İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

LARGE DEFORMATION OBJECT MODELING USING FINITE ELEMENT METHOD AND PROPER ORTHOGONAL DECOMPOSITION FOR HAPTIC ROBOTS

M.Sc. Thesis by

Yaşar PAÇA, B.Sc.

Department : Mechanical Engineering

Programme: System Dynamics and Control

JUNE 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

LARGE DEFORMATION OBJECT MODELING USING FINITE ELEMENT METHOD AND PROPER ORTHOGONAL DECOMPOSITION FOR HAPTIC ROBOTS

M.Sc. Thesis by
Yaşar PAÇA, B.Sc.
(503051615)

Date of submission : 5 May 2008

Date of defence examination: 9 June 2008

Supervisor (Chairman): Prof. Dr. Ata MUĞAN
Co-Supervisor: Assoc. Prof. Dr. Serhat YEŞİLYURT

Members of the Examining Committee Assist.Prof. Dr. Z. Yağız BAYRAKTAROĞLU

Prof. Dr. Metin GÖKAŞAN

JUNE 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

HİSSEDİCİ ROBOTLAR İÇİN SONLU ELEMANLAR VE UYGUN ORTOGONAL AYRIŞTIRMA METOTLARINI KULLANARAK YÜKSEK DEFORMASYONA SAHİP OBJE MODELLEME

YÜKSEK LİSANS TEZİ

Müh. Yaşar PAÇA
(503051615)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008
Tezin Savunulduğu Tarih : 9 Haziran 2008

Tez Danışmanı :	**Prof. Dr. Ata MUĞAN**
Eş Danışmanı :	**Doç. Dr. Serhat YEŞİLYURT**
Diğer Jüri Üyeleri	**Yrd. Doç. Dr. Z. Yağız BAYRAKTAROĞLU**
	Prof. Dr. Metin GÖKAŞAN

HAZİRAN 2008

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor Prof. Dr. Ata Mugan for let me to study such an interdisciplinary subject that enhanced my vision. I would also like to thank Prof. Dr. Tuncer Toprak for providing the equipments that are used in this thesis work.

This thesis would not have been possible without the support of Research Assistants Cengiz BAYKASOĞLU, Can YABANSU, İsmail Hakkı ŞAHİN, Veli NAKŞİLER and Salih ÖZKESER Special thanks to my friend Eray AKYOL. They have always encouraged and supported this thesis work.

I would also like to thank Prof. Dr. Emin Faruk Keçeci from Izmir Institute of Technology who supplied important discussions and advices without considering the distance.

Also, special thanks to Dr. Erdal Bulgan who is always believed in me. If it would be possible, imitating his devotion to research and kindness were the best pieces of my personality.

Finally, for giving infinite support and patience, I would like to thank my family.

June, 2008 Yaşar Paça

TABLE of CONTENTS

ABBREVIATIONS

BV	: Bounding Volume
BVH	: Bounding Volume Hierarchy
AABB	: Axis Aligned Bounding Box
OBB	: Oriented Bounding Box
K-DOP	: K discrete Oriented Polytope
SVD	: Singular Value Decomposition
KLD	: Karhunen - Loève Decomposition
PCA	: Principle Component Analysis
POD	: Proper Orthogonal Decomposition
FEM	: Finite Element Method
API	: Application Programming Interface
HDAPI	: Haptic Device API
HLAPI	: Haptic Library API
OPENGL	: Open Graphics Library

LIST OF TABLES

Page No

LIST OF FIGURES

LIST OF SYMBOLS

K_C : Critical Stiffness
g_{im} : Force Exerted On Mass i by the Spring Between Masses i and m
u_i : Displacements Along Axes.
ε_{ij} : Strain Tensor
δW : Virtual Work
σ_{ij} : Stress Tensor
Q_i^e : Nodal Forces
$\delta\Delta_i^e$: Virtual Displacements
N_{xx} : Axial Force per Unit Length
M_{xx} : Moment per Unit Length
A_{xx}^e : Extensional Stiffness
B_{xx}^e : Extensional Bending Stiffness
D_{xx}^e : Bending Stiffness
Ψ_{ij} : Lagrange Interpolation Functions
Φ_j : Hermite Cubic Interpolation Functions
N_i : Interpolation Functions in General
K_{ij}^{mn} : Elements of Stiffness Matrix
F_i : Elements of Force Matrix
R : Residual Matrix
T_{ij} : Elements of Tangent Stiffness Matrix
φ : Orthonormal Basis Vectors
c_i : Coefficients of Basis Vectors
$U_{(mxm)}$: Left Singular Matrix
$V_{(nxn)}$: Right Singular Matrix
$S_{(mxn)}$: Singular Value Matrix
σ_i : Singular Values

HİSSEDİCİ ROBOTLAR İÇİN SONLU ELEMANLAR VE UYGUN ORTOGONAL AYRIŞTIRMA METOTLARINI KULLANARAK YÜKSEK DEFORMASYONA SAHİP OBJE MODELLEME

ÖZET

Hissedici uygulamalar mühendislik alanında gün geçtikçe hızla artan bir ilgiye sahiptir. Temel olarak bu sistemlerin amacı kullanıcı ve sanal ortam arasında uygun özelliklere sahip mekanik cihazlar kullanarak kuvvet etkileşimleri kurmaktır. Bu sistemler cerrahi benzetimlerde, fiziksel rehabilitasyon uygulamalarında, bilgisayar destekli tasarımda (CAD), insan hareket analizinde, montaj benzetimlerinde, yaygın olarak kullanılmaktadır. Bütün bu uygulamalardaki temel problem bu sistemlerin yüksek hesaplama hızı ihtiyaçlarıdır. Kullanıcı ve sanal ortam arasında sürekli bir etkileşim oluşturmak için bu kuvvetler yaklaşık olarak 1000 Hz oranında güncellenmelidirler. Bu yüksek oran, fizik temmeli olmayan ya da doğrusal problemlerde ulaşılabilir durumdadır. Ancak fizik temmeli ve doğrusal olmayan benzetimlerde bu oranı karşılayabilmek amacıyla ilave modelleme metotları kullanılmaktadır. Gerçekçi ancak gerçek zaman performansı yüksek modeller elde etmek amacıyla model mertebesi düşürme tekniklerine sık sık başvurulur.

Bu çalışmada, hissedici arabirimler ve hesaplama metotları incelenmiştir. Bu amaçla yüksek deformasyon özelliğine sahip doğrusal olmayan bir kiriş modeli sonlu elemanlar metodu kullanılarak elde edilmiştir ve bu elde edilen model PHANTOM® Premium 6 DOF hissedici arabirimi ile etkileşime geçirilmiştir. Etkileşimi elde etmek amacıyla, kiriş modeli OpenGL kullanılarak görselleştirilmiştir ve cihaza OpenHaptics kütüphanesinin HDAPI fonksiyonları kullanılarak hükmedilmiştir. Düşük mertebeli model elde etmek amacıyla, elde edilen sonlu elemanlar modeline, uygun ortogonal ayrıştırma metodu uygulanmıştır. Her iki modelin gerçek zaman performanslari incelenerek kıyaslama yapılmıştır.

LARGE DEFORMATION OBJECT MODELING USING FINITE ELEMENT METHOD AND PROPER ORTHOGONAL DECOMPOSITION FOR HAPTIC ROBOTS

SUMMARY

Haptic applications have an increasing interest in engineering field. It is basically the process of interfacing the user with the virtual world via force interactions that are created by a proper mechanical device. The haptic systems have wide application areas such as surgical simulations, physical rehabilitation, computer-aided design, human motion analysis, assembly simulation. In all of the applications, the common problem is the high computational demand of the haptic systems. The interaction forces between the user and the virtual world should be updated 1000 Hz rate in order to obtain a continuous feeling of touch. This force rate is admissible for non-physics based applications or linear problems, further approaches are needed for physics based and nonlinear problems. To obtain realistic models, model reduction methods are used to obtain approximate low order but computationally efficient models.

In this study, haptic systems are introduced with investigating haptic interfaces and haptic rendering. In additionally, a large deformation beam problem is formulated and solved using Finite Element Method. Then Proper Orthogonal Decomposition method is applied to the obtained discrete beam model and low order model is obtained. Both of the developed models are integrated with PHANTOM® Premium 6 DOF haptic device and their real time computational performance is discussed. In order to integrate the both models OpenGL is used as a visualization library and OpenHaptics with HDAPI is used to command the haptic robot.

1. INTRODUCTION

The term Haptic comes from the Greek word "*haptesthai"* meaning to touch. It is a general term that is used to refer things related to sense of touch. The science of haptic is investigated by different disciplines such as neurology, psychology and engineering. In the field of engineering, the concept is called as computer haptics and generally it is the process of adding touch interactions between virtual world and the user via force feedback. In haptic systems, the user is able to feel and manipulate the virtual object by a mechanical force reflecting device. Haptic applications have increasing usage areas. They are used as a computer aided design tool and in simulation assembly control applications. One of the most important usage areas for haptic systems is surgery simulation. Tissue cutting or knitting can be modeled for the training purposes. In addition, improving sensory-motor skills of a person for the rehabilitation purpose is another important usage area for the haptic systems. In addition to the implementations in engineering field, sculptures use these systems to form complex curves and surfaces that make them possible to finalize their art.

1.1 General Concept and Literature Review

The origins of the haptic systems are teleoperated robots that are also known as telerobotics. In these systems, two robots are controlled from a specific distance. These robots are called as the master and the slave. The movement of the master robot is somehow transmitted to the slave robot to accomplish a specific task. Early systems have mechanical ties between the master and slave robots. The idea of substituting the slave robot with a virtually modeled system has started the progress in haptic systems [1]. The first application that contains force interactions between the user and virtual model can be seen in the work of Minsky [2]. Textural properties of the virtual model are investigated by using a planar haptic device. In later works, degrees of freedom (DOF) of the haptic systems are increased. 3 DOF systems are designed by Ziles, Sallisbury and Massie[3,4]. Then, the research on haptic is focused on 6 DOF systems [5].

Haptic systems can be grouped according to the type of the virtually modeled object. In some applications, the interaction between the user and virtual object does not produce any deformations. These types of virtual models are called as rigid body models [6, 7]. Rigid body simulations are important for the virtual environments in which collision event exists. In some applications, deformations and related reaction forces are critically important. In surgery simulation applications, trainee interacts with the model and the response is reflected to the user [8]. In cutting or knitting of the tissues, observed deformations add physical realism to the virtual environment [9]. The performance of the overall haptic system is greatly dependent on the complexity of the virtually modeled object and its response. The balance between the accuracy demand and the computational efficiency must be established to obtain smooth interactions. According to research studies, interaction forces between the user and virtual world should be updated at the rate of 1000 Hz in order to obtain a continuous feeling of touch [10]. This force rate is admissible for non-physics based applications or linear problems, further approaches are needed for physics based and nonlinear problems [11,12]. The common approach is to approximate high order model with a low order model that resembles the high order model but in a more computationally efficient way. In some applications, the response of the system is obtained and these solutions are processed to obtain dominant behaviors of the system [13].

1.2 Scope of Study

In this work, a low order model is investigated for a large deformation beam problem using Proper Orthogonal Decomposition (POD) method. In Chapter 2, haptic systems and their components are investigated. The concepts of haptic interfaces and haptic rendering explained. Types of the haptic interfaces and their properties are briefly introduced. Then, commonly used collision detection algorithms are defined. Bounding Volume Hierarchy method is further explained. The part of the haptic rendering algorithm that adds physical reality to the virtual world is the modeling section. The created models may be dependent on geometrical facts or physical relations which are briefly explained and compared with each other. Physically based models are discussed in further details. Finite Element Method based models and particle based models are also examined.

The governing equations of 2D nonlinear beam are obtained and solved using finite element method in Chapter 3. Reduced order model of the beam is also obtained using POD and compared with the finite element model.

In Chapter 4, High and low order beam models are integrated with the Phantom Premium 1.5 High Force 6DOF haptic device and real time simulation algorithms are developed. Integration producers and tools are introduced. Conclusion of the study is discussed in Chapter 5.

2. HAPTIC SYSTEMS

2.1 Haptic Interfaces

Haptic interfaces are mechanical devices that enable touch interactions between the user and virtual models. These devices generate mechanical effects that resembles human kinesthetic and touch which gives a feeling that the user acts as in real environment. Haptic interfaces accomplish this task by changing its mechanical properties under computer control. Computer mouse can be seen as the simplest human-computer interaction device. The computer mouse catches 2D planar motion of the user and sends it to the computer to enable interaction of the user with the software. As depicted in the Figure 2.1, there is no sensory feedback between the mouse and user. This type of interface can be named as passive interface.

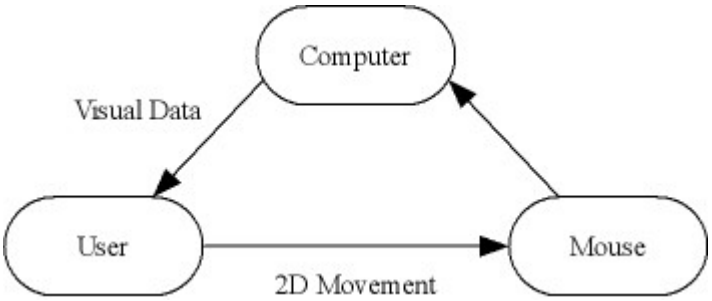

Figure 2.1: Computer Mouse Interactions

However, haptic interfaces change their mechanical properties under computer control to give touch feedback to the user. In general, haptic interfaces catch 3D movements of the user and send it to the computer for physical calculations. Based on these calculations, the mechanical properties of the device are changed to give the touch feeling to the user. The interaction can be shown in Figure 2.2. In animations or movies, human eye cannot detect the delay time between frames. The visual perception system is not fast enough to detect the time interval between two subsequent frames and the user perceives the animations continuously. Similar to visual systems, in order to give a continuous feeling of touch to the user, forces should be updated and transmitted in at a rate of 1000 Hz [10]. This fact shows that

capability of touch sense is more precise. The designed mechanical interface should not impede this sensitivity.

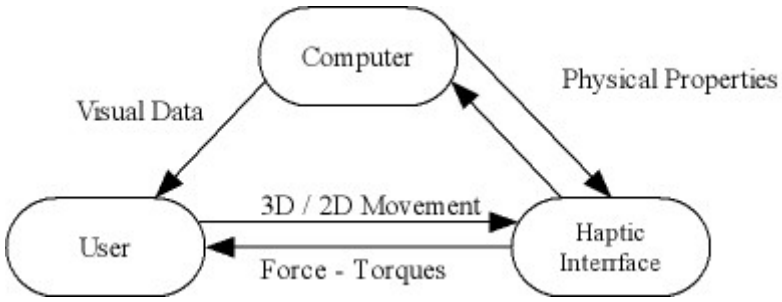

Figure 2.2: Haptic Interface Interactions

Haptic interfaces can be classified differently. A classification can be made according to the grounding locations of the interfaces [10]. Body based interfaces are worn by the user. Exoskeletons and haptic gloves are typical examples of this type of interfaces. Force feedback gloves are used to stimulate single point contact interactions. These types of devices are not capable of producing weight and inertial effects. Burdea [14] developed a pneumatically driven haptic hand shown in Figure 2.3.

Figure 2.3: Haptic Hand

In applications where higher degrees of freedom are needed to obtain a more realistic touch interaction, exoskeleton type haptic interfaces are widely used. These devices provide higher degrees of freedom and larger workspaces. Because of the fact that the user wears these devices, they give a better interaction feeling with the virtual environment. Especially exoskeletons are used to enhance the physical capabilities of the user. Weight lifting capacity, pace speed or jumping limits of the user can be increased by using these devices. In addition, these devices are used for rehabilitation

and motion analysis of human joints [15]. A typical example of a haptic exoskeleton is developed by Yang and Yeo [16] and given in Figure 2.4.

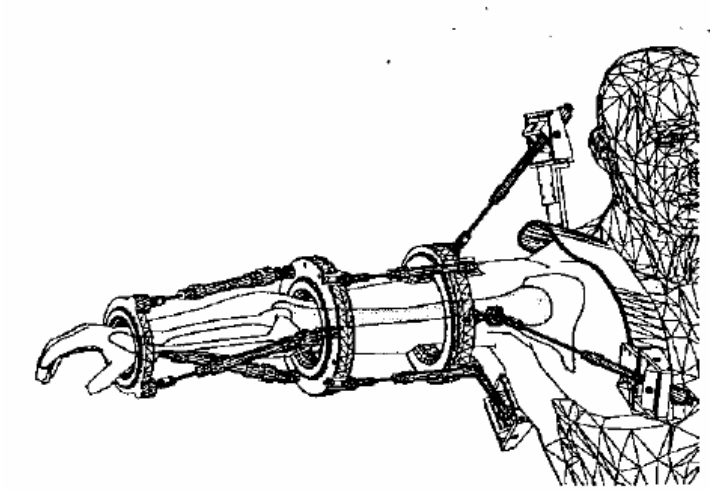

Figure 2.4: Wearable Haptic Device

Although they have large workspace volumes, they have lower applicable force and torque values compared with desktop devices.

In most of the applications, haptic interface is grounded on a fixed place. These devices are similar to robot manipulators and called as desktop type haptic interface. The workspace of these devices is relatively small compared with wearable haptic devices. Due to the fact that they have a fixed ground, the magnitudes of limits on force and torque values are greater than the wearable haptic interfaces.

An attempt to compare these devices is unavailable since their properties depend on the application areas and designed interfaces have different design goals. The research of Hayward and Oliver [17] is an attempt to determine the common expected properties and performance measures of these devices. In general, the expected properties of a haptic interface can be listed as,

- Low back-drive inertia and minimum friction during movement
- Constraints of the device should not impede the movement of the user within the workspace. The user should make his movements as free as possible.
- Symmetrical physical properties such as inertia, friction, stiffness, resonance frequency are needed to supply a stable movement.
- Appropriate range and resolution of position sensing and force reflection
- Ergonomics is needed for a user friendly interface. Uncomfortable interfaces can reduce performance of the device and user.

- Appropriate degrees of freedom are needed depending on the application.
- Bidirectionality must be supplied between the user and interface
- Proper structural response.

In Figure 2.5, the HapticMaster of FCS Control Systems is given. It is a commercial device with remarkable maximum exertable force and torque values.

Figure 2.5: Haptic Master

In Figure 2.6, the desktop device designed in McGill University Haptics Laboratory is depicted. The striking feature of the device is its high resolution. It can measure displacements up to 10 μm.

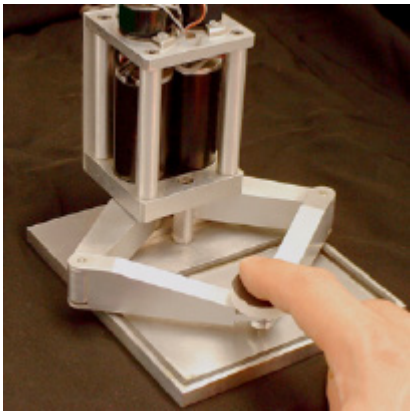

Figure 2.6: Pantograph Type Haptic Interface, McGill University

Haptic interfaces can also be categorized according to the force producing methods. Passive interfaces have controllable breaks. The energy of the device is dissipated by using these breaks. Active interfaces exchange mechanical energy between the user and device by using a feedback function. The actuators of the device behave like a force or position source. Two distinct feedback control methods exist for the active devices. These feedback methods ensure that the calculated force is correctly

transmitted to the user. In impedance controlled haptic devices, positions are measured and force is transferred. The most common example of impedance controlled device is the Phantom Haptic Device of Sensable Technologies. Phantom Omni Desktop device is shown in Figure 2.7,

Figure 2.7: Phantom Desktop Omni Device

Admittance controlled haptic devices measure the applied force and send the position information. The actuators on the device behave like a position source. They are especially used in applications where high force and large workspace are needed. The HapticMaster of FCS Control Systems is an example of such a device. In this thesis, Phantom Premium 1.5, 6 DOF High Force haptic interface is used. The specifications of the device are given in Table 2.1.

2.2 Haptic Rendering

The term haptic rendering is used to refer touch interactions between human and virtual environment via a haptic interface. Haptic rendering gives the user a chance of feeling, touching and manipulating the virtual object by using a haptic interface. As stated before, haptics is a multidisciplinary area including psychology, neuroscience and engineering. In engineering field, haptics is called as "computer haptics" to make the distinction from the other disciplines. The terms computer haptics is closely related to computer graphics. In general, computer graphics deal with the generating only visual objects. However, the computer haptics does not only deal with the visual object alone, it is also interested in the physics of the virtual object being modeled. Haptic rendering algorithm is responsible for calculating the interaction forces and the response of the model like deformations. Haptic rendering is mainly composed of 3 important subdivisions as shown in the Figure 2.8.

Table 2.1: Specifications of Phantom Premium 1.5, 6 DOF, High Force

Workspace	Translational(mm)	381W x 267H x 191D
	Rotational(degrees) (Yaw/Pitch/Roll)	297/260/335
Nominal Position Resolution	Translational(mm)	0.007
	Rotational(degrees) (Yaw/Pitch/Roll)	0.0023/0.0023/0.008
Backdrive Friction	0.2 N	
Max. exertable force and torque with nominal position	Translational(N)	37.5
	Rotational(mNm) (Yaw/Pitch/Roll)	(515/515/170)
Con. exertable force and torque with nominal position	Translational(N)	6.2
	Rotational(mNm) (Yaw/Pitch/Roll)	(27/27/7)
Stiffness	3.5 N mm^{-1}	
Inertia (at tip)	<210 g	
Force Feedback	x, y, z, Tx, Ty, Tz	
Position Sensing	x, y, z, roll, pitch, yaw	

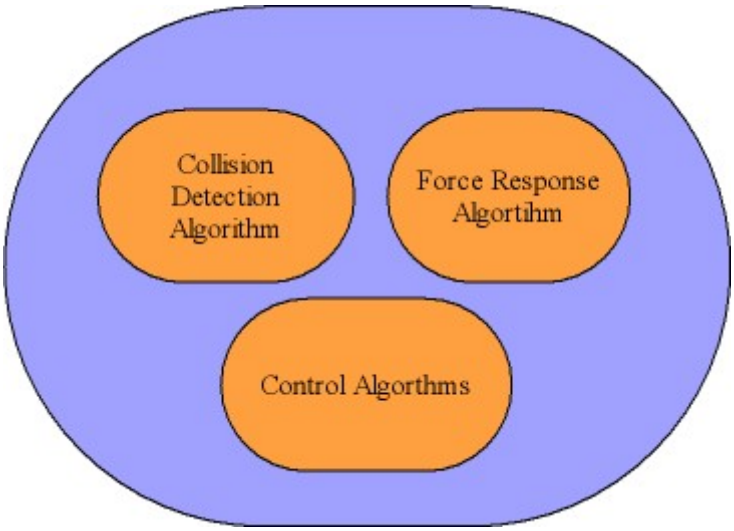

Figure 2.8: Haptic Rendering in General

In the following sections, each component of haptic rendering concept is explained. All the three algorithms run depending on each other. The interrelations of these algorithms will be explained in the following sections.

2.2.1 Collision Detection

Collision detection algorithms are responsible for controlling whether or not the virtual objects share the same location at any time of simulation.

.

The haptic interfaces are represented by a virtual object called *avatar* in virtual environment. This can be thought as the cursor of a computer mouse. Movements of the user are transformed to the screen coordinates that enable the user to interact with computer programs. In haptic applications, movements are not restricted to 2D planer transformations. 3D transformations of the interface are traced by the movements of the avatar on the virtual world namely on screen. The avatar in virtual environment may be a single point or it may have some complex geometry such as representation of surgery tools. The algorithm used to detect the collision event depends on the application. Single point collision detection algorithms are usually used for stimulating tool tip interactions. Multi point contact algorithms have wider usage areas. They are used for multi-body object interactions such as surgery tool implementations [8,9].

According to requirements of the applications, appropriate collision detection algorithm must be selected. In some cases, the detection of the collision is simply adequate without any additional information. In such cases Boolean type collision detection algorithms would be appropriate. However, in applications where realistic response effects are needed, only the information of collision event is not adequate. Additional information such as penetration distance, contact points, contact normal, contact area are needed to produce more realistic responses. This type of algorithms is referred as “enumerative collision detection algorithms” [18]. Collision detection algorithms can be classified depending on the geometrical primitives that are used to build models in virtual environment [19] and typical primitives are given in Figure 2.9.

In this thesis, polygonal models are used and non-polygonal models are not considered. Detailed description of non-polygonal models can be found in [19]. Polygonal models are most widely used objects in 3D computational geometry. If the polygons are not geometrically connected, they are called as “polygon soups”. In the case that polygons define a closed surface, it is called as “proper solid”. The inner and outer regions of these solids are well defined which is a desired property in collision detection algorithms.

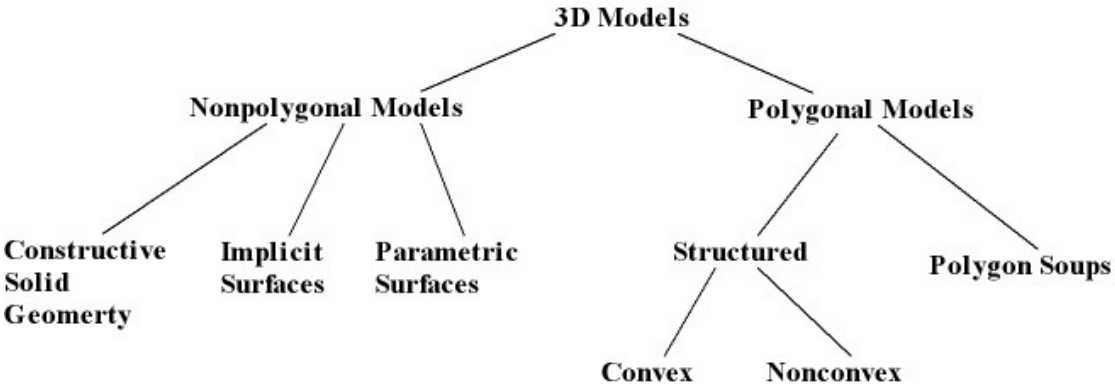

Figure 2.9: Types of 3D Representations**.**

In some applications, the equations of motions of the objects are readily available in a closed form of functions. In this case, the collision detection event can be predetermined by solving the closed form equations simultaneously as in Figure 2.10.

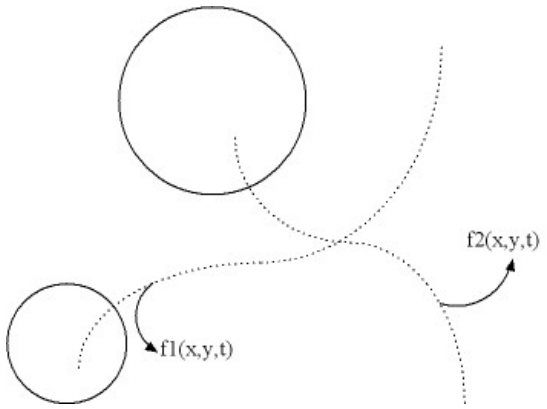

Figure 2.10: Closed-Form Equations Used For Collision Predetermination

The functions describing the motion are not present as in most applications but in need of only the instantaneous locations of the objects in virtual scene between time steps of the simulation procedure. In these applications, instead of determining the track of objects to catch the collision effect, the minimum distances between objects are investigated. If the minimum distance is below a given threshold value, the collision event is determined. The important point with these algorithms is the time steps used in the simulations. If the time step is too small, the overall performance of the operation greatly reduces. In the case of fast moving objects and large time steps in the simulation, collision event can be missed between two subsequent steps. The appropriate balance between the performance and simulation time must be established.

In many applications, the object primitives such as edges, vertices or faces are used for detection and query purposes. In some applications, a primitive called “ray” is used [20]. The defining function of the ray can be represented as the following formulation [21]. Intersection of the ray with an object is searched in the scene.

$$r(t) = m + td \qquad t \subset [-\infty, +\infty] \tag{2.1}$$

m : Origin of the ray

t : Time parameter

d : Unit vector along ray.

Collision detection algorithms are widely used and have been studied for various purposes. They are used in simulated based design, tolerance verification, engineering analysis, video games and animation and motion planning. In some of these applications, the virtual geometries under consideration are quite complex. In particular, when the real-time performance is inevitable for the application, this complexity becomes a burden. To increase real-time performance of the algorithms, complex objects and scenes are subdivided into smaller but more definable geometries with primitive shapes. To this end, Bounding Volume Hierarchy (BVH) methods are a frequently used method in collision detection and proximity queries. Bounding Volumes (BV) is used to represent geometric primitives. Complex geometries are approximated by simpler geometries. These algorithms are based on the idea of divide and conquer principle. Typical examples of bounding volumes are axis aligned bounding boxes (AABB), spheres, oriented bounding boxes (OBB) and k-discrete oriented boxes (k-DOP). AABB are one of the simplest bounding geometry as shown in Figure 2.11. Although the bounding volume is quite simple, empty space not covered by the inner geometry should be considered.

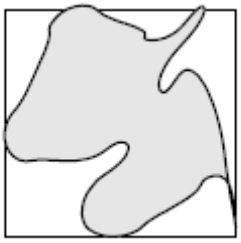

Figure 2.11: Axis Aligned Bounding Box

A similar case exists for sphere bounding box but in a more efficient way. In case that virtual models are bounded by spheres, the collision detection and proximity queries can be measured just comparing radii of BV's. Spheres are rotationally invariant. The location of center point and radius are the parameters used to define the BV. It provides fast computations.

Figure 2.12: Sphere Bounding Volume

Other important and widely used BV is OBB. The object is enclosed by an oriented parallel-piped. The parameters of the box are the orthogonal axes, center positions and extensions on each axis. This algorithm is generally slower than AABB's but provides tighter fit. Figure 2.13 shows a typical OBB.

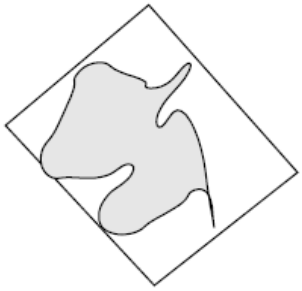

Figure 2.13: Oriented Bounding Box

In some applications, k-edged polygons are used to approximate geometry under consideration [22]. The principle is similar to the AABB's but this time more axes are used. The locations of the fixed axes are used as parameters of the box. A typical k-DOP BV is depicted in Figure 2.14.

In addition to geometrical approximation of the computational domain, a hierarchical structure is constructed. On top of the structure, broader BV is used meaning it encloses more than one primitive. The collision detection algorithm is firstly applied to the top most structure. If the collision event is detected in the BV's, then sub BV's are searched for more information like collision point or penetration amount [23]. A typical hierarchy and BV are depicted in the Figures 2.15 and 2.16.

Figure 2.14: K-DOP

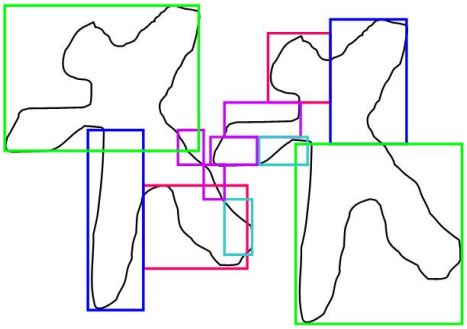

Figure 2.15: Typical Division of Event Scene

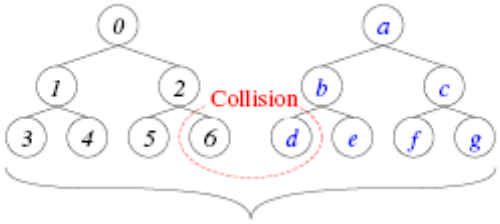

Figure 2.16: Bounding Volume Hierarchy Sample.

The collision detection algorithm used in haptics detects any contact between avatar and the objects virtually modeled and measure the depth of penetration. In the case of rigid object models, the algorithm detects the penetration amount since no deformations occur. Only the new position of the avatar is updated to the outer surface of the virtual model to provide the effect of rigidity.

In the case of deformable virtual object, the position of the avatar after the penetration is not updated. The penetration depth is sent to the force response algorithm to calculate the reaction force and corresponding deformation values. In general, the avatar is assumed as rigid and interacting objects are deformable.

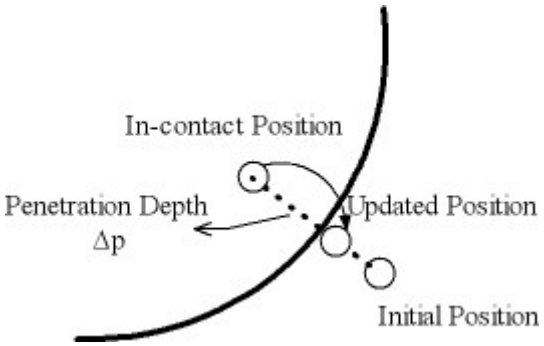

Figure 2.17: Rigid Object in Contact

However, in applications where long and flexible tool interactions are desired for simulation, the avatar is also modeled as deformable. The complexity of the problem increases dramatically in this case. In this thesis, avatar is modeled as rigid to reduce the complexity.

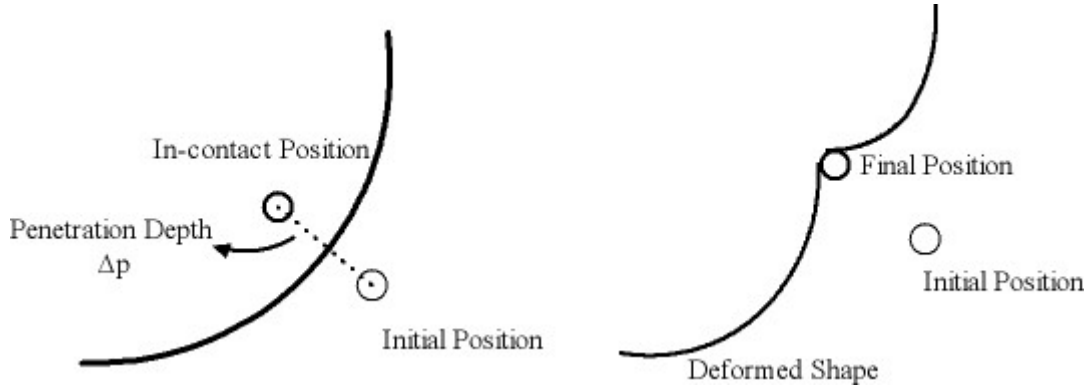

Figure 2.18: Deformable Object Before and After Contact.

2.2.2 Modeling in Haptic Environments

Modeling section of haptic systems is also called as Force Response Algorithms (FRA). It is the computationally most expensive part of the haptic rendering paradigm depending on modeling method chosen. Force response algorithms determine the reaction forces resulting from the interaction between the virtually modeled objects and the representation of the haptic interface called avatar. These algorithms can be seen as the part that adds physical reality to the virtual world. Frictional and gravitational effects, elastic or visco-elastic mechanical properties can be added to the environment to obtain more realistic models. The input to the FRA algorithms comes directly from collision detection algorithms. When a collision event is detected, collision queries such as penetration amount, collision point, and

contact normal are measured and transmitted to the FRA algorithms to obtain corresponding responses such as deformations and reaction forces. In FRA algorithms, 2 methods are used for deformable object modeling as follows.

2.2.2.1 Geometry Based Modeling

Geometry based models are computationally inexpensive applications. Real-time performance of these algorithms is high but they are not appropriate to obtain physically realistic simulations. Two main approaches exist. The first, in vertex based models; the vertices of the objects are moved to obtain a deformed model. Reaction forces are generally calculated by using simple Hook's law. Response of the objects are not based on continuum mechanics principles, they are just for visual purposes. The second, in spline based models; multiple control points are attained to surfaces and curves of the virtually modeled object as shown in Figure 2.19. Instead of transforming the vertices directly, the control points are manipulated to show deformation effects [24]. Spline based models provide smoother deformations visually.

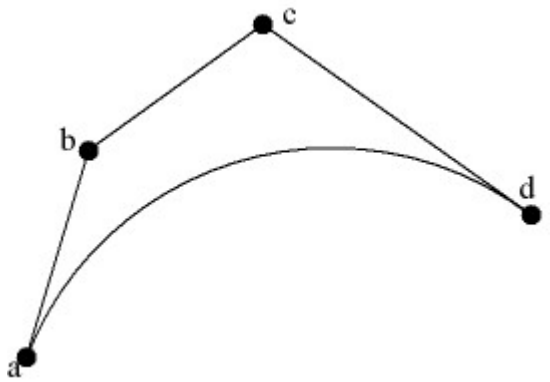

Figure 2.19: Control Points of Cubic Curve

Although the increasing number of control points allows proper shape control, the deformations do not rely on mechanics rules. In cases where visual representations are the primary target, these methods can be used efficiently.

2.2.2.2 Physics Based Modeling

In many applications, visual performance is not adequate and desirable. Physical principles and mechanical laws are needed to be embedded into simulation for realism. In order to obtain simulations with reality in an appropriate way, physics

based modeling techniques are used. Two distinct physics based modeling techniques exist namely Particle Based Modeling and Finite Element Method Based Modeling.

The main elements of the particle based modeling are the spring, damper and point masses. The modeling object is discretized using these concentrated mass particles. These point masses are connected to each other by spring-damper networks that complete the model as shown in Figure 2.20. The solution is obtained for each particle meaning that they have its own velocity and acceleration vectors. The main advantage of the spring-damper based modeling is the ease of implementation. They are both used in static and dynamics simulations efficiently. Although it provides efficient real-time performance, the deformation accuracy greatly depends on the topology of the spring-damper network. The nodes in the system are connected by several other members. Optimum positioning of the elements and appropriate number of spring and damper elements must be used to obtain realistic models. Inappropriate number of elements or non-optimal topology may lead the system to be over-constrained or under-constrained.

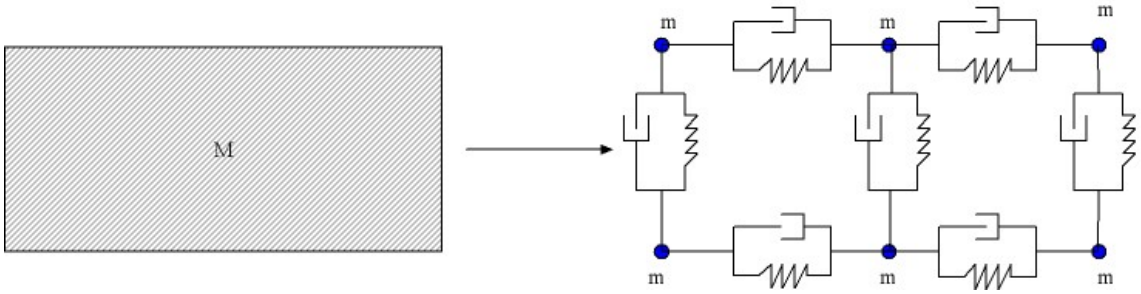

Figure 2.20: Spring – Damper Model

Deformations calculated by particle-based modeling have limited reliability because these networks are not based on continuum mechanics principles. In small deformation case, spring-damper model behaves like linear elastic continuum models. However in the case of large deformations, the accuracy of the spring-damper system decreases. Determining the correct stiffness and damping values for the each node is another important issue. These values are determined by experimental tests and given material behaviors. In addition, care should be taken in dynamical simulations. For a given time step Δt, number nodes n and mass $\mu = m_{total} / n$, there is a critical stiffness (K_c) value above which the system becomes unstable and divergent. The relationship is given in Equation 2.2. This

equation shows that time step should be decreased to increase the stiffness value. This relation limits the dynamic behavior of the models.

$$K_c = \frac{\mu}{\pi^2 (\Delta t)^2} \approx \frac{m_{total}}{n\pi^2 (\Delta t)^2} \tag{2.2}$$

In a dynamic system, the equation of a single particle can be formulated as,

$$m_i \ddot{x}_i = -c_i \dot{x}_i + \sum_m g_{im} + f_i \tag{2.3}$$

m_i represents the mass of the ith mass, x_i is its position vector, c_i is the related damping coefficient, g_{im} is the force exerted on the i'th mass by the spring between the ith and mth masses and f_i is the sum of external forces on the ith mass. Equations obtained for each mass is assembled into global equation system as given in Equation 2.4.

$$M\ddot{x} + C\dot{x} + Kx = f \tag{2.4}$$

The solution of Equation 2.4 can be obtained by solving following two equations.

$$\begin{aligned} \dot{v} &= M^{-1}(-Cv - Kx + f) \\ \dot{x} &= v \end{aligned} \tag{2.5}$$

The complexity of the spring-damper network and number of elements greatly affect the sizes of these matrices. In spring-damper based models, structures of the matrices are sparse and processing them is rather simple [24]. This simplicity is the primary reason for widely usage of type of models such as facial tissue modeling [25], animation [26] etc.

Finite Element Methods (FEM) are greatly used in engineering systems. The objects to be modeled are divided into finite number of elements (mesh) as seen on Figure 2.21.

Each element is formulated based on continuum mechanics principles. The formulated elements are assembled to obtain global set of equations. Boundary conditions, forces are then applied to the assembled system, and corresponding

deformations and reaction forces are calculated by solving the obtained set of equations.

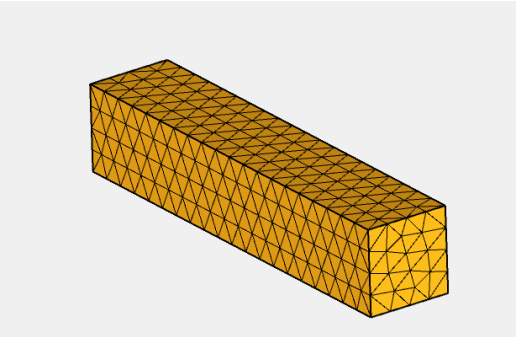

Figure 2.21: Finite Element Model

FEM are both applicable to surface based objects or volumetric models. They are suitable for realistic simulations because the elements are formulated according to the underlying physics. In addition, FEM based models have no restriction between stiffness values and time steps. The problem with the finite element method is the high expense in computational tasks [27]. The solution of the assembled system is obtained by matrix inversion procedures that greatly reduce the performance [28]. In addition, as the number of elements increased, the computational demand of the model also increases that makes the solution even more problematic. Also meshing the geometry reduces the performance as well. Especially in deformable modeling or in applications where topology changes occur, re-meshing is inevitable. Solving the equation sets and re-meshing the geometry in the same application greatly decreases the real-time performance. In order to overcome the performance drawbacks coming from the solution side, explicit methods and some decomposition methods are used especially in nonlinear models.

In this thesis, large deformation beam problem is formulated and discretized using finite element methods [29]. Developed models are integrated with Phantom Premium High Force 6 DOF Haptic Arm and proper orthogonal decomposition is used to obtain low order models.

3. LARGE DEFORMATION BEAM PROBLEM

Realistic modeling of objects is one of the primary goals of the researchers in the field of haptics. Due to computational burdens of haptic systems, most of the works in this field deals with linear elastic models. Misra's work [30] shows that without implementation of a nonlinear model, the user will receive noticeably different haptic feedback during interaction. In the following sections, large deformation beam model is formulated using the derivations of Reddy [29].

3.1 General Definitions

Euler-Bernoulli beam theory is used to formulate nonlinear beam problem. In this theory, plane sections perpendicular to the axis of the beam before deformation remains plane, rigid and perpendicular to the deformed axis of the beam. Transverse shear strains and the effect of Poisson effect is neglected in this theory. The difference between shear deformable beam theory and Euler-Bernoulli beam theory is depicted in Figure 3.1.

In order to obtain nonlinear beam formulation, appropriate displacement field that represents the problem should be chosen as in Equation 3.1.

$$u_1 = u_0(x) - z\frac{dw_0}{dx}, \quad u_2 = 0, \quad u_3 = w_0(x) \tag{3.1}$$

where u_1, u_2 and u_3 represent the displacements along x, y and z axis, respectively. u_0 and w_0 respectively represent axial and transverse displacements of a point on the neutral axis of the beam. By using Green strain tensor, nonlinear strain-displacement relation can be obtained as,

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_m}{\partial x_i}\frac{\partial u_m}{\partial x_j}\right) \tag{3.2}$$

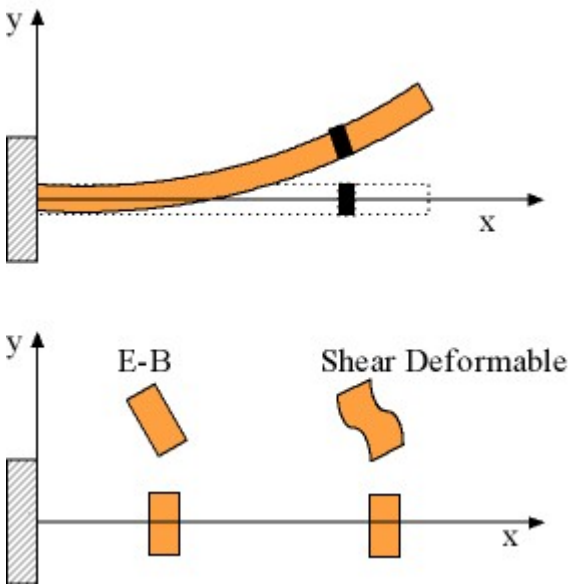

Figure 3.1: Euler-Bernoulli and Shear Deformable Beam Theories

Governing equations of the beam problem can be obtained by using principles of virtual displacements. According to the principle, the body is in equilibrium if the total virtual work done by the internal and external forces through their respective displacements are zero. The principle can be formulated as follows.

$$\delta W \equiv \delta W_I^e - \delta W_E^e = 0 \tag{3.3}$$

δW_I^e represents strain energy caused by actual stress σ_{ij} along the direction of virtual strain ε_{ij}. δW_E^e is the work done by the external loads. They can be expressed by using Equation 3.4 and 3.5.

$$\delta W_I = \int_V \delta\varepsilon_{ij}\sigma_{ij} d\mathrm{V} \tag{3.4}$$

$$\delta W_E = \int_{x_a}^{x_b} q\delta w_0 dx + \int_{x_a}^{x_b} f\delta u_0 dx + \sum_1^6 Q_i^e \delta\Delta_i^e \tag{3.5}$$

where V represents the volume of an element, q is the distributed load per unit length, f is the distributed axial load per unit length, Q_i^e are the nodal forces, $\delta\Delta_i^e$ are the virtual displacements. Nodal displacements and forces in Equations (3.4) and (3.5) can be expressed by using following relations.

$$\Delta_1^e = u_0(x_a) \qquad \Delta_2^e = w_0(x_a)$$
$$\Delta_3^e = -\left.\frac{dw_0}{dx}\right|_{x_a} = \theta(x_a)$$
$$\Delta_4^e = u_0(x_b) \qquad \Delta_5^e = w_0(x_b) \tag{3.6}$$
$$\Delta_6^e = -\left.\frac{dw_0}{dx}\right|_{x_b} = \theta(x_b)$$

$$Q_1^e = -N_{xx}(x_a) \qquad Q_3^e = -M_{xx} \qquad Q_4^e = -N_{xx}(x_b) \qquad Q_6^e = M_{xx}$$
$$Q_2^e = -\left[\frac{dw_0}{dx}N_{xx} + \frac{dMxx}{dx}\right]_{x_a} \qquad Q_5^e = \left[\frac{dw_0}{dx}N_{xx} + \frac{dMxx}{dx}\right]_{x_b} \tag{3.7}$$

N_{xx} and M_{xx} represent the axial force per unit length and moment per unit length, respectively. They are formulated defined as,

$$N_{xx} = \int_{A^e} \sigma_{xx} dA$$
$$M_{xx} = \int_{A^e} \sigma_{xx}\, z\, dA \tag{3.8}$$

Volume integral in Equation 3.4 can be transformed into line integral by using the following relation

$$\int_{Ve} (*) = \int_{x_a}^{x_b} \int_{A_e} (*)\, dA\, dx \tag{3.9}$$

Equation (3.4) can be rewritten by using the related strain relations and the property in Equation (3.9), and then one can obtain:

$$0 = \int_{x_a}^{x_b} \left[\left(\frac{d\delta u_0}{dx} + \frac{dw_0}{dx}\frac{d\delta w_0}{dx} \right) N_{xx} - z\frac{d^2\delta w_0}{dx^2} M_{xx} \right] dx$$
$$- \int_{x_a}^{x_b} q(x)\,\delta w_0(x)\, dx - \int_{x_a}^{x_b} f(x)\,\delta u_0(x) - \sum_{1}^{6} Q_i^e\, \delta\Delta_i^e \tag{3.10}$$

If the virtual terms, δu_0 and δw_0, are collected and their coefficients are set to zero, one can obtain the following two equations

$$0 = \int_{x_a}^{x_b} \left(\frac{d\delta u_0}{dx} N_{xx} - \delta u_0 f(x)\right) dx - Q_1 \delta\Delta_1^e - Q_4 \delta\Delta_4^e$$

$$0 = \int_{x_a}^{x_b} \left(\frac{d\delta w_0}{dx} \left(\frac{dw_0}{dx} N_{xx} \right) - \frac{d^2 \delta w_0}{dx^2} M_{xx} - \delta w_0 q(x) \right) dx \qquad \textbf{(3.11)}$$

$$- Q_2 \delta\Delta_2^e - Q_3 \delta\Delta_3^e - Q_5 \delta\Delta_5^e - Q_6 \delta\Delta_6^e$$

The nodal displacements and forces are depicted in the following figures

Figure 3.2: Nodal Displacements

Figure 3.3: Nodal Forces

The main of purpose of the derivation is to express Equation (3.11) in terms of nodal displacements. N_{xx} and M_{xx} force and moment terms should be written in terms of nodal displacements. Constitutive relation establishes the related relation. If linear elastic material behaviour is assumed, Hook's law can be used

$$\sigma_{xx} = E^e \varepsilon_{xx} \qquad \textbf{(3.12)}$$

By substituting Equation (3.12) and related strain relations into Equations (3.8), one can obtain

$$N_{xx} = \int_{A^e} \sigma_{xx} dA = \int_{A^e} E^e \varepsilon_{xx} dA = \int_{A^e} E^e \left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_0}{dx}\right)^2 - z\frac{d^2 w_0}{dx^2} \right] dA$$

$$= A^{xx} \left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_o}{dx}\right)^2 \right] - B_{xx}^e \frac{d^2 w_0}{dx^2} \qquad \textbf{(3.13)}$$

$$M_{xx} = \int_{A^e} \sigma_{xx} z dA = \int_{A^e} E^e \varepsilon_{xx} z dA = \int_{A^e} E^e \left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_0}{dx} \right)^2 - z\frac{d^2 w_0}{dx^2} \right] z\, dA$$
$$= B_{xx}\left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_0}{dx} \right)^2 \right] - D_{xx}^e \frac{d^2 w_0}{dx^2} \qquad \textbf{(3.14)}$$

In above relations, A_{xx}^e, B_{xx}^e and D_{xx}^e represent extensional, extensional bending and bending stiffnesses of the beam element, respectively. If the modeled beam is assumed to be made of an isotropic material and the x-axis is taken along the centroidal axis of the beam,

$$B_{xx}^e = 0$$
$$A_{xx}^e = E^e A^e \qquad \textbf{(3.15)}$$
$$D_{xx}^e = E^e I^e$$

where superscript e represents the element. By using Equations (3.13) and (3.14), Equation (3.11) can be expressed in terms of nodal displacements completely.

$$0 = \int_{x_a}^{x_b} A_{xx} \frac{d\delta u_0}{dx}\left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_0}{dx} \right)^2 \right] dx - \int_{x_a}^{x_b} f(x)\delta u_0 dx - Q_1 \delta u_0(x_a) - Q_4 \delta u_0(x_b)$$
$$0 = \int_{x_a}^{x_b} \left\{ A_{xx} \frac{d\delta w_0}{dx}\frac{dw_0}{dx}\left[\frac{du_0}{dx} + \frac{1}{2}\left(\frac{dw_0}{dx} \right)^2 \right] + D_{xx} \frac{d\delta^2 w_0}{dx^2}\frac{d^2 w_0}{dx^2} \right\} dx \qquad \textbf{(3.16)}$$
$$- \int_{x_a}^{x_b} q\delta w_0 dx - Q_2 \delta w_0(x_a) - Q_3 \delta\theta(x_a) - Q_5 \delta w_0(x_a) - Q_6 \delta\theta(x_b)$$
$$\theta = -\frac{dw_0}{dx}$$

3.2 Finite Element Formulation

In order to obtain finite element formulation of the beam model, axial and transverse displacements, u_0 and w_0, are approximated by using interpolation functions.

$$u_0(x) = \sum_{j=1}^{2} u_j \psi_j(x)\,,\; w_o(x) = \sum_{1}^{4} \Delta_j \phi_j(x) \qquad \textbf{(3.17)}$$

where ψ_j are the linear Lagrange interpolation functions and ϕ_j's are the Hermite cubic interpolation functions. Axial displacements are approximated using Lagrange interpolation functions and transverse displacements and rotations are approximated using Hermite cubic interpolation functions.

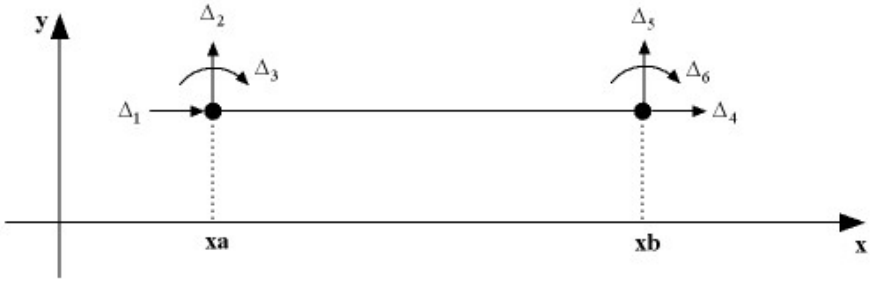

Figure 3.4: Derivation of interpolation functions

Lagrange interpolation functions are calculated using a linear polynomial given by

$$u(x) = a_0 + a_1 x \tag{3.18}$$

If the linear polynomial is evaluated at the nodal coordinates, then

$$\begin{aligned} x = xa \quad ; \quad u(x_a) &= a_0 + a_1 x_a = u_1 \\ x = xb \quad ; \quad u(x_b) &= a_0 + a_1 x_b = u_2 \end{aligned} \tag{3.19}$$

Solving the equations simultaneously for the coefficients a_0 and a_1, one gets

$$a_0 = -\frac{u_1 x_b - u_2 x_a}{x_a - x_b}, \quad a_1 = -\frac{u_2 - u_1}{x_a - x_b} \tag{3.20}$$

The solution in terms of nodal values can be expressed as,

$$u(x) = -\frac{u_1 x_b - u_2 x_a}{x_a - x_b} - \frac{u_2 - u_1}{x_a - x_b} x \tag{3.21}$$

Collecting the terms with respect to nodal values, one can obtain,

$$\begin{aligned} u(x) &= \frac{x - x_b}{x_a - x_b} u_1 + \frac{x - x_a}{x_b - x_a} u_2 \\ u(x) &= N_1 u_1 + N_2 u_2 \end{aligned} \tag{3.22}$$

The same procedure can be applied to obtain Hermite cubic interpolation functions. The starting polynomial and its derivative is formulated as,

$$u(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$
$$u'(x) = a_1 + 2a_2 x + 3a_3 x^2 \qquad \textbf{(3.23)}$$

If these equations are evaluated at the nodal coordinates as in the previous case,

$$\begin{aligned} x = x_a \quad ; \quad u_2 &= a_0 + a_1 x_a + a_2 x_a^2 + a_3 x_a^3 \\ x = x_a \quad ; \quad u_3 &= a_1 + 2a_2 x_a + 3a_3 x_a^2 \\ x = x_b \quad ; \quad u_5 &= a_0 + a_1 x_b + a_2 x_b^2 + a_3 x_b^3 \\ x = x_b \quad ; \quad u_6 &= a_1 + 2a_2 x_b + 3a_3 x_b^2 \end{aligned} \qquad \textbf{(3.24)}$$

In matrix form, the equations can be expressed as,

$$\underbrace{\begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ u_6 \end{bmatrix}}_{u} = \underbrace{\begin{bmatrix} 1 & x_a & x_a^2 & x_a^3 \\ 0 & 1 & 2x_a & 3x_a^2 \\ 1 & x_b & x_b^2 & x_b^3 \\ 0 & 1 & 2x_b & 3x_b^2 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}}_{a} \qquad \textbf{(3.25)}$$

If the equation is solved for matrix a and put into Equation (3.24), one can obtain the Hermite interpolation functions as,

$$u(x) = N_1 u_2 + N_2 u_3 + N_3 u_5 + N_4 u_6 \qquad \textbf{(3.27)}$$

$$N_1 = -\frac{(x - x_b)^2 (2x - 3x_a + x_b)}{(x_a - x_b)^3} \qquad \textbf{(3.28a)}$$

$$N_2 = \frac{(x - x_b)^2 (x - x_a)}{(x_a - x_b)^2} \qquad \textbf{(3.28b)}$$

$$N_3 = \frac{(x - x_a)^2 (2x - 3x_b + x_a)}{(x_a - x_b)^2} \qquad \textbf{(3.28c)}$$

$$N_4 = \frac{(x - x_a)^2 (x - x_b)}{(x_a - x_b)^2} \qquad \textbf{(3.28d)}$$

If we substitute interpolated values of $u_0(x)$ and $w_0(x)$ with calculated interpolation functions into Equation 3.16, one can obtain the following set of equations.

$$\begin{aligned} 0 &= \sum_{j=1}^{2} K_{ij}^{11} u_j + \sum_{J=1}^{4} K_{iJ}^{12} \overline{\Delta}_J - F_i^1 \qquad i = (1,2) \\ 0 &= \sum_{j=1}^{2} K_{Ij}^{21} u_j + \sum_{J=1}^{4} K_{IJ}^{22} \overline{\Delta}_J - F_i^2 \qquad I = (1,2,3,4) \end{aligned} \tag{3.29}$$

where,

$$K_{ij}^{11} = \int_{x_a}^{x_b} A_{xx} \frac{d\psi_i}{dx} \frac{d\psi_j}{dx} dx \tag{3.30a}$$

$$K_{iJ}^{12} = \frac{1}{2} \int_{x_a}^{x_b} \left(A_{xx} \frac{dw_0}{dx} \right) \frac{d\psi_i}{dx} \frac{d\phi_J}{dx} dx \tag{3.30b}$$

$$K_{Ij}^{21} = \frac{1}{2} \int_{x_a}^{x_b} \left(A_{xx} \frac{dw_0}{dx} \right) \frac{d\psi_i}{dx} \frac{d\phi_J}{dx} dx = 2K_{jI}^{12} \tag{3.30c}$$

$$K_{IJ}^{22} = \int_{x_a}^{x_b} D_{xx} \frac{d^2\phi_I}{dx^2} \frac{d^2\phi_J}{dx^2} dx + \frac{1}{2} \int_{x_a}^{x_b} A_{xx} \left(\frac{dw_o}{dx} \right)^2 \frac{d\phi_I}{dx} \frac{d\phi_J}{dx} dx \tag{3.30d}$$

$$F_i^1 = \int_{x_a}^{x_b} f\psi_i dx + Q_i \tag{3.31a}$$

$$F_I^2 = \int_{x_a}^{x_b} q\phi_I dx + Q_I \tag{3.31b}$$

If the equations are written in matrix form, one can obtain the element formulation as,

$$\begin{bmatrix} K^{11} & K^{12} \\ K^{21} & K^{22} \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} F^1 \\ F^2 \end{bmatrix} \tag{3.32}$$

$$[K(\Delta_i)][\Delta_i] = [F] \tag{3.33}$$

In order to obtain the solution vector Δ_i of the nonlinear equation, Newton-Raphson iterative method is used. In this method, the nonlinear sets of equations are linearized around a previously known solution. The residual of the sets of equation can be expressed as,

$$R(\Delta) = -(\,[F] - (\,[K(\Delta_i)][\Delta_i]\,)\,) \tag{3.34}$$

Equation 3.34 is expanded around a known solution into Taylor's series and one can obtain,

$$R(\Delta) = R(\Delta^{(r-1)}) + \left(\frac{\partial R}{\partial \Delta}\right)\Big|_{\Delta^{(r-1)}} . \delta\Delta + \frac{1}{2}\left(\frac{\partial^2 R}{\partial \Delta^2}\right)\Big|_{\Delta^{(r-1)}} .(\delta\Delta)^2 + \cdots = 0 \tag{3.35}$$

If the terms having the order higher than one are omitted, the sets of equations can be solved by using following relation,

$$\begin{aligned} \delta\Delta &= -(T(\Delta^{(r-1)}))^{-1}\, R(\Delta^{(r-1)}) \\ \delta\Delta &= (T(\Delta^{(r-1)}))^{-1}\, (F - K(\Delta^{(r-1)})\Delta^{(r-1)}) \end{aligned} \tag{3.36}$$

$$\Delta^r = \Delta^{(r-1)} + \delta\Delta^r \tag{3.37}$$

In the Equation 3.36, the term T is called as tangent stiffness matrix and calculated as,

$$T_{ij}^{ab} = \left(\frac{\partial R_i^a}{\partial \Delta_j^b}\right)^{(r-1)} \tag{3.38}$$

Components of the tangent stiffness matrix can be calculated as,

$$T^{11} = \int_{x_a}^{x_b} A_{xx} \frac{d\psi_i}{dx}\frac{d\psi_j}{dx} dx \tag{3.39a}$$

$$T^{12} = \int_{x_a}^{x_b} A_{xx} \frac{dw_0}{dx}\frac{d\psi_i}{dx}\frac{d\phi_J}{dx} dx = 2K^{12} \tag{3.39b}$$

$$T^{21} = \int_{x_a}^{x_b} A_{xx} \frac{dw_0}{dx}\frac{d\phi_I}{dx}\frac{d\psi_j}{dx} dx = K^{21} \tag{3.39c}$$

$$T^{22} = K^{22} + \int_{x_a}^{x_b} A_{xx}\left(\frac{du_0}{dx} + \frac{dw_0}{dx}\frac{dw_0}{dx}\right)\frac{d\phi_I}{dx}\frac{d\phi_J}{dx}dx \qquad \textbf{(3.39d)}$$

As an application, a full beam model hinged at both ends is used as shown in the Figure 3.5. Since the model is symmetric, only the half of the model is used for the computational efficiency. The response of the midpoint of the full beam model is shown in the Table 3.1 for various load cases.

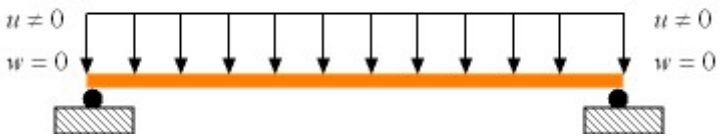

Figure 3.5: Hinged-Hinged Beam

The length of the modeled beam is taken as 2.54 meters in, the cross-sectional area is (25.4x25.4 mm^2), and elasticity module is taken as 206 GPA. The model consists of 4 elements. The vertical displacements of the each node are shown in the Figure 3.6.

In this model various forces are applied to the beam model and the resulting displacements are calculated. However, in haptic applications, the forces applied to the virtual models are unknown. The only information is the displacements resulting from collision detection phenomena.

Table 3.1: Mid-Point Deflection

Applied Load(N/mm)	*Tip Deflection (mm)*
0.2	-15.1081
0.7	-52.8783
1.2	-90.6486
1.7	-128.4188
2.2	-166.1891
2.7	-203.9593
3.2	-241.7296
3.7	-279.4998
4.2	-317.2701
4.7	-355.0403
10	-755.4
15	-1133.1

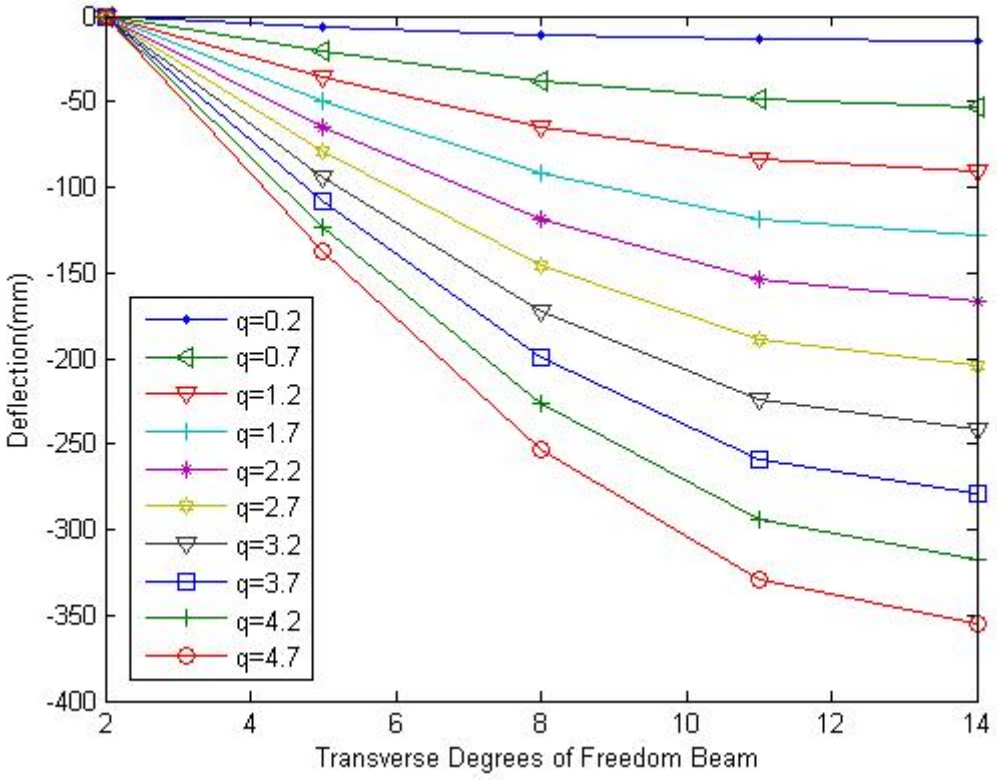

Figure 3.6: Nodal Beam Deflections for Load Cases q

The model is modified as to take the nodal displacement information as input and to calculate the resulting reaction force. Recall that the system is formulated as,

$$\underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}}_{F} = \underbrace{\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix}}_{K(\Delta)} \underbrace{\begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{bmatrix}}_{\Delta} \tag{3.40}$$

Let's assume that the displacement boundary condition is applied to the m^{th} degree of freedom. The equation above can be rearranged such that the column and the row of the related degree of freedom are extracted from the global set of equations. The reduced system of equations can be represented as,

$$\underbrace{\begin{bmatrix} f_1 - k_{1m}u_m \\ f_2 - k_{1m}u_m \\ \vdots \\ f_{(m-1)} - k_{1m}u_m \\ f_{(m+1)} - k_{1m}u_m \\ \vdots \\ f_n - k_{1m}u_m \end{bmatrix}}_{F} = \underbrace{\begin{bmatrix} k_{11} & k_{21} & \cdots & k_{1(m-1)} & k_{1(m+1)} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2(m-1)} & k_{2(m+1)} & \cdots & k_{2n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ k_{(m-1)1} & k_{(m-1)2} & \cdots & k_{(m-1)(m-1)} & k_{(m-1)(m+1)} & \cdots & k_{(m-1)n} \\ k_{(m+1)1} & k_{(m+1)2} & \cdots & k_{(m+1)(m-1)} & k_{(m+1)(m+1)} & \cdots & k_{(m+1)n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{n(m-1)} & k_{n(m+1)} & \cdots & k_{nn} \end{bmatrix}}_{K(\Delta)} \underbrace{\begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{(m-1)} \\ \Delta_{(m-2)} \\ \vdots \\ \Delta_n \end{bmatrix}}_{\Delta} \quad \textbf{(3.41)}$$

The extracted degrees of freedom and its reaction force can be expressed as,

$$f_m = k_{m1}\Delta_1 + k_{m2}\Delta_2 + \cdots + k_{mm}\Delta_m + \cdots + k_{nn}\Delta_n \quad \textbf{(3.42)}$$

The nodal displacements are obtained by solving Equation 3.41 and the reaction force is calculated by substituting the obtained displacements into Equation 3.42.

3.3 Proper Orthogonal Decomposition using Singular Value Decomposition

Proper Orthogonal Decomposition method is used to obtain low dimensional models of the original systems. The application of POD in engineering field can be seen in various areas including turbulent flow modeling, image processing [31], structural vibrations [32], chaotic dynamical systems, identification of nonlinear systems [33], structural dynamics and micro electro-mechanical systems (MEMS) [34]. In POD, the high dimensional system under consideration is approximated by using a set of orthonormal basis vectors that are obtained by processing the subspace created from the response of the system. These responses are either obtained from the results of experiments or numerical simulations. In order to obtain the orthonormal basis vectors, Karhunen-Loève Decomposition (KLD), Principle Component Analysis (PCA) and Singular Value Decomposition (SVD) methods are used. Although the three different methods approach the problem in different perspectives, they result in the same set of orthogonal base vector. The equivalence of these methods is given in [35]. KLD and PCA methods are based on statistical principles. In the following section of this thesis, POD in the sense of SVD is used.

Assume that $x \in R^m$ is a random vector in the subspace of R^m. The main objective of the POD is to represent x with minimum number of parameters using a few of the ordered orthonormal basis vectors ϕ_i and related coefficients c_i

$$x = \sum_{i=1}^{m} c_i \phi_i = \widetilde{\phi}\,\widetilde{c} \tag{3.43}$$

where,

$$\begin{aligned} &y_i = \phi_i^T x \ (i = 1,2,\cdots,m) \\ &c = (c_1, c_2 \ldots, c_m)^T \\ &\widetilde{\phi} = [\phi_1, \phi_2, \cdots, \phi_m] \end{aligned} \tag{3.44}$$

To represent the random vector x, assume that only l number of base vectors are used and can be approximated by using the following relation.

$$x \cong x(l) = \sum_{i=1}^{l} c_i \phi_i = \widetilde{\phi}\,\widetilde{c} \tag{3.45}$$

The main aim of the POD is to find the basis vectors that satisfy the following extreme value problem given by

$$\begin{aligned} &\min \varepsilon^2(l) = E\{\|x - x(l)\|^2\} \\ &\phi_i^T \phi_i = \delta_{ij} \quad i = 1,2,\cdots,m \\ &l \le m \end{aligned} \tag{3.46}$$

Singular value decomposition can be seen as an extension to the eigenvalue decomposition with an exception of that SVD is also applicable to non-square matrices. Assume X is a real (mxn) matrix. Then, X can be factored as,

$$X^T_{(mxn)} = U_{(mxm)} S_{(mxn)} V^T_{(nxn)} = (orthogonal)(diagonal)(orthogonal) \tag{3.47}$$

The columns of the matrix $U_{(mxm)}$ are called as left singular vectors and the columns of $V^T_{(nxn)}$ are called as right singular vectors of the matrix X. S is an (mxn) matrix with non-diagonal entries are zero and called as singular values of X^T. Diagonal entries of the matrix S are placed in decreasing order.

$$\begin{aligned} &s_{11} \ge s_{22} \cdots \ge s_{pp} \\ &p = \min(m,n) \end{aligned} \tag{3.48}$$

The matrix X can be expressed as a linear combination of the columns of the right and left singular vectors. Although SVD is a linear procedure, it is widely used in nonlinear system investigations.

$$X = col_1(U)s_{11}col_1(V)^T + col_2(U)s_{22}col_2(V)^T + \cdots + col_p(U)s_{pp}col_p(V)^T \qquad \textbf{(3.49)}$$

To determine the left and right singular vectors and singular values of X^T, following straight forward procedure is implemented. Semi definite $XX^T \in R^{mxm}$ matrix is created. Then, the eigenvalues of the XX^T are calculated and arranged in decreasing order such that,

$$\lambda_1 \geq \lambda_1 \geq \cdots \geq \lambda_r = \lambda_{r+1} = \cdots = \lambda_m = 0 \qquad \textbf{(3.50)}$$

The singular values are calculated as follows and placed in the diagonal entries of the matrix S,

$$\sigma_i = \sqrt{\lambda_i} \qquad (i = 1,2,\cdots,m) \qquad \textbf{(3.51)}$$

The columns of left singular vector U are formed by the solution of the following eigenvalue problem

$$\begin{aligned} &XX^T u_i = \lambda_i u_i \\ &U = (u_1, u_2, \cdots, u_m) \end{aligned} \qquad \textbf{(3.52)}$$

The right singular vector V is also calculated by a similar process. This time, the columns of the right singular vectors are formed by the solution of the related eigenvalue problem

$$\begin{aligned} &X^T X v_i = \lambda_i v_i \\ &V = (v_1, v_2, \cdots, v_n) \end{aligned} \qquad \textbf{(3.53)}$$

After calculating the right and left singular vectors, V and U respectively, and Singular Value Matrix S, any non-square matrix can be represented by the general SVD formulation given in Equation (3.47). Assume that in order to obtain approximate low order model of the system (R^m), first l basis vectors obtained from the SVD operation are used

$$x = \sum_{i=1}^{m} c_i \phi_i$$
$$x(l) = \sum_{i=1}^{l} c_i \phi_i \qquad \textbf{(3.54)}$$

If the error function is formulated as,

$$\varepsilon^2(l) = \sum_{i=1}^{n} \|x_i - x_i(l)\|^2 = \sum_{i=1}^{n} \left\| \sum_{j=l+1}^{m} c_{ji} \phi_j \right\|^2 = \sum_{i=1}^{n} \sum_{j=l+1}^{m} c_{ji}^2 = \sum_{j=l+1}^{m} \phi_j^T X X^T \phi_j \qquad \textbf{(3.55)}$$

In order to minimize the error function, the extreme value given in Equation 3.46 must be investigated. Lagrange multipliers method is used to solve the problem with given constraints. Lagrangian function of Equation 3.55 can be written as,

$$L = \sum_{j=l+1}^{m} \phi_j^T X X^T \phi_j - \sum_{i=l+1}^{m} \sum_{j=l+1}^{m} k_{ij} (\phi_j^T \phi_j - \delta_{ij}) \qquad \textbf{(3.56)}$$

The solution of the extreme value problem is searched for orthonormal vectors. The Lagrangian function is differentiated with respect to ϕ_i's

$$\frac{\partial L}{\partial \phi_j} = 2\left(X X^T \phi_j - k_{ij} \sum_{i=l+1}^{m} \phi_i \right) = 2 X X^T \phi_j - 2\Phi_{m-1} k_j \quad (j = l+1, l+2, \cdots, m)$$
$$k_j = (k_{(l+1)j}, k_{(l+2)j}, \cdots, k_{mj})^T \qquad \textbf{(3.57)}$$

In matrix form,

$$\frac{\partial L}{\partial \Phi_{(m-1)}} = 2 X X^T \Phi_{(m-1)} - 2\Phi_{(m-1)} K_{(m-1)}$$
$$K_{(m-1)} = (k_{(l+1)}, k_{(l+2)}, \cdots, k_m) \qquad \textbf{(3.58)}$$

If the derivative of the Lagrangian function is equated to zero, one can obtain

$$X X^T \Phi_{(m-1)} = \Phi_{(m-1)} K_{(m-1)} \qquad \textbf{(3.59)}$$

If the Equation 3.59 is pre-multiplied by $\Phi_{(m-1)}^T$, then we get

$$K_{(m-1)} = \Phi_{(m-1)}^T X X^T \Phi_{(m-1)} \qquad \textbf{(3.60)}$$

By the definition of $K_{(m-1)}$, it is a semi-definite matrix. Then, there exists an orthogonal matrix P that forms a diagonal matrix Λ such that,

$$\Lambda = P^T K_{(m-1)} P = P^T \Phi_{(m-1)}^T X X^T \Phi_{(m-1)} P \quad \textbf{(3.61)}$$

By post multiplying the Equation 3.59 by P, we obtain

$$\begin{aligned} X X^T \Phi_{(m-1)} P &= \Phi_{(m-1)} P P^T K_{(m-1)} P \\ X X^T \Phi_{(m-1)} P &= \Phi_{(m-1)} P \Lambda \end{aligned} \quad \textbf{(3.62)}$$

The form obtained in the equation above shows that $\Phi_{(m-1)} P$ contains the eigenvectors of the system related to the elements λ_i's of Λ that are the eigenvalues of XX^T. $\Phi_{(m-1)} P$ is called right singular vectors.

Theorem: Let $A \in R^{(nxm)}$ and $Q \in R^{(mxm)}$ be an orthogonal matrix, then the Frobenius norm $\|\cdot\|_F$ can be expressed as

$$\|A\|_F = \|AQ\|_F \quad \textbf{(3.63)}$$

The error function can be written by the help of the theorem and the definition of Frobenius norm as follows

$$\varepsilon^2(l) = \left\| X^T \Phi_{(m-l)} \right\|_F^2 = \left\| X^T \Phi_{(m-l)} P \right\|_F^2 = tr((X^T \Phi_{(m-l)} P)^T X^T \Phi_{(m-l)} P) = tr(\Lambda) \quad \textbf{(3.64)}$$

Note that, to obtain the minimum error, diagonal elements of Λ can only be the last $(m-l)$ singular values of matrix the X^T. The optimality is obtained when the right singular vectors are used as the orthogonal basis vectors and the error is the summation of the last $(m-l)$ singular values of X^T. That is,

$$\varepsilon^2(l) = tr(\Lambda) = \sum_{j=l+1}^{m} \sigma_j^2 \quad \textbf{(3.65)}$$

The POD method by using SVD is obtained when the right singular vectors are used as orthogonal base vectors such that

$$
\begin{aligned}
X^T_{(mxn)} &= U_{(mxm)} S_{(mxn)} V^T_{(nxn)} \\
X_{(nxm)} &= V_{(nxn)} S^T_{(nxm)} U^T_{(mxm)} \\
X_{(nxm)} &= V d_i \quad S^T_{(nxm)} U^T_{(mxm)} = \begin{bmatrix} d_1 & d_2 & \cdots & d_n \end{bmatrix}
\end{aligned}
\tag{3.66}
$$

In the case of nonlinear beam model, the columns x_i of system response matrix X are the displacements of the each node obtained by numerical simulations. For predetermined load scenarios, the corresponding coefficients d_i's are calculated for the each load case and stored, namely

$$
\begin{matrix} l_1 & l_2 & \cdots & l_n \\ \downarrow & \downarrow & \cdots & \downarrow \\ d_1 & d_2 & \cdots & d_n \end{matrix}
$$

Figure 3.7: Coefficients of Each Load Cases

When a load is applied which is different from the pre-determined load cases, corresponding coefficients must be calculated. In order to obtain the current coefficient, Lagrange interpolation functions are used such that,

$$
L_{n,m}(x) = \prod_{k \neq m}^{n} \frac{l - l_k}{l_m - l_k} \tag{3.67a}
$$

$$
L(x) = \sum_{m=1}^{n} l_m L_{n,m}(x) \tag{3.67b}
$$

3.4 Comparison of FEM and POD

In the following table, the comparison between the results of the finite element model and those of POD model is presented. The displacement boundary conditions are applied to the tip node of the beam model. The load cases are applied to a 4 element beam model.

The finite element model and POD model are solved for the following displacements

$$
l = 0.1, 0.2, 0.3 \cdots, 5 \quad (mm) \tag{3.68}
$$

Table 3.2: Comparison between FEM and POD

Trials	0.5		0.8		-4		3.75(Interpolation)		8(Extrapolation)	
Method	FEM	POD	FEM	POD	FEM	POD	FEM	POD	FEM	POD
DOF 1	0	-0.1524	0	-0.2667	0	-1.4859	0	-1.3906	0	-1.4435
DOF 2	0	0	0	0	0	0	0	0	0	0
DOF 3	0.0006	0.0006	0.0009	0.0009	0.0047	0.0047	0.0044	0.0044	0.0094	0
DOF 4	0	-0.1429	0	-0.2500	0	-1.3930	0	-1.3037	0	-1.3533
DOF 5	-0.1836	-0.1836	-0.2938	-0.2938	-1.4688	-1.4688	-1.3770	-1.3770	-2.9375	-0.0254
DOF 6	0.0006	0.0006	0.0009	0.0009	0.0044	0.0044	0.0042	0.0042	0.0089	0.0001
DOF 7	0	-0.1143	0	-0.2000	0	-1.1144	0	-1.0430	0	-1.0827
DOF 8	-0.3438	-0.3438	-0.5500	-0.5500	-2.7500	-2.7500	-2.5781	-2.5781	-5.5000	-0.0475
DOF 9	0.0004	0.0004	0.0007	0.0007	0.0035	0.0035	0.0033	0.0033	0.0071	0.0001
DOF 10	0	-0.0667	0	-0.1167	0	-0.6501	0	-0.6084	0	-0.6315
DOF 11	-0.4570	-0.4570	-0.7313	-0.7313	-3.6563	-3.6562	-3.4277	-3.4277	-7.3125	-0.0632
DOF 12	0.0003	0.0003	0.0004	0.0004	0.0021	0.0021	0.0019	0.0019	0.0041	0
DOF 13	0	0	0	0	0	0	0	0	0	0
DOF 14	-0.5	-0.5	-0.8000	-0.8000	-4.0000	-4.0000	-3.7500	-3.7500	-8.0000	-0.0691
DOF 15	0	0	0	0	0	0	0	0	0	0

When the transverse displacements are investigated, it can be seen that the POD model behaves exactly similar to the FEM model in the transverse direction. This fact can be seen as another advantage of the POD method. The specific dominant characteristics of the system can be extracted on purpose. The size of the high order model can be further decreased by eliminating the undesired degrees of freedom. Investigation of the table reveals that if the instantaneous applied displacement is in the interval of the load cases, POD matches FEM model. If the applied displacement is outside of the load cases defined in Equation 3.68, coefficients of the orthogonal base vectors are calculated using extrapolation. In that case, POD model fails to approximate FEM based model.

The computational time that is needed to complete a full simulation is investigated for both POD and FEM based models. The tip node deflection is taken as constant, the length of the beam is taken as 2.54 m with 25.4x25.4 mm^2 cross-sectional area and modulus of elasticity is taken as 206 GPa. The effect of increasing the number of elements in both POD and FEM based models are depicted in Figure 3.8, 3.9 and 3.10.

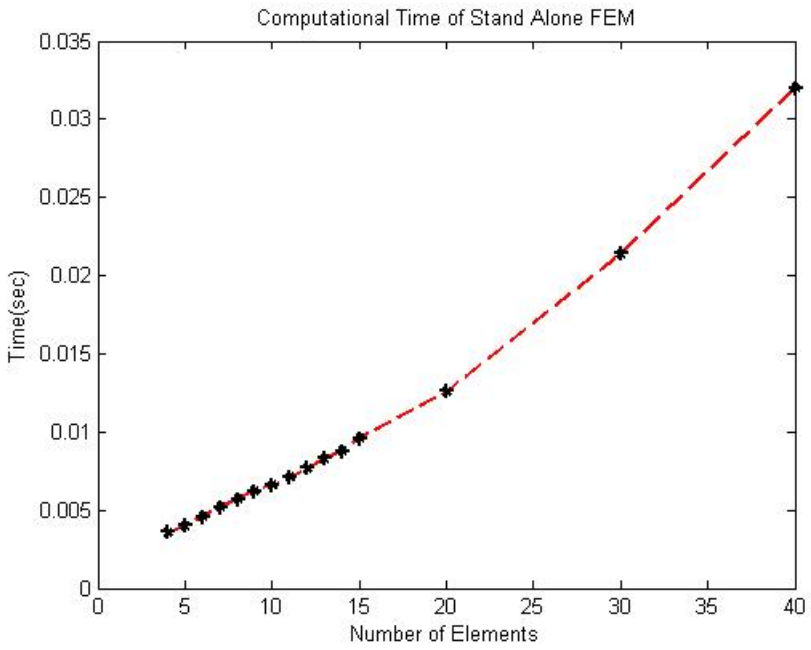

Figure 3.8: Computational Time Requirement of Stand Alone FEM Based Model

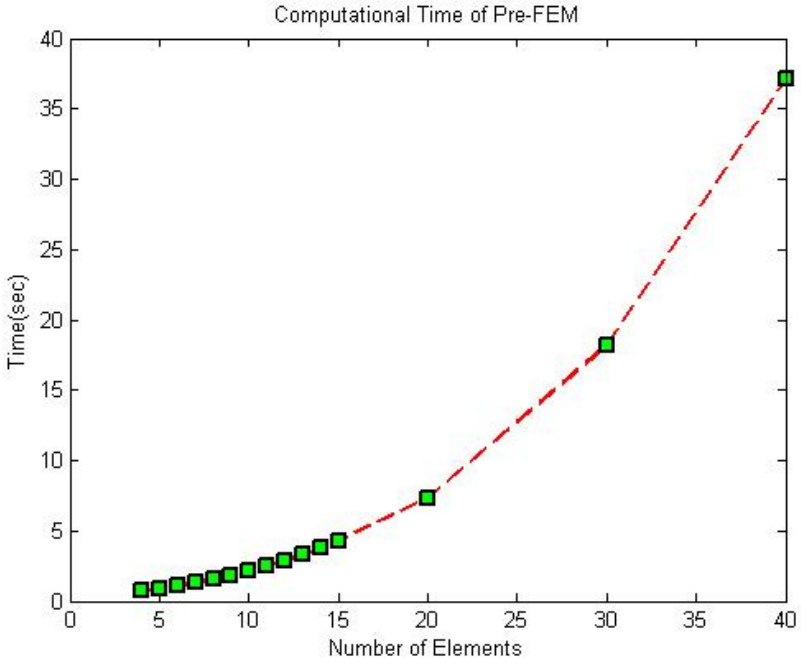

Figure 3.9: Computational Time Requirement of FEM Used As Preprocessor

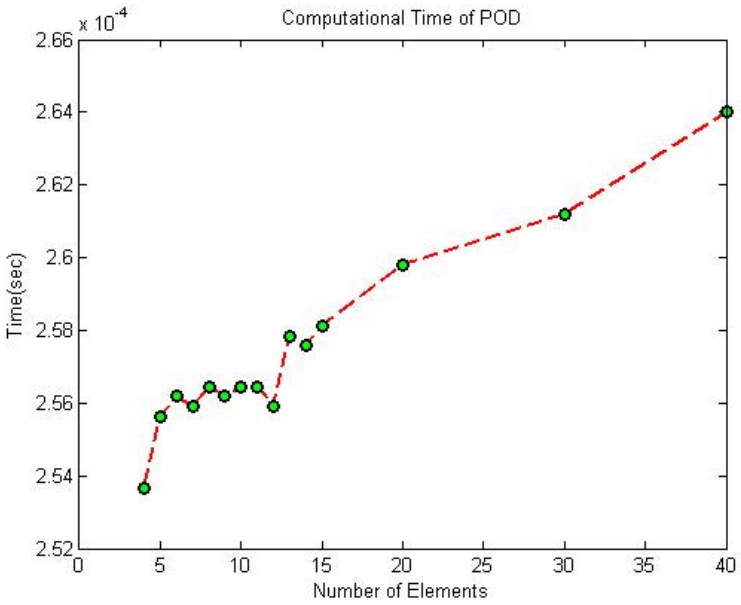

Figure 3.10: Computational Time Requirement of POD.

In numerical simulations of the reduced model, finite element model is used as a preprocessing stage and completed before the haptic simulation is started. Resulting forces and deformations are calculated using obtained coefficients and vectors. Computational time needed to calculate the response of the model is greatly reduced with this method. Investigations of the Figures 3.8, 3.9 and 3.10 revealed that the increase in the number of elements does not affect the computational demand of the POD based model. Preprocessing stage of the POD based model takes all the critical computational afford which does not affect the haptic simulation procedure. The fact that directly affects the computational afford needed for POD based model is the number of numerical simulations that are used for calculation of orthogonal base vectors and related coefficients. The computational behavior of POD based model is given in Figure 3.11.

All numerical simulations are completed on a desktop computer with Intel based CPU at 2 GHz and 2 GB of RAM on MATLAB environment.

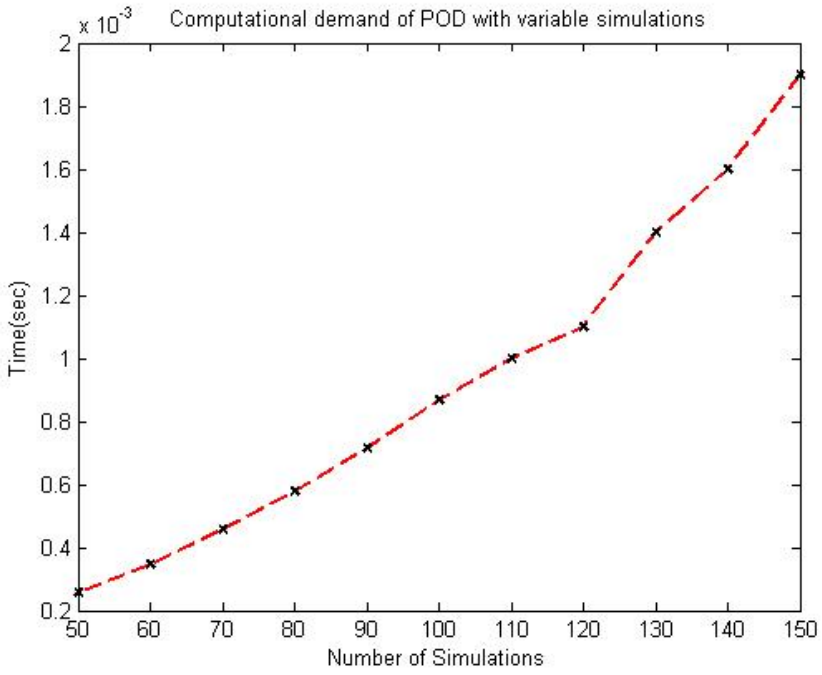

Figure 3.11: Time Requirement of POD with Variable Number Simulations

4. INTEGRATION OF BEAM MODEL TO A HAPTIC SYSTEM

In this chapter, developed models are integrated with Phantom Premium 1.5 High Force 6DOF haptic device. Device properties and integration algorithms are explained in detail. In a broad point of view, two distinct parts exist in integration procedure. The first part includes the mathematical modeling of beam and commanding the haptic interface. The second part is the visualization of both the model and haptic interface. The first part is programmed using OpenHaptics, Matlab and C++. OpenHaptics toolkit is a C++ library that is used to command the haptic interface. This library is composed of two application interface libraries (API), Haptic Device API (HDAPI) and Haptic Library API (HLAPI). The differences between these two APIs are the complexity and the granted level of access to the hardware of the haptic interface. HDAPI provides lower level of access than HLAPI. It enables the user to render forces directly, access to the encoders in bitwise manner. HLAPI provides high level access to the haptic interface. This API is built on top of HDAPI with an aim of hiding some complexity of low level programming. The users of HLAPI do not have to concern force calculation procedures or appropriate data structure creation and manipulations. Built-in functions of HLAPI deal with all the low level programming issues and the user is supposed to call these functions at right time in their algorithm. HLAPI also contains single point collision detection algorithm. This API is also easily implemented to existing graphics applications that is built using OPENGL. In some applications, HLAPI does not offer desired flexibility. In these cases, HDAPI should be used with high effort on programming issues. In this thesis, since a specific force response algorithm is developed, low level access to haptic interface is needed. HDAPI library is used for integration of the developed model with the specified haptic interface. The difference between two libraries is depicted in Figure 4.1.

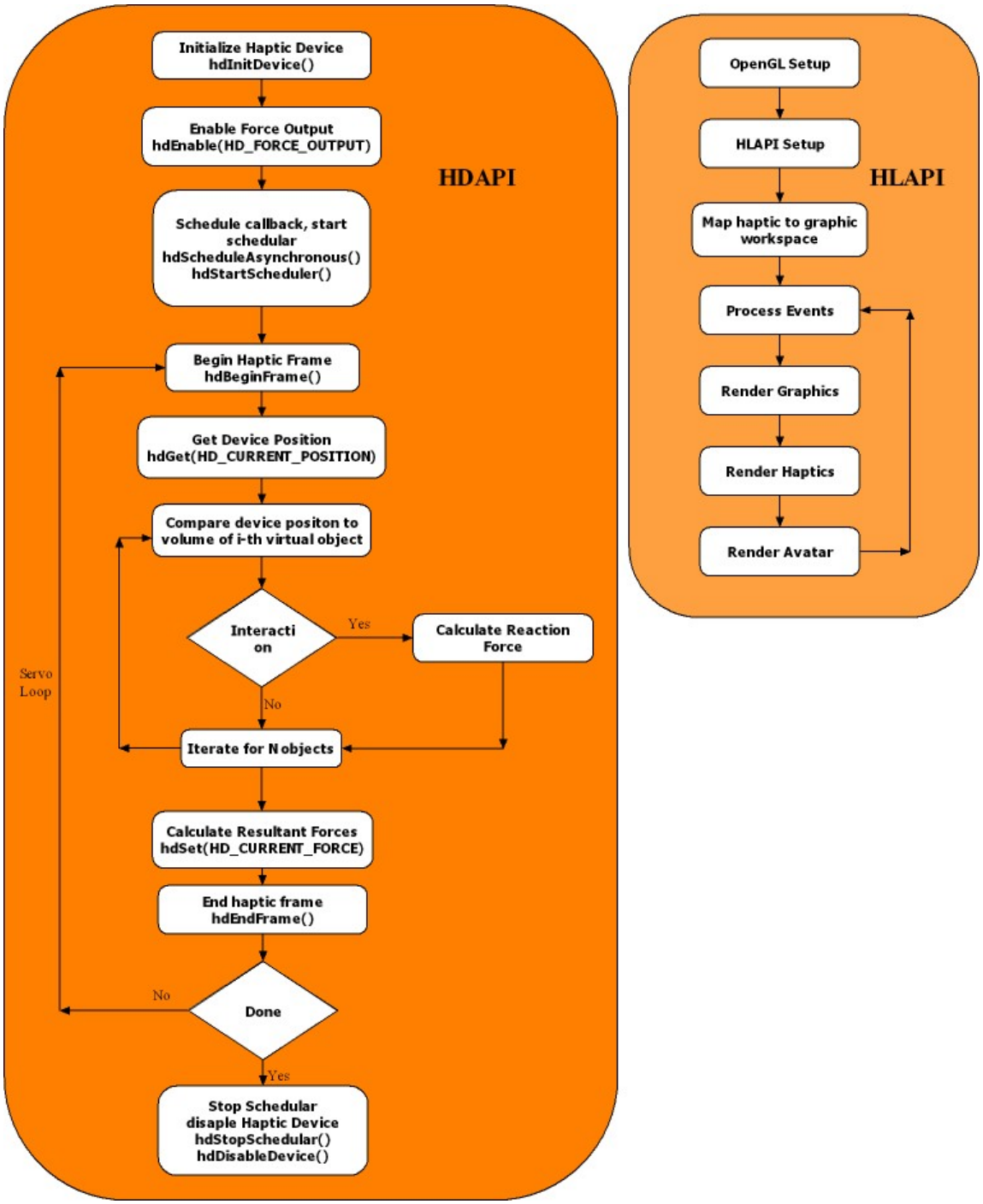

Figure 4.1: Complexity Comparison of HLAPI and HDAPI

4.1 Visualization of the Beam Model

Visualization of the beam model is programmed by OpenGL API that is based on C++. It is a cross-platform API used for generating 2D and 3D computer graphics. The basic operation of the algorithm is to accept geometrical primitives such as points, vertices, lines and transform them into pixels. In general, user creates virtual

objects and environments by describing specific properties such as light, perspective, color, dimensions and textures etc. In this thesis work, OpenGL is used to visualize both the modeled beam and haptic interface representation called avatar. In Figure 4.2 the modeled 2D environment and undeformed objects are depicted. The avatar is represented by a circular shape.

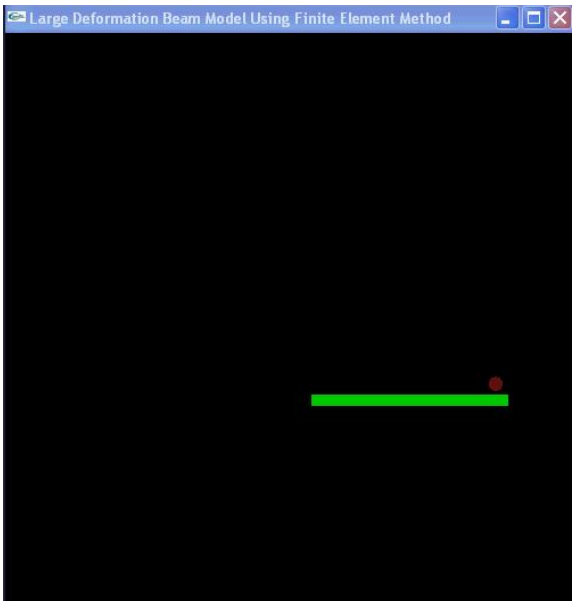

Figure 4.2: Visualization of Model

The results of the interactions between the avatar and beam model are depicted in the Figure 4.3. The calculated deformations are used to update the visual model.

4.2 Integration Algorithms

Large Deflection finite element model of nonlinear beam problem is implemented using diagram given in Figure 4.4. The movement of the user is measured by haptic interface and resulting positions and orientations are transferred to collision detection algorithm. If the collision event is detected, penetration amount is measured and sent to force response algorithms. Resulting deformations and reaction forces are calculated and sent to visual model and control algorithms respectively. Visual model is updated to give the effect of deformations on the model. The control algorithms embedded in HDAPI toolkit generate the resulting force and torque values on the tip of the haptic interface.

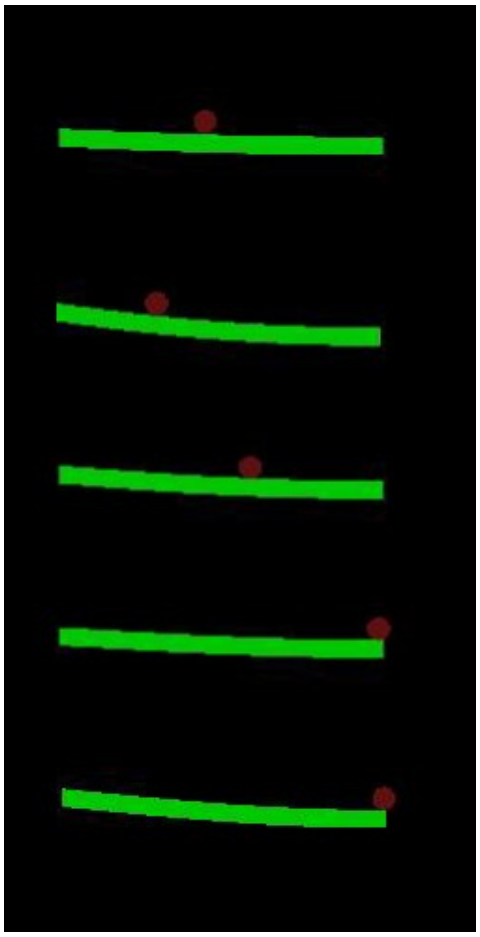

Figure 4.3: Various States of the Deformed Model

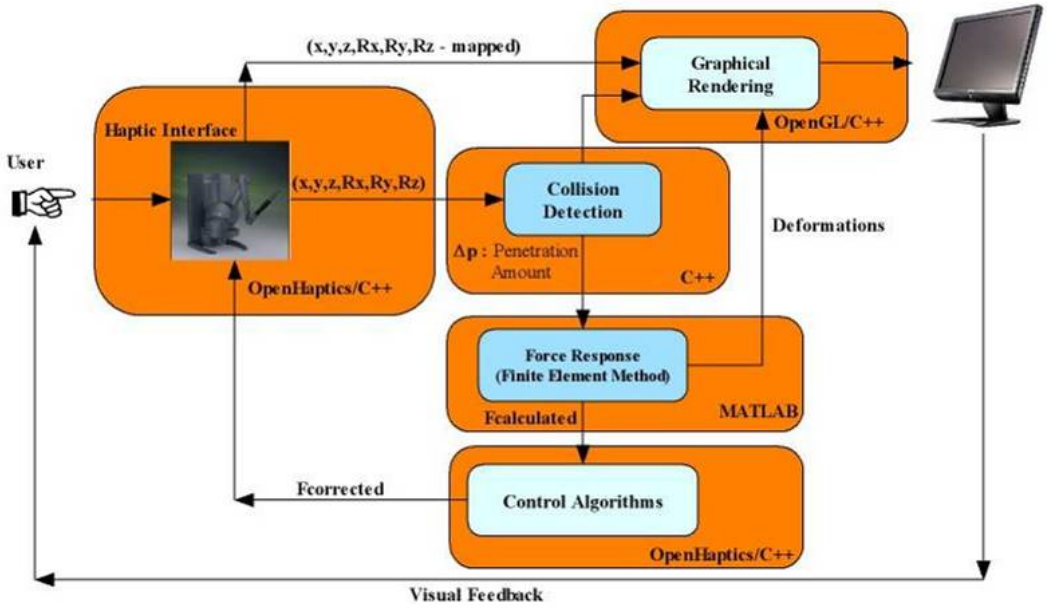

Figure 4.4: General Procedure of Interaction Diagram

The POD model is implemented in a similar structure. To this end, finite element algorithm is not used in real-time scheme. It is used as a pre-processing stage for the POD model. Force response algorithm has become calculation of related coefficients of orthonormal vectors as explained in Equation (3.67) and (3.68). Modified diagram is shown in Figure 4.4. The implementations of both models are programmed in object oriented fashion. The distinct parts of the general program are grouped into specific functions that ensure modularity of the software.

The integration procedures are implemented using Phantom Premium 1.5 High Force 6 DOF device shown in Figure 4.6

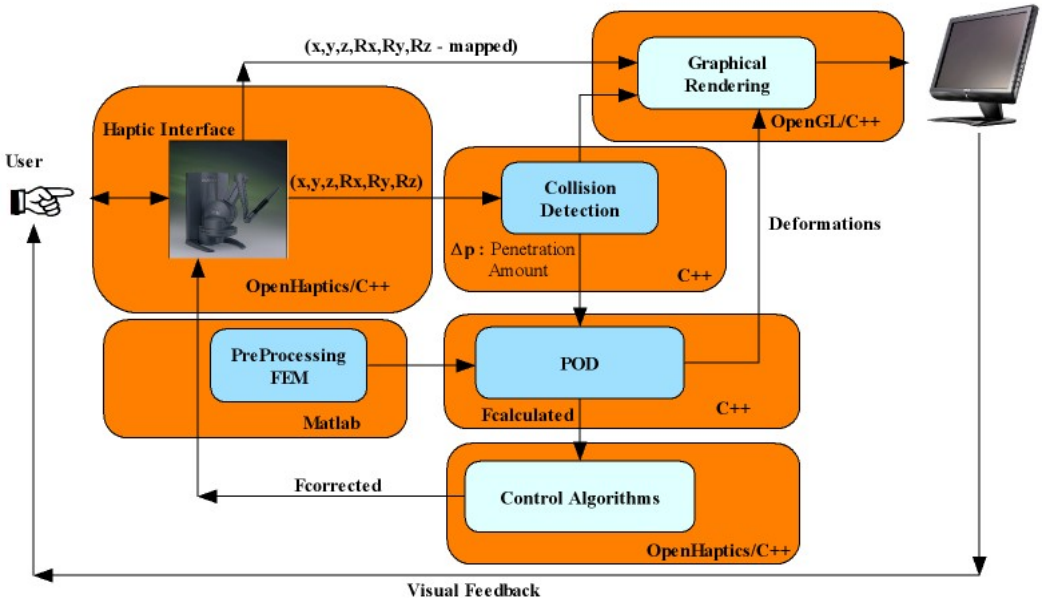

Figure 4.5: Modified Procedure of Interaction Diagram

Figure 4.6: Phantom Premium Desktop Device

5. CONCLUSION

In this study, haptic interfaces and haptic rendering concepts are introduced. The sub algorithms that compose the entire system are investigated.

In haptic applications, the most important problem that specifies the properties of the system modeled is the computational efficiency. High update rate demands of the haptic simulations limit the precision of virtually modeled objects. In case of non-physically based or linear models, the computational demands of the application are affordable. However, in order to obtain realistic haptic simulations, nonlinear models are needed with the drawback of additional burden on real time performance. As an application, nonlinear beam formulation is obtained and solved using finite element method and integrated successfully with the Phantom Premium 1.5 High Force 6DOF haptic device. Using Proper Orthogonal Decomposition method, high order beam model is approximated by a low order model. Investigations of the results have showed that POD method is successfully implemented and lower order approximation of the model is obtained. Lower order model has improved real-time performance of the overall system as expected. In addition, undesired degrees of freedoms of the model can be eliminated to further reduce the models that are intended to be used in haptic systems. The results of the numerical simulations have showed that POD fails to approximate FEM model when extrapolation is needed for the calculation of coefficients of orthonormal vectors. Extrapolation should not be allowed in POD based models. Since the haptic robot that is used to test models has force limitations due to the mechanics of the robot arm, the load case interval can be chosen appropriately that covers the safe working range of the robot.

REFERENCES

[1] **Otaduty, M.A. and Lin, M.C.,** 2005. Introduction to Haptic Rendering, *International Conference on Computer Graphics and Interactive Techniques*, Los Angeles, California, USA.

[2] **Minsky, M.**, 1995. Computational Haptics : The sandpaper system fo synthesizing texture for a force feedback display, *PhD Thesis*, MIT, Cambridge.

[3] **Zilles, C. and Sallisbury, K.**, 1995. A constraint based god object method for haptic display, *Proceedings of IEEE/RSJ International Conference on Inteligent Robotics and Systems*, Washington, DC, USA, August 1995, **3**, 3146.

[4] **Massie, T.M. and Salisbury J.K.**, 1994. The phantom haptic interface: A device for probing virtual objects, *Proceedings of ASME Haptic Interfaces for Visual Environment for Teleoperator Systems 1*, Chicago, IL, November 1994,**1** ,295-301.

[5] **Gosselin F., Martins, J.P, Bidard, C., Andriot, C., Brisset, J.**,2005. Design of a New Parallel Haptic Device for Desktop Applications, *Symposium on Haptic Interfaces for Virtual Environment and Teleoperated Systems*, 18-20 March, 189-194.

[6] **Barraf D.**, 1994. Fast Constact Force computation for non-penetrating rigid bodies, *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, Pittsburgh,USA, 23-24.

[7] **Daniela Constantinescu, Septimiu E. Salcudean, Elizabeth A. Croft**, 2004. Haptic Rendering of Rigid Body Collisions, *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'04)*, Vancouver, Canada, 27-28 March 2004, 2-8.

[8] **Basdogan C., De, S., Kim, J., Muniyandi, M, Kim, H., Srinivasan, M.**, 2004. Haptics in minimally invasive surgical simulation and training, *IEEE Computer Graphics and Applications*, **24**, 56-64.

[9] **Lindblad, A., Turkiyyah , G.**, 2007. A physically-based framework for real-time haptic cutting and interaction with 3D continuum models, *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, Beijing, China, 421-429.

[10] **K. Salisbury, F. Conti, F. Barbagli**, 2004. Haptic Rendering: Introductory Concepts, *IEEE Computer Graphics and Applications*, **24**, 24-32.

[11] **M. Bro-Nielsen,** 1998. Finite Element Modeling in Surgery Simulation, *Proceedings of the IEEE*, **86**, 490-503.

[12] **S. Cotin, H. Delingette**, 1998. Real-time Surgery Simulation with Haptic Feedback using Finite Elements, *Proceedings of the 1998 International Conference on Robotics and Automation*, Leuven, Belgium, 16-20 May, **4**, 3739-3744.

[13] **Basdogan C.,** 2001. Realtime simulation of dynamically deformable finite element models using model analysis and spectral lanczos decomposition methods, *Proceedings of the Medicine Meets Virtual Reality (MMVR'2001) Conference*, Irvine,CA, Jan 24-27.

[14] **Burdea, G. C., Zhuang J. A., Rosko, E., Silver, D., Langrama, N.**, 1992. A portable dextrous master with force feedback, *Presence: Teleoperators and Virtual Environments*, Cambridge, USA, Winter 1992, **1**, 18-28.

[15] **Bardorfer, A., Munih, M., Zupan, A., Primoęzi, A.**, 2001. Upper Limb Motion Analysis Using Haptic Interface, *IEEE/ASME Transactions on Mechatronics*, **6**, 253-260.

[16] **Yang, G., Ho,H.L., Chen, W., Lin, W., Yeo,S.H., Kurbanhusen, M.S.**, 2004, A haptic device wearable on a human arm, *IEEE Conference on Robotics, Automation and Mechatronics*, **1**, 243-247.

[17] **Hayvard, V. and Astley**, **O.R.**, 1996. Performance measures for haptic interfaces, *1996 Robotics Research: The 7th International Symposium*, Springer – Verlag, 195-207.

[18] **Lin M.C., Manocha D.**, 2003. Collision and proximity queries, in *Handbook of Discrete and Computational Geometry*, pp. 1-21, CRC Press, Boca Raton.

[19] **Lin M.C, Gottschalk, S.**, 1998. Collision detection between geometric models: a survey, *In Proc. of IMA Conference on Mathematics of Surfaces*, 37-56.

[20] **Basdogan C., C.H. and Srinivasan, M.A.,** 2001. Virtual Environments for Medical Training: Graphical and Haptic simulation of Common Bile Duct Exploration, *IEEE/ASME Transactions on Mechatronics*, **6**, 269- 285.

[21] **Eberle D.**, Primitive tests for collision detection, PDI/Dreamwork R&D.

[22] **Held, M., Klosowski, J., Mitchell, J.S.B.**, 1996. Real-time collision detection for motion simulation within complex environments, *In Proceedings of ACM SIGGRAPH 96 Visual*, Louisiana, USA, 151.

[23] **Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M.P., Faure, F., Magnenat-Thalmann, N., Strasser, W., Volino, P.**, 2005. Collision Detection for Deformable Objects, *Computer Graphics Forum*, **24**, 61-81.

[24] **Ginson, S.F.F, Mirtich, B.**, A survey of deformable modeling in computer graphics, 1997. *Mitsubishi Electric Research Laboratories*, Massachusetts, USA.

[25] **Lee, Y., Terzopoulas, D., Waters, K.**, 1993. Constructing physics based facial models for individuals, *In Proceedings of Graphic interface*, Toronto, May 1993, 1-8.

[26] **Chadwick, C., Haumann, D., Parent, R.**, 1989. Layered construction for deformable animated characters, *In Proceedings of the 1989 ACM SIGGRAPH conference*, NY, USA, July 1989, 243-252.

[27] **Liu A., Tendick F., Cleary K, Kaufmann C.**, 2006. A survey of surgical simulation : Applications, Technology and Education, *Presence*, **12**, 599-614.

[28] **Reddy J.N.,** 2006. An Introduction to Finite Element Analysis, MacGraw Hill, New York.

[29] **Reddy J.N.**, 2004. An Introduction to Nonlinear Finite Element Analysis, Oxford University Press, New York.

[30] **Misra S, Okamura A., Ramesh K.T.**, 2007. Force feedback is noticeably different for Linear versus Nonlinear elastic tissue models, *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*,USA, 519-524.

[31] **Holmes, P., Lumley, J. L., and Berkooz, G.**, 1996. Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, NY.

[32] **Fenny, B. F., Kappagantu, R.**, 1998. On the physical interpretation of Proper Orthogonal Modes in Structural Vibration, *Journal of Sound and Vibration*, **211**, 607-616,.

[33] **Lenaerts, V., Kerschen, G., Golinval, J. C.**, Proper Orthogonal Decomposition for Model Updating of Nonlinear Mechanical Systems, *Mechanical Systems and Signal Processing*, **15**, 31-33.

[34] **Hung, E. S., Senturia, S. D.**, Generating Efficient Dynamical Models for Microelectromechanical Systems from a few Finite Element runs, *Journal of Microelectromechanical Systems*, **8**, 280-289.

[35] **Liang, Y. C., Lee, H.P., Lim, S. P., Lin, W. Z., Lee, K. H., Wu, C. G.**, 2002, Proper Orthogonal Decomposition and Its Applications Part 1: Theory, *Journal of Sound and Vibration*, **252**, 527-544.

CURRICULUM VITAE

Yaşar Paça was born in Germany in 1982. He graduated from Istanbul Kabataş High School in 2000. He graduated from Izmir Institute of Technology, Mechanical Engineering Department in 2005. He has been a graduate student in System Dynamics and Control Department of ITU since 2005.