical and Biochemical Cytology* 8 (1960), October, Nr. 2, S. 467–482
- [22] CHEN, T.: Determining a Prony Series for a Viscoelastic Material From Time Varying Strain Data. In: *NASA STI Report Series* (2000), May, Nr. TM-2000-210123
- [23] CHEW, L. P.: Constrained Delaunay Triangulations. In: *Proceedings of the third annual symposium on Computational geometry (SCG ’87)*. Waterloo, Canada : ACM, 1987, S. 215–222
- [24] CHILDS, G. V.: *Microanatomy Web Atlas: Skin*. 1998. – Documentation for Call Biology Graduate Program, <http://cellbio.utmb.edu/microanatomy>

- [25] CHOI, T. D. ; ESLINGER, O. J. ; GILMORE, P. A. ; KELLEY, C. T. ; PATRICK, H. A.: *User's Guide to IFFCO*, May 2001. – <http://www4.ncsu.edu/ ctk/iffco.html>
- [26] CHRISTENSEN, R. M.: *Theory of Viscoelasticity*. 2. Mineola, N.Y. : Dover Publications, 2003. – Originally published: New York: Academic Press, 1982
- [27] CLARKE, S. ; SCHILLHUBER, G. ; ZAEH, M. ; ULBRICH, H.: The Effect of Simulated Inertial and Force Prediction on Delayed Telepresence. In: *Presence* 16 (2007), October, Nr. 5, S. 543–558
- [28] COSTA, K. D. ; HOLMES, J. W. ; MCCULLOCH, A. D.: Modelling cardiac mechanical properties in three dimensions. In: *Philosophical Transactions of the Royal Society A* 359 (2001), June, Nr. 1783, S. 1233–1250
- [29] DALLAS SEMICONDUCTOR CORPORATION (Hrsg.): *DS18S20 High Precision 1-Wire Digital Thermometer*. : Dallas Semiconductor Corporation, 1995. – www.dalsemi.com
- [30] DEMMEL, J.: LAPACK: A portable linear algebra library for supercomputers. In: *Proceedings of the 1989 IEEE Control Systems Society Workshop on Computer-Aided Control System Design*, 1989
- [31] DiMAIO, S. P. ; SALCUDEAN, S. E.: Needle Insertion Modeling and Simulation. In: *IEEE Transactions on Robotics and Automation* 19 (2003), October, Nr. 5, S. 864–875
- [32] DOUCET, A. ; S., Godsill ; ANDRIEU, C.: On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. In: *Statistics and Computing* 10 (2000), Nr. 3, S. 197–208
- [33] DUBIN, M. W.: *How the Brain Works*. 1. Blackwell Science, Inc., 2002
- [34] ENGELKE, R.: *Modellierung und Optimierung von Robotern mit einseitigen Bindungen und lokalen Verspannungen*. München, Technische Universität München, Dissertation, 2008
- [35] ERICKSON, D. ; WEBER, M. ; SHARF, I.: Contact Stiffness and Damping Estimation for Robotic Systems. In: *The International Journal of Robotics Research* 22 (2003), January, Nr. 1, S. 41–57
- [36] ESD ELECTRONIC SYSTEM DESIGN GMBH (Hrsg.): *CAN-PCI/331 PCI-CAN-Interface Hardware-Installation und technische Daten*. 1. : Esd electronic system design gmbh, 2001. – www.esd-electronics.com
- [37] ESD ELECTRONICS INC. (Hrsg.): *CAN-API Part 1: Function Description Software Manual*. 2. : Esd electronics Inc., 2006. – www.esd-electronics.com
- [38] ESD ELECTRONICS INC. (Hrsg.): *CAN-API Part 2: Installation Guide Software Manual*. 2. : Esd electronics Inc., 2006. – www.esd-electronics.com

- [39] FEATHERSTONE, R. ; ORIN, D.: Robot Dynamics: Equations and Algorithms. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2000, S. 826–834
- [40] FORTUNE, S.: A Sweepline Algorithm for Voronoi Diagrams. In: *Proceedings of the Second Annual Symposium on Computational Geometry*, Association for Computing Machinery (ACM), 1986, S. 313–322
- [41] FORTUNE, S.: Voronoi Diagrams and Delaunay Triangulations. In: *Handbook of discrete and computational geometry* (1997), August, S. 377–388
- [42] FÖRG, M.: *Mehrkörpersysteme mit mangelwertigen Kraftgesetzen - Theorie und Numerik*. München, Technische Universität München, Dissertation, 2007
- [43] FUNG, Y. C.: *Biomechanics: Mechanical Properties of Living Tissues*. 2. New York : Springer, 1993
- [44] GETZNER WERKSTOFFE GMBH (Hrsg.): *Comprehensive solutions with elastic materials*. : Getzner Werkstoffe Gmbh, 2008. – www.getzner.com
- [45] GILMAN, S.: Joint Position Sense and Vibration Sense: Anatomical Organisation and Assessment. In: *Journal of Neurology, Neurosurgery and Psychiatry* 73 (2002), S. 473–477
- [46] GILMORE, P. ; KELLEY, C. T.: An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima. In: *SIAM Journal on Optimization* 5 (1995), May, Nr. 2, S. 269–285
- [47] GILMORE, P. ; KELLEY, C. T. ; MILLER, C. T. ; WILLAMS, G. A.: Implicit Filtering and Optimal Design Problems. In: *Proceedings of the workshop on optimal design and control*, 1995, S. 159–176
- [48] GOLUB, G. H. ; VAN LOAN, C. F.: *Matrix Computations*. 3. Baltimore : Johns Hopkins Univ. Press, 1996
- [49] GURTIN, M. E.: The Linear Theory of Elasticity. In: *Encyclopedia of Physics, Mechanics Solids II* Bd. VIa-2. Springer Verlag, 1972, S. 1–295
- [50] HAGN, U. ; NICKL, M. ; JÖRG, S. ; PASSIG, G. ; BAHLS, T. ; NOTHHELFER, A. ; HACKER, F. ; LE-TIEN, L. ; ALBU-SCHÄFFER, A. ; KONIETSCHKE, R. ; GREBENSTEIN, M. ; WARPUP, R. ; HASLINGER, M. R. F. R. Frommberger ; HIRZINGER, G.: The DLR MIRO: a versatile lightweight robot for surgical application. In: *Industrial Robot: An International Journal* 35 (2008), Nr. 4, S. 324–336
- [51] HASHASH, Y. M. A. ; JUNG, S. ; GHABOUSSI, J.: Numerical implementation of a neural network based material model in finite element Analysis. In: *International Journal for Numerical Methods in Engineering* 59 (2004), S. 989–1005

- [52] HAUPT, P.: *Continuum Mechanics and Theory of Materials*. Berlin : Springer, 2000
- [53] HAYWARD, V. ; ASTLEY, O. R. ; GRANT, D. ; ROBLES-DE-LA-TORRE, G.: Haptic interface and devices. In: *Sensor Review* 24 (2004), Nr. 1, S. 16–29
- [54] HAZLEWOOD, G.: Approximating Constrained Tetrahedrizations. In: *Computer Aided Geometric Design* 10 (1993), Nr. 1, S. 67–87
- [55] HESTENES, M. R. ; STIEFEL, E.: Method of Conjugate Gradients for Solving Linear Systems. In: *Journal of Research of the National Bureau of Standards* 49 (1952), December, Nr. 6, S. 48–55
- [56] HONJO, T. ; LIM, E. M.: Visualization of Forest Landscapes by VRML. In: *Proceedings of the Third Asian Conference for Information Technology in Agriculture*. Beijing, China : China Agricultural Sciencetech Press, 2002, S. 26–28
- [57] HORST, R. ; PARDALOS, P. M. ; THOAI, N. V.: *Introduction to Global Optimization: Nonconvex Optimization and Its Application*. 2. Netherlands : Kluwer Academic Publishers, 2000
- [58] HORST, R. ; TUY, H.: *Global Optimization: Deterministic Approaches*. 2. Berlin, Heidelberg, Germany : Springer-Verlag, 1993
- [59] HOTELLING, H.: Analysis of a Complex of Statistical Variables into Principal Components. In: *Journal of Educational Psychology* 24 (1933), S. 417–441 and 498–520
- [60] HUSTY, M. ; KARGER, A. ; SACHS, H. ; STEIHILPER, W.: *Kinematik und Robotik*. 1. Berlin : Spring, 1997
- [61] JAMES, D. L. ; PAI, D. K.: ArtDefo Accurate Real Time Deformable Objects. In: *SIGGRAPH Computer Graphics*, 1999, S. 65–72
- [62] KALISKE, M. ; ROTHERT, H.: Formulation and Implementation of Three-Dimensional Viscoelasticity at Small and Finite Strains. In: *Computational Mechanics* 19 (1997), S. 228–239
- [63] KALLMANN, M. ; BIERI, H. ; THALMANN, D.: Fully Dynamic Constrained Delaunay Triangulation. In: *Geometric Modelling for Scientific Visualization* 73 (2003), S. 241–257
- [64] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME-Journal of Basic Engineering* (1960), Nr. 82, S. 35–45
- [65] KITWARE INC. (Hrsg.): *VTK Manual: Documentation*. 5. : Kitware Inc., 2008. – www.vtk.org/doc/release/5.2/html

- [66] KOCH, R.: 3-D surface reconstruction from stereoscopic image sequence. In: *Proceedings of the Fifth International Conference on Computer Vision*, 1995, S. 109–114
- [67] KOCH, R. ; POLLEFEYS, M. ; VAN GOOL, L.: Realistic surface reconstruction of 3D scenes from uncalibrated image sequence. In: *The Journal of Visualization and Computer Animation* 11 (2000), S. 115–127
- [68] LI, T. ; ZHU, S. ; MITSUNORI, O.: Using Discriminant Analysis for Multi-class Classification. In: *Proceedings of the Third IEEE International Conference on Data Mining*, IEEE Computer Society, 2003
- [69] MACIEL, A. ; BOULIC, R. ; THALMANN, D.: Deformable Tissue Parameterized by Properties of Real Biological Tissue. In: *IEEE Computer Graphics and Applications* 2673/-1 / 2003 (2003), S. 74–87
- [70] MACIEL, A. ; SARNI, S. ; BUCHWALDER, O. ; BOULIC, R. ; THALMANN, D.: Multi-Finger Haptic Rendering of Deformable Objects. In: *Proceedings of Eurographics Symposium on Virtual Environments*, The Eurographics Association, 2004, S. 105–111
- [71] MACKERLE, J.: Rubber and rubber-like materials, finite-element analyses and simulations, an addendum: a bibliography (1997-2003). In: *Modelling and Simulation in Materials Science and Engineering*
- [72] MAGENUS, K. ; MÜLLER, H. H.: *Grundlagen der Technischen Mechanik*. 6. Stuttgart : B. G. Teubner, 1990
- [73] MAHVASH, M. ; HAYWARD, V.: Haptic Simulation of a Tool in Contact with a Nonlinear Deformable Body. In: *Surgery Simulation and Soft Tissue Modeling* 2673/2003 (2003), S. 311–320
- [74] MAHVASH, M. ; HAYWARD, V.: High-Fidelity Haptic Synthesis of Contact with Deformable Bodies. In: *IEEE Computer Graphics and Applications* (2004), April, S. 48–55
- [75] MARDIA, K. V. ; KENT, J. T. ; BIBBY, J. M.: *Multivariate Analysis*. London : Academic Press, 1989
- [76] THE MATHWORKS, INC. (Hrsg.): *VRML97 Functional Specification and VRML97 External Authoring Interface (EAI)*. The MathWorks, Inc., 2003. – ISO/IEC 14772 standard document
- [77] THE MATHWORKS, INC. (Hrsg.): *The MathWork: Accelerating the pace of engineering and science*. The MathWorks, Inc., 2009. – Matlab Documentation, www.mathworks.com
- [78] MAYBECK, P. S.: *Stochastic Models, Estimation, and Control*. 6. New York : Academic Press, 1979

- [79] MAYER, H. ; NAGY, I. ; KNOLL, A. ; BRAUN, E. U. ; BAUERNSCHMITT, R. ; LANGE, R.: Haptic Feedback in a Telepresence System for Endoscopic Heart Surgery. In: *Presence Teleoperators and Virtual Environments* 16 (2007), October, Nr. 5, S. 459–470
- [80] MERWE, R. van d. ; WAN, E. A.: Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. In: *Proceedings of the Workshop on Advances in Machine Learning*. Montreal, Canada, June 2003
- [81] MICRO-EPSILON MESSTECHNIK GMBH AND CO KG (Hrsg.): *OptoNCDT 1605 Berührungslose optische Wegmeßsysteme Bedienungsanleitung*. Ortenburg: Micro-Epsilon Messtechnik GmbH and Co KG
- [82] MILLER, G. L. ; TALMOR, D. ; TENG, S. ; WALKINGTON, N.: A Delaunay Based Numerical Method for Three Dimensions: generation, formulation, and partition. In: *Proceeding of 27th Annual ACM Symposium on the Theory of Computing*, ACM, 1995, S. 683–692
- [83] MILLER, G. L. ; TALMOR, D. ; TENG, S. ; WALKINGTON, N. ; WANG, H.: Control Volume Meshes Using Sphere Packing: Generation, Refinement and Coarsening. In: *Proceeding of 5th International Meshing Roundtable*, 1996, S. 47–61
- [84] MOLA, F. ; SICILIANO, R.: Discriminant Analysis and Factorial Multiple Splits in Recursive Partitioning for Data Mining. In: *Lecture Notes in Computer Science* 2364/2002 (2002), S. 7–25
- [85] MURPHY, M. ; MOUNT, D. M. ; GABLE, C. W.: A Point-Placement Strategy for Conforming Delaunay Tetrahedralization. In: *Proceedings of the Eleventh Annual Symposium on Discrete Algorithms*, Association for Computing Machinery, 2000, S. 67–74
- [86] NELSON, R. J.: *The Somatosensory System: Deciphering the Brain's Own Body Image*. 1. Florida : CRC Press LLC, 2002
- [87] NEUMANN, L.: *Optimierung von CVT-Ketten*. München, Technische Universität München, Dissertation, 2007
- [88] NOCEDAL, J. ; WRIGHT, S. J.: *Numerical Optimization*. New York : Springer, 1999
- [89] NOWACKI, W.: *Theorie des Kriechens lineare Viskoelastizität*. 1. Wien : Franz Deuticke, 1965
- [90] OKUDA, H. ; YOSHIMURA, S. ; YAGAWA, G. ; MATSUDA, A.: Neural network-based parameter estimation for non-linear finite element Analyses. In: *Engineering Computation* 15 (1998), Nr. 1, S. 103–138
- [91] PIEPER, D. L. ; ROTH, B.: The Kinematics of Manipulators Under Computer Control. In: *Proceedings of the 2nd International Congress on Theory of Machines*, 1969, S. 159–169

- [92] POSTEL, J.: *User Datagram Protocol*, August 1980. – RFC 768, standard
- [93] POSTEL, J.: *Transmission Control Protocol*. DARPA, September 1981. – RFC 793, DARPA internet program, protocol specification
- [94] RAGNO, G. ; DE LUCA, M. ; LOELE, G.: An Application of Cluster Analysis and Multivariate Classification Methods to Spring Water Monitoring Data. In: *MicroChemical Journal* 87 (2007), December, Nr. 2, S. 119–127
- [95] RAIKO, T. ; ILIN, A. ; KARHUNEN, J.: Principal Component Analysis for Large Scale Problems with Lots of Missing Values. In: *Proceedings of the 18th European Conference on Machine Learning*. Warsaw, Poland : Springer Verlag, 2007, S. 691–698
- [96] ROBLES-DE-LA-TORRE, G.: The Importance of the Sense of Touch in Virtual and Real Environments. In: *IEEE MultiMedia, Special issue on Haptic User Interfaces for Multimedia Systems* 13 (2006), Nr. 3, S. 24–30
- [97] ROHRMEIER, M.: Interactive Simulation Using Virtual Systems: Web Based Robot Simulation Using VRML. In: *Proceedings of the 32nd Conference on Winter Simulation*, Society for Computer Simulation International, 2000, S. 1525–1528
- [98] ROJAS, R.: *Neural Networks A systematic Introduction*. Berlin : Springer, 1996
- [99] RUURDA, J. P. ; DRAAISMA, W. A. ; HILLEGERSBERG, R. van ; RINKES, I. H. M. B. ; GOOSZEN, H. G. ; JANSSEN, L. W. M. ; SIMMERMACHER, R. K. J. ; BROEDERS, I. A. M. J.: Robot-Assisted Endoscopic Surgery: A Four-Year Single-Center Experience. In: *Digestive Surgery* 22 (2005), September, Nr. 5, S. 313–320
- [100] SALISBURY, K. ; CONTI, F. ; BARBAGLI, F.: Haptic Rendering: Introductory Concepts. In: *Computer Graphics and Applications* 24 (1993), March/April, Nr. 2, S. 24–32
- [101] SCHILLHUBER, G.: *Roboter-gestützte Modellidentifikation und Simulation von deformierbaren Körpern für haptische Anwendungen*. München, Technische Universität München, Dissertation, 2009
- [102] SCHWARTZ, J.-M. ; DENNINGER, M. ; RANCOURT, D. ; MOISAN, C. ; LAURENDEAU, D.: Modelling Liver Tissue Properties Using a Non-Linear Visco-Elastic Model for Surgery Simulation. In: *Medical Image Analysis* 9 (2005), December, S. 103–112
- [103] SEDEF, M. ; SAMUR, E. ; BASDOGAN, C: Real-Time Finite Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data. In: *IEEE Computer Graphics and Applications* (2006), November/December, Nr. 0272-1716/06, S. 58–68

- [104] SEIFERT, J. W.: *Data Mining: An Overview*. 2004. – CRS Report for Congress, Congressional Research Service, The Library of Congress
- [105] SENSABLE TECHNOLOGIES, INC. (Hrsg.): *Specifications for the Phantom Desktop and Phantom Omni Haptic devices*. 01. SensAble Technologies, Inc., January 2009. – www.sensable.com/support-overview.htm.
- [106] SHEWCHUK, J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: *Applied Computational Geometry: Towards Geometric Engineering*, 1996, S. 203–222
- [107] SHEWCHUK, J. R.: Tetrahedral mesh generation by Delaunay refinement. In: *Proceedings of the fourteenth annual symposium on Computational geometry*, 1998, S. 86–95
- [108] SHEWCHUK, J. R.: Constrained Delaunay Tetrahedralizations and Provably Good Boundary Recovery. In: *Eleventh International Meshing Roundtable*, 2002, S. 193–204
- [109] SHEWCHUK, J. R.: General-Dimensional Constrained Delaunay and Constrained Regular Triangulations I: Combinatorial Properties. In: *Discrete and Computational Geometry* 39 (2008), Nr. 1, S. 580–637
- [110] SI, H.: *TetGen A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. 1. Berlios, 2006. – tetgen.berlios.de
- [111] SI, H. ; GAERTNER, K.: Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. In: *Proceedings of the 14th International Meshing Roundtable*, 2005, S. 147–163
- [112] SRINIVASAN, M. A. ; BASDOGAN, C.: Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. In: *Computers and Graphics* 21 (1997), Nr. 4, S. 393–404
- [113] SRIVASTAVA, S. ; GUPTA, M. R. ; FRIGYIK, B. A.: Bayesian Quadratic Discriminant Analysis. In: *Journal of Machine Learning Research* (2007), S. 1277–1305
- [114] TERAN, J. ; BLEMKER, S. ; HING, V. N. T. ; FEDKIW, R.: Finite Volume Methods for the Simulation of Skeletal Muscle. In: *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003, S. 68–75
- [115] TESCHNER, M. ; HEIDELBERGER, B. ; MUELLER, M. ; POMERANETS, D. ; GROSS, M.: Optimized Spatial Hashing for Collision Detection of Deformable Objects. In: *Proceedings of Vision, Modeling, Visualization (VMV '97)*, 2003, S. 47–54
- [116] TESCHNER, M. ; KIMMERLE, S. ; HEIDELBERGER, B. ; ZACHMANN, G. ; RAGHUPATHI, L. ; FUHRMANN, A. ; CANI, W. P. ; FAURE, F. ; MAGNENAT-THALMANN, N. ; STRASSER, W. ; VOLINO, P.: Collision Detection for

- Deformable Objects. In: *Eurographics State of the Art Report*, Eurographics Association, 2004, S. 119–139
- [117] TEWARSON, R. P.: *Sparse Matrices*. 1. New York : Academic Press Inc., 1973
- [118] TIMOSHENKO, S. P. ; GOODIER, J. N.: *Theory of Elasticity*. 3. Auckland : McGraw-Hill, 2005
- [119] TRUESDELL, C. ; TOUPIN, R. A.: The Classical Field Theories. In: *Encyclopedia of Physics* Bd. III-3. Springer Verlag, 1960
- [120] VIDAL, R. ; MA, Y. ; SASTRY, S.: Generalized Principal Component Analysis (OPCA). In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), December, Nr. 12, S. 1945–1959
- [121] YUCESOY, C. A. ; KOOPMAN, B. H. ; HUIJING, P. A. ; GROOTENBOER, H. J.: Three-dimensional finite element modeling of skeletal muscle using a two-domain approach: linked fiber-matrix mesh model. In: *Journal of Biomechanics* 35 (2002), S. 1253–1262
- [122] ZACHMANN, G. ; LANGETEPE, E.: Geometric Data Structures for Computer Graphics, ACM Transactions of Graphics, July 2003
- [123] ZARCHAN, P. ; MUSOFF, H.: *Fundamentals of Kalman Filtering: A Practical Approach*. 1. Reston : American Institute of Aeronautics and Astronautics, 2000
- [124] ZHAO, C. ; SCHILLHUBER, G. ; ULBRICH, H.: Force Modeling with Parameter Estimation for Real Time Force Simulation. In: *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Xi'an, CHINA, 2008, S. 764–769
- [125] ZHAO, C. ; ULBRICH, H.: Force Modeling of Inhomogeneous Material Using Unsupervised Learning and Model Identification. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. Bangkok, Thailand, 2009, S. 1319–1324
- [126] ZIENKIEWICZ, O. C. ; TAYLOR, R. L.: *The Finite Element Method for Solid and Structural Mechanics*. 6. Oxford : Elsevier, 2005
- [127] ZIENKIEWICZ, O. C. ; TAYLOR, R. L. ; ZHU, J. Z.: *The Finite Element Its Basis and Fundamentals*. 6. Oxford : Elsevier, 2005