

**CONTACT DETECTION, COLLISION FORCES AND
FRICTION FOR PHYSICALLY BASED VIRTUAL
WORLD MODELING**

by

Irfan A. Essa

Bachelor of Science,
Illinois Institute of Technology,
May 1988

Submitted to the Department of Civil Engineering
in partial fulfillment of the requirements for the degree of

Master of Science

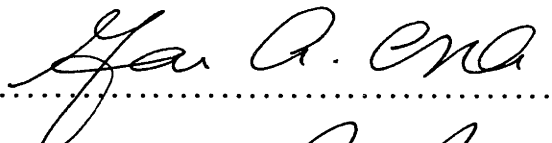
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1990

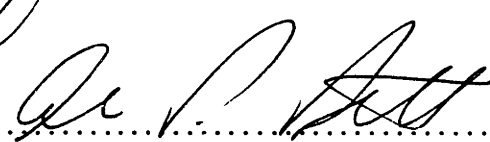
© Massachusetts Institute of Technology 1990

Signature of Author



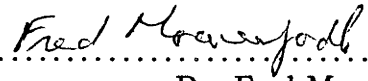
May 15, 1990

Certified by



Dr. Alex Pentland
Associate Professor of Computers, Communication and Design Technology,
M. I. T. Media Laboratory
Thesis Supervisor

Certified by



Dr. Fred Moavenzadeh
Professor of Civil Engineering
Thesis Reader

Accepted by



MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

Chairman, Departmental Committee on Graduate Students

JUN 15 1990

ARCHIVES

CONTACT DETECTION, COLLISION FORCES AND FRICTION FOR PHYSICALLY BASED VIRTUAL WORLD MODELING

by

Irfan A. Essa

Submitted to the Department of Civil Engineering
on May 15, 1990, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Detection of contact and calculation of collision forces is an important problem in any kind of physical multi-body simulation. For computer graphics and physically based animation it is especially important to devise methods that combine efficient computational methods with powerful existing graphics tools if one is to obtain a realistic, real-time virtual world. Most of physical simulations are computationally expensive, and thus, it is difficult to set up any simulations that are stable and have a real-time response.

Efficient methods for contact detection and response for physical interactions of deformable objects in physically based virtual world environments are presented. Contact, collision and friction of objects in virtual worlds are specifically addressed in the framework of differential geometry and finite element modeling. A statistical approach is introduced for estimation and control of the physical simulation. These methods employ statistical estimation of contact between stochastically defined surfaces and linear control theory for estimation and control to obtain stable forward time simulations. By mapping from the statistical domain to the geometric domain and then to the physical domain, we have been able to obtain efficient physical simulations of multi-body systems.

Thesis Supervisor: Dr. Alex Pentland

Title: Associate Professor of Computers, Communication and Design Technology,
M. I. T. Media Laboratory

Thesis Reader: Dr. Fred Moavenzadeh

Title: Professor of Civil Engineering

Laboratory: M.I.T. Media Laboratory

*Dedicated to
my Mom and Dad
for twenty four years of love, dedication and inspiration*

Acknowledgments

This thesis and the research which it describes would not have been accomplished had it not been for the guidance and aid of multitude of individuals.

First, and perhaps foremost, I wish to acknowledge the contribution of Professor Alex Pentland for providing invaluable guidance, advice, support and criticism from the conception of this research idea to the compilation of this thesis.

Secondly, I would like to thank Professor Fred Moavenzadeh for his support and guidance and Professors David Zeltzer and Jerome Connor for their valuable technical and moral support.

A great many thanks to Dr. John Williams for introducing me to this project and for his invaluable guidance through it.

I would like to thank each and every member of my group, the Vision and Modeling Group at the M. I. T. Media Lab, the group I have worked with while doing this research, for their help in every aspect of this research from concepts to writing of the thesis. I would also like to thank the members of the Computer Graphics and Animation Group at the M. I. T. Media Lab, with whom some technical debates helped me clarify many of my problems.

Special thanks to Anand Bodapati, as it was he who initiated my interest in the stochastic and estimation approach to physically based modeling .

A special note of thanks to Army Research Office, Under the University Initiatives Program, for funding this research.

And last but not the least, a hearty thanks to the most important people in my life, my family. If it were not for the caring, love, affection and continuous support of my mom and dad, my sisters and brother, none of this would be at all possible. A special thanks to a very special person who carries a special meaning to my life, my fiancée Shani.

Contents

1	Introduction	13
1.1	Background	14
1.2	Contact and Collision	14
1.3	FEM Perspective	16
1.4	Multi Body Dynamics	17
1.5	Stochastic Approach	17
1.6	Control Theory Approach	17
1.7	Current Implementations	19
2	Contact Detection	20
2.1	Shape and surface representation	20
2.1.1	Mathematical formulation	21
2.1.2	Superquadric Representation	22
2.1.3	Implicit Representation of Superquads	23
2.1.4	Surface normals of superquadric functions	24
2.2	Contact for Deformed and Undeformed Objects	24
2.3	Curvatures for Improved Contact Detection	27
2.3.1	Formulation for curvatures of parametric curves	27
2.3.2	Lowest point of penetration	28
2.3.3	Local minimum point of the target	29
3	Finite Element Method Formulation	32
3.1	Formulation of Displacement Based Finite Element Method	33

3.1.1	Basic Steps of the Formulation	33
3.2	General Derivation Of Finite Element Equations	33
3.2.1	Principle of Virtual Work	35
3.2.2	Shape or Interpolation Functions	35
3.2.3	Develop Stiffness and Load Matrices	36
3.2.4	Include Inertia Forces	37
3.2.5	Include Damping Forces	38
3.3	Static Analysis	38
3.4	Dynamic Analysis	39
3.4.1	Direct Integration Methods	39
3.4.2	Mode Superposition	41
3.4.3	Determination of minimum time step	47
4	Collision Response	50
4.1	Previous Work	50
4.2	Penalty Method	52
4.3	FEM Formulation of the contact problem	53
4.4	Response for Deformable Bodies	56
4.5	Computational Geometry : The Correct Direction Of Response	61
4.6	Surface Penetrations	62
5	Friction	63
5.1	Previous work	64
5.2	Coulomb's Law of Friction	64
5.2.1	Friction Cone and Friction Angle	65
5.3	Formulation of Frictional Resistance	67
5.3.1	Translation in Friction	69
5.3.2	Rotation in Friction	70
5.4	FEM formulation	70
5.5	Friction for deformable objects	72
5.6	Surface Patches and Roughness	73

6	Stochastic Theories For Measure Of Roughness	75
6.1	Texture, Fractals, and MRF	76
6.1.1	Fractals	76
6.1.2	Markov Random Fields	77
6.2	Surface Roughness as a Random Process	78
6.3	Interpenetration of two rough surfaces	79
6.4	Stochastic Estimation of Stress Distribution on Contact	85
7	Application of Control Theory	87
7.1	Problems with Physical Simulations	87
7.2	Feedback Control	90
7.3	Filtering	91
7.3.1	One Dimensional Filtering	92
7.4	Wiener Filtering	94
7.4.1	Wiener Filter : Formulation	96
7.4.2	Linear Prediction with Wiener filters	99
7.5	Kalman Filtering	101
7.5.1	Kalman Filter : Observer	102
7.5.2	Gain Factor and Variance Equation of Kalman filters	103
7.5.3	Steady state Kalman filter	105
7.5.4	Algorithm for optimal control using Kalman filters	105
7.6	Filtering In Forward Simulations	108
8	Conclusions	110
A	Deformations and Modal Values	113
A.1	Implicit Function with Deformation Mapping for Modeling Primitives . . .	113
A.2	From Deformed Space to Undeformed Space	115
B	Finite Element Method Formulations	117
B.1	Setting up H and B matrices	118
B.1.1	To set up Interpolation functions and the Interpolation matrix . . .	121

B.1.2	Jacobians	124
B.2	Numerical Integration	126
B.3	Implementation with Superquads	127
C	Stiffness Matrices and Stiffness values	129
C.1	Stiffness values at a node	129
C.2	Stiffness values at any point inside segment	131
D	Random Variables : Definitions	133
D.1	Random Variables and Random Processes	133
D.2	Statistical Characteristics of Random Processes	135
D.3	Convolution Identities	136
D.4	Gaussian Distribution	137
E	Control Theory : Definitions	139
E.1	System Dynamics : Definitions	139
E.2	Control Theory : Definitions	140
E.3	White Noise and Linear System Response	141

List of Figures

1-1	A Typical THINGWORLD session	18
2-1	Representation of Coordinate System of Superquads.	22
2-2	Superquads with various squareness values	23
2-3	Deformed Superquads	25
2-4	Target and Contactor.	26
2-5	Objects in Contact.	26
2-6	Objects in Contact and Deformed.	28
2-7	Point of Contact (two-dimensional view).	29
2-8	Ellipsoid for a contact point (two-dimensional view).	30
3-1	General Three Dimensional Body.	34
4-1	Penalty Method.	51
4-2	Conditions prior to contact	53
4-3	Conditions at contact	54
4-4	Forces acting on target and contactor	55
4-5	Model for Collision and Contact Forces	57
4-6	A Simplified Model for Collision and Contact Forces	58
4-7	Deformed Contact.	60
4-8	Directions of Response.	61
5-1	Coulomb's Law	66
5-2	Types of Frictional Resistance	66
5-3	Friction Cone and Friction Angle	67

5-4	Force and Moment of Sliding Friction	68
5-5	Normal and tangential tractions on the contactor	71
5-6	The Mass-Spring-Dashpot Model for Collision/Contact/Friction	72
5-7	Normal and Surface Stress distributions	73
6-1	Two Dimensional Shape with Stochastic Distribution	78
6-2	Contact of two “rough” objects	80
6-3	Rough Surface, from two principle directions of the object.	81
6-4	Using a random variable to estimate penetration on contact.	82
6-5	Contact of two spheres with Gaussian distribution for surface roughness . .	83
6-6	Stress distributions on collision and stochastic penetration.	85
7-1	System response; expected and observed	88
7-2	An example of contact, A cube lying on the floor (a) Before simulation (time step = 0) and (b) during simulation (time step = 20) [no contact]	89
7-3	A 50 time step simulation of contact	90
7-4	Forces on the 26 nodes of a cube lying on the floor	91
7-5	Repeated box filters of order 1-4	92
7-6	Forces on the node 11 of a cube lying on the floor, 50 time steps (a) No <i>filtering</i> and (b) First order <i>filtering</i>	93
7-7	Forces on the node 11 of a cube lying on the floor, 50 time steps (a) Second order <i>filtering</i> and (b) Third order <i>filtering</i>	94
7-8	Forces on the node 11 of a cube lying on the floor, 50 time steps, First, Second and Third order <i>filtering</i> and no <i>filtering</i>	95
7-9	Block diagram representation of Kalman Filter	102
B-1	A simple bar element	118
B-2	One Dimensional Element	122
B-3	Two Dimensional Element (Nine nodes shown)	123
B-4	Three Dimensional Element (Twenty nodes shown)	124
C-1	Distribution of Stiffness from nodes to a point on the segment.	131

D-1 Convolutions of two *random variables* 134

List of Tables

2.1	The inside-outside function of a superquad	24
3.1	Central Difference Method	41
3.2	Newmark Integration method	42
7.1	Step by step method for designing a Kalman filter.	105
A.1	Modal Deformation Matrix Values in terms of Modal Amplitudes	114
A.2	Physical Basis of Modal Amplitudes	115
B.1	The interpolation functions of four to nine variable number nodes for two dimensional elements	123
B.2	The interpolation functions of eight to twenty variable number nodes for three dimensional elements	125

Chapter 1

Introduction

A physically based virtual world, is a synthetic world based on the principles of physics that we see around us in everyday life. This “world” contains virtual objects that are expected behave in the virtual environment as real objects would in the real world. This emphasis leads into a study of physics and mechanics, as these are the essential problems for this kind of modeling.

One of the most important issues for a virtual world with multiple objects is physical interactions of these objects. The problem of contact detection, though intensely studied by researchers in the computer graphics, robotics, materials and mechanical engineering, has not been completely solved. However there are a lot of results that will help characterize the contact problem and solve it in certain cases. In a physically based virtual world it does not, however, suffice to just detect if there is a contact. Further work to see what kind of contact exists and how much surface penetration has resulted from the contact is essential. Contact response due to a contact and forces due to sliding constitute especially important aspects of a physically based environment.

Efficient detection of collisions and characterization of response are the chief goals of this thesis.

1.1 Background

As mentioned above, contact detection and collision response is essential in multi-body simulations. As interest in developing virtual worlds escalates [1, 2, 3, 4] in computer graphics, computer animation and vision and modeling, much research has been devoted to the study of the theories of contact detection, collisions and surface penetrations from the fields of civil and mechanical engineering, and implementing them for computer animations and modeling. The problem faced here is that most of the collision detection algorithms developed in these fields, and there are a large number of them, are computationally expensive and in many cases non-linear [5]. The focus of the computer graphics community has been to develop time and cost efficient algorithms, keeping in mind their need for realism, approximate accuracy and visual acceptability. This has led to approximate methods and linearizing techniques for non-linear systems. Some of these techniques have been adopted from various engineering disciplines [6, 7, 8].

In this thesis most of the emphasis will be on the physical interactions of objects in virtual worlds. For this purpose some formulations for dynamic simulations using finite element method will be presented. These formulations are required, because the essence of this work is to develop concepts of virtual modeling with deformable objects.

1.2 Contact and Collision

For computer animations of interactions of bodies in multi-body systems, two specific issues are at hand, detection and response [9]. Detection is primarily a *kinematic* problem, which involves the relative positions of objects (bodies) in the virtual world while response is a *dynamic* problem, which involves predicting behavior according to laws of physics (Beer and Johnson [10] Chap 12). Moore and Wilhelms [9] discuss at length the problems with contact detection and response for computer animations and present algorithms for collision detection that deal with triangulated surface representations of objects and rigid polyhedra. These algorithms are simple and robust but for most cases are too computationally expensive to be feasible. In Chapter 2 various methods of contact detection will be presented and it will be shown how applications of implicit function geometric descriptions and differential

geometry can be adopted to reduce computation. Implicit functions will be used describe the geometric and modeling primitives, and the advantages of these will be emphasized. Most of the concepts presented can easily be generalized for explicit functions , with an additional cost that goes with these functions for calculations in differential geometry [9, 11].

The *response* issue comes to play after the contact has been detected. Any system that uses dynamic simulations must respond to collisions realistically and automatically. The collision response has essentially two facets, the magnitude and the direction of response. The collision response must satisfy the *laws of conservation of angular and linear momentum*. Moore and Wilhelms presented [9] algorithms for collision response for rigid bodies. In Chapter 4 new and efficient methods for collision response for non-rigid deformable objects, which will account for the stress distributions across the surfaces due to deformations, directly relating them to the material properties of the objects [12] will be presented. These methods can easily be generalized to rigid-bodies.

An specially important type of response is the frictional force generated by interpenetrating or touching bodies. If contact or surface penetration is detected a force (response) against the direction of motion parallel to the surface needs to be automatically imparted on graphic elements of our virtual world for realistic behavior. The concepts of friction have not been completely implemented in the computer graphics/animation fields due to problems in determination of contact patches and the computational complexity involved with it [13]. These concepts are explained in detail, from the basics of *Coulomb's law* to stress distributions on the surfaces on contact in Chapter 5. The friction response problem will be presented in detail in the following chapters (5,6 and 7) and an analytical description of the frictional forces and statistical estimates of roughness of colliding/sliding objects will be derived and used to enhance these simulations.

A very important consideration in animations and especially physically based forward time simulations, is control. The ability to control the instability that has been added on to the system by approximations that were made by the dynamic simulation module is a problem that has not been addressed in virtual world simulations. A solution to this problem will be presented in Chapter 7.

1.3 FEM Perspective

For a force based virtual environment it is essential to develop a system for transfer of forces between bodies and within bodies. This is especially important for deformable objects. For deformable objects the stress and strain behavior of an individual object has to be modeled. This requires discretization of the objects and study of the forces (stresses) and displacements (strains) of these discrete elements of the object. For the modeling of deformable objects this work extensively relies on the in depth work done in Finite Element Analysis and Discrete Element Method [14, 15, 16, 17, 18].

Finite Element Method (FEM) ¹ is a numerical procedure for obtaining solutions of differential equations in engineering analysis. It has two main subdivisions. First, discretize an object and obtain its displacements and forces as per its structural framework, and second, use the continuum of the discretized elements to get the solutions for the whole body [19].

Fields of civil, mechanical, aerospace and materials engineering are enriched in methods employing in FEM and Discrete Element Modeling (DEM) for these kind of applications. The problem, however with FEM is that it is computationally quite expensive. Though a structural or mechanical designer can wait a few minutes to get the results of his analysis a user of a virtual world system would only be pleased with a system that simulates a physical process as he would expect it in real world. Implementation of FEM for virtual worlds is becoming an important topic now as the computer hardware is becoming more powerful, however the ability to do all the in depth computation, that is required for FEM, at every time step is still not possible. The problems of the computational complexity become more and more apparent as we think of virtual worlds with complex scenes [13]. Some formulations of FEM for *physically based virtual world systems* will be discussed and at the same time some methods to deal with these in more approximate method will be presented in Chapter 3.

¹Finite Element Method will be mostly referred to as FEM and Discrete Element Method will be referred to as DEM.

1.4 Multi Body Dynamics

For multi-body dynamics for virtual world systems, time and cost efficient methods that are accurate within reason are desired. One special method of FEM that has received a lot of attention in virtual world modeling in the last couple of years is the modal dynamics formulation as presented by Pentland and Williams [6, 16]. This formulation of dynamic simulation will be used for most of the implementations discussed here and a discussion of this method with respect to the mode superposition method that is used for dynamic analysis of systems in FEM will be presented [18]. This discussion will emphasize on the importance of adopting FEM principles for *virtual world* modeling and how computationally expensive methodologies of FEM could be adopted, with some modifications for applications in *physically based modeling*.

1.5 Stochastic Approach

For a multi-body dynamics system interaction of bodies and their response to this interaction is a function of the roughness of the surface. This surface roughness in a virtual world is typically modeled using a stochastic. The intention here is not only to model the roughness but to use this roughness to estimate surface penetrations and determine the contact patch. This surface roughness model could be used to “shape” the stress distribution of the interpenetrated surfaces and hence, could be used for the *response* computations as discussed in Chapters 2 and 6.

1.6 Control Theory Approach

One major problem with a force based forward dynamics system is the stability of the solution. To keep a virtual world environment real time or at least time efficient, we are forced to use large time-steps for simulations and coarse sampling of the virtual (graphic) objects. Therefore, in forward dynamics systems, with the constraints as mentioned above, it is almost impossible to avoid introducing a significant error into the dynamic simulation. Hence to get the stability that will make the system more realistic, it is necessary to consider

concepts from control theory.

By using simple concepts of overshoot, time delay and lag, controllability and observability filtering, prediction and control, the dynamic response of the system can be controlled or “reshaped” to discount the error and noise that was introduced by the above mentioned restrictions. Some simplified methods of controlling the dynamic response of the system that will solve the problem of instability will be presented in Chapter 7.

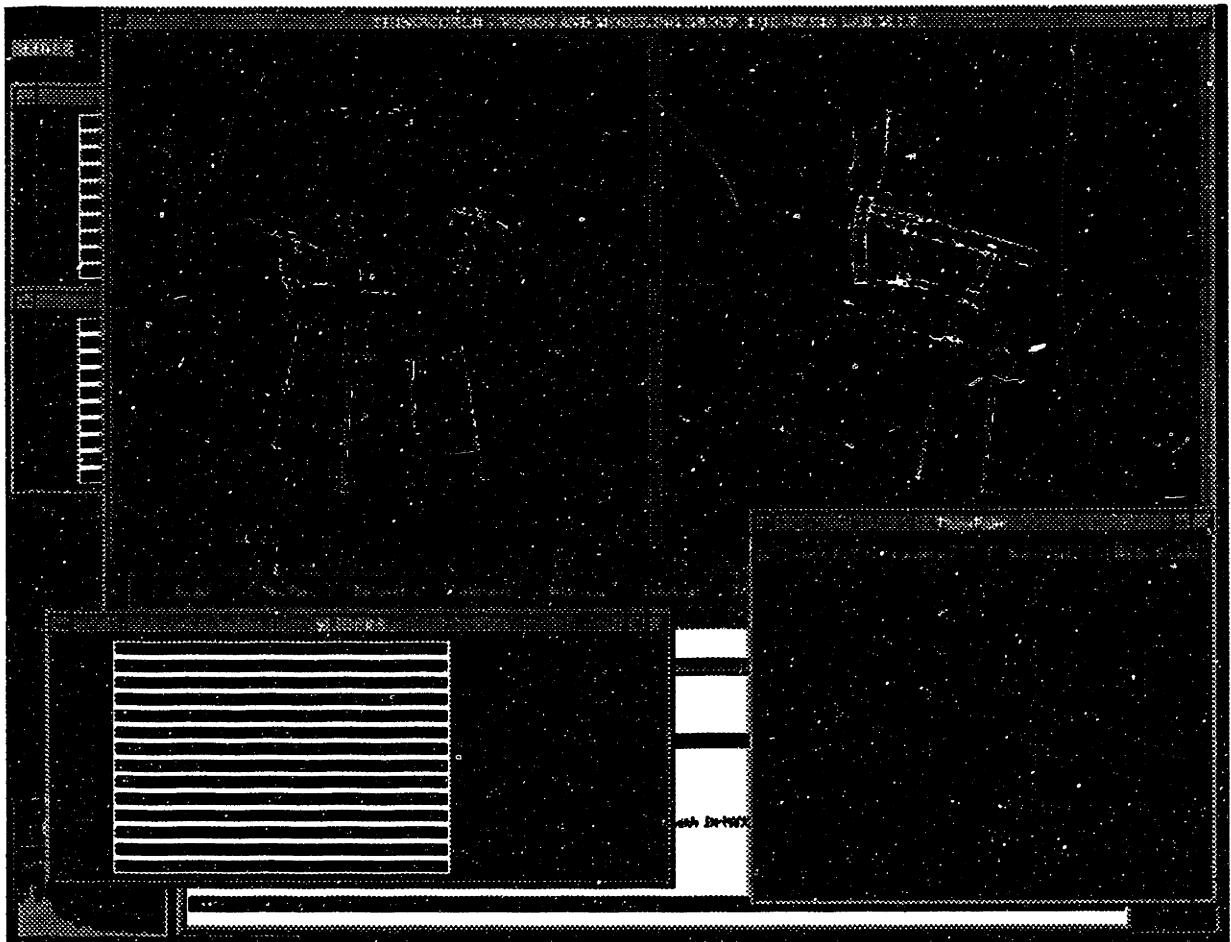


Figure 1-1: A Typical THINGWORLD session

1.7 Current Implementations

Most of the concepts that are discussed here are implemented on the *Thingworld Modeling System* [20, 21].² This system uses the Modal Vibrations Method [6, 16] for its dynamic simulations and the superquadric functions [22] are its geometric/shape primitives. *Thingworld* runs on Sun3s (with or without the floating point 68881 coprocessor) and Sun4s (Sparc chip) with a TAAC Board. Most of the code is a mixture of Lucid Lisp and C, where C is used for numerical calculations and Lisp for object orientated control. The graphics are run on X11 Release 4.³ A typical session of Thingworld on a Sparcstation is shown in Figure 1-1.

²Thingworld Modeling System is a ©1985,86,87,88,89,90 Alex Pentland, ©1989,90 Vision and Modeling Group, The Media Lab, M.I.T..

³Sun3, Sun4, TAAC and Sun Lucid Lisp are trademarks of Sun Microsystems X11 R4 is ©X Consortium M.I.T.

Chapter 2

Contact Detection

The basic principle of contact detection in virtual world modeling is to determine if the bodies are penetrating each other. This requires knowledge of the surface and shape representations of objects and their relationships to each other in the geometric domain. Computational geometry is used to calculate the curvatures and normals of these surfaces, which then helps resolve the exact locations of penetrations. Since it is desired to deal with deformable objects, a deformation mapping on these surfaces is required, which in turn requires application of a deformation map to the computations of surface normals and curvatures.

2.1 Shape and surface representation

With the increase in interest of numerical methods for three dimensional design and representation, parametric shape representations are proving to be of more and more importance in computer graphics, animation, vision and modeling and many other disciplines. These parametric representations provide a mathematical formulation, which gives more intuitive detail in setting up of three dimensional environments, which are easily accessible for shading, texture, displacement maps, imaging, finite element meshes and many other numerical systems. Using simple three dimensional parametric modifiers for transformations extends the ranges of geometric shapes, deformations can also applied in the same way [23, 6, 16]. The *superquadric functions* with transformations and deformations extend the geometric primitives of quadric surfaces and parametric patches to give a whole new family of flexible

and deformable three dimensional forms. The biggest advantage of this representation is the ability to alter the shape by modifying few interactive parameters.

2.1.1 Mathematical formulation

A formulation for three dimensional parametric shape functions is developed as follows. Consider two 2d parametric curves,

$$\begin{aligned} \mathbf{h}(\omega) &= \begin{bmatrix} h_1(\omega) \\ h_2(\omega) \end{bmatrix} & \omega_0 \leq \omega \leq \omega_1 \\ \mathbf{m}(\eta) &= \begin{bmatrix} m_1(\eta) \\ m_2(\eta) \end{bmatrix} & \eta_0 \leq \eta \leq \eta_1 \end{aligned} \quad (2.1)$$

The spherical product of these two curves gives the three dimensional surface,

$$\mathbf{X} = \mathbf{m} \otimes \mathbf{h}$$

$$\mathbf{X}(\eta, \omega) = \begin{bmatrix} m_1(\eta)h_1(\omega) \\ m_1(\eta)h_1(\omega) \\ m_2(\eta) \end{bmatrix} \quad \begin{array}{l} \eta_0 \leq \eta \leq \eta_1 \\ \omega_0 \leq \omega \leq \omega_1 \end{array} \quad (2.2)$$

Geometrically, $\mathbf{h}(\omega)$ is a horizontal curve vertically modulated by $\mathbf{m}(\eta)$ where $m_1(\eta)$ changes the relative scale of \mathbf{h} while $m_2(\eta)$ raises and lowers it. η is a north-south parameter, like latitude, and ω is a east-west parameter, like longitude (see Figure 2-1),. The spherical product can be scaled by a vector $\mathbf{A} = [a_1 \ a_2 \ a_3]^T$ which carries the sizes of the object in three directions. ¹

$$\mathbf{X}(\eta, \omega) = \begin{bmatrix} a_1 m_1(\eta)h_1(\omega) \\ a_2 m_1(\eta)h_1(\omega) \\ a_3 m_2(\eta) \end{bmatrix} \quad \begin{array}{l} \eta_0 \leq \eta \leq \eta_1 \\ \omega_0 \leq \omega \leq \omega_1 \end{array} \quad (2.3)$$

¹For a detailed description of this derivation and spherical products see Barr [23, 24, 25, 26].

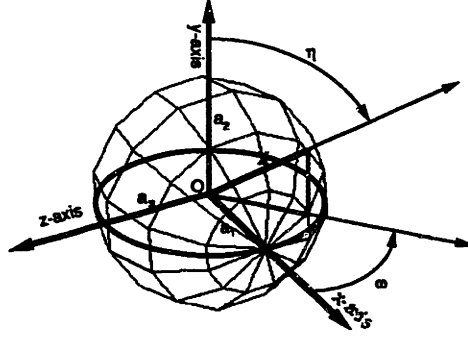


Figure 2-1: Representation of Coordinate System of Superquads.

2.1.2 Superquadric Representation

Superquadrics can be compared to lumps of clay that can be deformed into realistic looking three dimensional models ² [27]. This analogy becomes apparent on using the Thingworld Modeling System. Superquadrics are a family of parametric shapes that were invented by a Danish designer Peit Hein as an extension of basic quadric surface and shapes as is explained above [22, 23, 24, 28]. Superquadric surface is defined by setting up the following relationships in Equation (2.1).

$$\begin{aligned} \mathbf{h}(\omega) &= \begin{bmatrix} \cos^{e_2}(\omega) \\ \sin^{e_2}(\omega) \end{bmatrix} & -\pi \leq \omega \leq \pi \\ \mathbf{m}(\eta) &= \begin{bmatrix} \cos^{e_1}(\eta) \\ \sin^{e_1}(\eta) \end{bmatrix} & -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \end{aligned} \quad (2.4)$$

Therefore the superquadric surface is defined by:

$$\mathbf{X}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{e_1}(\eta) \cos^{e_2}(\omega) \\ a_2 \cos^{e_1}(\eta) \sin^{e_2}(\omega) \\ a_3 \sin^{e_1}(\eta) \end{bmatrix} \quad \begin{array}{l} -\pi \leq \omega \leq \pi \\ -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \end{array} \quad (2.5)$$

Where e_1 and e_2 are squareness parameters (see Figure 2-2).

²clay is considered to be a fastest traditional modeling medium [20]

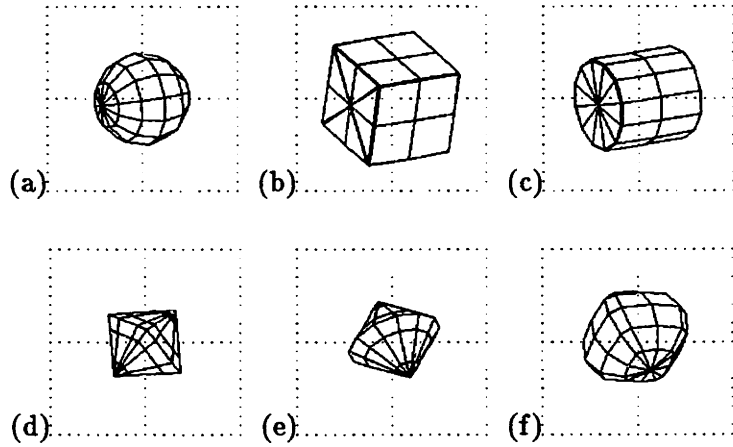


Figure 2-2: Superquads with various squareness values (a) $e1 = 1.0, e2 = 1.0$, (b) $e1 = 0.05, e2 = 0.05$, (c) $e1 = 1.0, e2 = 0.05$, (d) $e1 = 2.0, e2 = 2.0$, (e) $e1 = 1.0, e2 = 2.0$, and (f) $e1 = 0.5, e2 = 1.0$

2.1.3 Implicit Representation of Superquads

Eliminating the parameters ω and η from Equation (2.3) using the trigonometric identity $\sin^2 x + \cos^2 x = 1$, we get an implicit equation of a superquadric surface [22].

$$\left[\left[\frac{x}{a_1} \right]^{\frac{2}{e_2}} + \left[\frac{y}{a_2} \right]^{\frac{2}{e_2}} \right]^{\frac{e_2}{e_1}} + \left[\frac{z}{a_3} \right]^{\frac{2}{e_1}} = 1.0 \quad (2.6)$$

This equation could then be written as an *inside-outside* function.

$$F(x, y, z) = \left[\left[\frac{x}{a_1} \right]^{\frac{2}{e_2}} + \left[\frac{y}{a_2} \right]^{\frac{2}{e_2}} \right]^{\frac{e_2}{e_1}} + \left[\frac{z}{a_3} \right]^{\frac{2}{e_1}} \quad (2.7)$$

Following is the functionality of the *inside-outside* function when a given point (x_p, y_p, z_p) is plugged into the inside-outside function of a superquad: Hence the superquadric functions provides an inherently better mathematical formulation for contact detection then the non-parameterized representations where the *inside-outside* is dreadfully expensive to compute and sometimes techniques for calculating bounding boxes of models and casting rays from one element to the other are used ³ [13].

³See [9] and [13] for comparisons.

<i>Inside-outside function of a superquad</i>		
$F(x_p, y_p, z_p)$	> 1.0	outside the surface
$F(x_p, y_p, z_p)$	$= 1.0$	on the surface
$F(x_p, y_p, z_p)$	< 1.0	inside the surface

Table 2.1: The inside-outside function of a superquad

2.1.4 Surface normals of superquadratic functions

Normals to the two parametric curves in Equation (2.4) are;

$$\begin{aligned}
\mathbf{n}_h(\omega) &= \begin{bmatrix} \frac{1}{a_1} \cos^{2-e_2}(\omega) \\ \frac{1}{a_2} \sin^{2-e_2}(\omega) \end{bmatrix} \\
\mathbf{n}_m(\eta) &= \begin{bmatrix} \cos^{2-e_1}(\eta) \\ \frac{1}{a_3} \sin^{2-e_1}(\eta) \end{bmatrix}
\end{aligned} \tag{2.8}$$

After

$$\mathbf{N} = \mathbf{n}_m \otimes \mathbf{n}_h$$

we get the surface normals of the superquadratic functions:

$$\mathbf{N}(\eta, \omega) = \begin{bmatrix} \frac{1}{a_1} \cos(\eta)^{2-e_1} \cos(\omega)^{2-e_2} \\ \frac{1}{a_2} \cos(\eta)^{2-e_1} \sin(\omega)^{2-e_2} \\ \frac{1}{a_3} \sin(\eta)^{2-e_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{x} \cos(\eta)^2 \cos(\omega)^2 \\ \frac{1}{y} \cos(\eta)^2 \sin(\omega)^2 \\ \frac{1}{z} \sin(\eta)^2 \end{bmatrix} \tag{2.9}$$

2.2 Contact for Deformed and Undeformed Objects

Barr [24, 26]) presented transformations and deformations of solid primitives specifically applied to parametric three dimensional shapes. Nguyen [29], Pentland [27] Bajcsy [22] and Solina [30] use the superquadrics with deformations and transformations for object recognition and description. These deformations and transformations extend the functionality of these parametric superquadratic functions as geometric primitives. Pentland [6] presented another method of applying deformations on the superquadratic functions which quite effi-

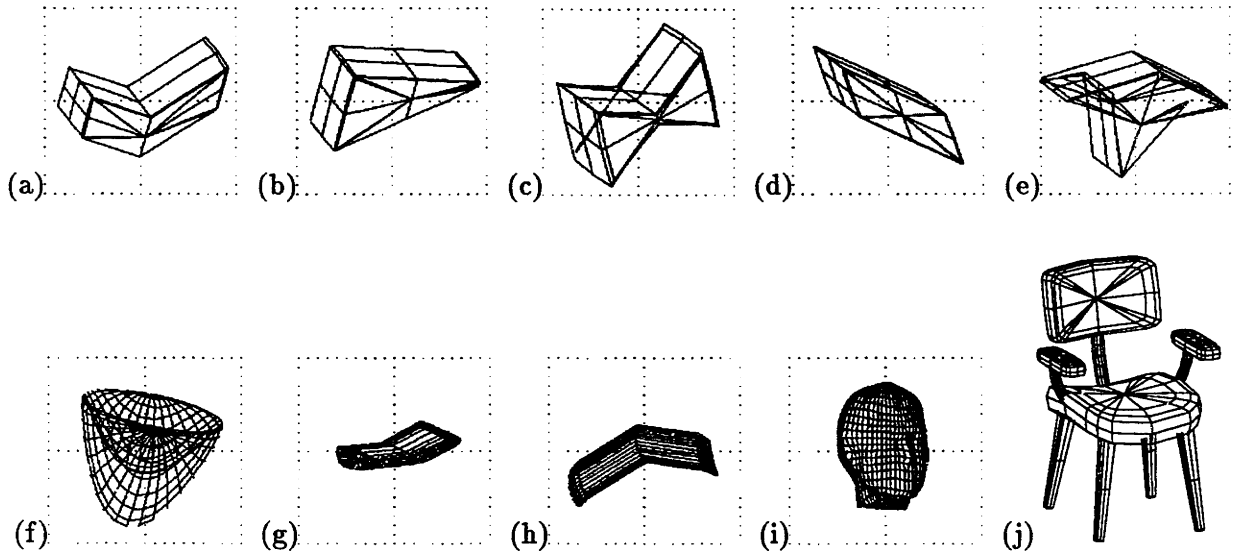


Figure 2-3: Deformed Superquads; (a) bending only (b) pinch only (c) taper only (d) shear only (e) bending, taper and pinch superimposed. (f) cup (g) wing (aerofoil) (h) bent pipe (i) deformations with a displacement map; from range data and (j) joining parts to form a chair.

ciently work with the clay sculpting analogy [20]. This deformation mapping is based on the different vibration modes of the body transformations (see Appendix A). Using only the linear and quadratic modes of vibration, a large variety of deformations could be incorporated into the system. Since the modes of vibrations are uncoupled ⁴ [6], superposition of these modes is also possible and extends the functionality of superquads into a deformable domain. Let \mathbf{D} be the deformation matrix of a superquad, then

$$\mathbf{X}_d = \mathbf{D}\mathbf{X} \quad (2.10)$$

where \mathbf{X}_d is vector for the deformed superquadric surface.

The deformation matrix should be always invertible (i.e. $\det \mathbf{D} \neq 0$) to allow calculations of undeformed surface from the deformed surface. However there are cases in which this might not be true and an iterative process must be used to determine the undeformed space from the deformed (see Appendix A).

Figure 2-3 shows the different types of modes of vibrations that form the deformed

⁴discussed in the Chapter 3

superquads. Figures 2-3 (a) to (d) show the modes of bending, taper, pinch and shear separately, while Figures 2-3 (e) (f) (g) and (h) show the superposition of these modes. Figure 2-3 (i) shows a face that was generated by deformations of a single superquad and then a displacement map was applied on top. The displacement and the deformation maps were based on range data from a real face. Superquads have been used extensively for data fitting and recognition of parts. Pentland [31] has done a lot of work of object recognition using deformable superquads, and this figure is an example of that work. Figure 2-3 (j) shows how various shapes could be added together to form complex scenes.

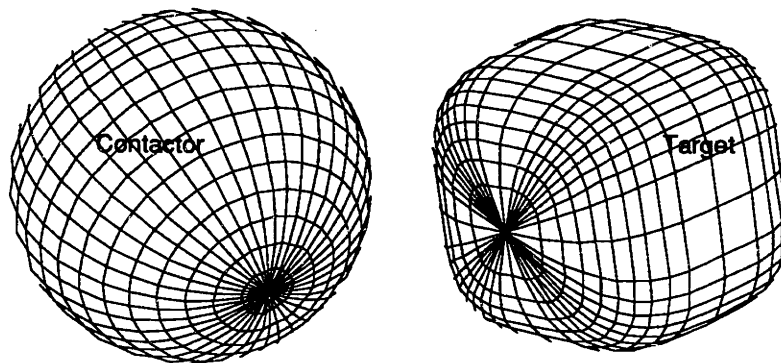


Figure 2-4: Target and Contactor.

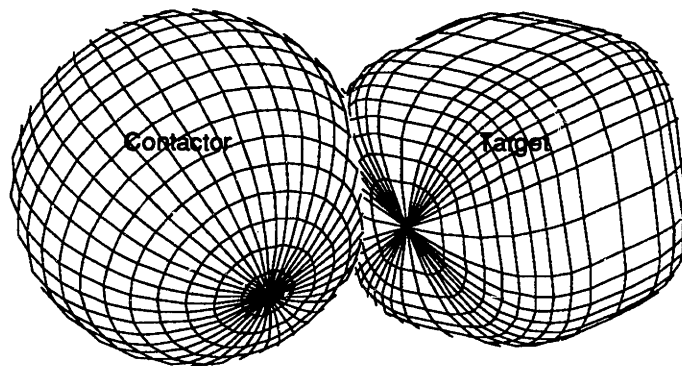


Figure 2-5: Objects in Contact.

2.3 Curvatures for Improved Contact Detection

In this section a formulation will be developed to improve contact detections for parametric shape representations, even with less sampling points on the surface. For our formulation, the colliding bodies will be referred to as contactor and target, the target being the one with whom the collision is taking place and the contactor the body itself (see Figure 2-4). First the formulation of curvatures of parametric curves.[29, 22]

2.3.1 Formulation for curvatures of parametric curves

Equation of curvature of a parametric curve is given by:

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{[(\dot{x})^2 + (\dot{y})^2]^{\frac{3}{2}}} \quad (2.11)$$

Applying Equation (2.11) to the parametric curves in (2.4) we obtain the curvatures of the two parametric curves of the superquadric surfaces. These essentially are the *principle curvatures* of the superquadric surface.

$$\kappa_h(\omega) = \frac{1}{\dot{h}_1^2 + \dot{h}_2^2} \begin{vmatrix} \dot{h}_1 & \ddot{h}_1 \\ \dot{h}_2 & \ddot{h}_2 \end{vmatrix} \quad (2.12)$$

$$\kappa_m(\eta) = \frac{1}{\dot{m}_1^2 + \dot{m}_2^2} \begin{vmatrix} \dot{m}_1 & \ddot{m}_1 \\ \dot{m}_2 & \ddot{m}_2 \end{vmatrix} \quad (2.13)$$

Since both $\kappa_h(\omega)$ and $\kappa_m(\eta)$ are perpendicular (\perp) to each other the curvature of the surface at any point is given by

$$\kappa(\alpha) = \kappa_m \cos^2(\alpha) + \kappa_h \sin^2(\alpha) \quad (2.14)$$

where α is the angle from the $h(\eta)$ curve, traversing the surface at that point.

The *Gaussian curvature* of a superquad as calculated by Nguyen [29] is:

$$\mathbf{K}(\eta, \omega) = m_1^3 m_2 \frac{|\mathbf{n}_m(\eta)|^2 |\mathbf{n}_h(\omega)|^2}{|\mathbf{N}(\eta, \omega)|^4} \begin{vmatrix} h_1 & \dot{h}_1 \\ h_2 & \dot{h}_2 \end{vmatrix} \kappa_m(\eta) \kappa_h(\omega) \quad (2.15)$$

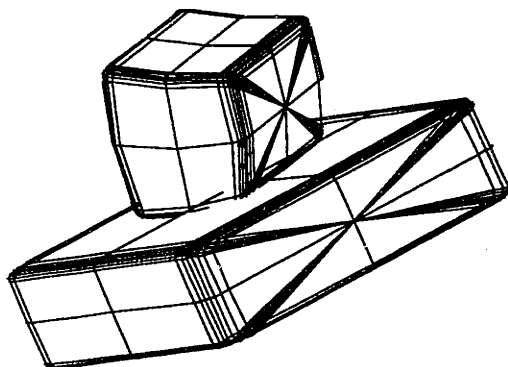


Figure 2-6: Objects in Contact and Deformed.

2.3.2 Lowest point of penetration

As mentioned above, one very good way of determining if the bodies are in contact is to check if any point of one body is inside the surface of the other. The use of *inside-outside* function Table 2.1 for parametric surfaces presents a definite edge. However it is not feasible to check each and every point of one body with the *inside-outside* of the other as it is computationally too expensive. For this reason sampling points of one object (the contactor) are checked with the *inside-outside* of the target. The problem with this method is that if the sampling is too coarse then the penetration value returned will be true for that specific point but most probably not the point of most penetration(see Figure 2-7). By using the principles of differential and computational geometry to exactly locate that point (see Figure 2-8) this problem could be solved with less computation. The following is the algorithm.

1. Check for contact of a sample point, referred to as a node. This check is simply testing the inside-outside function for the specific node.

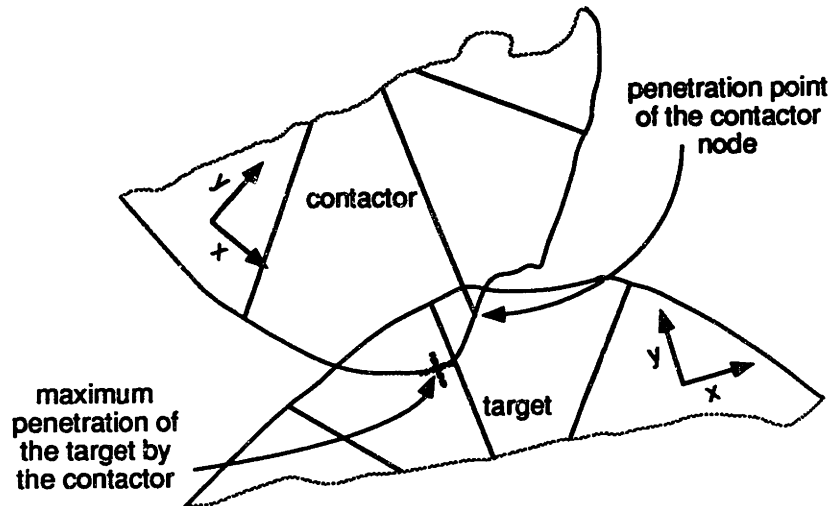


Figure 2-7: Point of Contact (two-dimensional view).

2. If the check is positive and there is contact then the lowest point should be located. This can be done by two methods

- (a) A brute force method of incrementally changing the values of x , y and z to get the lowest value of the *inside-outside* function. OR
- (b) Traversing down the two parametric curves Equation (2.4) using differential geometry and do a local minimization of the contactor body inside the target body to locate the most penetrated point of the object.

The latter method is discussed in more detail in the next section.

2.3.3 Local minimum point of the target

To determine the local minima of a target body inside the contactor,

1. Calculate the curvatures at that point using Equations (2.13) and (2.12).

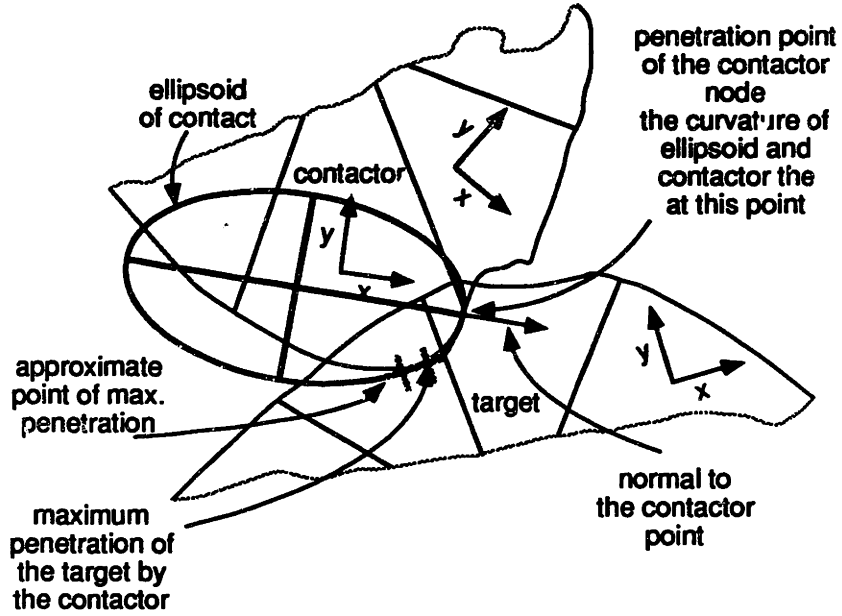


Figure 2-8: Ellipsoid for a contact point (two-dimensional view).

2. Form an ellipsoid with a vertex of the ellipse at the node and the centroid of the ellipsoid lying on the surface normal of the contactor node, (see Figure 2-8) with the following equation

$$\mathbf{X}_e(\eta, \omega) = \begin{bmatrix} \kappa_m \cos(\eta) \cos(\omega) \\ \kappa_h \cos(\eta) \sin(\omega) \\ \kappa_o \sin(\eta) \end{bmatrix} \quad \begin{array}{l} -\pi \leq \omega \leq \pi \\ -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \end{array} \quad (2.16)$$

where

$$\kappa_o \simeq \frac{\kappa_m + \kappa_h}{2}$$

This ellipsoid can now be transformed (translated and rotated) into the local coordinates of the target and then minimized in the local coordinates system to get the lowest point within the target.

$$\mathbf{X}_e^t = \mathbf{TX}_e \quad (2.17)$$

$$\nabla \mathbf{X}_e^t = 0 \quad (2.18)$$

where

$$\nabla = \left[\begin{array}{ccc} \frac{d}{dx} & \frac{d}{dy} & \frac{d}{dz} \end{array} \right]$$

This should give a point which is the minimum point on the ellipsoid that is inside the target.

Simplifications As it is obvious that the above method is computationally expensive. More speed could be achieved at the loss accuracy by assuming a sphere rather than an ellipsoid. The major problem that exists with this method is that the curvature at flat locations is very large. Singularity checks are required to make sure that curvature calculations do not blow up (diverge). It might be noted that all of these calculations of curvatures and normals are in undeformed space. The deformation map has to be applied on top to get the right values (see Appendix A).

So far, the discussion has been restricted to detection of contact. The next phase is the determination of the response due to this contact. For this information about the force and energy distributions over the surface of the colliding bodies is necessary. In the next chapter formulations, using FEM, will be developed for this purpose. The concepts of dynamic analysis will be introduced and efficient methods for the dynamic simulations will be presented. In the following chapters (4 and 5) these formulations will then be applied for collision and friction.

Chapter 3

Finite Element Method Formulation

In this chapter some introductory concepts of finite element methods from their formulation to their applications are presented. Since the intention is to use deformable models in virtual world modeling, it is essential to develop and use, formulations for relating the forces on the surface and within the body and its deformations. Finite Element Method, which is a numerical procedure for solving differential equation systems for engineering analysis, provides a very efficient formulation for this kind of application. The primary subdivisions of FEM are:

- idealize the system into a form that can be analyzed (discretize).
- formulate the governing equilibrium equations of the idealized system (formulate).
- solve these equilibrium equations (obtain node and element characteristics).
- interpret the results to the entire system (backsolve, utilize continuum of elements to obtain solution for the whole system).

It will be shown in this chapter, how these principles could be used to develop a finite element basis for an object, specifically for objects that have a parametric surface representations (superquadric functions). The formulations of the governing equilibrium equations

are presented and then a discussion on the use of the different methods for dynamic analysis will be presented. ¹

3.1 Formulation of Displacement Based Finite Element Method

This formulation is based on the fact that when certain part (node) of a body is allowed some displacement, it is possible to calculate the applied loads on it. The displacement based finite element is just an extension to the displacement method of analysis.

3.1.1 Basic Steps of the Formulation

1. Idealize the structure (or whatever) as an assemblage of elements that are interconnected.
2. Identify the unknown displacements that completely define the displacement response.
3. Establish force balance equations corresponding to the unknown joint displacements (from item 2 above) and solve.
4. Calculate internal stress distributions.
5. Interpret the displacements and stresses calculated above based on the idealizations and assumptions used.

3.2 General Derivation Of Finite Element Equations

Consider a three dimensional body (see Figure 3-1). In this figure the the coordinate system, displacements variables and the different types of forces and their notations are specified, where;

¹Most of the formulations presented here are based on the formulations presented by Bathe [18] and the concepts of FEM from [19]. The notations used are also compatible with Bathe [18].

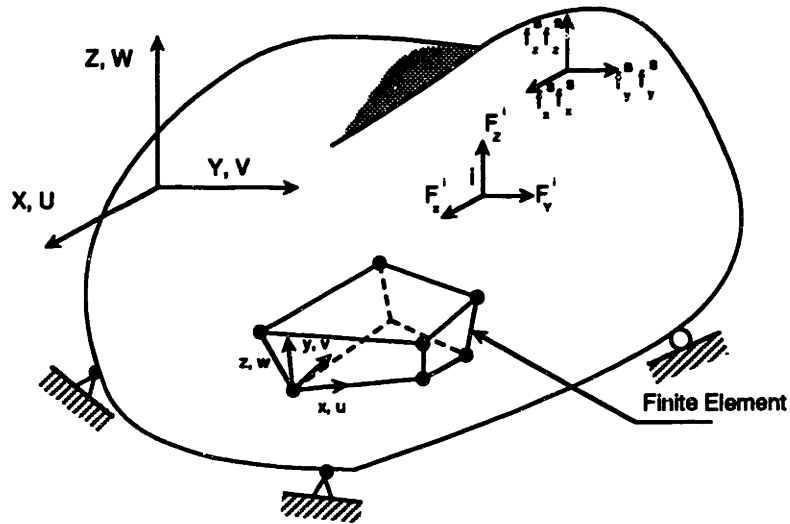


Figure 3-1: General Three Dimensional Body.

X, Y and Z , = global coordinate system
 U, V and W , = global displacements
 x, y and z , = local coordinate system
 u, v and w , = local displacements

$$\mathbf{f}^B = \begin{bmatrix} f_x^B \\ f_y^B \\ f_z^B \end{bmatrix}, \quad \mathbf{f}^S = \begin{bmatrix} f_x^S \\ f_y^S \\ f_z^S \end{bmatrix}, \quad \mathbf{F}^i = \begin{bmatrix} F_x^i \\ F_y^i \\ F_z^i \end{bmatrix}$$

here:

\mathbf{f}^B = Body Forces (Force/unit volume)
 \mathbf{f}^S = Surface Forces (Force/surface area)
 \mathbf{F}^i = Concentrated Forces

$$\mathbf{U} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

The strains (ϵ) corresponding to \mathbf{U} and the corresponding stresses (τ) are:

$$\epsilon = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix}; \quad \tau = \begin{bmatrix} \tau_{xx} \\ \tau_{yy} \\ \tau_{zz} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{bmatrix}$$

3.2.1 Principle of Virtual Work

Definition 3.1 *The Equilibrium of a body requires that for any compatible, small virtual displacements, which satisfy the essential boundary conditions, imposed on the body, the total virtual work is equal to the total external virtual work [18].*

$$\int_V \bar{\epsilon} \tau dV = \int_V \bar{\mathbf{U}}^{BT} \mathbf{f}^B dV + \int_V \bar{\mathbf{U}}^{ST} \mathbf{f}^S dS + \sum_i \bar{\mathbf{U}}^{iT} \mathbf{F}^i \quad (3.1)$$

Note: A bar on the vector signifies “virtual”

For the above to be true the following conditions have to be met:

- displacements should be compatible and continuous.
- must satisfy the displacement boundary conditions.
- must satisfy constitutive relationships (i.e. stresses could be evaluated from strains).

3.2.2 Shape or Interpolation Functions

As per the definition of Principle of Virtual work, it is necessary to set up a system of shape functions that relate the displacement of one node of the body with the the relative displacements of all the other node. Similar functions are required for the distribution of strains across the body of the finite element model. These functions called the *displacement interpolation functions* and *strain-displacement functions* are described in vector form as;

$$\mathbf{u}^{(m)}(x, y, z) = \mathbf{H}^{(m)}(x, y, z) \mathbf{U} \quad (3.2)$$

and

$$\epsilon^{(m)}(x, y, z) = \mathbf{B}^{(m)}(x, y, z)\mathbf{U} \quad (3.3)$$

where;

- H** = The displacement interpolation matrix
- B** = The strain displacement matrix
- U** = The three dimensional displacement vector.
- m** = The element number

H and **B** are related and **B** be calculated by appropriate differentiations of the functions in **H**. Setting up of the **B** and **H** matrices is explained in detail in Appendix B.

3.2.3 Develop Stiffness and Load Matrices

By using equations (3.2) and (3.3) the assemblage process of the local stiffness matrices to the global stiffness matrix, referred to as the *direct stiffness method* is automatically performed in the finite element application of the principle of virtual displacements. This application is done as follows.

Relation between strains and initial stresses is:

$$\tau^{(m)} = \mathbf{E}_m^{(m)} \epsilon^{(m)} + \tau^{I(m)} \quad (3.4)$$

where:

- E** = The elasticity matrix
- τ^I = The initial stress

Converting Equation (3.1) into element formulation:

$$\begin{aligned} \int_{V^{(m)}} \bar{\epsilon}^{(m)} \tau^{(m)} dV^{(m)} &= \int_{V^{(m)}} \bar{\mathbf{U}}^{(m)BT} \mathbf{f}^{(m)B} dV^{(m)} \\ &+ \int_{V^{(m)}} \bar{\mathbf{U}}^{(m)ST} \mathbf{f}^{(m)S} dS^{(m)} \\ &+ \sum_i \bar{\mathbf{U}}^{iT} \mathbf{F}^i \end{aligned} \quad (3.5)$$

Now substituting Equations (3.2) and (3.3) into equation (3.5) and comparing with the following equilibrium equation;

$$\mathbf{K}\mathbf{U} = \mathbf{R} \quad (3.6)$$

We get;

$$\mathbf{K} = \sum_m \int_{V^{(m)}} \mathbf{B}^{(m)T} \mathbf{E}^{(m)} \mathbf{B}^{(m)} dV^{(m)} \quad (3.7)$$

also,

$$\mathbf{R} = \mathbf{R}^B + \mathbf{R}^S - \mathbf{R}^I + \mathbf{R}^C \quad (3.8)$$

$$\mathbf{R}^B = \sum_m \int_{V^{(m)}} \mathbf{H}^{(m)T} \mathbf{f}^{(m)B} dV^{(m)} \quad (3.9)$$

$$\mathbf{R}^S = \sum_m \int_{S^{(m)}} \mathbf{H}^{(m)ST} \mathbf{f}^{(m)S} dS^{(m)} \quad (3.10)$$

$$\mathbf{R}^I = \sum_m \int_{V^{(m)}} \mathbf{B}^{(m)T} \boldsymbol{\tau}^{(m)I} dV^{(m)} \quad (3.11)$$

$$\mathbf{R}^C = \mathbf{F} \quad (3.12)$$

3.2.4 Include Inertia Forces

In the above equations of motion, the inertia effects were neglected. If the loads are to be applied rapidly, then the inertia forces have to be considered using d'Alambert's principle [18]. We can simply include the element inertia forces as a part of the body forces. Assuming that the element accelerations are in the same directions as the element displacements, the contribution of the inertia force to the load vector is:

$$\mathbf{R}^B = \sum_m \int_{V^{(m)}} \mathbf{H}^{(m)T} [\mathbf{f}^{(m)B} - \rho^{(m)} \mathbf{H}^{(m)} \ddot{\mathbf{U}}] dV^{(m)} \quad (3.13)$$

again on comparing with

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \quad (3.14)$$

we get,

$$\mathbf{M} = \sum_m \int_{V^{(m)}} \mathbf{H}^{(m)T} \rho^{(m)} \mathbf{H}^{(m)} dV^{(m)} \quad (3.15)$$

3.2.5 Include Damping Forces

Subsequently, in measuring time dependent response, it is observed that the energy is dissipated which in time dependent response is taken into account by “velocity dependent damping forces”. Based on this:

$$\mathbf{R}^B = \sum_m \int_{V^{(m)}} \mathbf{H}^{(m)T} [\mathbf{f}^{(m)B} - \rho^{(m)} \mathbf{H}^{(m)} \ddot{\mathbf{U}} - \kappa^{(m)} \mathbf{H}^{(m)} \dot{\mathbf{U}}] dV^{(m)} \quad (3.16)$$

For equation:

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{C} \dot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{R} \quad (3.17)$$

and hence:

$$\mathbf{C} = \sum_m \int_{V^{(m)}} \mathbf{H}^{(m)T} \kappa^{(m)} \mathbf{H}^{(m)} dV^{(m)} \quad (3.18)$$

A complete method for setting up the stiffness \mathbf{K} and mass \mathbf{M} matrices for superquads is presented in Appendix B from the development of the shape interpolation functions to the deformation matrices. The rest of this chapter will concern itself with the use of these matrices, which define the governing equations of our system.

Two important methods for analysis of physical systems are available, they are;

1. Static Analysis.
2. Dynamic Analysis.

These methods are explained in detail in the following sections.

3.3 Static Analysis

The basic difference between static analysis and dynamic analysis is that in static analysis the velocity and acceleration dependent forces are neglected, while in dynamic analysis they are not. The decision between dynamic and static analysis is purely an “engineering” decision, the object being to reduce the analysis effort. However, it should be realized that the assumption of static analysis must be justified, otherwise the results are meaningless. In the case of non-linear analysis (which is not dealt with in this thesis) neglecting the effects of inertia and damping forces may be so severe that a solution may be difficult or not possible

to obtain. For *virtual world* simulations inertia and damping forces are important and hence enforcing the use of dynamic analysis. The two important methods for static analysis are *Gauss Elimination Methods* and *Orthogonal Matrices Method*² and these methods have to be used in dynamic analysis for step-by-step integration methods as described in the next section.

3.4 Dynamic Analysis

The equations of equilibrium governing the linear dynamic response of a system of finite elements as shown in the formulations above is:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \quad (3.19)$$

Recall that this equation (same as (3.17)) was derived from the considerations of statics at time t . Rewriting the above equation:

$$\mathbf{F}_I(t) + \mathbf{F}_D(t) + \mathbf{F}_E(t) = \mathbf{R}(t) \quad (3.20)$$

where $\mathbf{F}_I(t)$ are the inertia forces, $\mathbf{F}_D(t)$ are the damping forces and $\mathbf{F}_E(t)$ are elastic forces at time t .

Mathematically, Equation (3.19) represents a system of linear differential equations of second order. The solution could be obtained by standard methods of solving differential equations with constant coefficients [18, 32]. However, these methods are computationally expensive, hence a few effective techniques are required suitable to our systems. Two of these techniques are discussed below:

3.4.1 Direct Integration Methods

In direct integration methods the Equation (3.19) is integrated using an numerical iterative procedure. By direct integration it is meant that prior to numerical integration, no transformation of equation into a different form is carried out. The direct integration techniques

²see Bathe [18] and Strang [32, 33] for detailed analysis of these methods

are aimed at satisfying the system Equation (3.19) only at discrete time intervals Δt apart, rather than at any time t and that the variation of displacement, velocities and accelerations within a time step is assumed [18]. It is this assumption of the variation of displacements, velocities and accelerations that determine the accuracy, stability and cost efficiency of the system.

A few commonly used direct integration methods are:

1. Central Difference Method.
2. Houbolt Method
3. Wilson Method
4. Newmark Method.

Only two of the above methods are discussed below (for more detail on these methods see Bathe [18] or any other Numerical Methods in Engineering Text).

Central Difference Method Central difference method is based on the following assumption:

$$\ddot{\mathbf{U}}_t = \frac{1}{\Delta t^2}(\mathbf{U}_{t-\Delta t} - 2\mathbf{U}_t + \mathbf{U}_{t+\Delta t}) \quad (3.21)$$

$$\dot{\mathbf{U}}_t = \frac{1}{2\Delta t}(-\mathbf{U}_{t-\Delta t} + \mathbf{U}_{t+\Delta t}) \quad (3.22)$$

The algorithm for Central Difference Method (CDM) for dynamic analysis is shown in Table 3.1.

Newmark Method The Newmark method is based on the following assumptions:

$$\dot{\mathbf{U}}_{t+\Delta t} = \dot{\mathbf{U}}_t + \left[(1 - \gamma)\ddot{\mathbf{U}}_t + \gamma\ddot{\mathbf{U}}_{t+\Delta t} \right] \Delta t \quad (3.23)$$

$$\mathbf{U}_{t+\Delta t} = \mathbf{U}_t + \dot{\mathbf{U}}_t\Delta t + \left[\left(\frac{1}{2} - \alpha\right)\ddot{\mathbf{U}}_t + \alpha\ddot{\mathbf{U}}_{t+\Delta t} \right] \Delta t^2 \quad (3.24)$$

The algorithm for Newmark Integration Method for dynamic analysis is shown in Table 3.2.

A.	<i>Initial Calculations</i>
	<ol style="list-style-type: none"> 1. Form stiffness matrix \mathbf{K}, mass matrix \mathbf{M} and damping matrix \mathbf{C} 2. Initialize \mathbf{U}_0, $\dot{\mathbf{U}}_0$ and $\ddot{\mathbf{U}}_0$. 3. Select time step Δt, $\Delta t < \Delta t_{cr}$, and calculate integration constants: $a_0 = \frac{1}{\Delta t^2}$; $a_1 = \frac{1}{2\Delta t}$; $a_2 = 2a_0$; $a_3 = \frac{1}{a_2}$ 4. Calculate $\mathbf{U}_{-\Delta t} = \mathbf{U}_0 - \Delta t \dot{\mathbf{U}}_0 + a_3 \ddot{\mathbf{U}}_0$. 5. Form effective mass matrix $\hat{\mathbf{M}} = a_0 \mathbf{M} + a_1 \mathbf{C}$. 6. Triangularize $\hat{\mathbf{M}}$: $\hat{\mathbf{M}} = \mathbf{LDL}^T$.
B.	<i>For each step</i>
	<ol style="list-style-type: none"> 1. Calculate effective load at time t; $\hat{\mathbf{R}}_t = \mathbf{R}_t - (\mathbf{K} - a_2 \mathbf{M})\mathbf{U}_t - (a_0 \mathbf{M} - a_1 \mathbf{C})\mathbf{U}_{t-\Delta t}$ 2. Solve for displacements at time $t + \Delta t$: $\mathbf{LDL}^T \mathbf{U}_{t+\Delta t} = \hat{\mathbf{R}}_t$. 3. If required, evaluate accelerations and velocities at time t: $\ddot{\mathbf{U}}_t = a_0(\mathbf{U}_{t-\Delta t} - 2\mathbf{U}_t + \mathbf{U}_{t+\Delta t})$ $\dot{\mathbf{U}}_t = a_1(-\mathbf{U}_{t-\Delta t} + \mathbf{U}_{t+\Delta t})$

Table 3.1: Step by step solution using central difference method, from Bathe [18], \mathbf{U}_t is \mathbf{U} at time t

3.4.2 Mode Superposition

The direct integration methods discussed in the previous section show that if a diagonal mass matrix is used for inertance effects in the analysis³ and no damping is assumed, then the number of operations per time step, roughly, is greater than $2nm_k$ where n is the order of the stiffness matrix and m_k is half bandwidth⁴ of the stiffness matrix. In CDM $2nm_k$ operations are required for the product of the stiffness times the displacement vector and in Houbolt, Wilson and Newmark methods, $2nm_k$ operations are required for the solution of the system at every time step. If consistent mass matrix is used and damping is assumed then about nm_k additional operations are required per time step.

The above discussion shows that the number of operations required in dynamic analysis are directly proportional to the number of time steps used in the analysis. Therefore, direct integration techniques are good for short duration simulations, and hence there is a need for

³It has been proved that the assumption of a diagonal mass matrix is a good one and converges to the same solution as the actual mass matrix, except after some oscillations about the correct solution (monotonic convergence).

⁴See Bathe[18] Appendix A.2.2 and Segerlind [19] for complete discussion on bandwidth of a stiffness matrix.

A.	<i>Initial Calculations</i>
1.	Form stiffness matrix \mathbf{K} , mass matrix \mathbf{M} and damping matrix \mathbf{C}
2.	Initialize \mathbf{U}_0 , $\dot{\mathbf{U}}_0$ and $\ddot{\mathbf{U}}_0$.
3.	Select time step Δt , parameters α and γ and calculate integration constants: $\alpha \geq 0.50$; $\alpha \geq 0.25(0.50 + \gamma)^2$ $a_0 = \frac{1}{\alpha \Delta t^2}$; $a_1 = \frac{\gamma}{\alpha \Delta t}$; $a_2 = \frac{1}{\alpha \Delta t}$; $a_3 = \frac{1}{2\alpha} - 1$; $a_4 = \frac{\gamma}{\alpha} - 1$; $a_5 = \frac{\Delta t}{2} \left(\frac{\gamma}{\alpha} - 2 \right)$; $a_6 = \Delta t(1 - \gamma)$; $a_7 = \gamma \Delta t$
4.	Calculate $\mathbf{U}_{-\Delta t} = \mathbf{U}_0 - \Delta t \dot{\mathbf{U}}_0 + a_3 \ddot{\mathbf{U}}_0$.
5.	Form effective stiffness matrix $\hat{\mathbf{K}} = \mathbf{K} + a_0 \mathbf{M} + a_1 \mathbf{C}$.
6.	Triangularize $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{LDL}^T$.
B.	<i>For each step</i>
1.	Calculate effective load at time $t + \Delta t$; $\hat{\mathbf{R}}_t = \mathbf{R}_t - \mathbf{M}(a_0 \mathbf{U}_t + a_2 \dot{\mathbf{U}}_t + a_3 \ddot{\mathbf{U}}_t) \mathbf{C}(a_1 \mathbf{U}_t + a_4 \dot{\mathbf{U}}_t + a_5 \ddot{\mathbf{U}}_t)$
2.	Solve for displacements at time $t + \Delta t$; $\mathbf{LDL}^T \mathbf{U}_{t+\Delta t} = \hat{\mathbf{R}}_{t+\Delta t}$.
3.	If required, evaluate accelerations and velocities at time t ; $\ddot{\mathbf{U}}_{t+\Delta t} = a_0(\mathbf{U}_{t+\Delta t} - \mathbf{U}_t) - a_2 \dot{\mathbf{U}}_t - a_3 \ddot{\mathbf{U}}_t$ $\dot{\mathbf{U}}_{t+\Delta t} = \dot{\mathbf{U}}_t + a_6 \ddot{\mathbf{U}}_t + a_7 \ddot{\mathbf{U}}_{t+\Delta t}$

Table 3.2: Step by step solution using Newmark integration method, from Bathe [18], \mathbf{U}_t is \mathbf{U} at time t

a method which transforms the Equation (3.19) into a form which leads to a less costly step by step solution. Since the number of operations, as mentioned above, is proportional to the half bandwidth m_k of the stiffness matrix, a reduction in m_k ⁵, would greatly reduce the cost of step-by-step solution [18]. The mode superposition method is used in the *Thingworld Modeling System* and has gained a lot of interest in the computer graphics and animation fields, for physically based modeling after its introduction by Pentland and Williams [6].

Change of Basis to Generalized Displacements To accomplish the purpose described above the following transformation on the finite element nodal point displacements \mathbf{U} is used.

$$\mathbf{U}(t) = \mathbf{P}\mathbf{X}(t) \quad (3.25)$$

⁵emphasis should be paid to the fact that the bandwidth of the system matrices depends on the topology of the finite element mesh (how the nodes and elements are numbered), however, there is a limit on a minimum bandwidth that can be achieved for a system, therefore another technique to establish this criteria has to be adopted

where \mathbf{P} is the square transformation matrix and $\mathbf{X}(t)$ is a time-dependent vector of generalized displacements. Combine this with our Equation (3.19). Substitute Equation (3.25) into Equation (3.19) and premultiply by \mathbf{P}^T :

$$\tilde{\mathbf{M}}\ddot{\mathbf{X}} + \tilde{\mathbf{C}}\dot{\mathbf{X}} + \tilde{\mathbf{K}}\mathbf{X} = \tilde{\mathbf{R}} \quad (3.26)$$

where:

$$\tilde{\mathbf{M}} = \mathbf{P}^T\mathbf{M}\mathbf{P}; \quad \tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{C}\mathbf{P}; \quad \tilde{\mathbf{K}} = \mathbf{P}^T\mathbf{K}\mathbf{P}; \quad \tilde{\mathbf{R}} = \mathbf{P}^T\mathbf{R}\mathbf{P} \quad (3.27)$$

Now we can express the elemental displacements in terms of generalized displacements:

$$\mathbf{u}^{(m)}(x, y, z, t) = \mathbf{H}^{(m)}\mathbf{P}\mathbf{X}(t) \quad (3.28)$$

With this transformation of the basis a new system of stiffness, mass and damping matrices is obtained which has a smaller bandwidth than the original system. Usually the transformation matrix is established using the displacement solutions of free vibration equilibrium equations with damping neglected [18].

Hence assuming no damping the system of equations is:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \quad (3.29)$$

the solution to this equation is of the form:

$$\mathbf{U} = \phi \sin [\omega(t - t_0)] \quad (3.30)$$

where ϕ is a vector of order n , t is the time variable, t_0 a time constant and ω the frequency of vibration of vector ϕ . From this an eigenvalue problem could be set up as follows, from which ϕ and ω must be determined.

$$\mathbf{K}\phi = \omega^2\mathbf{M}\phi \quad (3.31)$$

The eigenvalue problem in Equation (3.31) yields the n eigensolutions

$$(\omega_1^2, \phi_1), (\omega_2^2, \phi_2), \dots, (\omega_n^2, \phi_n)$$

where all the eigenvectors are \mathbf{M} -orthonormalized ⁶. Hence we have:

$$\phi_i^T \mathbf{M} \phi_j \begin{cases} = 1; & i = j \\ = 0; & i \neq j \end{cases} \quad (3.32)$$

and

$$0 \leq \omega_1^2 \leq \omega_2^2 \leq \omega_3^2 \leq \dots \leq \omega_n^2 \quad (3.33)$$

The vector ϕ_i is the i th-mode shape vector and ω_i is the corresponding frequency of vibration. Now defining a matrix Φ , which has for its columns the eigenvectors ϕ_i , and a diagonal matrix Ω^2 , with the eigenvalues ω_i^2 on its diagonal:

$$\Phi = [\phi_1, \phi_2, \phi_3, \dots, \phi_n]; \quad \Omega^2 = \begin{bmatrix} \omega_1^2 & & & & \\ & \omega_2^2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \omega_n^2 \end{bmatrix} \quad (3.34)$$

Using (3.34) we can write (3.31) as:

$$\mathbf{K}\Phi = \mathbf{M}\Phi\Omega^2 \quad (3.35)$$

and since the eigenvectors are \mathbf{M} -orthonormal:

$$\Phi^T \mathbf{K} \Phi = \Omega^2 \quad \Phi^T \mathbf{M} \Phi = \mathbf{I} \quad (3.36)$$

From the above formulations it becomes apparent that matrix Φ is an appropriate

⁶as per the properties of eigenvalues and eigenvectors, see Bathe [18] or Strang [32, 33]

transformation matrix \mathbf{P} , therefore replacing in (3.25) and (3.26) we get;

$$\ddot{\mathbf{X}} + \Phi^T \mathbf{C} \Phi \dot{\mathbf{X}} + \Omega^2 \mathbf{X} = \Phi^T \mathbf{R}(t) \quad (3.37)$$

Equation (3.37) shows that if the damping matrix is not included in the analysis, then the finite element equations are decoupled on using a transformation matrix \mathbf{P} with the free vibration mode shapes of FEM system to be its columns [6] [18]. The implementation of this method by Pentland [6] on the Thingworld Modeling System and how this Φ is used for deformations of an object is explained in Appendix A. Now we will briefly touch on the concepts of systems with or without damping.

Analysis without Damping: For systems in which damping effects are negligible and could be neglected the equations of equilibrium for analysis are reduced to;

$$\ddot{\mathbf{X}} + \Omega^2 \mathbf{X} = \Phi^T \mathbf{R}(t) \quad (3.38)$$

or to n independent and individual equations of the form

$$\begin{aligned} \ddot{x}_i(t) + \omega_i^2 x_i(t) &= r_i(t) \\ r_i(t) &= \phi_i^T \mathbf{R}(t) \\ \text{for } i &= 1, 2, 3, \dots, n \end{aligned} \quad (3.39)$$

Note here that in the above Equations (3.40), each equation is an equilibrium equation of single degree of freedom with a unit mass and stiffness of ω_i^2 . The solution to each equation in (3.40) can be obtained by any integration method described in the direct integration methods or by using the following *Duhamel integral*.

$$x_i(t) = \frac{1}{\omega_i} \int_0^t r_i(\tau) \sin \omega_i(t - \tau) d\tau + \alpha_i \sin \omega_i t + \beta_i \cos \omega_i t \quad (3.40)$$

where α_i and β_i are constants depending on the initial conditions. For complete response all the n -equations above are solved and finite element nodal point displacements are obtained

by superposition of response in each mode [18], using

$$\mathbf{U}(t) = \sum_{i=1}^n \phi_i x_i(t) \quad (3.41)$$

Analysis with Damping: In the analysis of systems in which the effects of damping cannot be neglected it would be preferable, if at all possible, to deal with the same functionality of decoupled equilibrium equations as in the undamped case. The damping matrix, essentially estimates the overall energy dissipation during the system response, and hence, cannot be generally obtained from element assemblage, like the mass and stiffness matrices. If assumed that damping is proportional to response then

$$\phi_i^T \mathbf{C} \phi = 2\omega_i^2 \xi_i \delta_{ij} \quad (3.42)$$

where ξ_i is a modal damping parameter and δ_{ij} is the *Kronecker delta* ($\delta_{ij} = 1$ for $i = j$, $\delta_{ij} = 0$ for $i \neq j$). Now using Equation (3.42) and assuming that the eigenvectors ϕ_i are C-orthogonal. Equation (3.37) becomes

$$\begin{aligned} \ddot{x}_i(t) + 2\omega_i \xi_i \dot{x}_i(t) + \omega_i^2 x_i(t) &= r_i(t) \\ r_i(t) &= \phi_i^T \mathbf{R}(t) \\ \text{for } i &= 1, 2, 3, \dots, n \end{aligned} \quad (3.43)$$

Compare this equation with Equation (3.40) and note that an extra damping term is introduced. A damping ratio, ξ_i , is a part of the single degree of freedom system. Since the formulation of damping case was set up with the intent of maintaining the functionality of the damped and the undamped cases similar, therefore similar procedures could be used to solve the damped case. The *Duhamel integral* now becomes.

$$x_i(t) = \frac{1}{\bar{\omega}_i} \int_0^t r_i(\tau) e^{-\xi_i \omega_i (t-\tau)} \sin \bar{\omega}_i (t-\tau) d\tau + e^{-\xi_i \omega_i t} \{ \alpha_i \sin \bar{\omega}_i t + \beta_i \cos \bar{\omega}_i t \} \quad (3.44)$$

where

$$\bar{\omega}_i = \omega_i \sqrt{1 - \xi_i^2}$$

Note that if $\xi_i = 0$ Equation (3.40) becomes (3.44).

For a system with damping it is important to consider the following two points.

1. The total damping of the structure is equal to the sum of individual damping of each mode. AND
2. In modal analysis for numerical solution of the system, using the decoupled equations, only stiffness \mathbf{K} and mass \mathbf{M} matrices are calculated from the elemental assemblage. Damping \mathbf{C} is computed as follows.

The damping matrix that satisfies the relation in Equation (3.42) is obtained by using the *Caughey series* [18],

$$\mathbf{C} = \mathbf{M} \sum_{k=0}^{p-1} a_k [\mathbf{M}^{-1}\mathbf{K}]^k \quad (3.45)$$

where the coefficients $a_k, k = 1, \dots, p$, are calculated from p simultaneous equations. At $p = 2$ Equation (3.45) reduces to *Rayleigh damping* ($\mathbf{C} = a_1\mathbf{M} + a_2\mathbf{K}$). If $p > 2$ then matrix \mathbf{C} is, in general, a full matrix, which increases the cost of computation by big factor, therefore, *Rayleigh damping* is assumed. A limitation of this assumption is that higher modes are considered more damped than the lower modes. The magnitudes of the *Rayleigh damping* coefficients, a_1 and a_2 , are to a large extent dependent on the energy dissipation characteristics of the structural materials.

3.4.3 Determination of minimum time step

In the preceding sections two principal procedures for solutions of dynamic equations were presented. An important observation was that the cost of a direct integration analysis is directly proportional to the number of time steps. It follows that the selection of an appropriate time step in direct integration techniques is of extreme importance. The time step should be small enough to obtain the desired accuracy and not smaller than necessary, which would make the solution more costly. The aim of this section is to discuss in detail the problem of selecting an appropriate time step Δt . The analysis of the stability and accuracy characteristics of the integration methods results in essential guidelines for selection of an appropriate time step.

Mode Superposition versus Direct Integration: As mentioned in preceding sections using either mode superposition or direct integration procedures the solution is essentially still obtained by numerical integration. However, for mode superposition a change of basis from finite element coordinate basis to the basis of eigenvectors of the generalized eigenproblem (Equation (3.31)) is performed prior to time integration. Hence, essentially Equation (3.37) or

$$\ddot{\mathbf{X}} + \mathbf{\Delta}\dot{\mathbf{X}} + \mathbf{\Omega}^2\mathbf{X} = \mathbf{\Phi}^T\mathbf{R}(t) \quad (3.46)$$

where $\mathbf{\Delta} = \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi}$ is the differential equation that needs to be solved. Here the columns in $\mathbf{\Phi}$ are the \mathbf{M} -orthonormalized eigenvectors (free vibration modes) $\phi_1, \phi_2, \dots, \phi_n$ and $\mathbf{\Omega}^2$ is a diagonal matrix with the eigenvalues (free vibration frequencies squared) $\omega_1^2, \omega_2^2, \dots, \omega_n^2$ of the eigenproblem (Equation (3.31)) as described in Section 3.4.2. Assuming damping is proportional, $\mathbf{\Delta}$ is a diagonal matrix as mentioned in previous section.

The equations (3.46) (3.37) consist of n decoupled equations, which can be solved by any of the previously mentioned integration techniques in the preceding sections. However, as the periods of vibration, $T_i, i = 1, 2, \dots, n$ are know, where, $T_i = \frac{2\pi}{\omega_i}$, we can choose, in the numerical integration of each equation in (3.46), an appropriate time step that assures a required level of accuracy. On the other hand, if all the n equations in (3.46) were integrated using the same time step Δt , then the mode superposition method is completely equivalent to direct integration, in which the same time step is used. The essence of the mode superposition solution is that frequently only a small fraction of the total number of decoupled equations need to be considered, in order to obtain a good approximate solution to the actual response. Mostly, only the first p equilibrium equations need to be used; i.e. we need to include only $i = 1, 2, \dots, p$ where $p \ll n$ (3.40) or (3.44) equations, in order to get a reasonably accurate solution. This means that only p eigenvalues and eigenvectors in Equation (3.31) need to be solved and superposition of modes Equation (3.41) is carried out by adding up only the first p modes.

The reason that only the lowest modes are considered lies in the complete modeling process for dynamic analysis. In general, when *monotonic* convergence conditions are not satisfied, the finite element analysis approximates the lowest exact frequencies, and little or no accuracy can be expected in approximating the higher frequencies and mode shapes.

Therefore, there is little justification in including the dynamic response of high frequencies in the analysis. This can be achieved by considering only the important modes of the finite element system.

While considering the problem of selecting the number of modes to be included in the mode superposition analysis, it should be remembered that an approximate solution of equilibrium equations in Equation (3.19) is desired. If not enough modes are considered then the equation in Equation (3.19) are not solved accurately enough and the equilibrium, including inertia forces is not satisfied for the appropriate response calculated. Denoting by \mathbf{U}_p the response predicted by mode superposition when p modes are considered, an indication of the accuracy of the analysis at any time t is obtained by calculating the error measure ϵ_p ;

$$\epsilon_p(t) = \frac{|\mathbf{R}(t) - [\mathbf{M}\ddot{\mathbf{U}}_p(t) + \mathbf{K}\mathbf{U}_p(t)]|}{|\mathbf{R}(t)|} \quad (3.47)$$

Assuming that $|\mathbf{R}(t)| \neq 0$. If a good approximate solution of the equilibrium systems has been obtained then $\epsilon_p(t)$ will be small at any time t . Note that $\mathbf{U}_p(t)$ must be obtained by accurate calculations at the response in each of the p modes considered, so that the error is only due to not including enough modes ⁷

In this chapter, the formulations to set up a three dimensional objects which can deform and has internal and surface stresses, have been presented. In the following chapters these three dimensional objects will be used to develop formulations for interactions of these objects in a *virtual world*.

⁷For a more detailed discussion on stability and accuracy of integration methods described here see Strang [33, 32], Bathe [18] and Segerlind [19]

Chapter 4

Collision Response

Many analytical methods for characterizing and calculation of *collision response* of rigid and non-rigid colliding bodies have been presented for computer animation and simulation. In the following sections these methods will be discussed and a new method for collision response for deformable objects based on the formulations developed in previous chapters, will be derived.

4.1 Previous Work

Moore and Wilhelms [9] presented a method for rigid body collisions which relied on the calculation of impulse of the colliding bodies at a single point. Laws of conservation of angular and linear momentum were used to determine the collision response. This force was then applied by introducing springs at the collision points. Hahn [34] presented similar methods for rigid-body contacts using *non-holonomic* and *holonomic constraints*. “Penalty” Methods, that introduce restraining forces against penetrations were presented by Terzopoulos [7] for rigid and non-rigid penetrations. Baroff [35] presented an analytical methods for contact response for non-penetrating rigid bodies in continuous contact. In many respects Moore and Wilhelms [9] method, which uses springs for response forces is in the same lines as to the penalty method, as referred to by Terzopoulos [7].

Here it is important to get into the discussion of contact response and collision response. Even though sometimes these terms are used interchangeably [35]. Collision response is

actually a discontinuous impulse force that acts instantaneously on a body, which has just collided with another body. On the other hand contact response is a continuous response that is shared by the contacting bodies from the the moment they are in contact. These both responses are calculated/determined analytically on collision and/or contact to prevent bodies from penetrating each other. Continuous contact (resulting in contact response) is a little difficult to characterize, specially for a simulation. Moore and Wilhelms [9] model simultaneous collisions as staggered series of single collisions, while Hahn [34] models them as a series of frequently occurring collisions.

As Baroff [35] has noted that for most collision response calculations, size of the critical time step plays an important role, specially when the objects are moving with high velocities, (impulse is high). Also for the systems, that deal with continuous contact as mentioned above, stability becomes a big problem, and the objects will never become stationary on contact unless the time step is extremely small. The stability problem will dealt with briefly in this chapter and then the discussion will be continued in Chapter 7.

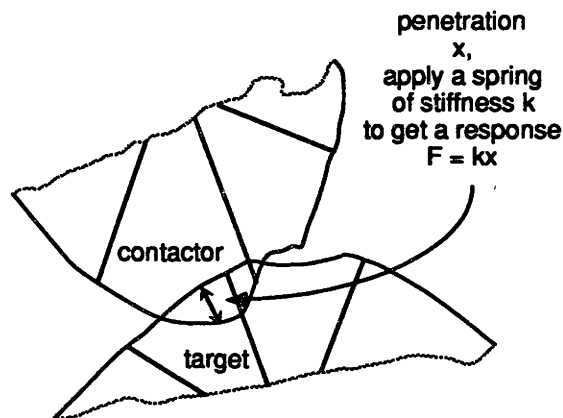


Figure 4-1: Penalty Method.

4.2 Penalty Method

In the “penalty” method as presented by Moore and Wilhelms [9] and Terzopoulos [7], the penetration of bodies is prevented by applying a repulsive force on the surface using springs. For best results these repulsive forces should be surface forces. For deformable objects the force should be equivalent to the surface forces that caused the deformations, i.e. the force forces required to undeform the object. However as mentioned earlier, the above mentioned methods apply spring forces at the points of penetrations by determining the spring constant that would be needed to negate the penetration. As it is obvious, in this case the contact patch is not at all characterized and the repulsive force is not at all distributed over the surface, and hence realism is compromised. Baroff [35] discusses that various analytical methods have a win over the “penalty” methods, specially for rigid bodies. Baroff’s discussion on expensive computations and approximate results due infinite rigidity of hard bodies and infinitely hard surfaces for rigid bodies is true, however, the use of analytical methods that are presented to solve indeterminate rigid systems for rigid bodies is highly approximate. It should however be noted that the results of that system are extremely good, and surprisingly look quite realistic.

Pentland [6, 13] and Williams [16] have time and again emphasized the need of surface patch characterization for collisions and contact. This kind of surface patch characterization is considered extremely important in engineering analysis of contact as discussed by Bathe [5]. Pentland [6, 13] recommends the use of analytic representations of geometric shapes for surface characterization. In Chapter 2 the geometric formulations for surface representations were presented. The formulations could very well be used to get the shape of the interpenetration region, as was discussed in that chapter.

Using the parametric surface representations with implicit equations for defining deformable models that are also finite element models, a new method of determining collision and contact response will be presented. In essence, this method will figure out the deformations in contacting bodies on the basis of their material properties, and amount of surface forces exerted by these bodies on deformations. These deformation forces will then be applied as collision or contact forces. First a FEM formulations will be presented then a simplified approach will be discussed.

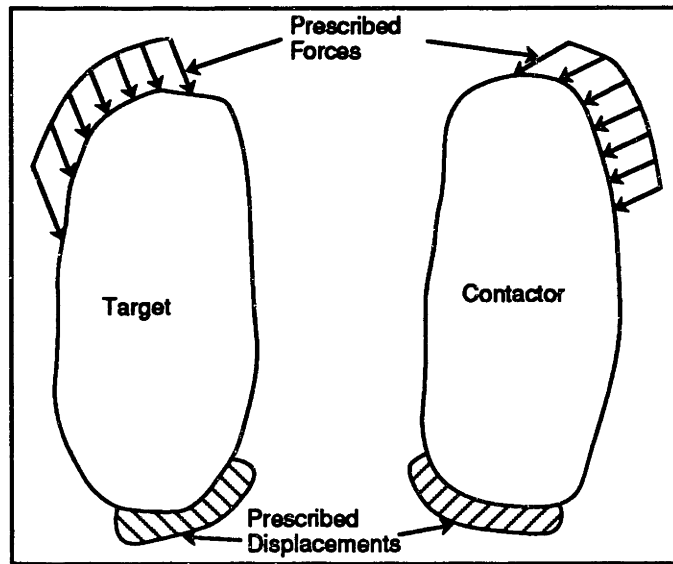


Figure 4-2: Contact Problem : Conditions prior to contact [5].

4.3 FEM Formulation of the contact problem

In this section the contact problem will be viewed with an FEM perspective and a formulation will be presented for collision response for a deformable object. Consider Figures 4-2, 4-3 and 4-4. These figures shows two generic bodies colliding called the contactor and the target. Contactor contains the nodes that come in contact with the segments (polygons) and nodes of the target. The most important conditions of contact along the contact surface is that no material overlap should occur, which as a result require contact forces along the region of contact. Following is true about these contact forces.

- These forces are equal and opposite (Newton's third law of motion).
- Contact forces are only compressive (they can only push).
- Traction on the surface are frictional forces and obey the laws of friction (This point will be considered in detail in the next chapter and for now we will work with the assumption of no friction).

Now lets characterize the different kinds of contact and discuss the response for these.

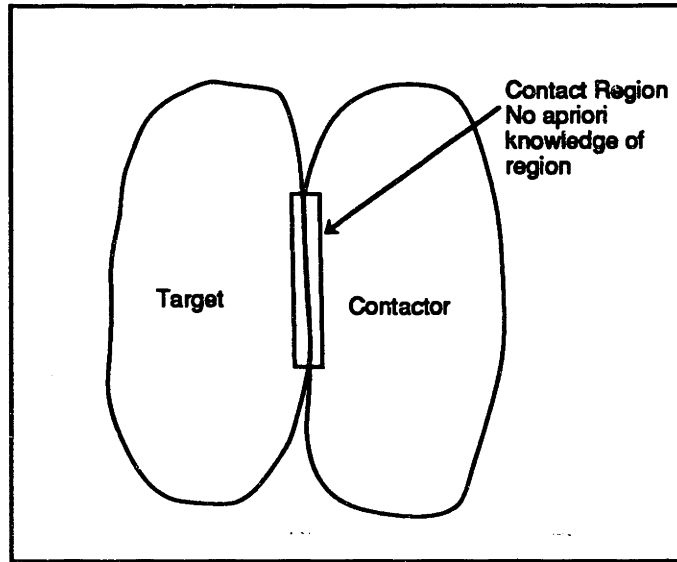


Figure 4-3: Contact Problem : Conditions at contact [5].

Definition 4.1 (Contact forces for sticking contact) *Sticking contact can be defined as a contact in which, (a) the contactor node has penetrated the target body in iteration $i - 1$, whereas it was not in contact in iteration $i - 2$; (b) frictional resistance is enough to resist sliding, i.e. the node does not slide. In case (a) the contact force at a specific node, say k is zero at iteration i and the contact force is generated during iteration i when the penetration is eliminated [5].*

Definition 4.2 (Contact forces for sliding contact) *A contactor node k is assumed to be in sliding contact if the tangential traction, as discussed in Chapter 5 exceeds the frictional capacity.*

Bathe [5] presents an algorithm for *axisymmetric contact analysis* using the incremental finite element equations of motion, by adding the contact conditions with static conditions.

The governing finite element equations are:

$$\left\{ \begin{bmatrix} \mathbf{K}_{\mathbf{t}(t+\Delta t)}^{(i-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + [\mathbf{K}_{\mathbf{c}(t+\Delta t)}^{(i-1)}] \right\} \begin{bmatrix} \Delta \mathbf{U}^{(i)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{t+\Delta t} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_{t+\Delta t}^{(i-1)} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{R}_{\mathbf{c}(t+\Delta t)}^{(i-1)} \\ \Delta \mathbf{c}_{\mathbf{c}(t+\Delta t)}^{(i-1)} \end{bmatrix} \quad (4.1)$$

where;

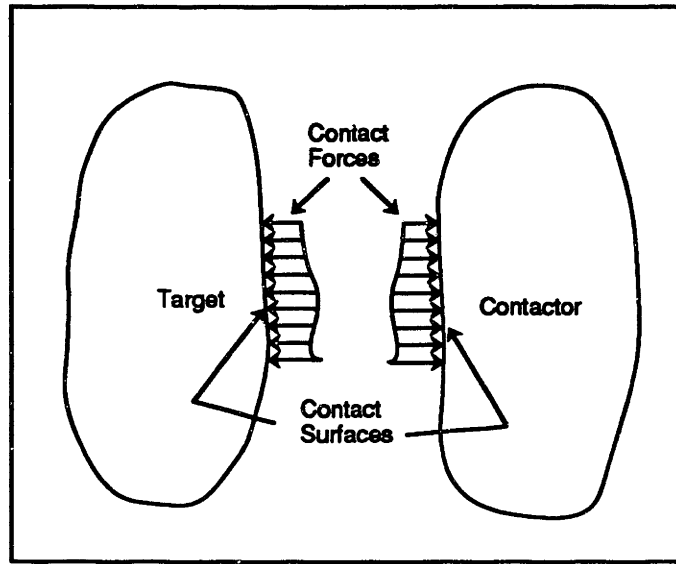


Figure 4-4: Contact Problem : Forces acting on target and contactor [5].

- $\Delta \mathbf{U}^{(i)}$ = vector of incremental displacement in iteration i .
- $\Delta \lambda^{(i)}$ = vector of increments i contact forces iteration i .
- $\mathbf{K}_{\mathbf{t}}^{(i-1)}$ = target stiffness matrix, after iteration $i - 1$.
- $\mathbf{K}_{\mathbf{c}}^{(i-1)}$ = contact stiffness matrix, for effect of contact after iteration $i - 1$.
- $\mathbf{F}_{\mathbf{t}+\Delta t}^{(i-1)}$ = vector of nodal point forces equivalent to element stresses after iteration $i - 1$.
- $\mathbf{R}_{\mathbf{t}+\Delta t}$ = vector of total applied external forces at time $t + \Delta t$.
- $\mathbf{R}_{\mathbf{c}}^{(i-1)}$ = vector of updated contact forces after iteration $i - 1$.
- $\Delta_{\mathbf{c}}^{(i-1)}$ = vector of overlaps or penetrations of bodies.

Each contactor node k contributes to $\mathbf{K}_{\mathbf{c}}^{(i-1)}$, $\mathbf{R}_{\mathbf{c}}^{(i-1)}$, and $\Delta_{\mathbf{c}}^{(i-1)}$. These terms can be considered separately for single contactor nodes as the contributions for a number of nodes are obtained by adding the individual contributions using direct stiffness method's assemblage principles.

Note that the above iterative system for the contact problem corresponds to a full Newton Iteration. Bathe [5] suggests the use of Newton Raphson method just for this reason. ¹

¹For further details on this formulation and worked out examples see Bathe [5, 18], For Newton Raphson method see Strang [32].

Most of the difficulty in solving the above described contact problem lies in numerically updating the contact conditions at the contactor node. In other words a system should automatically decide whether the node is in contact, and if in contact, is it sliding or sticking contact, so that the matrices for sliding or sticking contact could be incorporated in the system at that iteration.

The contact scenario proceeds as follows. The contactor node penetrates the target body in an iteration. This is the contact detection stage. An efficient method using parametric shape functions and *inside-outside* functions was presented in the Chapter 2. The next problem is the geometric overlap of these bodies. It is essential to know the shape of the contact patch [6] to get an accurate description of the contact scenario. This is an extremely difficult and time consuming operation, specially for an interactive virtual world system. Some simple ways of using the geometric shape descriptions of the objects could be used for the surface patch characterization. Some methods for approximate and stochastic methods for this description on basis of the roughness of the surface are presented in Chapter 6. Also in each every iteration, checks for sticking or sliding contact are required. The case of motion after collisions/contact will be discussed in detail when the discussion on friction is taken up in the next chapter. Now we will discuss how a simplified approach to the method described in this section could be employed for our purposes.

4.4 Response for Deformable Bodies

To develop a formulation of collision and contact of bodies, specifically for deformable bodies, first we need a suitable model. Figure 4-5 shows two bodies connected by springs and dampers. This represents a simplistic model of collision of two bodies. The springs and dampers, account for the deformability and the response of the object. In this figure, k_c and k_t represent the stiffness of the springs, which accounts for the stiffness of the contactor and target (nodes, segments). In case of three dimensional deformable bodies, these represent the stiffness matrices \mathbf{K} of the target and the contactor. However for simplicity, and ease of computation, we will assume that each node of the object has been decoupled and has its own stiffness value. We will consider collisions between nodes of the objects or the node of the contactor with the segment of the target. The stiffness value of each and every node

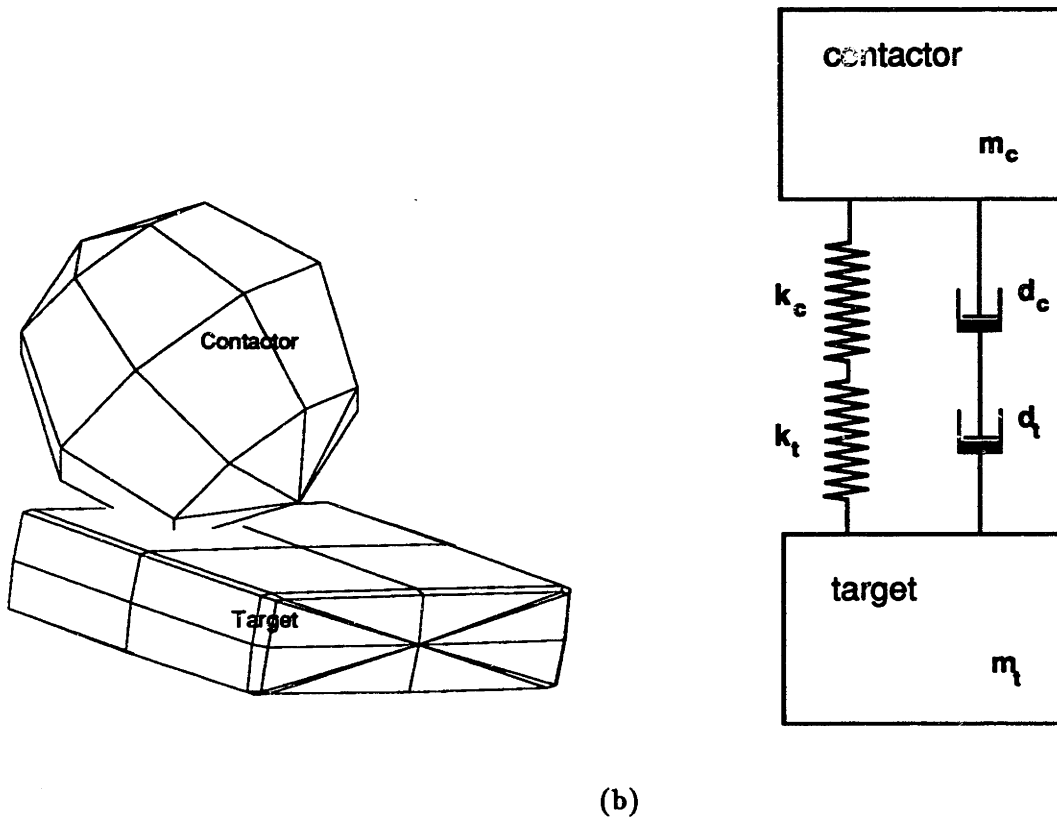


Figure 4-5: Model for Collision and Contact Forces

essentially makes up the stiffness matrix \mathbf{K} of the three-dimensional body. Similarly the damping coefficients c_c and c_t can be related to the members of the the damping matrix \mathbf{C} and m_c and m_t from the mass matrix.² Appendix C shows approximate methods of determining the stiffness values of the the nodes and segments from the stiffness matrix. The mass and damping values of each node could be determined using diagonal mass and damping matrices, which as mentioned in previous chapter lead to a good, but *monotonic* convergence to a solution.

Now using the following constitutive relationships the model in Figure 4-5 can be mod-

²In system dynamics $\frac{1}{k}$ is referred to as capacitance (stores energy), c as resistance (resists flow) and m as inertance (induces flow).

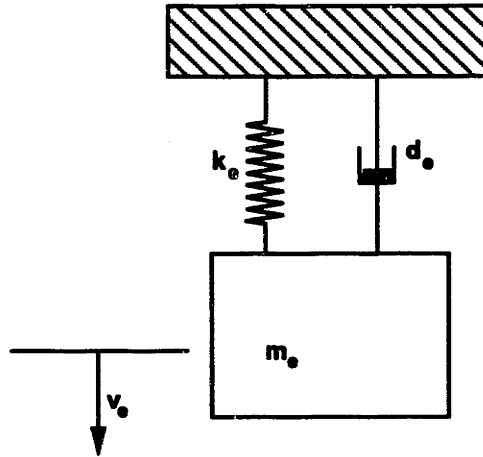


Figure 4-6: A Simplified Model for Collision and Contact Forces using the constitutive relationships from Equation (4.3)

ified to Figure 4-6.

$$\begin{aligned}
 m_e &= \sum_{i=0}^j m_i \\
 c_e &= \sum_{i=0}^j c_i \\
 k_e &= \frac{\prod_{i=0}^j k_i}{\sum_{i=0}^j k_i}
 \end{aligned}
 \tag{4.2}$$

In this system the state variables are k_e and m_e . Let v be the relative velocity of the system and f the force on the mass and if F_a is the applied force then:

$$\begin{bmatrix} \dot{f} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & k_e \\ -\frac{1}{m_e} & -\frac{c_e}{m_e} \end{bmatrix} \begin{bmatrix} f \\ v \end{bmatrix} + F_a \begin{bmatrix} 0 \\ \frac{1}{m_e} \end{bmatrix}
 \tag{4.3}$$

The above equation is a second order differential equation with two system variables, f and v and is equivalent to ;

$$m_e \ddot{x} + c_e \dot{x} + k_e x = F_a
 \tag{4.4}$$

as on solving the above equation for \ddot{x} which equals \dot{v} we get;

$$\ddot{x} = \frac{F_a}{m_e} - \frac{c_e}{m_e} \dot{x} - \frac{k_e}{m_e} x$$

and as;

$$\begin{aligned} f &= k_e x \\ \dot{f} &= k_e \dot{x} \\ \ddot{f} &= k_e v \end{aligned}$$

The notation in Equation (4.3) is known as *state-space notation* and the reason for using this notation is that the system equation can be generally written as;

$$\dot{\mathcal{X}} = A\mathcal{X} + BU \quad (4.5)$$

where \mathcal{X} is the vector of system variables, \mathcal{U} is the vector of inputs, while A and B are the matrices system parameters as can be seen on comparison with Equation (4.3). The order of the system is defined by the number of variables in vector \mathcal{X} , which in turn is dependent on the integral constraints on the system. This kind of notation is typically used for estimation and control theory applications and since, estimation and control are some major parts of this thesis, it is important to introduce this notation.

Using Equation (4.3) with dynamic analysis (as discussed in previous chapter) gives us the change in velocity and forces for each time step. Hence, this formulation provides us with an simplified method for collision and contact response at each iteration. However it should be noted that it works with one big assumption that it suffices to consider a collision in which the properties for stiffness, damping and mass of the nodes of the target and contactor could be used. Essentially the assumption is based on the decoupled and local information of the collision, and disregards the global effects. What makes this assumption feasible is that on collision, when forces are picked up, these are then applied as external forces to the bodies, at this point the forces are distributed over the body. Hence this formulation defines the collision on the local level, determines the approximate forces that caused the deformation, then applies them to the bodies. Since the material properties of the nodes are used the deformation is completely dependent on the material characteristics

in the region of collision of the object.

The assumption fails when the mesh of the model is coarse. In that case the localized effect of the collision will not be efficiently transferred, as the estimation of the stiffness values itself would be inaccurate and the collision deformation would be highly approximate. However, the error is still low in these cases and the response realistic, as the surface forces are being distributed. Appendix C , shows a simple method to distribute the stiffness values over the surface and to increase the number of nodes.

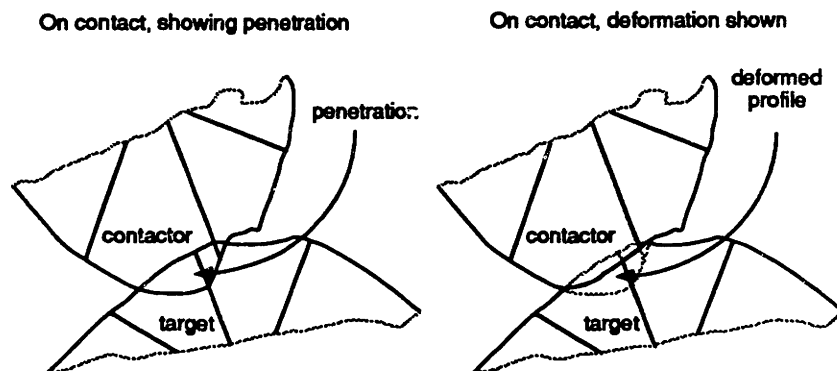


Figure 4-7: Deformed Contact.

One of the main features of this method is that the deformations of the colliding objects are dependent on the material properties of the objects and the uses of the principles of energy minimization to figure out the relevant deformations of both the target and the contactor. The other advantage of this method is that it really does not require a complete surface description as the distribution of the material stiffness over the mesh approximates the stress distributions over the surface. By using the stiffness and damping values, the deformation is accounted for and hence the response of the body to the collision is predictable, which is then used for undeforming, and hence applied as a response for collision.

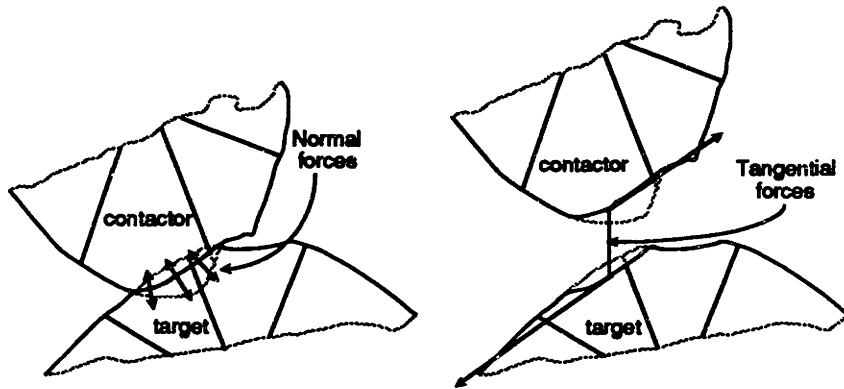


Figure 4-8: Directions of Response.

4.5 Computational Geometry : The Correct Direction Of Response

The direction of response for the contactor and the target in collision as per Newton's Third Law should be completely opposite. This is however, not an easy thing to determine as we still do not have the complete profile of the object after collision, specially as penetration is used to check for contact. For a completely non-penetrating contact, the surfaces are touching each other and they are not deforming, hence the normals to the bodies and the tangents can be determined. Using the magnitudes of tangential and normal force responses, the direction can be calculated [35]. For deformable objects, to get the direction of the response the shape of the surface after deformation is needed, so that the normals and tangents to it could be calculated.

Figure 4-7 shows a two-dimensional view of a collision. A check for contact, and the estimation of deformation requires penetration, as described in Chapter 2. The deformed profile of the contactor and the target is also shown. Figure 4-8 shows the normal and tangents the surface. By using parametric surface representations with *inside-outside* function, with deformations mapped on top, it is possible to determine the directions of the normals and tangents to the surface if the deformed object can be transformed back into

undeformed space. Appendix A shows a method by which a body that is in deformed space is transformed into undeformed space [15, 14, 16].

4.6 Surface Penetrations

In the previous section, a study of the response of collision and/or contact of objects in a multi-body simulation environment is presented. The problem that was briefly touched was the characteristic description of the contact patch. This characterization is extremely important for non-rigid body cases as the deformable bodies on collision undergo change in shape, hence the contact patch is not a simple geometric shape, but quite complex. Analytical description of this shape is difficult. It is this problem of contact patches of non-rigid bodies that has led us into the study of approximate methods. A discussion of penetrated surface (overlapped surface) in terms of stochastic estimations of the roughness of surfaces will be carried out in detail in Chapter 6. However, it is wise to at least open the doors to it now.

Every surface has some roughness. For most of these surfaces the roughness is basically a random process mapped on top of a smooth surface. Assuming this to be true, then it is also true that if a statistical domain could be mapped on top of a geometric domain, we will get a geometric surface with roughness. Even though we have discussed in detail on how to figure out if the bodies are interpenetrating, in Chapter 2, if a roughness is mapped on to the surface then it is required to check for interpenetration of the roughness too. This takes us into the sum and differences of random processes and convolution of the random processes. This theory will be developed more formally in the later chapters, but it suffices to say now that by mapping from statistical to geometric domain, it will help us to get an estimate of the surface patch that could be used to estimate the stress distribution over the surface and/or the collision response.

In the next chapter a discussion on the tangential (frictional) forces will be presented, a topic which was avoided in this chapter. The formulations that were presented in this chapter will be generalized to handle frictional forces.

Chapter 5

Friction

In the preceding chapters (Chapters 2, 4) it was assumed that the surfaces in contact were *frictionless*. If the surfaces are frictionless then the surface forces exerted by the contactor and the target (Figure 4-2) are normal to the surfaces, and the objects can move freely with respect to each other along their surfaces. In actuality, this is a far from a realistic assumption, as friction is a very important part of the world. Hence its importance and significance in *physically based virtual world modeling* is quite apparent. Friction has been one of the more difficult problems to solve for people trying to make *virtual and synthetic worlds*. Frictional forces, their analysis, and their application in design of mechanical models is an integral part of machine and part designs.

Friction is usually described as a non-conservative force which resists motion. In the world around us it is easy to see, with the knowledge of Newton's First Law of Motion in the back of our minds, the importance of frictional resistance. The reason for interest in friction, by researchers working on the concept of virtual reality are quite obvious too. Similarly, Robotics research has also concerned itself with friction, as the design of a robot that will move in the real world will have to account for the resistance to its motion. Friction also plays an important role in design of control systems for dynamic processes because its irregular contributions tend to) disturb to the system stability.

5.1 Previous work

So far in the world of computer graphics and animation, friction has been either extensively approximated or simply neglected. Baroff [35] completely ignores friction in his dynamic simulations of non-rigid contact, mainly due to the reasons of it causing instability to the system. Schroder [36] approximates the effects of friction by incorporating rigid-body motion damping. This results in a resistance to the motion internally, not due to the external resistance of traction and penetrating forces. Hahn [34] and Moore and Wilhelms [9] present methods for continuous contact, but their formulation is based on a repeated application of forces due to collision. These forces are not restricted to being normal to the surface and hence the forces in surface direction are also applied at contact. The reasons for need of approximations for friction (and/or ignoring it) are:

- computationally it is very expensive to determine the traction forces over the surface (requires integration over the contact patch),
- the contact patch of the two surfaces is difficult to characterize, and
- it is difficult to estimate the stress distributions across the surface of contact.

All of the above mentioned problems are within the main interest of this thesis and have been addressed for various aspects of virtual world modeling in Chapters 2, 3 and 4. Further discussion of estimation and control of systems in contact is pursued in the following chapters. In this chapter the discussion will be restricted to formulations for friction in the dynamic models and then the application of these formulations and the ones developed in previous chapters. Theoretical background of frictional resistance will be introduced and then various kind of contact conditions will be evaluated.

5.2 Coulomb's Law of Friction

Coulomb conducted the first thorough investigation of sliding friction, in which he searched for dependencies of the frictional force on every conceivable parameter, including the time of repose before sliding commences, elapsed time of sliding, speeds of sliding, types of surfaces,

cleanliness of surfaces and the magnitude of the normal force. Although all of the above effect the friction response in one way or other, Coulomb's conclusion was;

Definition 5.1 (Coulomb's Law of Friction) *Over an enormous range of materials and normal forces to the surface, the frictional force is directly dependent on the normal force and the material of the objects (surfaces) [37].*

The tangential force of friction during sliding (see Definition 4.2) is directed opposite to the direction of motion, with the magnitude proportional to the normal force. The constant of proportionality is called the *coefficient of dynamic friction* (μ_k) which depends on the contacting materials, but not on the speed of the motion. If there is no motion between the contacting bodies then sticking contact (Definition 4.1) conditions exist and different constant (usually greater) is used, called the *coefficient of static friction* (μ_s). The tangential force is constrained to be no greater than the product of the normal force and the *coefficient of static friction*.

$$F_f \leq \mu_s F_n \quad (5.1)$$

During sliding, Coulomb's law completely determines the frictional force,

$$F_f = \mu_k F_n \quad (5.2)$$

however, in case of sticking contact (no motion), the frictional force may not be completely determined. In some cases, the potential indeterminacy is resolved by the tendency of the frictional force to resist any impending motion at the point in question [10, 37].

5.2.1 Friction Cone and Friction Angle

While calculating friction forces, it is essential to characterize the kind of contact and hence the friction. There are basically three type of frictional forces as described in Figure 5-2. Static friction is for a sticking contact while the sliding friction, for a sliding contact. The Limiting friction is the phase change from static to sliding friction. The limiting friction comes into play when the motion starts or ends. A more intuitive idea of friction can be developed using the geometric interpretation of Coulomb's Law as developed by Moseley

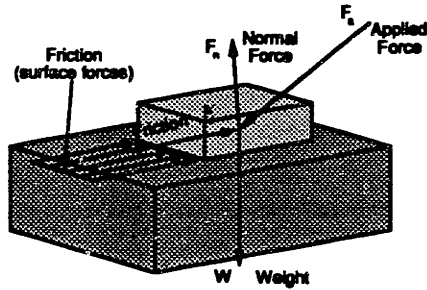


Figure 5-1: Coulomb's Law

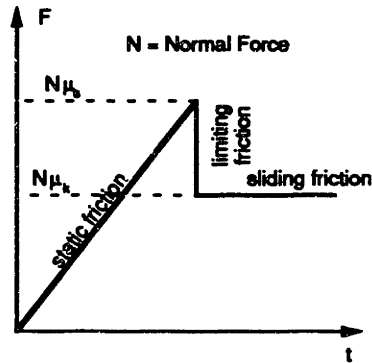


Figure 5-2: Types of Frictional Resistance

[37]. Consider a point p moving on a surface (Figure 5-3). Construct a vector F , which is the total contact force on that point, comprising the normal force F_n due to surface stiffness and tangential force F_f due to friction. The angle between the normal force F_n and F_f is given by

$$\alpha = \tan^{-1} \frac{F_f}{F_n}$$

where α is the *friction angle*. By Coulomb's law $F_f = \mu F_n$ where μ is the constant of friction, (type of contact establishes if it is kinetic or static friction). Therefore $\alpha = \tan^{-1} \mu$. The set of vectors subtending an angle of α from the normal, form a cone called a *friction cone* [37]. The *friction angle* specifies the minimum angle of attack (direction of force) that will create motion, that is the body will be in sliding contact. The *friction cone* describes a whole region, in which if the applied force vector lies, then the case of static friction

is controlling the motion along the surface. If the vector is outside this region, then the case of either limiting or sliding friction is in effect. The concept of *friction cones* and *friction* is extensively used by Mason [38] for robotic manipulator pushing operations. For fixture planning and clamping forces Cutkosky [39] suggests the use of limit surfaces in force/moment space to check whether parts will slip or stick.

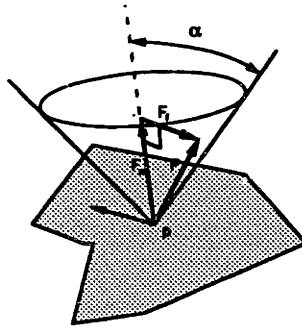


Figure 5-3: Friction Cone and Friction Angle

5.3 Formulation of Frictional Resistance

If all the applied forces, excluding the support contact forces, reduce to a single force, then the problem of planar motion of a body reduces to the problem of the motion of a point in plane. However, that is generally not the case, specially for deformable objects, which are as we have emphasized, important for a virtual modeling. Friction force, being a non-conservative force, renders most of the techniques of classical mechanics inapplicable, and most of the discussion of planar motion problems with friction are restricted to single degree of freedom systems, or to systems that reduce to the motion of a point. In this section expressions for forces and moments due to friction with supports during general planar motion, will be developed. These formulations, will then be studied with respect to FEM formulations for contact (Section 4.3) and then attempt will be made to generalize to more approximate methods for deformable objects (like Section 4.4). First a formal definition of planar motion,

Definition 5.2 (Planar Motion) *Planar motion is either translation or rotation about some instantaneously motionless point [37].*

Translation and rotation are handled separately for friction. For translation, the system of forces reduces to a single force through a point. This point is independent of the direction of translation and is called the *center of friction*. This type of reduction, however, is not possible for rotation, as will be shown. For the purpose of this formulation consider that $\mu_k = \mu_s = \mu$ and sliding contact is the only contact possible.

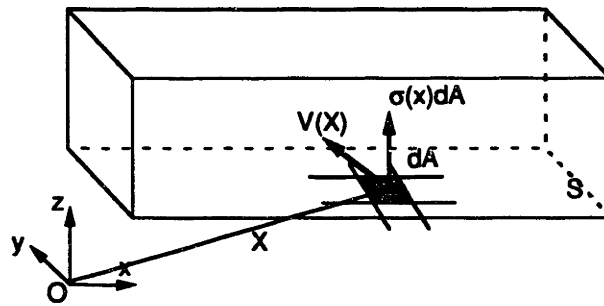


Figure 5-4: Force and Moment of Sliding Friction

Consider an object in planar motion (Figure 5-4). Let S be the region of contact with the surface, and let dA be the differential element of the area of S . Let X be the position of dA , $\sigma(X)$ be the pressure (stress) at X and V_X be the velocity of the object relative to the supporting surface at X . The normal force at X is $\sigma(X)dA$ and using Coulomb's law, the friction force at X is

$$F_f|_X = -\mu \frac{V_X}{|V_X|} \sigma(X) dA$$

by integrating over the entire support region S ;

$$F_f = \int_S -\mu \frac{V_X}{|V_X|} \sigma(X) dA \quad (5.3)$$

The total frictional moment M_f is

$$M_f = \int_S X \otimes -\mu \frac{V_X}{|V_X|} \sigma(X) dA \quad (5.4)$$

where \otimes signifies cross product of two vectors.

5.3.1 Translation in Friction

During pure translation, all points on an object have the same relative velocity (not true for deformable bodies), hence the constants could be factored out from the above Equations (5.3) and (5.4) to get:

$$F_f = -\mu \frac{V_X}{|V_X|} \int_S \sigma(X) dA \quad (5.5)$$

The total frictional moment M_f is

$$M_f = -\mu \left\{ \int_S X \sigma(X) dA \right\} \otimes \frac{V_X}{|V_X|} \quad (5.6)$$

Let F_o be the total normal contact force and X_o be the centroid of pressure distribution $\sigma(X)$:

$$F_o = \int_S \sigma(X) dA$$

$$X_o = \frac{\int_S X \sigma(X) dA}{F_o}$$

substitute the above in Equations (5.5) and (5.6) to get

$$F_f = -\mu \frac{V_X}{|V_X|} F_o$$

$$M_f = X_o \otimes F_f$$

From these equations it can be concluded that system of frictional forces of an translating object reduces to a single force, applied at the center of the pressure distribution (center of friction), whose direction is opposite the direction of translation. At this moment it is important to note that the problem now is to determine the pressure distribution at the contact region, which is usually indeterminate. One of the main purposes of this thesis has been to identify and formulate approximate methods for this pressure and stress distribution, and is presented in Chapters 3, 4, 6 and 7.

5.3.2 Rotation in Friction

Let X_r be the instantaneous center of rotation and $\dot{\Theta}$ be the angular velocity. The velocity at X , V_X is therefore given by

$$V_X = \dot{\Theta} (\hat{k} \times (X - X_r))$$

so

$$\frac{V_X}{|V_X|} = \text{sgn}(\dot{\Theta}) \hat{k} \times \frac{X - X_r}{|X - X_r|}$$

where \hat{k} is a unit normal in xy -plane. ¹ Hence $\hat{k} \times X$ indicates a rotation of X by $\frac{\pi}{2}$. Substitute into Equations (5.3) and (5.4) and simplify to get;

$$F_f = -\mu \text{sgn}(\dot{\Theta}) \hat{k} \times \int_S \frac{X - X_r}{|X - X_r|} \sigma(X) dA \quad (5.7)$$

$$M_f = -\mu \text{sgn}(\dot{\Theta}) \int_S X \cdot \frac{X - X_r}{|X - X_r|} \sigma(X) dA \quad (5.8)$$

Comparing Equations (5.5) and (5.6) with Equations (5.7) and (5.8) it is seen that the simplification obtained in the case of translation does not apply to rotation

5.4 FEM formulation

The formulation for contact presented in Section 4.3 is applicable in the case of friction. While presenting that formulation it was assumed that the contact surfaces were frictionless and the only forces exerted on contact were normal forces. Using the above mentioned formulations of friction forces based on Coulomb's law, traction forces can also be incorporated into the model. In keeping with the notations of Section 4.3, Figure 5-5 shows the friction and normal forces on segments of the contactor. ²

One of the major difficulties in modeling friction is the decision on whether the contactor

¹ $\text{sgn}(x)$ is defined as following

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases}$$

² For a more detailed description of these derivations see Bathe [5].

node is releasing (limiting friction case) or is in sticking or sliding condition. Considering the total and relative magnitudes of nodal point forces leads to numerical difficulties and it is preferred to establish the condition at a contactor node from accumulated effects and conditions of the contactor segments, adjacent to the node. The state of the segment is determined by the tangential and normal forces on the segment, which are calculated from the distributed tangential and normal forces over a segment, which are in turn calculated from the nodal point forces. Coulomb's law (Equations (5.1) and (5.2)) is then applied over the segment to decide on the type of contact. This numerical difficulty was approached in another way by Lötstedt [40] who recommends a modification to the Coulomb's law. This modification states that the tangential force is proportional to the *estimate* of the normal force, extrapolated from previous values. This modification resolves the ambiguity and inconsistency of the problem, though it does not exactly reproduce the behavior as expected of Coulomb's law. For the impact problem Lötstedt assumes a fixed impulsive coefficient of friction, and then solves for the impulsive force that minimizes the system kinetic energy, yielding a perfectly elastic collision.

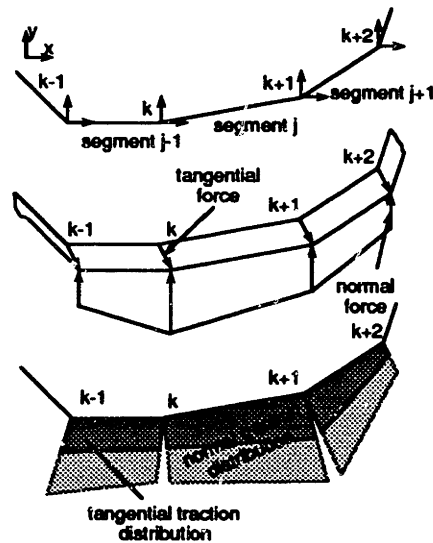


Figure 5-5: (top) Forces on contactor (middle) Friction and Normal on contactor and (bottom) Friction and Normal force distribution over segments.

5.5 Friction for deformable objects

In dealing with non-deformable objects it is extremely easy to apply the translational model of friction, where all the forces reduce to one force acting at the center of friction. For rigid bodies the velocities of all the nodes are the same and hence the relative motion of the body in translational motion is the same. In case of non-rigid bodies both of the above statements are good assumptions if the deformations are small. However as described in the previous chapter, for larger deformations the contact and interpenetration of surfaces effect the distribution of the normal and frictional forces over the surface. In Section 4.4 a formulation for localized collision and contact response was presented. Since we assumed frictionless surfaces, the forces considered in the equation were only the normal surfaces. The same equation is presented here again;

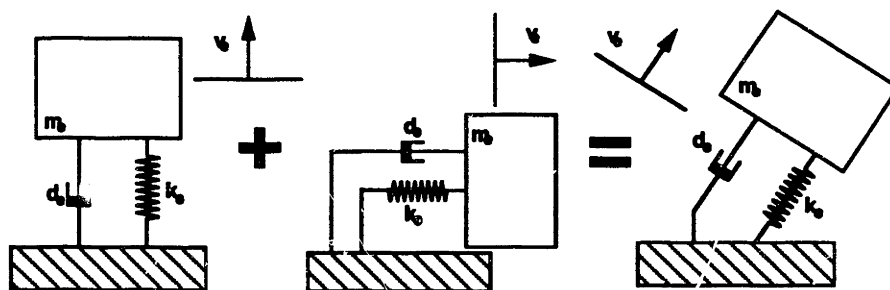


Figure 5-6: The Mass-Spring-Dashpot Model for Collision/Contact/Friction

$$\begin{bmatrix} \dot{f} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & k_e \\ -\frac{1}{m_e} & -\frac{c_e}{m_e} \end{bmatrix} \begin{bmatrix} f \\ v \end{bmatrix} + F_a \begin{bmatrix} 0 \\ \frac{1}{m_e} \end{bmatrix} \quad (5.9)$$

Here the force has components both in the normal and tangential directions and the Coulomb's law (Equations (5.1) and (5.2)) is then applied to determine the kind of contact per contactor segment. By making the contact problem localized the reduction of all the forces to one is possible and then these forces are applied to the surface in dynamics simulation of the model, hence the forces are distributed over the whole surface. A crude and

approximate model of the contact model with friction and normal forces is shown in Figure 5-6.

5.6 Surface Patches and Roughness

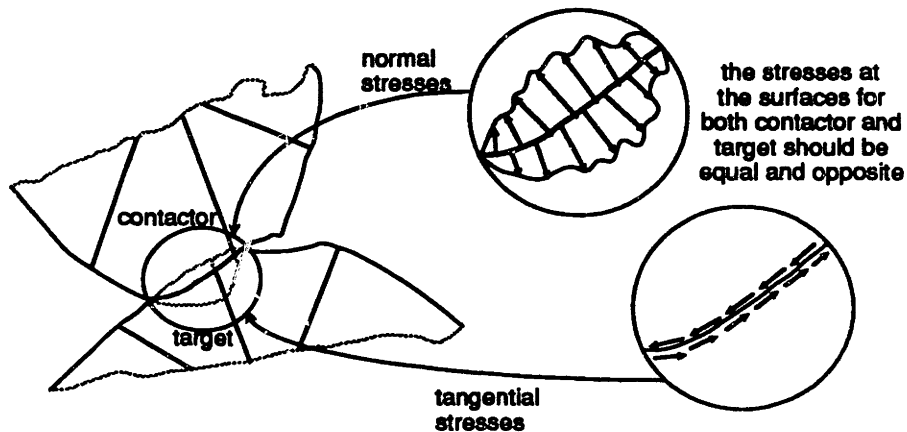


Figure 5-7: Normal and Surface Stress distributions

The essential problem in case of friction is the same as for contact. The surface region of contact needs to be characterized so that the integration as shown in Equations (5.3) and (5.4) can be carried out. By using methods of computational and differential geometry as described in Chapter 2 the surface patches of regular geometric shapes can be determined. The indeterminacy of the surface pressure or the stress distribution under a contact region gets to be a bigger problem in the realm of deformable objects. Figure 5-7 shows a two dimensional view of stress distributions on a contact region. The following chapters will present some approximate methods for estimation of these stresses. In addition to the stress distributions, another aspect, which is extremely important for virtual world modeling is surface roughness. The physical response of a contact of surfaces of various roughness should account for the perturbations on the surface of the object. This concept is studied in a stochastic estimation domain and is presented in the following chapter. The use of estimation techniques for this purpose is warranted if one deems it necessary to obtain surface and contact regions after collision of deformable objects, specially for virtual world

modeling.

Chapter 6

Stochastic Theories For Measure Of Roughness

The previous chapters presented a discussion of contact detection and response and the importance of characterizing the contact patch and the stress distribution was discussed. Throughout this discussion it was emphasized that the computational complexity of these methods is extremely high. Even though the methods of FEM and DEM are suitable to completely solve these problems to a reasonable accuracy, the system would be far from cost-effective. Hence there is a need for efficient algorithms with better computational complexity.

Currently, extensive research efforts have been directed to describe roughness of surfaces in terms of random processes. For example *fractals*, *Markov Random fields* (MRF) and *Gibb's distribution functions* are being used extensively for this, in both computer graphics, vision, and in material science. This reasoning has led us into pursuing this concept a little further and using these surface descriptions for interpenetration and interaction of objects in a multi-body system. The formulation and explanations of this concept follow.

This chapter will first present the simple concepts of textures and random variables then discuss the mapping from the stochastic domain to the geometric domain, and then finally from the geometric domain to the physical domain will be presented. It should be emphasized here that in this formulation we are presenting a method based on stochastic

and estimation theories, with the intention that these methods will improve the time and cost effectiveness of *physically based virtual world systems*.

6.1 Texture, Fractals, and MRF

Texture is qualitatively defined as a stochastic description of surface geometry. Synthetic texture was introduced into computer graphics by Catmull in 1975 and since then there have been many significant contributions for correct and efficient applications of texture to surfaces [41]. One of the important applications of texture has been to simulate shape detail that would be inefficient or difficult to model directly, either by using normal perturbation [42], or displacement mappings [43]. Texture has also been used by incorporating it into synthesized images, as a post process, to enhance the understanding of shapes [44]. Texturing may be described in many ways, with a varying degrees of theoretical and practical considerations. The general approach is to assign a texture value to a point on the surface based on the distribution of the texture function within the surface region surrounding that point. This texture value could be a randomly assigned value using, e.g., MRF, fractals, or other distributions. Note that the intention here is simply not to generate rough surfaces, as is done computer graphics but to use the surface roughness from the textures for characterizing the fine detail of collisions and contact for use in multi-body simulations.

Before getting into the details of stochastic and geometric tools for modeling surface penetrations some discussion on *fractals* and *MRFs* is desirable. Definitions of random variables, random processes and their stochastic properties are given in Appendix D.

6.1.1 Fractals

The defining characteristic of a *Fractal*¹ is called the *fractal dimension* and is qualitatively defined as a measure of the degree of roughness or irregularity or brokenness of a surface. A fractal, technically is defined as a set for which the Hausdorff-Biesicovich dimension is strictly larger than the topological dimension. A general characterization of a fractal is that they are the end result of a physical processes that modify shape through local action.

¹The word *fractal* originates from a Latin adjective, *fractus* from the verb *frangere*, which means "to break"

Such processes, after repeated applications, typically produce a fractal description [45]. The quantitative definition of a fractal is in terms of probability distribution of a generated function. For example, a random function $I(x)$ is a fractal Brownian function if for all x and Δx

$$P \left\{ \frac{I(x + \Delta x) - I(x)}{\|\Delta x\|^H} \leq y \right\} = F(y) \quad (6.1)$$

where $F(y)$ is a cumulative distribution function (see Definition D.2, Appendix D) [46]. Note that x and $I(x)$ can be interpreted as vector quantities and therefore provide an extension to two or more topological dimensions [45]. If $I(x)$ is defined as a scalar, then the fractal dimension D of the fractal $I(x)$ is $D = 2 - H$. Hence, it is obvious that if $H = \frac{1}{2}$ and $F(y)$ comes from zero-mean Gaussian with unit variance, then $I(x)$ is the classical Brownian function. (For further reading on fractals see [45, 46, 47, 48, 49])

6.1.2 Markov Random Fields

Markov Random Fields (MRF) are exactly described by a random probabilistic process in which all interaction is local and the probability that the cell is in given state is determined by the probability for the states of neighboring cells [50]. This high sense of locality results in a distribution energy, which is a summation of terms dependent only on a few variables. Formally MRFs are defined [51] as; Let Ω be the set of all configurations:

$$\Omega = \{\omega = (x_{s_1}, x_{s_2}, \dots, x_{s_N}) : x_{s_i} \in \Lambda, 1 \leq i \leq N\}$$

the event $\{X_{s_1} = x_{s_1}, \dots, X_{s_N} = x_{s_N}\}$ is abbreviated as $\{X = \omega\}$, then X is an MRF with respect to \mathcal{G} if

$$P \{X = \omega\} > 0 \text{ for all } \omega \in \Omega \quad (6.2)$$

$$P \{X_s = x_s \mid X_r = x_r, r \neq s\} = P \{X_s = x_s \mid X_r = x_r, r \in \mathcal{G}_s\} \quad (6.3)$$

Now we will start our discussion of characterizing surfaces and contacts using random variables and random processes, like the ones described above. The definitions of random variables and their stochastic and statistical properties, which will be used in the following formulations are given in Appendix D.

6.2 Surface Roughness as a Random Process

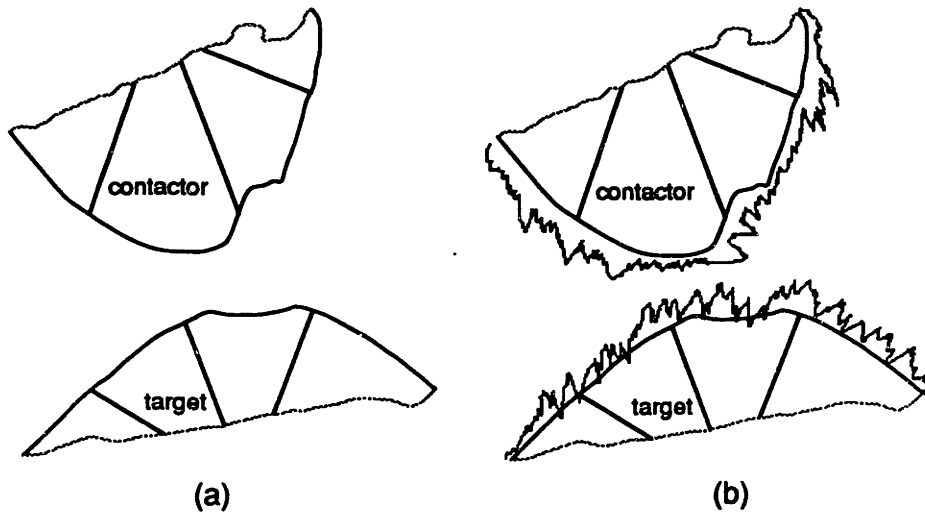


Figure 6-1: Two Dimensional Shape with Stochastic Distribution (a) Without surface roughness (b) With a random distribution mapped on.

As mentioned above, the standard approach for texture mapping on to a surface is to assign a texture value to each point on the surface by drawing samples from appropriate distributions. It should be emphasized here that when a random variable is mapped onto a surface, in essence we are performing a mapping of a statistical domain onto the geometric domain. In Chapter 2 geometric formulations for surfaces were established. By mapping on a statistical surface on top of the geometric surface we get another surface that has some geometric surface with surface characteristics that on the localized domain are samples from RVs distribution. Figure 6-1 shows a two dimensional shape with and without a statistical mapping. Note that these are exaggerated figures used to introduce the formulations. Consider two principle directions of a surface x_{g_1} and x_{g_2} . (The definition of the principle surfaces requires orthogonality between the two directions). Consider two random variables $\langle x_{s_1} \rangle$ and $\langle x_{s_2} \rangle$ (We will denote a random variable by putting $\langle \rangle$ around it). Now if the surface can be described by $\langle x_1 \rangle$ and $\langle x_2 \rangle$ after the surface roughness mapping then the

equation of the surface becomes;

$$\begin{aligned}\langle x_1 \rangle &= x_{g_1} + \langle x_{s_1} \rangle \\ \langle x_2 \rangle &= x_{g_2} + \langle x_{s_2} \rangle\end{aligned}\tag{6.4}$$

In the above equations the terms $\langle x_{s_1} \rangle$ and $\langle x_{s_2} \rangle$ could be either a fractal, MRF, random noise, any other random process.

Once the surface description is available, (which, as has been previously emphasized, is not the problem that we are attempting to solve) the problem that comes up is how to use this surface description for the characterization of surface penetrations and contact patches. One approach is to use the geometric shape of the object, with the mapping for the surface and try to ascertain the geometric intersection of the surfaces. This, as has been mentioned in Chapter 2 is not so easy for most cases, specially when dealing with non-rigid objects. Even though it may be possible to get geometric intersections for surfaces, the complexity of the problem explodes ² when some roughness has been mapped onto the surface. The following sections will describe simpler methods, which will with less computation estimate the contact surface and stress distributions on the surface.

6.3 Interpenetration of two rough surfaces

The approach that is usually taken to determine the contact surface is to check for the intersecting surface for two, 3-D bodies. In this section an estimation or stochastic method for approximating this contact is presented.

For this formulation consider the 2-D contact model of a target and a contactor (see Figure 6-2). Let the random variables that describe the stochastic surface distribution along the principle directions of the contactor and the target be $\langle z_1^c \rangle$, $\langle z_2^c \rangle$, $\langle z_1^t \rangle$ and $\langle z_2^t \rangle$ respectively.

Since we are using the the principle directions of the surface we can use the geometric concepts of *spherical products* and distributions of curvatures over surfaces as detailed in

²Buchberger [52] presents a non-linear computational geometry method using Gröbner Bases for the exact calculation of the interpenetrations of superellipsoids and states the problems to be extremely expensive. For more details on this see [53, 54].

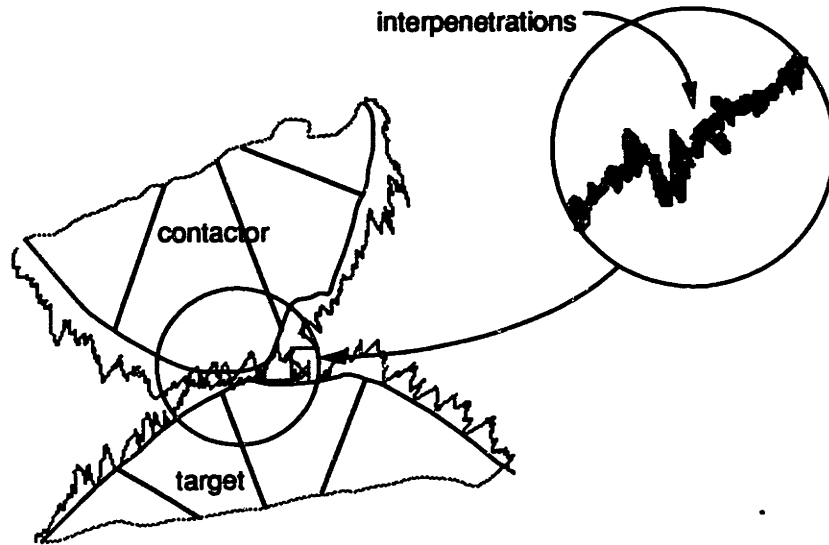


Figure 6-2: Contact of two “rough” objects

Chapter 2 to get the estimates of roughness of the surface at any point, in any direction, by estimating the penetrations in the two orthogonal directions on the surface. This linear combination of estimates in the two orthogonal directions gives the stochastic characterization for any direction from a point on the surface of the object (see Figure 6-3).

$$\langle r_p \rangle = \langle r_1 \rangle \cos^2(\alpha) + \langle r_2 \rangle \sin^2(\alpha) \quad (6.5)$$

where α is the angle traversing the surface at that point from one of the principle directions of the surface. Figure 6-3 shows a 3-D body and shows two principle directions on the surface. Both these directions could have different distributions, which are specified, allowing the surface distribution in any specific direction to be calculated.

For our formulations, we are principally concerned with the direction of motion. Assume that the relative direction of motion of the target and the contactor is in u direction in space, which subtends angles of α_c and α_t with respect to the principle axes of the contactor and target respectively. Since we know the roughness along the principle directions, the

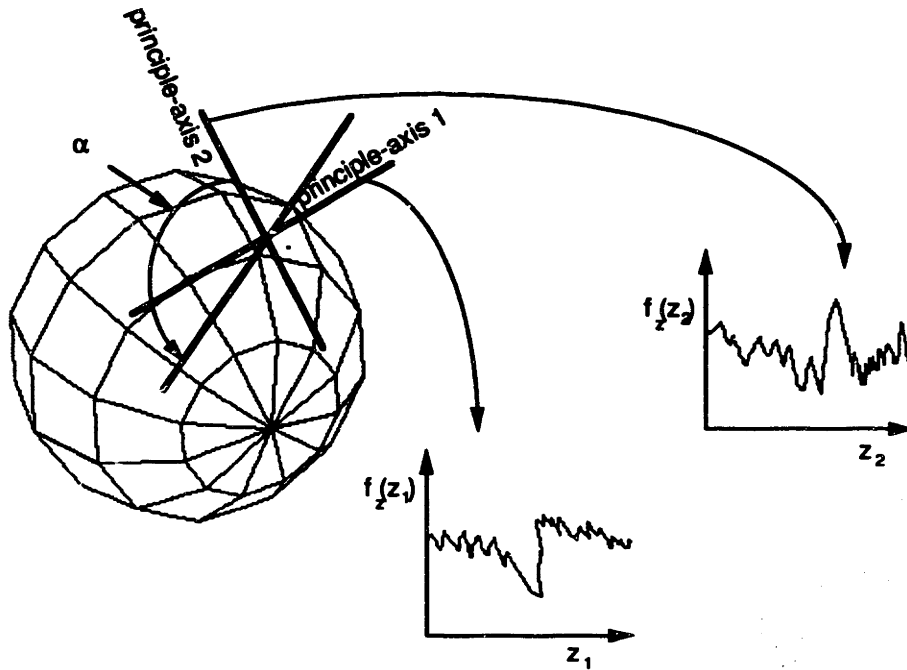


Figure 6-3: Rough Surface, from two principle directions of the object.

roughness in the direction of motion can be calculated using Equation (6.5);

$$\begin{aligned} \langle z_u^c \rangle &= \langle z_1^c \rangle \cos^2(\alpha) + \langle z_2^c \rangle \sin^2(\alpha_c) \\ \langle z_u^t \rangle &= \langle z_1^t \rangle \cos^2(\alpha) + \langle z_2^t \rangle \sin^2(\alpha_t) \end{aligned} \quad (6.6)$$

The above operations are carried out by scalar multiplications of random variables and then additions. The random variables are added by using convolutions as described in Appendix D. Now we are interested in the intersection between the random variables $\langle z_u^c \rangle$ and $\langle z_u^t \rangle$, which will tell us the penetrations between the rough surfaces of the target and the contactor.

$$\langle z_u \rangle = \langle z_u^c \rangle + (-\langle z_u^t \rangle) \quad (6.7)$$

where $\langle z_u \rangle$ is the random variable for the interpenetration of the target and contactor (see Figure 6-2). As mentioned above, for a simple addition of two random variables the convolution of their density function is a most efficient method of calculating the density

of the sum (see Definition D.4). However, in this case the difference is required, so it is necessary to flip the density function of one of the random variables. Flipping the density is not a difficult operation and is explained in detail by DeGroot [55] and Papoulis [56] (see Appendix D). Method for flipping is built into the convolution algorithms which effectively uses the covariance matrices to determine the differences of random variables. Applying convolution we get;

$$f_{z_u}(z_u) = f_{z_u^c}(z_u^c) * f_{z_u^t}(z_u^t) \tag{6.8}$$

where $*$ means convolution (see Appendix D). Now we need to map the random variable

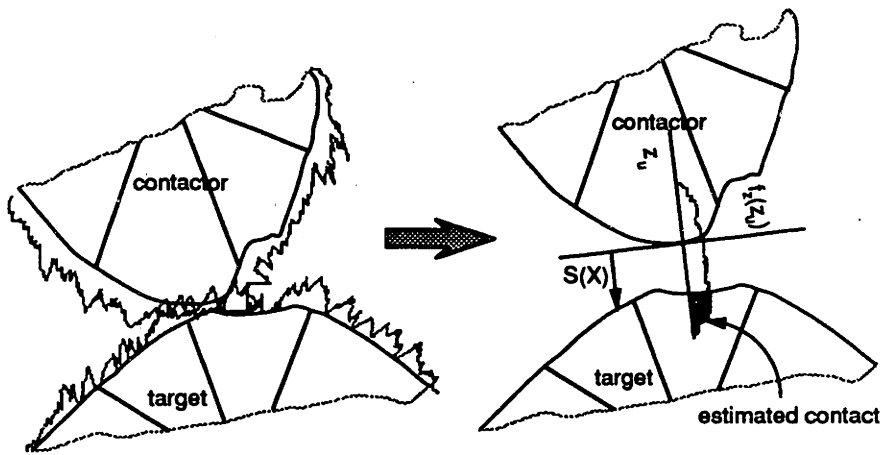


Figure 6-4: Using a random variable to estimate penetration on contact.

$\langle z_u \rangle$ on to the underlying geometric surface and determine the amount of penetration on the contactor during contact. Figure 6-4 shows the the density function of the estimated roughness due to contact. A probabilistic estimate of the amount of penetration of this roughness with the contactor is desired. To estimate the penetration of $\langle z_u \rangle$ inside the contactor the expression for the surface description is needed. Essentially, this is where the stochastic domain is mapped onto the geometric domain. From geometry of the surfaces an expression for the surface $S(X)$ can be setup (see Chapter 2). Using this relation we can

estimate the penetrations of the objects as shown by Figure 6-4 using the following relation;

$$F(z_u) = P \{S(X) \leq z_u \leq \infty\} = \int_{S(X)}^{\infty} f(\mu) d\mu \quad (6.9)$$

which gives the estimated interpenetration due to roughness. Now we will present an example, which explain all the above concepts, assuming Gaussian random variables for roughness.

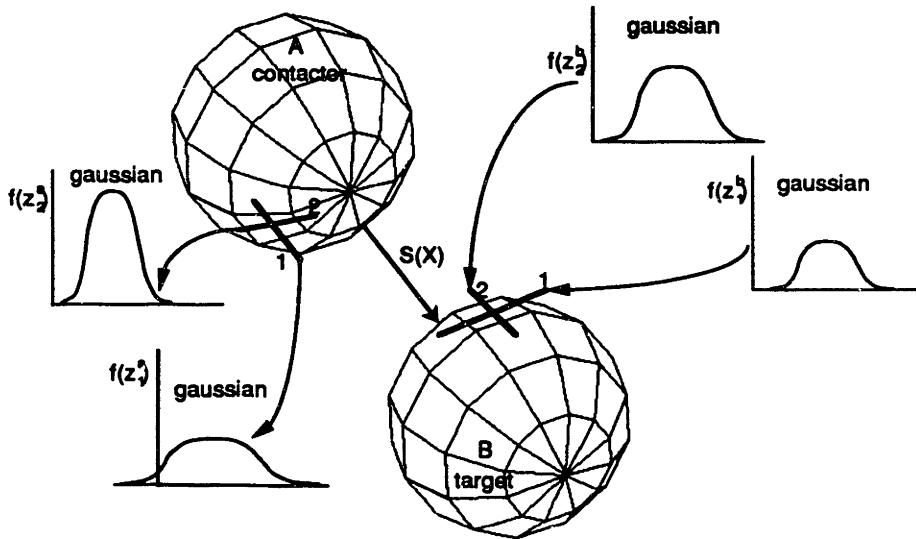


Figure 6-5: Example 6-5: Contact of two spheres, using Gaussian as roughness distributions.

Example 6.1 Consider a collision between two spheres, X_a and X_b (see Figure 6-5). The texture mapping along their principle directions, on both these spheres is a Gaussian function (see Appendix D) as given below;

$$z_1^a = k_1 e^{-c_1 x^2}, \quad z_2^a = k_2 e^{-c_2 x^2}$$

$$z_1^b = k_3 e^{-c_3 x^2}, \quad z_2^b = k_4 e^{-c_4 x^2}$$

where superscript "a" denotes sphere A, superscript "b", sphere B and $k_1, k_2, k_3, k_4, c_1, c_2, c_3, c_4$ are constants. Assume, the direction of motion to be in direction u . Calculate the roughness distribution of the spheres in the direction of motion using Equation (6.6). As we are using Gaussian random variables the linear combination of the random variables in the principle directions is still going to return a Gaussian with a different mean and deviation

(k_a, k_b, c_a and c_b are constants), hence

$$z_u^a = k_a e^{-c_a x^2}, \quad z_u^b = k_b e^{-c_b x^2}$$

are the surface distributions in the direction of motion.

Now using Equation (6.8) and convolving the the densities of the two random variables (z_u^a) and (z_u^b), the density of the random variable of the intersection is obtained. Again, a convolution of a Gaussian is a Gaussian hence we have another Gaussian with some constants (k, c) as deviations and means.

$$z_u = k e^{-c x^2}$$

This Gaussian distribution can now be used to estimate the penetration in geometric domain using Equation (6-4).

$$F(z_u) = P \{S(X) \leq z_u \leq \infty\} = \int_{S(X)}^{\infty} k e^{-c x^2} d\mu$$

integrating and using the properties of Gaussians;

$$F(z_u) = \int_{S(X)}^{\infty} k e^{-c x^2} d\mu$$

$$F(z_u) = k \int_{S(X)}^{\infty} e^{-c x^2} d\mu$$

$$F(z_u) = G(\infty) - G(KS(X))$$

where; $G(\infty) = 1$ and $G(x) = \varepsilon(x) + \frac{1}{2}$ (see Appendix D). Where $\varepsilon(x) = \text{erf}(x)$ Hence;

$$F(z_u) = 1 - \varepsilon(KS(X)) + \frac{1}{2}$$

$$F(z_u) = \frac{1}{2} - \varepsilon(KS(X))$$

where K is a constant, and $S(X)$ is to be determined as per the surface description. From Equation (D.16) (Appendix D)

$$\varepsilon(\rho) = \frac{1}{\sqrt{2\pi}} \int_0^\rho e^{-\frac{\xi^2}{2}} d\xi$$

Now our system of equations becomes;

$$F(z_u) |_{S(X)} = \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \int_0^{\xi=S(X)} e^{-\frac{\xi^2}{2}} d\xi$$

or;

$$F(z_u) |_{S(X)} = \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \int_{S(X)} \left\{ \int_0^{\xi=S(X)} e^{-\frac{\xi^2}{2}} d\xi \right\} dX$$

or in the case of two principle directions of the surface $S(X) = s(x_1, x_2)$;

$$F(z_u) |_{S(X)} = \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \int_{x_1} \int_{x_2} \left\{ \int_0^{\xi=s(x_1, x_2)} e^{-\frac{\xi^2}{2}} d\xi \right\} dx_1 dx_2$$

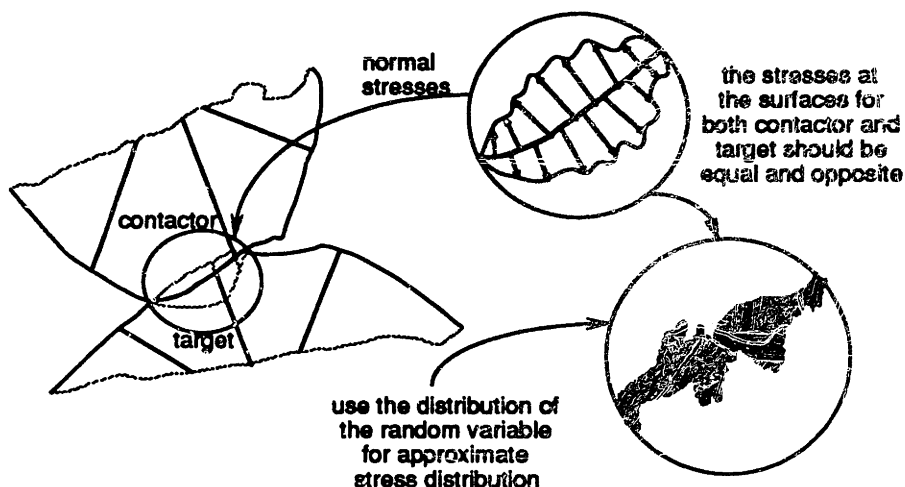


Figure 6-6: Stress distributions on collision and stochastic penetration.

6.4 Stochastic Estimation of Stress Distribution on Contact

So far the discussion was related to how a stochastic function could be used for characterizing the contact problem. The same formulation could be extended to determine forces and stress distributions over surfaces. As mentioned in Chapter 4 and Chapter 5 the difficult problem with non-rigid body contact and collision response is to determine the contact patch and the stress distribution at contact. Essentially the equation

$$F = \int_A \sigma(A) dA \quad (6.10)$$

has to be solved, where σ specifies the stress at the contact area and varies from point to point. Hence it is essentially dependent on contact region A (see Figure 5-4). Under most circumstances, specially for rigid contacts it is assumed that the stress distribution

is constant, and the integration over the area is used to determine the collision or contact response. Because of the stochastic component of the surface, traditional methods of geometry and mechanics are almost useless when it comes to determine the area over which to integrate. Fortunately, the above formulation, when used for approximating the area of contact using the texture distributions of the contactors, also provides a good estimation technique for complementing the information that can be derived from the underlying geometry and mechanics, without going in to the details of exact penetration patch or surface of intersection. As it is obvious from seeing the formulation, the computational complexity of this stochastic method is much lower than that required to characterize a surface patch using traditional methods.

For this estimation approach it is important to recognize that essentially we are simply pushing the idea of stochastic approximation /abstraction one step forward. The main purpose in characterizing a contact patch was to determine the penetration. This penetration is the deformation on the surface of the contactors and hence develops surface strains which sets up a stress distributions across the surface where normally stress is proportional to strain ($\sigma \propto \epsilon$). By using the random variables $\langle z_1 \rangle$ and $\langle z_2 \rangle$ as above to determine the profile of the stress distributions and variation, the integration over the area can be extremely simplified, computational complexity reduced and effects of deformed rough surfaces incorporated into real time systems. Thus for each point on the surface we calculate the surface interpenetration area by Equation (6.9) and applying stress times area, approximate the total response for a contact (see Figure 6-6).

In this chapter a stochastic estimation approach for a *physically based virtual world* was presented. The main desire for introducing this method was spawned by a need for cost-efficient modeling system, without compromising the realism of simulation. The methods discussed above are approximate and efficient but quite practical. In the next chapter another set of techniques will be presented, the emphasis there also to achieve a physical system that is efficient in response. Procedures for filtering out the error incorporated into the system due to computational restrictions will be introduced. The concepts of stochastic estimates and dynamic time response with control theory will be studied.

Chapter 7

Application of Control Theory

In Chapter 3, formulations for dynamic and static analysis were presented and in Chapters 4 and 5 it was shown how these formulations are used for dynamic simulations. However, at the same time it was also mentioned that to keep the systems time efficient and real time the time-step size has to be overly large and the spatial sampling of the objects coarse. This results in erroneous inputs to the system and hence, instability. The available options are to either use control theory methods to remove the instabilities or to decrease the time-step. Since decreasing the time step is not a feasible option for physical modeling in a virtual world, the control and filtering methods are presented in this chapter [57, 58].

7.1 Problems with Physical Simulations

The major problem with dynamic physical simulations for real time physically based virtual worlds is that the input to the system, which are forces, i.e. the contact, collision and friction forces as described in Chapters 4 and 5, have a very broad frequency distribution. Unfortunately because of real time restrictions, we are restricted to constant and relatively large time steps. This results in the inability of the system to adequately describe the response due to these inputs. This can be considered as an error and approximated as an input of a noise signal to the system (see Figure 7-1), resulting in instability of the system and hence unrealistic behavior. This problem can be solved by smaller time steps or super time sampling at the time of force input (collision). Similar steps could be taken in terms

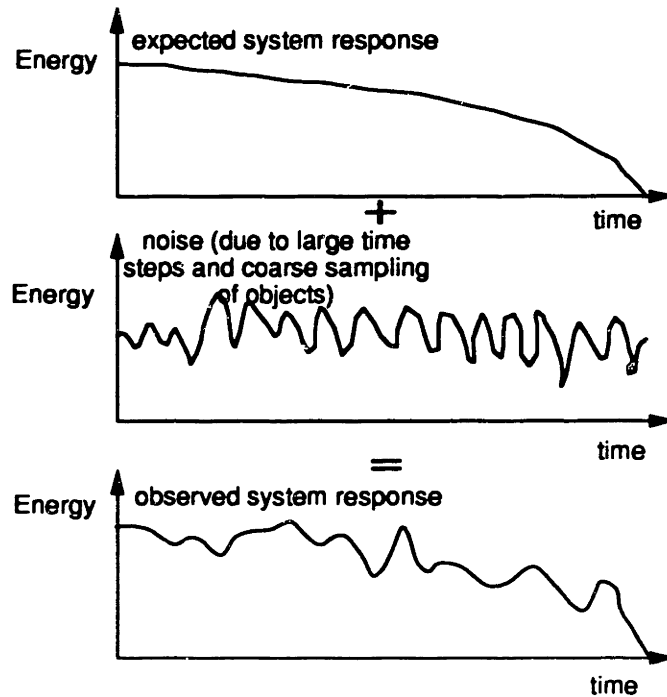


Figure 7-1: System response; expected and observed.

of sampling points of objects about to collide. However the compromise in doing so is loss of real time response as the simulations slows down every time there is a force input to the system, e.g., a collision. To minimize these problems, control mechanisms and processes with discrete-time dynamics, which will impart stability to the system at every time step are introduced. For this control, the theories of stochastic control using Gaussian filters, Wiener filters, Kalman Filters and Optimal Observers will be used.

Application of control theory in physically based modeling has been quite limited. Stekete and Badler [59] presented methods for *kinetic adjustments and phrasing control for keyframe animation*. Brotman and Netravali [60] used *optimal control* methods for motion interpolation for *keyframe animation*. Both these are strictly dealing with *keyframe animations* and hence are far from control of dynamic simulations.

In this thesis only some of the basic and simple concepts of control theory will be presented. An interested reader is referred to see Karnop and Rosenberg [61], Friedland [57] and Aoki [58].

For the purpose of discussion of control and filtering we will use examples from the Thingworld Modeling System. First a simple example using box filters will be used to reduce the effects of the error introduced. Then concepts of Wiener filtering will be presented to impart an improved predictive control, and finally the concept of Kalman filters will be presented and a Kalman Filter will be designed for our collision model.

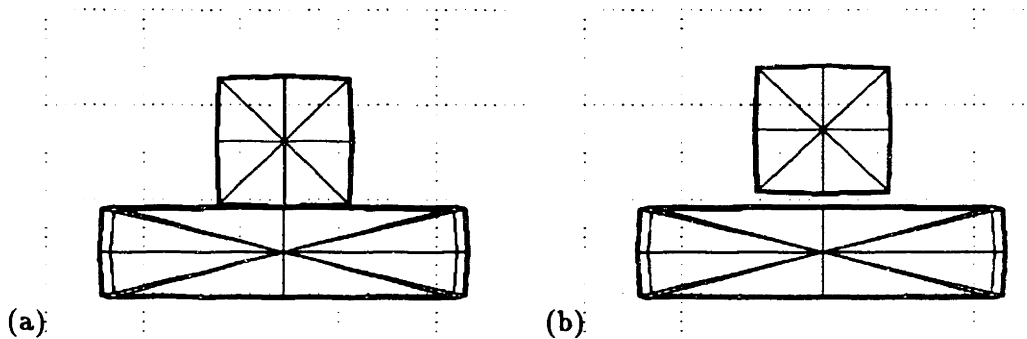


Figure 7-2: An example of contact, A cube lying on the floor (a) Before simulation (time step = 0) and (b) during simulation (time step = 20) [no contact]

Example 7.1 Figure 7-2 shows a simulation of a cube lying on a flat surface on the Thingworld Modeling System. This simulation ran for 50 time steps and using Mathematica¹ [62] the plots shown in figures 7-4 were generated. Figures 7-4(a),(b) and (c) show the x , y and z direction forces respectively, and 7-4(d) shows the total force magnitude on the 26 nodes of the element for the 50 time steps of simulation. Only few of the nodes appear to have been excited in this simulation. These are the nodes that are in contact with the flat surface. Note that the forces at each node, as the simulation time increases, fluctuate. For example at about the time step of 10, the forces at the nodes appear to be zero. This is because the contact has been lost. This fluctuation in forces is due to the reasons explained above and in this chapter methods will be described to smooth down the plots shown in Figure 7-4. Note that the expected behavior is a smoothly descending f vs. t curve. The sudden kinks in the plot are due to the extraneous forces that were picked up due to coarse quantizations during the simulations.

Now some important concepts of control theory will be presented.

¹Trademark, Wolfram Research

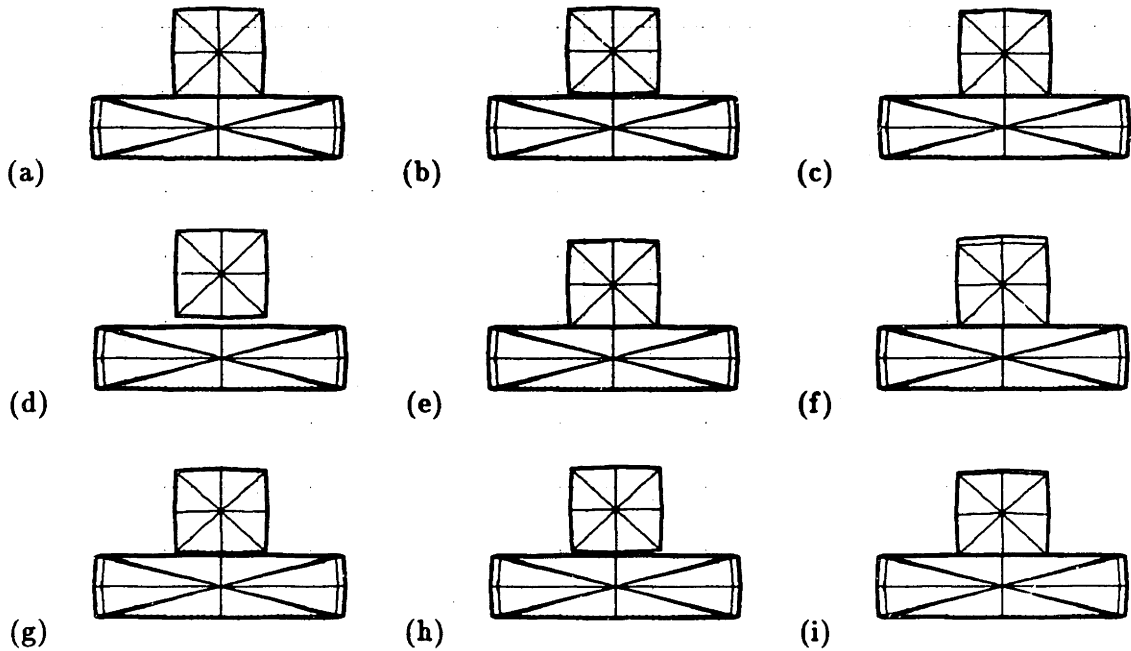


Figure 7-3: An example of contact, A cube lying on the floor (a) time step = 5 (b) time step = 10 (c) time step = 15 (d) time step = 20 (e) time step = 25 (f) time step = 30 (g) time step = 35 (h) time step = 40 and (i) time step = 45.

7.2 Feedback Control

One of the most important aspects of control theory is Feedback Control, which is defined as a conscious, intentional use of feedback to control the behavior of a dynamic system.

Feedback control plays an essential role in the kind of control that is needed by the dynamic systems that are used for virtual environments. It is required that a feedback mechanism should be set up that keeps a check on outputs of the simulation and sends information if the output is getting unstable. With reference to our Example 7.1, a feedback mechanism to check if the forces picked up by nodes are getting bigger and/or if the contact will be lost in the next step is required. Essentially, a system to check what the output was at the current step and then to adjust and estimate the output at the next step is required. In next few sections the topics of filtering and adding noise to the system to gain optimal control will be addressed.

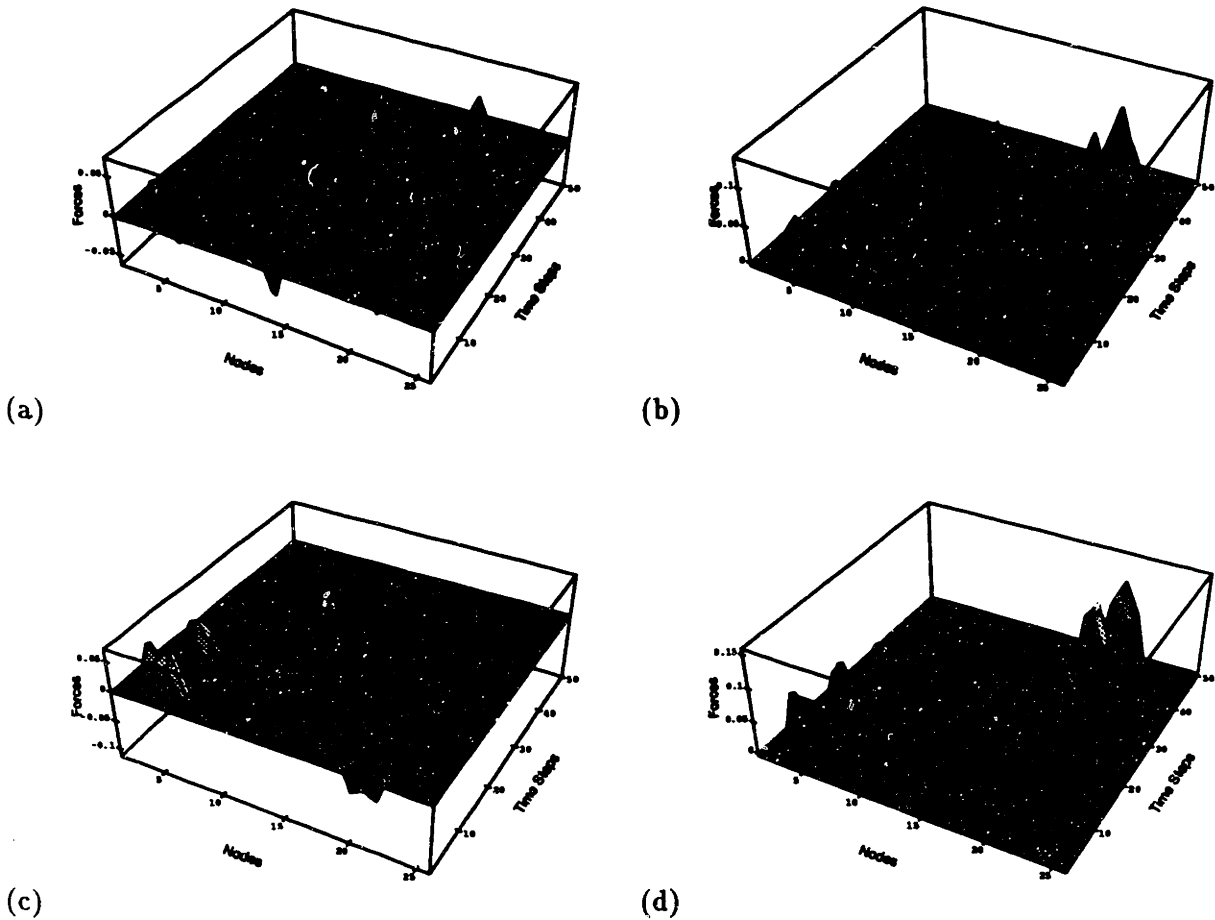


Figure 7-4: Forces on the 26 nodes of a cube lying on the floor for 50 time steps (a) Forces in x-direction (b) Forces in x-direction (c) Forces in y-direction and (a) Magnitude of Forces.

7.3 Filtering

Filtering is one of the most common operations in signal processing, image processing, raster image synthesis and system dynamics. Its applications include image enhancements, low pass filtering (blur), antialiasing, image, warping, texture mapping etc. It is used where the signal (input) needs to be modified to achieve a desired output (filtered). The fundamental filtering operation is *convolution* (see Section D.1), in which a weighted function or *kernel* is passed over an input signal and a weighted average is computed for each output sample.

As in the case of filtering of dynamic response, the signal is only one dimensional, hence the formulations for *one dimensional filtering* are presented.

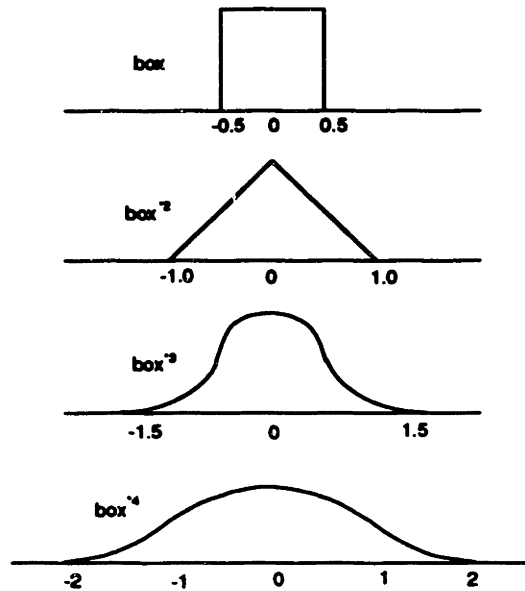


Figure 7-5: Repeated box filters of order 1-4 [63]

7.3.1 One Dimensional Filtering

An unweighted average is considered equivalent to a convolution with a box. The average value of a signal f between a and b is:

$$\int_b^a f(x)dx = \frac{F(b) - F(a)}{b - a} \quad (7.1)$$

where F is the indefinite integral of f . This formula can be used to convolve f with a box of width W , if $a = x + \frac{W}{2}$ and $b = x - \frac{W}{2}$. The above described method, albeit not the most elegant, would satisfy the filtering requirements of the dynamic simulations as desired. This same method could now be generalized to higher quality filters. Following is a definition of repeated filtering, which is essential for higher order filtering methods. Consider a box with a kernel of width W :

$$box_w(x) = \begin{cases} \frac{1}{W} & |x| < \frac{W}{2} \\ 0 & |x| \geq \frac{W}{2} \end{cases}$$

When a box is convolved with itself n times, the result is called a *repeated box filter*

$$box^{*n} = box * box * \dots * box \text{ (ntimes)}$$

For notations and identities of convolution see Section D.3.

Figure 7-5 shows the first order to the fourth order repeated box filter.

The filters that are shown above, and the filters of higher orders can be ranked in quality as follows:

$$box < box^{*2} < box^{*3} < \dots < Gaussian < sine$$

By the *Central Limit Theorem* as n increases the repeated box filters broaden and approach Gaussian [64]. As can be seen from Figure 7-5, where the fourth order box filter almost resembles as Gaussian.

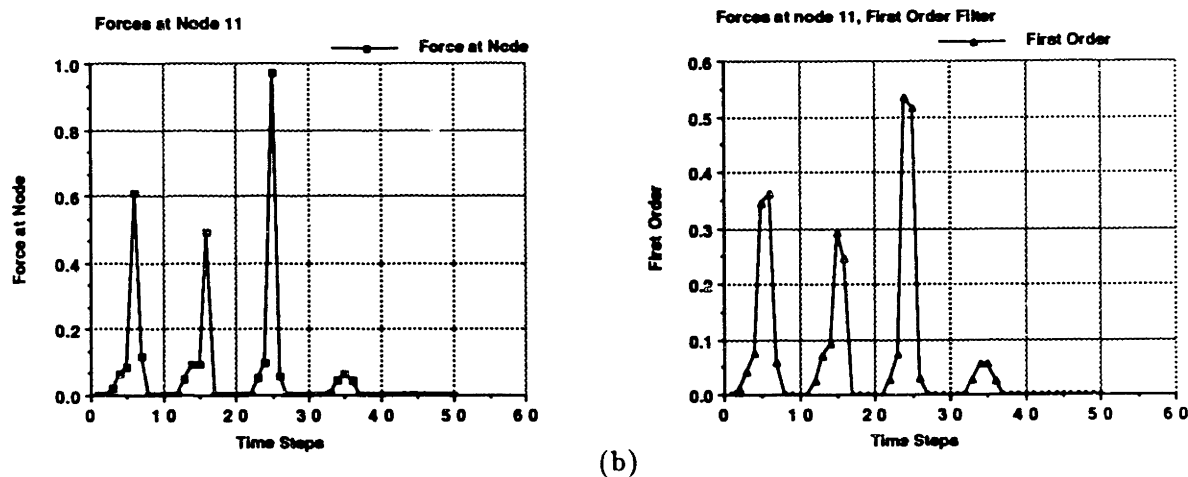


Figure 7-6: Forces on the node 11 of a cube lying on the floor, 50 time steps (a) No *filtering* and (b) First order *filtering*.

Example 7.2 In the previous example of the cube lying on a flat surface (Example 7.1) the force variations were shown as the simulation was going on. Following from the same example, repeated box filtering will be applied to the same simulation. The graphs of forces on only one node (node 11) are shown in Figures 7-6, 7-7 and 7-8. These graphs show how the response was smoothed as filtering was applied (see the difference between 7-6(a) and 7-6(b), 7-7). All of the repeated filters with the initial unfiltered response are plotted on

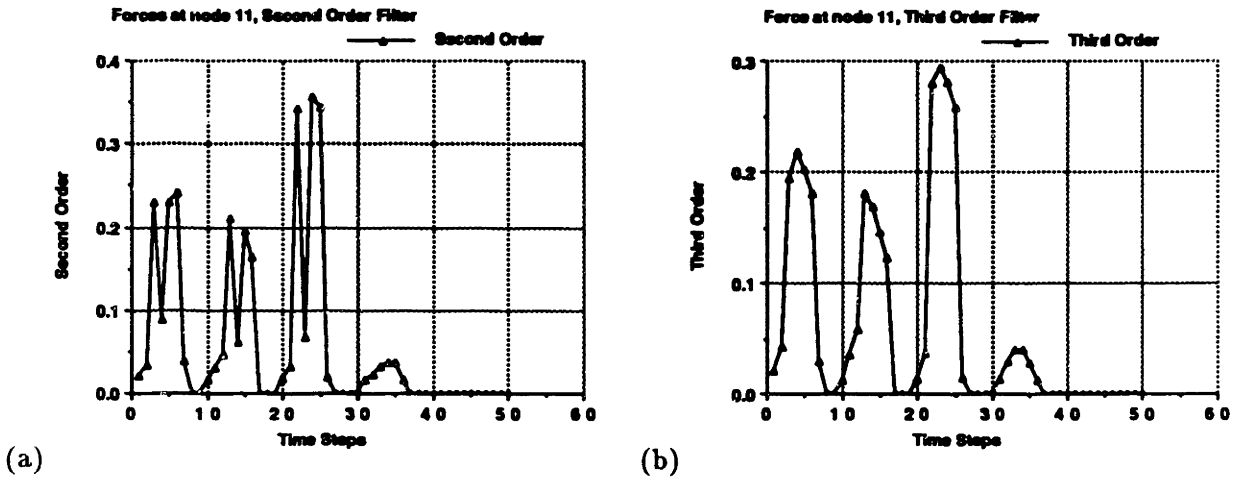


Figure 7-7: Forces on the node 11 of a cube lying on the floor, 50 time steps (a) Second order *filtering* and (b) Third order *filtering*.

the same graph 7-8 for a more quantitative comparison. Note that the response has become smoother and smoother per repeated applications of the box filter.

In the next section, a special type of filtering technique from the the realms of *Control Theory* and *Signal Processing* is described and its feasibility for dynamics systems is presented.

7.4 Wiener Filtering

The Wiener filter is a special type of a filter where the goal is to produce a best possible estimation of a signal from a noise corrupted observation [65]. The discrete form of the problem that as considered by Wiener is as following. Consider a signal $x(n)$,

$$x(n) = d(n) + w(n) \quad (7.2)$$

where $d(n)$ is the signal that is to be recovered from $x(n)$ and where $w(n)$ is contaminating the observation $x(n)$. Assuming that both $d(n)$ and $w(n)$ are random processes Wiener presented methods to design filters that would produce a minimum mean square error

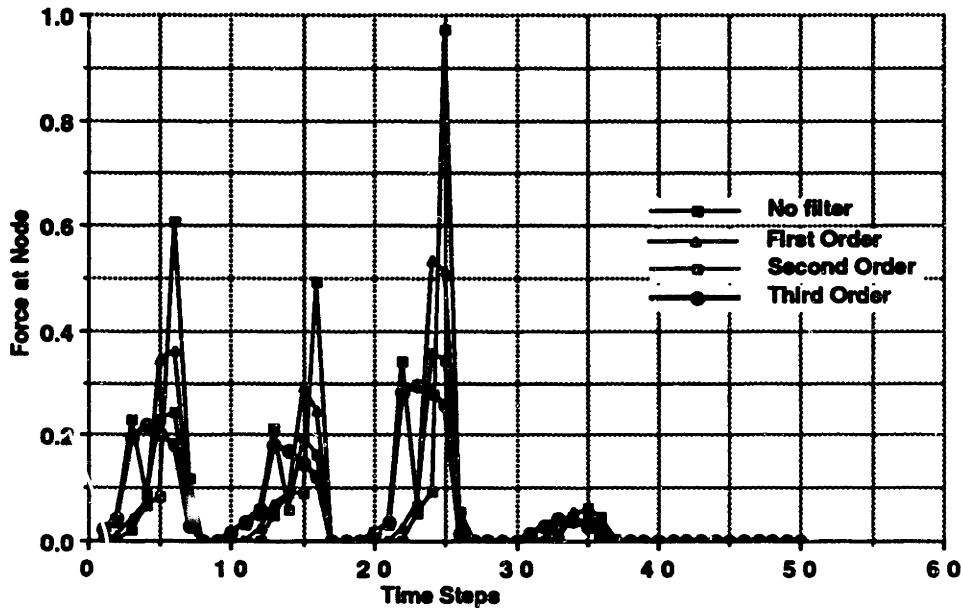


Figure 7-8: Forces on the node 11 of a cube lying on the floor, 50 time steps, First, Second and Third order *filtering* and no *filtering*.

estimate of $d(n)$. Specifically, with

$$\varepsilon = E \{ e^2(n) \} \tag{7.3}$$

where;

$$e(n) = d(n) - \hat{d}(n) \tag{7.4}$$

the problem is to design an optimum filter $\mathcal{H}(s)$, that minimizes ε . A wide range of problems like smoothing and linear prediction can be solved, assuming that $x(n)$ is some signal from which an estimate $\hat{d}(n)$, of a desired signal, $d(n)$, is to be found using a linear shift-invariant Wiener filter $\mathcal{H}(s)$. Now we will present a simple formulation for a design of a Wiener filter, by deriving discrete forms of Wiener-Hopf equations which defines optimum filter coefficients as the solution to a set of linear equations.

7.4.1 Wiener Filter : Formulation

Lets $x(n)$ be a given signal, from which we want to estimate another signal $d(n)$. Assuming that $x(n)$ and $d(n)$ are random processes with correlation functions (see Appendix D) $r_x(k)$ and $r_d(k)$ and a cross-correlation of $r_{dx}(k)$. Consider a linear estimate of $d(n)$ by taking a linear combination of $(p + 1)$ values of $x(n)$,

$$\hat{d}(n) = \sum_{i=0}^p a(i)x(n-i) \quad (7.5)$$

Therefore, $\hat{d}(n)$ is generated as an output of the wiener filter having a unit sample response $a(n)$,

$$\hat{d}(n) = a(n) * x(n) \quad (7.6)$$

and our goal is to design a filter,

$$A(s) = \sum_{i=0}^p a(i)z^{-i} \quad (7.7)$$

with the error

$$e(n) = d(n) - \hat{d}(n) \quad (7.8)$$

as small as possible. Since both $d(n)$ and $\hat{d}(n)$ are random processes the error functional that has to be minimized is the mean square error

$$\varepsilon = Ee^2(n) = E \left\{ [d(n) - \hat{d}(n)]^2 \right\} \quad (7.9)$$

A necessary condition for the set of filter coefficients $a(0), a(1), \dots, a(p)$ to minimize the mean square error in the estimate of $d(n)$ is that the derivatives of ε with respect to each of the coefficients vanish,

$$\frac{\delta \varepsilon}{\delta a(k)} = 2E \left\{ e(n) \frac{\delta e(n)}{\delta a(k)} \right\} = 0 \quad ; \quad k = 0, 1, 2, \dots, p \quad (7.10)$$

Since,

$$e(n) = d(n) - \sum_{i=0}^p a(i)x(n-i) \quad (7.11)$$

therefore

$$\frac{\delta e(n)}{\delta a(k)} = -x(n-k) \quad (7.12)$$

and Equation (7.10) could be rewritten as

$$\begin{aligned} \frac{\delta \varepsilon}{\delta a(k)} &= -2E\{e(n)x(n-k)\} = 0 \quad ; \quad k = 0, 1, 2, \dots, p \\ E\{e(n)x(n-k)\} &= 0 \quad ; \quad k = 0, 1, 2, \dots, p \end{aligned} \quad (7.13)$$

which is known as the orthogonality principle or the projection theorem. From Equations (7.12) and (7.13) we get

$$E\{d(n)x(n-k)\} - \sum_{i=0}^p a(i)E\{x(n-i)x(n-k)\} = 0 \quad (7.14)$$

Therefore, with $r_x(k) = E\{x(n)x(n-k)\}$ and $r_{dx}(k) = E\{d(n)x(n-k)\}$ we get

$$\sum_{i=0}^p a(i)r_x(k-i) = r_{dx}(k) \quad ; \quad k = 0, 1, 2, \dots, p \quad (7.15)$$

Equation (7.15) describes a set of $(p+1)$ linear equations in the $(p+1)$ unknowns $a(l)$, $l = 0, 1, 2, \dots, p$ and may be written in matrix form as follows;

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \dots & r_x(p) \\ r_x(1) & r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \dots & r_x(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \\ r_{dx}(2) \\ \vdots \\ r_{dx}(p) \end{bmatrix} \quad (7.16)$$

This equation is known as the Wiener-Hopf equation and is written in matrix form as;

$$\mathbf{R}_x \mathbf{a} = \mathbf{r}_{dx} \quad (7.17)$$

where \mathbf{R}_x is a $(p+1) \times (p+1)$ symmetric correlation matrix, \mathbf{a} is the vector of Wiener filter coefficients and \mathbf{r}_{dx} is the cross-correlations of between the desired $d(n)$ signal and the observed signal $x(n)$.

We now evaluate the the mean square error between the estimate, $\hat{d}(n)$ and $d(n)$.

$$\varepsilon = Ee^2(n) = E\{e(n)[d(n) - \sum_{l=0}^p a(l)x(n-l)]\} \quad (7.18)$$

$$= E\{e(n)d(n)\} \sum_{l=0}^p a(l)E\{e(n)x(n-l)\} \quad (7.19)$$

from the orthogonality principle Equation (7.13) we see that for the Wiener coefficients $E\{e(n)x(n-k)\} = 0$. Therefore the second term in above equation is zero;

$$\varepsilon_{min} = Ee(n)d(n) = E\left\{\left[a(n) - \sum_{l=0}^p a(l)x(n-l)\right]d(n)\right\} \quad (7.20)$$

taking expected values into account

$$\varepsilon_{min} = r_d(0) - \sum_{l=0}^p a(l)r_{dx}(l) \quad (7.21)$$

in vector notation

$$\varepsilon_{min} = r_d(0) - \mathbf{r}_{dx}^T \mathbf{a} \quad (7.22)$$

Since

$$\mathbf{a} = \mathbf{R}_x^{-1} \mathbf{r}_{dx} \quad (7.23)$$

then the minimum error maybe written in terms of the correlation matrix \mathbf{R}_x and the cross-correlation matrix \mathbf{r}_{dx} as follows

$$\varepsilon_{min} = r_d(0) - \mathbf{r}_{dx}^T \mathbf{R}_x^{-1} \mathbf{r}_{dx} \quad (7.24)$$

Note if no filtering was to be performed then the mean square error in the estimate, which is denoted by ε_0 is

$$\varepsilon_0 = E\{d^2(n)\} = r_d(x) \quad (7.25)$$

then Equation (7.22) becomes

$$\varepsilon_{min} = \varepsilon_0 - \mathbf{r}_{dx}^T \mathbf{a} \quad (7.26)$$

This completes the formulation of the Wiener filters under the assumptions that we

began with. The application of interest to us in multi-body dynamics for virtual world modeling is linear prediction and will be explained in the next section.

7.4.2 Linear Prediction with Wiener filters

Linear prediction is an important problem in many applications which require forecasting the behavior. It is concerned with predicting (estimating) the output $x(n)$ at any time $n+1$ from the current and previous values of $x(n)$. Specifically, the idea is determine the value of $x(n+1)$, given $x(k)$ where $k < n$, using the predictor of the form,

$$\hat{x}(n+1) = \sum_{k=0}^p a(k)x(n-k) \quad (7.27)$$

The linear filtering problem can be cast into the framework of Wiener filter if $d(n) = x(n+1)$ in the formulation developed in the previous section. In setting up the Wiener-Hopf equation all that is needed is to evaluate the cross-correlation, $r_{dx}(k)$, between $d(n)$ and $x(n)$. Therefore, with

$$r_{dx}(k) = E\{d(n)x(n-k)\} = E\{x(n+1)x(n-k)\} = r_x(k+1) \quad (7.28)$$

the Wiener-Hopf equation for the optimum linear predictor is

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \dots & r_x(p) \\ r_x(1) & r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \dots & r_x(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} r_x(1) \\ r_x(2) \\ r_x(3) \\ \vdots \\ r_x(p-1) \end{bmatrix} \quad (7.29)$$

and the mean square error is

$$\epsilon_{min} = r_x(0) - \sum_{k=0}^p a(k)r_x(k+1) \quad (7.30)$$

Now Lets consider a couple of examples of applying a Wiener filter to a dynamic system

Example 7.3 Consider Figure 7-1, which shows the actual and observed responses and a disturbance process that introduces the error. We will try to develop a system using a Wiener filter for that model.

Using the same notations as above and assuming both the observed and error signals to be Gaussians we have;

$$d(t) = \mathcal{G}[\mu(t), \sigma_d^2]$$

$$w(t) = \mathcal{G}[0, \sigma_w^2]$$

where the observed response has both mean and variance, while the contaminating response is zero-mean. From Equation (7.2) we get;

$$x(t) = d(t) + w(t) \simeq \mathcal{G}[\mu(t), \sigma_x^2]$$

the correlations are given by

$$r_x(\tau) = E \{x(t) + x(t - \tau)\}$$

$$r_x(\tau) = E \{d(t) + w(t)\} \{d(t - \tau) + w(t - \tau)\}$$

Simplifying and throwing away terms for no correlation between $d(t)$ and $w(t)$, we get;

$$r_x(\tau) = r_d(\tau) + r_w(\tau)$$

as $w(t)$ is zero mean the correlation function is simply determined and is in term of its variance. The correlation function $r_d(\tau)$ can be determined using Equation (D.6). However it should be noted here that for a dynamic system $d(t)$ is deterministic, in which case the problem is much simplified and the need for $r_d(\tau)$ vanishes. Therefore the $r_x(\tau)$ can be easily determined and the linear prediction of Wiener filters can be applied to get the actual response.

Now another example in which another formulation of Wiener filters is applied [66]

Example 7.4 Consider the case of dynamic simulation, specifically the one of a cube lying on a flat surface, as was discussed in Examples 7.1 and 7.2. A very good assumption of the profile of the power spectrum of this simulation would be a Gaussian, and we can assume a random variable of constant density function as the disturbance input to the system. As per the methods of Wiener filters, we need to determine the actual response of the system by getting rid of the corrupted input that was incorporated into the system due to large time steps and coarse sampling of the object.

We assume the observed signal $d(v)$ to be a Gaussian $k_1 e^{-k_2 v}$ and the noise signal $w(v)$ to be k_3 , where k_1 , k_2 and k_3 are constants.

Now using the Wiener filtering to remove the noise components from the system in a

form linear prediction format and using the following equation for a Wiener filter [66];

$$P_{actual}(v) = \mathcal{H}^{-1}(v) \|\mathcal{H}(v)\|^2 \left[\|\mathcal{H}(v)\|^2 + \frac{\|w(v)\|}{\|d(v)\|} \right]^{-1} P_{observed}(v) \quad (7.31)$$

Assuming $\mathcal{H}(v)$ to be equal to 1 in our case we get;

$$P_{actual}(v) = \left[1 + \frac{\|k_3\|}{\|k_1 e^{k_2 v^2}\|} \right]^{-1} P_{observed}(v)$$

$$P_{actual}(v) = \left[\frac{\|k_1 e^{k_2 v^2}\| + \|k_3\|}{\|k_1 e^{k_2 v^2}\|} \right]^{-1} P_{observed}(v)$$

as adding an constant density to a Gaussian RV can be considered as adding another Gaussian with zero mean and some deviation, and hence returns a Gaussian and a Gaussian divided by a Gaussian is also a Gaussian RV, therefore the output of the system after the filter is a Gaussian as expected.

7.5 Kalman Filtering

Now another method of imparting control on a dynamic system is approached. The emphasis from now on would be adaptive control procedures and attempt would be made to define formulations to achieve optimal control of the system. Essentially, this formulation will deal with both the estimation and control parts of control systems, both of which are discussed below.

Kalman filters (see Appendix E) are used both to forecast and to obtain estimates of state vectors of dynamic models given a set of data over some fixed period of time. Estimation part is usually referred to as the smoothing of data [58]. In both of the above applications, the knowledge of the model dynamics is required and at least the means and the second moments of noise statistics are needed. Outputs from Kalman filter appear to be the same as that of approximate estimates, when noises are *Gaussian* and same as the (weighted) least squares with non-*Gaussian* noises. *Kalman filters* provide a computationally efficient ways for incorporating new or additional information for revising estimates. ²

²For further reading on Kalman filter and estimation and optimal control, see [58, 57].

7.5.1 Kalman Filter : Observer

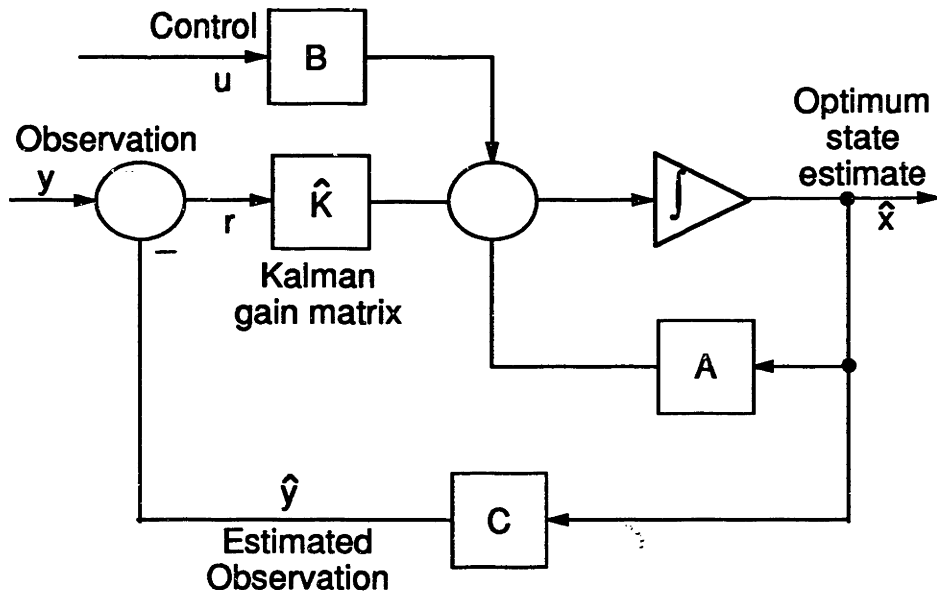


Figure 7-9: Block diagram representation of Kalman Filter as an optimum observer, from Friedland [57].

Given a dynamic process

$$\dot{\mathcal{X}} = A\mathcal{X} + BU + FV \quad (7.32)$$

where U is a known input and the observations are given by

$$y = C\mathcal{X} + W \quad (7.33)$$

where V and W are white noise processes (see Section E.3 and Definition E.9), having known spectral density matrices. The *optimum* observer as per Kalman [67, 68] is given by the following differential equation.

$$\dot{\hat{\mathcal{X}}} = A\hat{\mathcal{X}} + BU + \hat{K}(y - C\hat{\mathcal{X}}) \quad (7.34)$$

which defines an observer (see Definition E.6), provided that the gain matrix \hat{K} is optimally chosen. One of the remarkable properties of a *Kalman filters* is that it is optimum under any

reasonable performance criterion, provided the random processes are white and *gaussian* [58].

The gaussian requirement is a condition on the first-order probability density functions (see Section D.2) of \mathcal{V} and \mathcal{W} , e.g.;

$$f(\mathcal{W}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathcal{W}|^{\frac{1}{2}}} \exp^{-\frac{1}{2} \mathcal{W}^T \mathcal{W}^{-1} \mathcal{W}}$$

which is the multidimensional gaussian probability density function. Equation (7.34) is the best possible filter, when the dynamic process and observations are linear, and the random processes \mathcal{V} and \mathcal{W} are gaussian white noise. (If the random processes were not *gaussian* then a nonlinear filter is better) [57].

7.5.2 Gain Factor and Variance Equation of Kalman filters

The gain factor $K(t)$ in Equation (7.34) which makes the covariance matrix (Definition D.9) of the error small, is determined by the following method. The optimum value of $K(t)$ will be designated by $\hat{K}(t)$.

Define the error as $e = \mathcal{X} - \hat{\mathcal{X}}$ and using Equations (7.32), (7.33) and (7.34), the differential equation for the error is;

$$\begin{aligned} \dot{e} &= \dot{\mathcal{X}} - \dot{\hat{\mathcal{X}}} \\ &= A\mathcal{X} + F\mathcal{V} - A\hat{\mathcal{X}} - K(C\mathcal{X} + \mathcal{W} - \hat{C}\hat{\mathcal{X}}) \\ &= (A - KC)e + F\mathcal{V} - K\mathcal{W} \end{aligned} \tag{7.35}$$

Since \mathcal{V} and \mathcal{W} are white noise processes, their weighted sum is

$$\gamma = F\mathcal{V} - K\mathcal{W} \tag{7.36}$$

which is also white noise with a covariance matrix Q_γ . From Equation (D.7)

$$\begin{aligned} E\{[\gamma(t), \gamma^T(\tau)]\} &= F(t)E\{\mathcal{V}(t)\mathcal{V}^T(\tau)\}F^T(\tau) \\ &\quad - K(t)E\{\mathcal{W}(t)\mathcal{V}^T(\tau)\}F^T(\tau) \end{aligned}$$

$$\begin{aligned}
& - F(t)E \{ \mathcal{V}(t)\mathcal{W}^t(\tau) \} K^T(\tau) \\
& - K(t)E \{ \mathcal{W}(t)\mathcal{W}^t(\tau) \} K^T(\tau)
\end{aligned} \tag{7.37}$$

Assuming that the expected values on the right hand side of Equation (7.37) are of white noise then;

$$\begin{aligned}
Q_\gamma(t) &= F(t)V(t)F^T(\tau) - K(t)X^T(t)F^T(\tau) \\
&\quad - F(t)X(t)K^T(\tau) - K(t)W(t)K^T(\tau)
\end{aligned} \tag{7.38}$$

Let P_γ be the covariance matrix of the error, and by using the variance equation (Equation (D.10), Definition D.10),

$$\begin{aligned}
\dot{P} &= (A - KC)P + P(A^T - C^T K^T) + Q_\gamma \\
&= (A - KC)P + P(A^T - C^T K^T) + FV F^T \\
&\quad - KX^T F^T - FX K^T + FW K^T
\end{aligned} \tag{7.39}$$

If the cross-variance X between the excitation noise \mathcal{V} and the observation noise \mathcal{W} were absent, then the above equation would have complete correlation with the optimal control equations and therefore the analysis as used by Friedland [57] (Chapter 9) for optimal control formulations ³ can be employed to give

$$\hat{K} = (\hat{P}C^T + FX)W^{-1} \tag{7.40}$$

where the observation noise spectral density matrix W must be nonsingular. The differential equation of \hat{P} is

$$\dot{\hat{P}} = \tilde{A}\hat{P} + \hat{P}\tilde{A}^T - \hat{P}C^T W^{-1} C \hat{P} + F\tilde{V}F^T \tag{7.41}$$

where

$$\tilde{A} = A - FXW^{-1}C$$

³For a detailed handling of this formulation and derivation see Chapter 9, 10 and 11 of [57]

$$\tilde{V} = V - XW^{-1}X^T \quad (7.42)$$

7.5.3 Steady state Kalman filter

The Equation (7.41), known as the *Ricaati equation*, is valid for any finite time interval. If time is allowed to become infinite, the solutions may tend to infinity or they may remain finite. If all the matrices on the right-hand side of Equation (7.41) remain constant then constant steady state solution might exist, which is given by solving the matrix quadratic equation, also called the *algebraic Ricaati equation* [57]

$$0 = \tilde{A}\hat{P} + \hat{P}\tilde{A}^T - \hat{P}C^TW^{-1}C\hat{P} + F\tilde{V}F^T \quad (7.43)$$

The *algebraic Ricaati equation* (ARE) has a unique positive definite solution [57] if

1. The system is asymptotically stable, OR
2. The system defined by the pair $[A, C]$ is observable (Definition E.7) and the system defined by the pair $[A, FV\frac{1}{2}]$, (i.e., the system $\dot{x} = Ax + FV\frac{1}{2}v$ is controllable (Definition E.8)

7.5.4 Algorithm for optimal control using Kalman filters

A generalized algorithm for using Kalman to design a filter for control of the dynamic model is given by Table 7.1. This table is for steady-state Kalman filter. In the following example,

Algorithm for Design of Kalman filters	
1.	Set up the dynamic model as a dynamic state space system in the form of Equations (7.32) and (7.33).
2.	Determine the A, B, C and F matrices.
3.	Let P be the covariance matrix and apply Equation (7.43) for steady state to get p_1, p_2, \dots
4.	Calculate the Kalman filter gain matrix using Equation (7.40)
5.	Use Equation (7.34) and solve the differential equations to get the transfer functions and the filters

Table 7.1: Step by step method for designing a Kalman filter.

we design a Kalman filter for the collision response of two bodies using the algorithm shown

in Table 7.1. The formulation of collision response was presented in Chapter 4 and will be used here with some simplifying assumptions to show how a Kalman filter can be designed for a specific application.

Example 7.5 Consider the collision of two objects as described in Section 4.4. Let f and v (force and velocity) be the state variables. We will also assume damping is zero ($c = 0$), (see Section 3.4 for the validity of this assumption). The governing equations are;

$$m\ddot{x} + kx = F_a \quad (7.44)$$

after further simplification;

$$\ddot{x} + \Omega^2 x = f_a \quad (7.45)$$

where $\Omega^2 = \frac{k}{m}$ and $f_a = \frac{F_a}{m}$. In state-space notation we have;

$$\begin{bmatrix} \dot{f} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & k \\ \Omega^2 & 0 \end{bmatrix} \begin{bmatrix} f \\ v \end{bmatrix} + f_a \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7.46)$$

Comparing with Equations 7.32 and 7.33 we get;

$$A = \begin{bmatrix} 0 & k \\ \Omega^2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The reason control is needed in this system is because unnecessarily large forces are being introduced, because of coarse sampling and large time steps. In this example it would be better to introduce this additional force as an acceleration disturbance z , which we will consider to be white noise. We would like to observe the forces at every iteration hence our observation equation is

$$\mathcal{Y} = f + w \quad (7.47)$$

where w observation noise. Hence from above, C and F are;

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Let the optimum covariance matrix be;

$$\hat{P} = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$$

The by using Equation (7.43) the elements of P satisfy;

$$(k + \Omega^2)p_2 - \frac{1}{W}p_1^2 = 0 \quad (7.48)$$

$$\Omega^2 p_1 + \Omega^2 p_3 - \frac{1}{W}p_1 p_2 = 0 \quad (7.49)$$

$$(k + \Omega^2)p_2 - \frac{1}{W}p_2^2 + Z = 0 \quad (7.50)$$

where W and Z are the spectral density (1×1) matrices of the observation noise w and excitation noise z , respectively.

Solving Equation (7.50) we get;

$$p_1 = (k + \Omega^2)W\sqrt{\frac{\gamma}{2}} \quad (7.51)$$

$$p_2 = (k + \Omega^2)^2W\frac{\gamma}{2} \quad (7.52)$$

where;

$$\gamma = 1 + \sqrt{1 + \frac{4Z}{W(k + \Omega^2)^2}}$$

we don't need p_3 to calculate the gain matrix and will avoid the computation. Note that in the above computation it was necessary to ensure that $p_1 \geq 0$ and $p_2 > 0$.

Now we will determine the Kalman gain factor, which is given by Equation (7.40)

$$\hat{K} = (\hat{P}C^T + FX)W^{-1} = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \frac{1}{W} = \begin{bmatrix} (k + \Omega^2)\sqrt{\frac{\gamma}{2}} \\ (k + \Omega^2)^2\frac{\gamma}{2} \end{bmatrix} \quad (7.53)$$

The closed loop filter poles and transfer functions from the observed forces \mathcal{Y} to the estimated state $\hat{\mathcal{X}}_1 = \hat{f}$ and $\hat{\mathcal{X}}_2 = \hat{v}$ are of interest. Assuming input $\mathcal{U} = f_a$ is zero, and then by Equation (7.34),

$$\begin{aligned} s\hat{\mathcal{X}}(s) &= A\hat{\mathcal{X}}(s) + \hat{K}[\mathcal{Y}(s) - C\hat{\mathcal{X}}(s)] \\ &= (A - \hat{K}C)\hat{\mathcal{X}}(s) + \hat{K}\mathcal{Y}(s) \\ \hat{\mathcal{X}}(s) &= (sI - A_0)^{-1}\hat{K}\mathcal{Y}(s) \end{aligned} \quad (7.54)$$

where $A_0 = A - \hat{K}C$ is the closed loop observer dynamics matrix;

$$A_0 = \begin{bmatrix} 0 & k \\ \Omega^2 & 0 \end{bmatrix} - \begin{bmatrix} (k + \Omega^2)\sqrt{\frac{\gamma}{2}} \\ (k + \Omega^2)^2\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} -(k + \Omega^2)\sqrt{\frac{\gamma}{2}} & k \\ \Omega^2 - (k + \Omega^2)^2\frac{\gamma}{2} & 0 \end{bmatrix}$$

and

$$(sI - A_0)^{-1} = \frac{1}{\Delta s} \begin{bmatrix} s & \Omega^2 - (k + \Omega^2)^2\frac{\gamma}{2} \\ k & s - (k + \Omega^2)\sqrt{\frac{\gamma}{2}} \end{bmatrix} \quad (7.55)$$

where Δs is the closed-loop characteristic polynomial

$$\Delta s = s^2 - s(k + \Omega)\sqrt{\frac{\gamma}{2}} - k(\Omega^2 - (k + \Omega^2)^2\frac{\gamma}{2})$$

Combining Equation (7.55) with Equation (7.54) the two filters $\mathcal{H}_1(s)$ and $\mathcal{H}_2(s)$ are obtained as follows;

$$\hat{\mathcal{X}}(s) = \begin{bmatrix} \hat{f}(s) \\ \hat{v}(s) \end{bmatrix} = \hat{K}\mathcal{Y}(s)$$

and

$$\begin{aligned} \mathcal{H}_1(s) &= \frac{\hat{f}(s)}{\mathcal{Y}(s)} && \text{for force} \\ \mathcal{H}_2(s) &= \frac{\hat{v}(s)}{\mathcal{Y}(s)} && \text{for velocity} \end{aligned} \tag{7.56}$$

from the characteristic equations and the above equations the closed-loop poles and the zeros of the filters can also be determined. The excitation noise covariance Z is part of the equation in terms of γ . If Z tends to zero then γ tends to 2 and the filters are simplified into first-order filters. If Z tends to infinity then the transfer function $\mathcal{H}_2(s)$ for velocities behaves like a high pass filter, while $\mathcal{H}_1(s)$ for forces behaves like a low pass filter. For further details on Kalman filter design see Friedland [57].

7.6 Filtering In Forward Simulations

One of the essential feature of the control procedures described in the previous sections is that the filtering is allowed in forward simulation dynamic models (is causal). Convolution filtering methods described in Section 7.3 filter out the error as the signal (input) is given to the system, with a lag time enforced on the system, while Kalman filters estimates the the state by observing the data over some fixed input time and then based on the first order statistics of the that data, introduces control to the system via means of Kalman filter.

For dynamic simulation models of contact and collision response Equation (4.3) the application of control systems as defined and formulated above becomes feasible due the error (noise) that has been incorporated into the system due to restrictions on the size of time steps. As super sampling of time and the mesh of the bodies is not possible the instability of the system is essentially inherent. As it is obvious from Figures 7-4 and 7-6 the disturbances or the collision response are extremely random. Now basing it on the formulation of *Kalman filters*, if these random disturbances could be assumed to be *white noise* then an extremely good filter could be designed, for both linear and non-linear systems. For the dynamic simulations for virtual world modeling, where time of response is essential, the assumption of linear system is an extremely good one.

It should be noted that in all essentiality both the above methods of control are used to filter out the noise using their own random behavior. Hence in other words this method is designed to reduce to noise that system itself introduced into itself for the purpose of dynamic simulations.

Chapter 8

Conclusions

The preceding chapters have presented many diverse and wide-ranging ideas, for the physical interactions of the virtual elements in a physically based virtual world. The desire is to develop and formulate concepts and methods that would give the user a physically based environment to enter and interact with as he would in the real world. The virtual world that is desired therefore, has the goals of physical realism, with controlled complexity. Realism has been defined with respect to response of the system to the user perception of realism, hence, computationally efficient methods leading to real time or close to real time systems have been emphasized

A crucial component of such a virtual world is the physical behavior of interactions of objects. During dynamic simulations, the objects in the virtual world will interact (collide, touch), and if the methods used to impart a physical response are not correct, then the realism of the virtual world will be compromised. However, in a graphic environment, an object is nothing but a collection of lines points and polygons. To develop a system in which a set of lines and polygons would become real (or virtual, in this case) object, we need to simulate material and physical properties, like mass, volume, surface, etc. For this object to be physical, however, and by being “physical” it is meant that it has shape, size, mass, inertia and a non-penetrable surface, an energy or force based description of this object needs to be wrapped around it. In other words, it is desired that a physical domain be mapped on to this graphic domain so that if any other object tries to penetrate this object, a resistance is provided by the object itself. This is the underlying principle that is

presented in this thesis.

It is important to note that the general and well accepted methods of using polyhedra for surface and body descriptions are extremely expensive for mapping of physical domains. A new family of parametric surface representations has been introduced that has proven to be quite efficient for both the purposes of mapping of physical domains and checking for interpenetrations. As shown in Chapter 2 superquadrics can be very efficiently used for contact detection and for mapping of deformations. Since the computations for these contact detections and deformation mappings are much faster than compared to the point- or surface-based polyhedra representations, where most of the effort is employed in solving geometric problems, a major part of the interaction problem is resolved.

The second part of the object interaction problem is characterizing the response to the interaction after it has been detected. In the past, in *virtual world* modeling most of the work done was restricted to rigid bodies, however, many simulations are just not possible without non-rigid deformable objects. This realization has led to a lot of interest in the force-based models of objects derived by finite discretization of the models. Though finite element modeling is not a new concept in the theoretical and application worlds of engineering, it has found new enthusiasts in the graphics and modeling fields. A virtual-work and displacement based formulation of three-dimensional shapes is presented in Chapter 3 and its mapping on to a superquadric is presented in Appendix B. The methods of dynamic simulation of a virtual environments with computational efficiency in mind has been described. The essence of these formulations is that the response of interactions should be based on the material and physical properties of the bodies.

In contact and collision, emphasis has to be paid to the types of contact and then the characteristic transferral of forces between interacting bodies. The methods described in Chapter 4 deal with this problem in various forms of complexity. Procedures to impart on bodies the right forces on collision, depending on the their material and deformation properties are defined. Frictional resistance to motion, a problem that has been avoided to date in virtual world modeling, is introduced and formulations are presented for both translational and rotational frictions. The main emphasis here is to develop a method that is on the same lines as collision response, which characterizes the contact and then applies

tractional resistance.

The problem with accurate considerations of contact and friction is that the contact patch for a contact is not easy to ascertain. Specially in the case of deformable objects, where the contact patch also deforms at contact. For this purpose it is recommended that the effort of developing visually realistic, surface visualization techniques by surface texture mapping be employed. A stochastic estimation approach is presented to determine the roughness of a body with reference to its surface texture and carrying on this estimation to calculate the interpenetration of two or more surfaces, which is then related to the contact area or the stress distribution over that area.

The underlying principle, as mentioned above for virtual world modeling is to create a synthetic physical environment that is physically realistic and of low computational complexity. The desire is to achieve a close-to-real-time, if not a real-time, environment. For this, certain approximations must be made, which can on occasion introduce instability into the system. To remove this instability from the system control theory concepts of observing, filtering and control are advocated. Since the simulation of a virtual worlds is a forward simulation problem, only causal filtering approaches are applicable. Emphasis has been paid on the use of adaptive filtering techniques, because of its ability to impart optimal control to achieve the desired behavior, by removing from the system the error that was introduced by approximations. Kalman filtering is introduced as an optimal filter because of its features of imparting to impart adaptive forward control by using a random noise input (white noise), which is inherent to the system.

For an overall perspective, this thesis presents ideas and concepts for physically based virtual world modeling based on concepts picked up from a wide range of disciplines. The emphasis, however, is on computational efficiency with realism. The decisions regarding approximations versus loss of realism are user defined and the methods of varying computational complexity are presented.

Appendix A

Deformations and Modal Values

A.1 Implicit Function with Deformation Mapping for Modeling Primitives

A major drawback of any point- or surface-based representation is the expense of solving geometric problems such as a check for a point in space with respect to the surface. This problem can be much improved by using an implicit function $f(x, y, z) = d$ to describe object geometry, as for such representations the computational complexity of these computations is $O(r)$, a significant improvement over that of $O(nr)$ where r is the number of points to check for and n is the points for surface representation.

The modal representation of shape can be mapped onto the implicit function shape primitives by first describing each mode by an appropriate polynomial function, and then using global deformation mapping to warp the shape primitive into the appropriate form. The polynomial deformation mappings that correspond to each of the modes are ascertained by a linear regression of a polynomial with m terms in appropriate powers of x , y , and z , against the n triples of x , y and z that compose a vector ϕ_i ($3n \times 1$) for the i th node. Φ then forms a matrix ($3n \times m$) containing the m , ϕ_i vectors. The regression is;

$$\alpha = (\beta^T \beta)^{-1} \beta^T \phi_i \quad (\text{A.1})$$

where α is an ($m \times 1$) matrix of the coefficients of the desired deformation polynomial, β is

Values of the Modal Deformation Matrix	
d_{00}	$= m_6 + ym_{12} + zm_{15} - (m_{13} + m_{16})sgn(x) - m_{14} - m_{17}$
d_{01}	$= m_{11} + 2y(m_{13} + sgn(x)m_{14})$
d_{02}	$= m_{10} + 2z(m_{16} + sgn(x)m_{17})$
d_{10}	$= m_{11} + 2x(m_{19} + sgn(y)m_{20})$
d_{11}	$= m_7 + xm_{18} + zm_{21} - (m_{19} + m_{22})sgn(y) - m_{20} - m_{23}$
d_{12}	$= m_9 + 2z(m_{22} + sgn(y)m_{23})$
d_{20}	$= m_{10} + 2x(m_{25} + sgn(z)m_{26})$
d_{21}	$= m_9 + 2y(m_{28} + sgn(z)m_{29})$
d_{22}	$= m_8 + xm_{24} + ym_{27} - (m_{25} + m_{28})sgn(z) - m_{26} - m_{29}$
where;	
$\mathbf{X} = [x, y, z]^T = \text{Undeformed Point}$	

Table A.1: Modal Deformation Matrix Values in terms of Modal Amplitudes

a $(3n \times m)$ matrix whose first column contains the elements of

$$u = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_m, y_m, z_m)$$

and whose remaining columns consist of the modified versions of u where the x , y , and/or z components have been raised to the various powers to achieve higher order modes (See Pentland [6] for more details).

By linearly superimposing the various deformation mappings one can obtain an accurate accounting of the object's non-rigid deformation. In the *Thingworld Modeling System* [6, 20, 21] the set of polynomial deformations is combined into a (3×3) matrix of polynomials, \mathbf{D} , that is referred to as the *modal deformation matrix*. This matrix transforms pre-deformation point positions \mathbf{X} into the deformed coordinates \mathbf{X}_d

$$\mathbf{X}_d = \begin{bmatrix} d_{00} & d_{01} & d_{02} \\ d_{10} & d_{11} & d_{12} \\ d_{20} & d_{21} & d_{22} \end{bmatrix} \mathbf{X} \quad (\text{A.2})$$

Because low-order modes change slowly as a function of object shape, the matrix can be used for a very wide range of shapes, and thus may be precomputed.

Modal Amplitudes					
m_0	x-translation	m_1	y-translation	m_2	z-translation
m_3	x-rotation	m_4	y-rotation	m_5	z-rotation
m_6	x-size	m_7	y-size	m_8	z-size
m_9	x-shear	m_{10}	y-shear	m_{11}	z-shear
m_{12}	xy-taper	m_{13}	xy-bend	m_{14}	xy-pinch
m_{15}	xz-taper	m_{16}	xz-bend	m_{17}	xz-pinch
m_{18}	yx-taper	m_{19}	yx-bend	m_{20}	yx-pinch
m_{21}	yz-taper	m_{22}	yz-bend	m_{23}	yz-pinch
m_{24}	zx-taper	m_{25}	zx-bend	m_{26}	zx-pinch
m_{27}	zy-taper	m_{28}	zy-bend	m_{29}	zy-pinch

Table A.2: Physical Basis of Modal Amplitudes

Typical entries of \mathbf{D} are in Table A.1 where \mathbf{m} represents a $(m \times 1)$ vector of

$$m_1, m_2, \dots, m_m$$

which are the *modal amplitude* of the objects and describe the amplitudes of the various modes of vibrations.

The modal amplitudes m_i have intuitive meanings and are shown in Table A.2. In $m_0 - m_5$ are the rigid body modes of translation and rotation, $m_6 - m_8$ are the x, y, and z sizes, $m_9 - m_{11}$ are shears about the x, y, and z axes and the rest are bends, tapers and pinches in various axes. Note that because the rigid body modes are calculated in the object's coordinate system, they must be rotated to global coordinates before being integrated with the remainder of any dynamical simulation system.

A.2 From Deformed Space to Undeformed Space

The modal deformation matrix as shown above in Table A.1 is a function of undeformed space \mathbf{X} . This is because the deformations are a linear combinations and superpositions of different modes of vibrations for a canonical undeformed superquad which are mapped onto a undeformed superquad to get a deformed model.

In this case even though the deformation matrix is invertible, to change from deformed

space to undeformed space requires an iterative process of solving a (3 x 3) system of quadratic polynomials. Newton-Raphson and Secant Methods are extremely good for convergence, except that for Newton-Raphson it is required to calculate the inverse of the Jacobian of the 3x3 system at every iteration, which is extremely expensive, however the convergence is faster. This method is described below;

We are trying to calculate \mathbf{X} from \mathbf{X}_d , hence from Equation (A.2) we have $\mathbf{X}_d = \mathbf{D}\mathbf{X}$. As explained above \mathbf{D} is itself a function of \mathbf{X} , hence inverting \mathbf{D} does not solve the problem. Essentially we have a system of three polynomials;

$$\begin{aligned}x_d &= g_1(x, y, z) \\y_d &= g_2(x, y, z) \\z_d &= g_3(x, y, z)\end{aligned}$$

or since x_d , y_d and z_d are already known and are constants;

$$\begin{aligned}G_1(x, y, z) &= 0 \\G_2(x, y, z) &= 0 \\G_3(x, y, z) &= 0\end{aligned}$$

Using Newton's Method [32];

$$\mathbf{J}^k(\mathbf{X}^{k+1} - \mathbf{X}^k) + \mathbf{G}^k = \mathbf{0} \quad (\text{A.3})$$

where \mathbf{J} ($n \times n$) is called the *Jacobian matrix*¹ and it contains the first derivatives of $\frac{\delta G_i}{\delta x_j}$ of the n ($n = 3$ in our case) functions G_i with respect to the n variables x_j . The multiplication $\mathbf{J}^k(\mathbf{X}^{k+1} - \mathbf{X}^k)$ imparts a first order correction to \mathbf{G} . k represents iteration. The resulting \mathbf{X}^{k+1} after every iteration has to be checked for the error threshold. For *Modified Newton* method, *Quasi-Newton* method and method of *steepest descent* see Strang [32].

¹The definition of this Jacobian Matrix is the same as the one presented in Appendix B, application a little different

Appendix B

Finite Element Method Formulations

In this Appendix a method will be shown to develop the mass \mathbf{M} , damping \mathbf{C} and stiffness \mathbf{K} matrices of three dimensional body. It will be assumed that the bodies are *isoparametric*. Examples of one-dimensional bodies are given to give a more intuitive idea of this method.

The equations for calculations of mass \mathbf{M} , damping \mathbf{C} and stiffness \mathbf{K} matrices from Chapter 3 are;

$$\mathbf{M} = \sum_{i=1}^m \int_{V^{(i)}} \mathbf{H}^{(i)T} \rho^{(i)} \mathbf{H}^{(i)} dV^{(i)} \quad (\text{B.1})$$

$$\mathbf{C} = \sum_{i=1}^m \int_{V^{(i)}} \mathbf{H}^{(i)T} \kappa^{(i)} \mathbf{H}^{(i)} dV^{(i)} \quad (\text{B.2})$$

$$\mathbf{K} = \sum_{i=1}^m \int_{V^{(i)}} \mathbf{B}^{(i)T} \mathbf{E}^{(i)} \mathbf{B}^{(i)} dV^{(i)} \quad (\text{B.3})$$

where

- H** = The displacement interpolation matrix
- B** = The strain displacement matrix
- U** = The three dimensional displacement vector.
- E** = Elasticity (stress-strain) matrix.
- i* = The element number
- m* = The number of elements in the assemblage
- κ = Damping Constant
- ρ = Density

and the integrations are over the whole volume V of the element. First we need to determine the **H** and **B** matrices.

B.1 Setting up H and B matrices

To understand how these matrices could be set up lets study a very simple example.

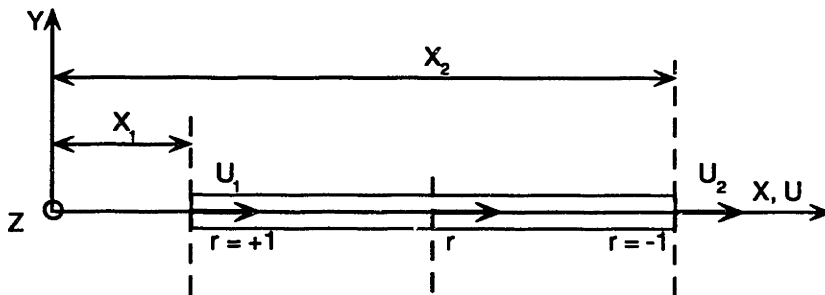


Figure B-1: A simple bar element

Example B.1 A one dimensional bar element In the figure of a beam element shown in Figure B-1 a new form of local coordinates is introduced, which sets up a system that is easy to integrate. The local coordinates vary from -1.0 to 1.0 in 'r' in the direction of X.

Transformation from the local r-coordinate system to the global X-coordinate system is given by

$$X = \frac{1}{2}(1 - r)X_1 + \frac{1}{2}(1 + r)X_2$$

Now let;

$$h_1 = \frac{1}{2}(1 - r)$$

$$h_2 = \frac{1}{2}(1 + r)$$

therefore;

$$X = h_1 X_1 + h_2 X_2$$

$$X = \sum_m h_i X_i$$

Similarly, as we are dealing with an isoparametric element, the displacements are;

$$U = h_1 U_1 + h_2 U_2$$

Using Equation (3.2) we see that h_1 and h_2 are simply the interpolation functions for node 1 and 2 respectively, hence;

$$\mathbf{H} = \left[\frac{1}{2}(1-r) \quad \frac{1}{2}(1+r) \right]$$

Substituting into Equations (B.1) and (B.2);

$$\mathbf{M} = \int_L \left[\begin{array}{c} -\frac{1}{2}(1-r) \\ \frac{1}{2}(1+r) \end{array} \right] \rho \left[-\frac{1}{2}(1-r) \quad \frac{1}{2}(1+r) \right] A dX$$

and;

$$\mathbf{C} = \int_L \left[\begin{array}{c} -\frac{1}{2}(1-r) \\ \frac{1}{2}(1+r) \end{array} \right] \kappa \left[-\frac{1}{2}(1-r) \quad \frac{1}{2}(1+r) \right] A dX$$

where A is the cross-sectional area of the beam. For \mathbf{K} we need to determine \mathbf{B} , For strains we have;

$$\epsilon = \frac{dU}{dX}$$

$$\epsilon = \frac{dU}{dr} \frac{dr}{dX}$$

from Figure B-1;

$$\frac{dU}{dr} = \frac{U_2 - U_1}{2}$$

$$\frac{dX}{dr} = \frac{X_2 - X_1}{2} = \frac{L}{2} = \mathbf{J}$$

The above calculation is for a Jacobian \mathbf{J} as we will see later. Now from these expressions we can set up equations for ϵ and from Equation (3.3) consequently for \mathbf{B} ;

$$\epsilon = \frac{U_2}{L} - \frac{U_1}{L}$$

$$\mathbf{B} = \left[-\frac{1}{L} \quad \frac{1}{L} \right]$$

Substituting into Equation (B.3) for $m = 1$ and for one-dimensional bar $bfE = E$, where E is Young's modulus element in this case;

$$\mathbf{K} = \int_L \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix} E \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix} AdX$$

One of the reasons of defining a new coordinate system of r was the ease of integration. If in the above equation of \mathbf{M} , \mathbf{C} and \mathbf{K} we can replace dX by $\mathbf{J}dr$ as calculated above; then these equations become much easy to integrate;

$$\mathbf{M} = \int_{-1}^1 \begin{bmatrix} -\frac{1}{2}(1-r) \\ \frac{1}{2}(1+r) \end{bmatrix} \rho \begin{bmatrix} -\frac{1}{2}(1-r) & \frac{1}{2}(1+r) \end{bmatrix} \mathbf{A} \mathbf{J} dr$$

$$\mathbf{C} = \int_{-1}^1 \begin{bmatrix} -\frac{1}{2}(1-r) \\ \frac{1}{2}(1+r) \end{bmatrix} \kappa \begin{bmatrix} -\frac{1}{2}(1-r) & \frac{1}{2}(1+r) \end{bmatrix} \mathbf{A} \mathbf{J} dr$$

$$\mathbf{K} = \int_{-1}^1 \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix} E \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \mathbf{A} \mathbf{J} dr$$

where;

$$J = \frac{L}{2}$$

hence;

$$\mathbf{M} = \frac{2AL\rho}{3} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{C} = \frac{2AL\kappa}{3} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{K} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

From Example B.1 above we can develop generalized equations for coordinate interpolations.

$$\begin{aligned} x &= \sum_{i=1}^q h_i x_i \\ y &= \sum_{i=1}^q h_i y_i \end{aligned} \tag{B.4}$$

$$z = \sum_{i=1}^q h_i z_i$$

Where x , y , and z are the coordinates at any point on the element (local coordinates) and $x_i, y_i, z_i, i = 1, \dots, q$, are the coordinates of q element nodes. The interpolation functions h_i are defined in the natural coordinate system of the element, which has variables r , s , and t , each varying from -1 to +1.

Considering the geometry of two- and three-dimensional elements (see Figures B-3, B-4), note that by coordinate interpolation as per Equation (B.5), the elements can have curved boundaries. This gives it an added advantage over the generalized coordinate finite element formulation. Another advantage is the ease with which the element displacement functions can be constructed. In an *isoparametric* formulation, the element displacements are interpolated in the same way as the coordinate.

$$\begin{aligned} u &= \sum_{i=1}^q h_i u_i \\ v &= \sum_{i=1}^q h_i v_i \\ w &= \sum_{i=1}^q h_i w_i \end{aligned} \tag{B.5}$$

Note: In this formulation we are dealing with ordinary polynomial interpolants. For many cases it would be preferable to use Hermetian, Gaussian or Chebyshev's polynomials. [18] forms

B.1.1 To set up Interpolation functions and the Interpolation matrix

The two major postulates to set up the interpolation functions are :

1. h_i equals 1 at x_i and equals 0 at every other node.
2. Sum of interpolation functions is always equal to 1.0 ($\sum h_i = 1.0$)

Using the above mentioned postulates, the interpolation functions h_i , corresponding to a specific nodal point layout could be determined in a systematic manner. However, it is

convenient to set it up by inspection as shown below

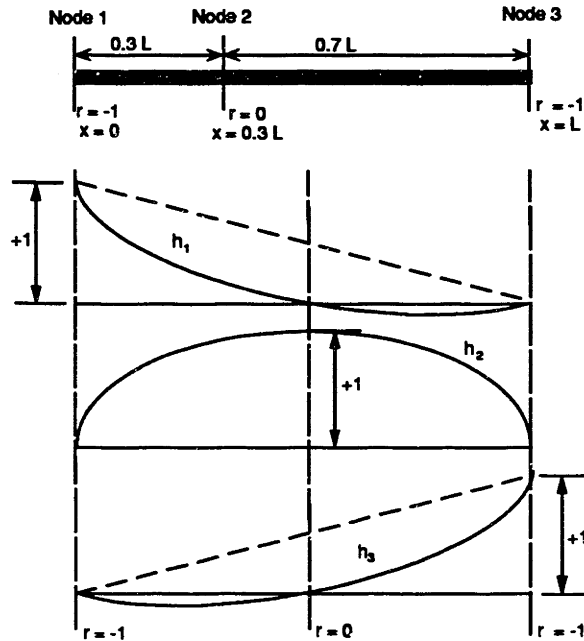


Figure B-2: One Dimensional Element

Example B.2 Interpolation functions for a 3 node one-dimensional element The Figure B-2 shows a simple one-dimensional element with three nodes. This figure also shows the displacement profiles for each node based on the conditions just mentioned above. If this was a two node element our interpolation functions would have been the same as in Example B.1. Notice how the local r -coordinates are set up, with the variation from -1 to $+1$. The displacement curves for unit displacements at nodes 1 and 2 show the interpolation functions as in the previous example. Condition 1 is satisfied by both if there was no third node. With the third node the interpolation functions become quadratic and half of the third interpolation function has to be subtracted from the first two so that the above conditions could be satisfied. The three interpolation functions are;

$$\begin{aligned} h_1 &= \frac{1}{2}(1-r) - \frac{1}{2}(1-r^2) \\ h_2 &= \frac{1}{2}(1+r) - \frac{1}{2}(1-r^2) \\ h_3 &= (1-r^2) \end{aligned}$$

From this we can set up an interpolation matrix for this three node one dimensional model.

$$\mathbf{H} = \left[\begin{array}{ccc} \frac{1}{2}(1-r) - \frac{1}{2}(1-r^2) & \frac{1}{2}(1+r) - \frac{1}{2}(1-r^2) & (1-r^2) \end{array} \right]$$

		$i = 5$	$i = 6$	$i = 7$	$i = 8$	$i = 9$
$h_1 =$	$\frac{1}{4}(1+r)(1+s)$	$-\frac{1}{2}h_5$	\dots	\dots	$-\frac{1}{2}h_8$	$-\frac{1}{4}h_9$
$h_2 =$	$\frac{1}{4}(1-r)(1+s)$	$-\frac{1}{2}h_5$	$-\frac{1}{2}h_6$	\dots	\dots	$-\frac{1}{4}h_9$
$h_3 =$	$\frac{1}{4}(1-r)(1-s)$	\dots	$-\frac{1}{2}h_6$	$-\frac{1}{2}h_7$	\dots	$-\frac{1}{4}h_9$
$h_4 =$	$\frac{1}{4}(1+r)(1-s)$	\dots	\dots	$-\frac{1}{2}h_7$	$-\frac{1}{2}h_8$	$-\frac{1}{4}h_9$
$h_5 =$	$\frac{1}{2}(1-r^2)(1+s)$	\dots	\dots	\dots	\dots	$-\frac{1}{2}h_9$
$h_6 =$	$\frac{1}{2}(1-s^2)(1-r)$	\dots	\dots	\dots	\dots	$-\frac{1}{2}h_9$
$h_7 =$	$\frac{1}{2}(1-r^2)(1-s)$	\dots	\dots	\dots	\dots	$-\frac{1}{2}h_9$
$h_8 =$	$\frac{1}{2}(1-s^2)(1+r)$	\dots	\dots	\dots	\dots	$-\frac{1}{2}h_9$
$h_9 =$	$(1-r^2)(1-s^2)$	\dots	\dots	\dots	\dots	\dots

Table B.1: The interpolation functions of four to nine variable number nodes for two dimensional elements

Using the the derivatives of interpolation functions and their linear combinations gives us the strain-displacement matrix, we'll however postpone the setting up of strain-displacement matrix until our discussion of Jacobians.

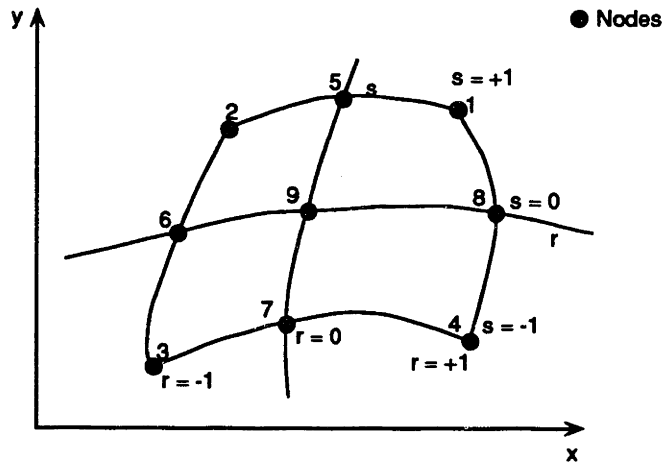


Figure B-3: Two Dimensional Element (Nine nodes shown)

In a manner similar to the one dimensional model, the interpolation functions for two and three dimensional models can also be calculated. Figure B-3 shows a two dimensional model

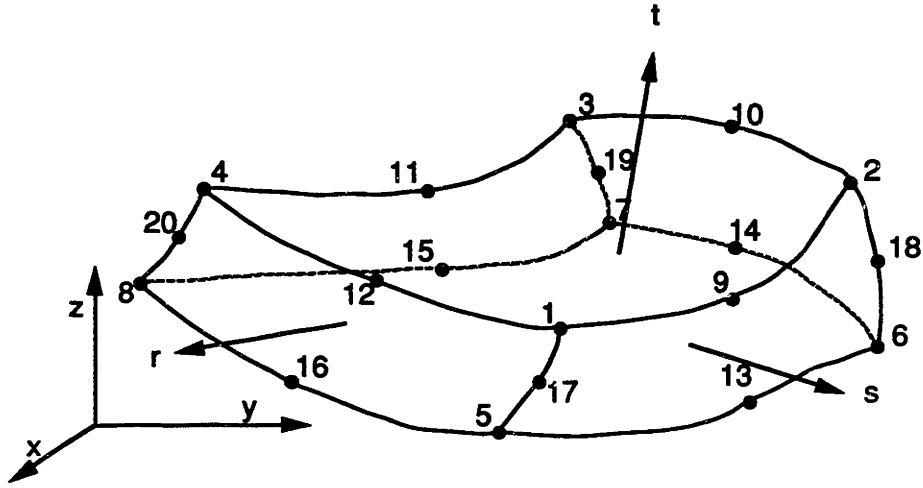


Figure B-4: Three Dimensional Element (Twenty nodes shown)

with its r and s local coordinates. The interpolation functions of four to nine nodes for two dimensional elements are given in Table B.1 Figure B-4 shows a three dimensional model with 20 nodes. The interpolation functions of eight to twenty nodes for three dimensional elements are shown in Table B.2. The three dimensional 20 node element is the one used for determining the mass M , damping C and stiffness K matrices of superquads.

B.1.2 Jacobians

To be able to evaluate the stiffness matrix of an element (using Equation (B.3)), we need to calculate the strain-displacement transformation matrix B . The element strains are obtained in terms of derivatives of element displacements with respect to the local coordinates.

$$\epsilon^T = \begin{bmatrix} \epsilon_{xx} & \epsilon_{yy} & \epsilon_{zz} & \gamma_{xy} & \gamma_{xz} & \gamma_{yz} \end{bmatrix}$$

or;

$$\epsilon^T = \begin{bmatrix} \frac{du}{dx} & \frac{dv}{dy} & \frac{dw}{dz} & \frac{du}{dy} + \frac{dv}{dx} & \frac{du}{dz} + \frac{dw}{dx} & \frac{dv}{dz} + \frac{dw}{dy} \end{bmatrix} \quad (B.6)$$

Since the strain displacements are related to x , y and z , we need to use the chain rule

$h_1 =$	$g_1 - (g_9 + g_{12} + g_{17})/2$
$h_2 =$	$g_2 - (g_9 + g_{10} + g_{18})/2$
$h_3 =$	$g_3 - (g_{10} + g_{11} + g_{19})/2$
$h_4 =$	$g_4 - (g_{11} + g_{12} + g_{20})/2$
$h_5 =$	$g_5 - (g_{13} + g_{16} + g_{17})/2$
$h_6 =$	$g_6 - (g_{13} + g_{14} + g_{18})/2$
$h_7 =$	$g_7 - (g_{14} + g_{15} + g_{19})/2$
$h_8 =$	$g_8 - (g_{15} + g_{16} + g_{20})/2$
$h_j =$	g_j for $j = 9, \dots, 20$
Using:	
$g_j =$	0 if node j is not included, otherwise,
$g_j =$	$G(r, r_j)G(s, s_j)G(t, t_j)$
where:	
$G(\beta, \beta_j) =$	$\frac{1}{2}(1 + \beta_j \beta)$ for $\beta_j = \pm 1$
$G(\beta, \beta_j) =$	$(1 - \beta^2)$ for $\beta_j = 0$

Table B.2: The interpolation functions of eight to twenty variable number nodes for three dimensional elements

to change $\frac{du}{dr}$ to $\frac{du}{dx}$;

$$\frac{d}{dx} = \frac{d}{dr} \frac{dr}{dx} + \frac{d}{ds} \frac{ds}{dx} + \frac{d}{dt} \frac{dt}{dx} \quad (\text{B.7})$$

Using Similar relationships for $\frac{d}{dy}$, and $\frac{d}{dz}$.

$$\begin{bmatrix} \frac{d}{dr} \\ \frac{d}{ds} \\ \frac{d}{dt} \end{bmatrix} = \begin{bmatrix} \frac{dx}{dr} & \frac{dy}{dr} & \frac{dz}{dr} \\ \frac{dx}{ds} & \frac{dy}{ds} & \frac{dz}{ds} \\ \frac{dx}{dt} & \frac{dy}{dt} & \frac{dz}{dt} \end{bmatrix} \begin{bmatrix} \frac{d}{dx} \\ \frac{d}{dy} \\ \frac{d}{dz} \end{bmatrix} \quad (\text{B.8})$$

or in matrix notation,

$$\frac{d}{dr} = \mathbf{J} \frac{d}{dx} \quad (\text{B.9})$$

where \mathbf{J} is the Jacobian operator relating the natural coordinate derivatives to the local coordinate derivatives. Note that using Equation (B.5) one can quite easily calculate the Jacobian operator. Since we require $\frac{d}{dx}$;

$$\frac{d}{dx} = \mathbf{J}^{-1} \frac{d}{dr} \quad (\text{B.10})$$

Hence by using equations (B.5) and (B.10) we can evaluate $\frac{du}{dx}, \frac{du}{dy}, \dots, \frac{dw}{dz}$ and construct the strain-displacement transformation matrix \mathbf{B} by using;

$$\epsilon^{(m)}(x, y, z) = \mathbf{B}^{(m)}(x, y, z)\mathbf{U} \quad (\text{B.11})$$

On converting to local element degrees of freedom, in Equations (B.3), (3.9), (3.10), (B.1) (B.2) have

$$dV = \det[\mathbf{J}]drdsdt \quad (\text{B.12})$$

B.2 Numerical Integration

An explicit integration of the volume integral in the above mentioned equations is not very effective, specially when higher order interpolation is used or when the elements are very distorted. For this reason numerical integrations should be preferred. All of the integrals for our formulation are of the forms shown below. These equations can be numerically integrated by the equations shown below.

$$\begin{aligned} \int \mathbf{F}(r)dr &= \sum_i \alpha_i \mathbf{F}(r_i) \\ \int \mathbf{F}(r, s)dr &= \sum_{i,j} \alpha_{ij} \mathbf{F}(r_i, s_j) \\ \int \mathbf{F}(r, s, t)dr &= \sum_{i,j,k} \alpha_{ijk} \mathbf{F}(r_i, s_j, t_k) \end{aligned} \quad (\text{B.13})$$

This integration can be carried out by either of the two following methods

- Newton-Cotes Formulas
- Gaussian Quadrature Formulas

In Newton-Cotes Formulas (which includes Trapezoidal and Simpson's Rules) we need $n + 1$ points to integrate exactly a polynomial of degree n . While in Gauss Formula one requires n points to integrate a polynomial of order $(2n - 1)$. It is for

this reason that Gauss method is preferred over the Newton-Cotes method. The simple formula for Gauss Method in three-dimensional integration is,

$$\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{F}(r, s, t) dr ds dt = \sum_{i,j,k} \alpha_i \alpha_j \alpha_k \mathbf{F}(r_i, s_j, t_k) \quad (\text{B.14})$$

For a more detailed outlook on numerical techniques for integration see Strang [32, 33], Bathe [18] or any other book in text in linear algebra and numerical methods.

B.3 Implementation with Superquad

To do any dynamic simulations using deformable superquads, it is essential to set up a FEM domain for the superquads so that the geometric and shape characteristics of the superquads could be used for mesh generation and hence for force distributions internal to the body. Since superquads are three-dimensional geometric shapes we can use an 8 node to as many nodes as desired, a purely “engineering” decision based on the requirements. Assuming the body to be isoparametric the mass \mathbf{M} , damping \mathbf{C} and stiffness \mathbf{K} and the load matrices of a superquad could be set up using the following step by step procedure;

1. Select the number of nodes n required to describe the body, number them, keeping in mind the minimum bandwidth requirements.
2. Using Table B.2 define the interpolant functions h_1, h_2, \dots, h_n and set up the displacement interpolation matrix \mathbf{H} . Check for the postulates as described in Section B.1.1.
3. Determine the Jacobian matrix, \mathbf{J} . In the case of superquads there are two forms of deformations, one is the deformation due to the squareness parameters, e_1 and e_2 in the equation

$$\mathbf{X}(\eta, \omega) = \begin{bmatrix} \cos^{e_1}(\eta) \cos^{e_2}(\omega) \\ \cos^{e_1}(\eta) \sin^{e_2}(\omega) \\ \sin^{e_1}(\eta) \end{bmatrix} \quad \begin{matrix} -\pi \leq \omega \leq \pi \\ -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \end{matrix} \quad (\text{B.15})$$

and other due to the deformation mapping of the various modes of vibration as described in Appendix A. Hence,

$$\mathbf{J} = \mathbf{D}_s \mathbf{D} \quad (\text{B.16})$$

where;

$$\mathbf{D}_s = \begin{bmatrix} \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) & 0 & 0 \\ 0 & \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) & 0 \\ 0 & 0 & \sin^{\epsilon_1}(\eta) \end{bmatrix} \quad (\text{B.17})$$

and \mathbf{D} is given by Equation (A.2) and the values are as in Table A.1. Now by setting up ϵ as in Equation (B.6) by using the Jacobian matrix \mathbf{J} and the differentials of the shape interpolation functions \mathbf{B} could be set up by using Equations (B.10) and (B.11).

4. The Elasticity matrix \mathbf{E} for a three-dimensional isoparametric element is;

$$\mathbf{E} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad (\text{B.18})$$

where

$$\begin{aligned} E &= \text{Young's modulus} \\ \nu &= \text{Poisson's ratio} \end{aligned}$$

5. Replace all of the above determined matrices and other material properties of the superquad into Equations (B.1), (B.2) and (B.3) and integrate using the Gaussian Quadrature methods described in the previous section, to get the mass \mathbf{M} , damping \mathbf{C} and stiffness \mathbf{K} matrices of a superquad.

Appendix C

Stiffness Matrices and Stiffness values

In this Appendix approximate methods of determining the stiffness values at different points on a body with a known stiffness matrix are given;

C.1 Stiffness values at a node

In this section, an approximate method for determining the stiffness (k) value of a node from the stiffness matrix (\mathbf{K}) of the body will be described. Before trying to approximate the stiffness value of a point from a stiffness matrix, it is important to understand the physical significance of the stiffness matrix.

For a spring the force is defined as

$$F = kx$$

where k is the stiffness value of the spring. In case of a 3-D deformable object the the same analogy follows, specially since a deformable object on micro level is a combination of many smaller springs

$$\mathbf{F} = \mathbf{K}\mathbf{U} \tag{C.1}$$

where \mathbf{F} is the force vector, \mathbf{K} is the stiffness matrix and \mathbf{U} is the displacement vector.

Explicitly for an ($n \times n$) system we have;

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_j \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} k_{1,1} & k_{1,2} & \dots & k_{1,j} & \dots & k_{1,n} \\ k_{2,1} & k_{2,2} & \dots & k_{2,j} & \dots & k_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{j,1} & k_{j,2} & \dots & k_{j,j} & \dots & k_{j,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{n,1} & k_{n,2} & \dots & k_{n,j} & \dots & k_{n,n} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_j \\ \vdots \\ u_n \end{bmatrix} \quad (\text{C.2})$$

Now to develop an intuitive idea of the stiffness matrix \mathbf{K} consider that only one point (point j) on the whole body has deformed by a unit displacement and all the other nodes remain fixed (no displacement). Now from Equation (C.2) we can calculate the forces at all the nodes as we already know the stiffness matrix. Multiplying the stiffness matrix with a ($n \times 1$) vector where all the values are zero except $u_j = 1.0$. This multiplication returns the j th column of the matrix. Hence from this we can conclude that the j th row or a column of a stiffness matrix (linear materials only as the stiffness matrix is not symmetric for non-linear material) represents the forces on the nodes of the object due to a unit displacement of the j th node.

A stiffness matrix is always has a heavy diagonal, and one good assumption of the stiffness value of a node would be the diagonal element for that node in the stiffness matrix. This assumption is acceptable to a certain extent, however some kind of a weighted sum of the values of the stiffness matrix for the column of the node would be a little better.

If the bandwidth considerations are taken into account then the approximation attempted by taking the diagonal elements is very good. However, the chances of error are higher, if the diagonal is not heavy and the stiffness matrix is full. For mass and damping matrices, since monotonic convergence is observed for diagonal matrices, therefore the approximations of diagonal matrices is pretty efficient. In case of modal superposition methods, where the equations are completely uncoupled, the method discussed here is also acceptable.

Note however, that this is an approximate method and should be used for problems that do not desire extreme accuracy or as initial guesses for more accurate methods.

C.2 Stiffness values at any point inside segment

In virtual world environments, our interest is in employing physically realistic models and at the same time adopt efficient and approximating algorithms. For time efficiency one of the concerns is sampling of graphic objects. In this section, an approximate method of distributing the stiffness values of the nodes to the segment defined by the nodes is presented. Note that since each node has a stiffness value in the three directions, therefore the problem of determining the stiffness of a point in a segment is essentially a three-fold process, in which each direction is calculated by its corresponding planes.

We will present a general method for any polygynous shape. Figure C-1 shows a 5 node

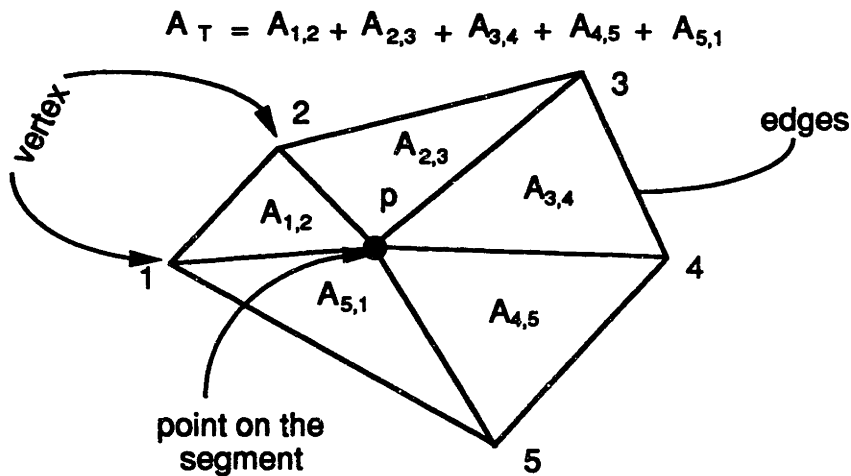


Figure C-1: Distribution of Stiffness from nodes to a point on the segment.

segment, and the point of concern is p .

$$\begin{aligned} \text{Point } i &\iff (x_i, y_i, z_i) \\ \text{Point } j &\iff (x_j, y_j, z_j) \\ \text{Point } p &\iff (x_p, y_p, z_p) \end{aligned}$$

Assuming that we are trying to calculate the stiffness in z -direction, we will determine the areas in xy -plane only.

The area of a triangle with known coordinates can be calculated by;

$$A(\Delta) = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \quad (\text{C.3})$$

The segment or the polygon under consideration can be broken up into triangles with the point p and the edges of the segment (see Figure C-1). The area of these triangles should be calculated using Equation (C.3). These areas are $A_{i,j}$ where i and j are the node numbers. The ratio of the each area with respect to the total area has to be calculated so the weightage can be established. Hence for n nodes per segment ($n = 3$ for a triangle)

$$A_T = \sum_{q=0}^n A_{i,j} \quad i \neq j$$

therefore the ratios are;

$$r_{i,j} = \frac{A_{i,j}}{A_T} \quad (\text{C.4})$$

and if the stiffness of a node q is k_q , then the stiffness of the point inside the segment is given by;

$$k_p = \sum_{q=0}^n \frac{r_{i,q} + r_{q,j}}{2} k_q \quad i \neq j \quad (\text{C.5})$$

This method has to be repeated in all the three planes to pick up the right k_x , k_y and k_z for each point.

Appendix D

Random Variables : Definitions

D.1 Random Variables and Random Processes

Here some basic concepts of random variables are defined which are be used in the formulation of roughness of surfaces: ¹

Definition D.1 *Consider a sample space denoted by S . A real valued function that is defined in space S is called a random variable.*

A random variable X is a function that assigns a real number $X(s)$ to each possible outcome $s \in S$ [55]. Random variables are usually denoted by $\langle X \rangle$. This is the notation that will be adopted in this chapter.

Definition D.2 *For any point in the subspace A to S (i.e. $A \in S$) the probability $P(x \leq A)$ of x is dependent on x and is called the cumulative distribution function of the RV x .*

$$F_x(x) = P\{x \leq A\}$$

¹For more details of random variable and random processes and the properties of distribution and density functions see [56, 55]

Definition D.3 The derivative of the distribution function is called the density function (also known as the frequency function) of the RV x is;

$$f(x) = \frac{dF(x)}{dx}$$

Now a definition for the sum of two random variables

Definition D.4 The sum of two random variables x and y , is $z = x + y$. The probability distribution of the function of two random variables, z is given by;

$$F(z) = \int_{-\infty}^{\infty} \int_{-\infty}^{z-y} f(x,y) dx dy$$

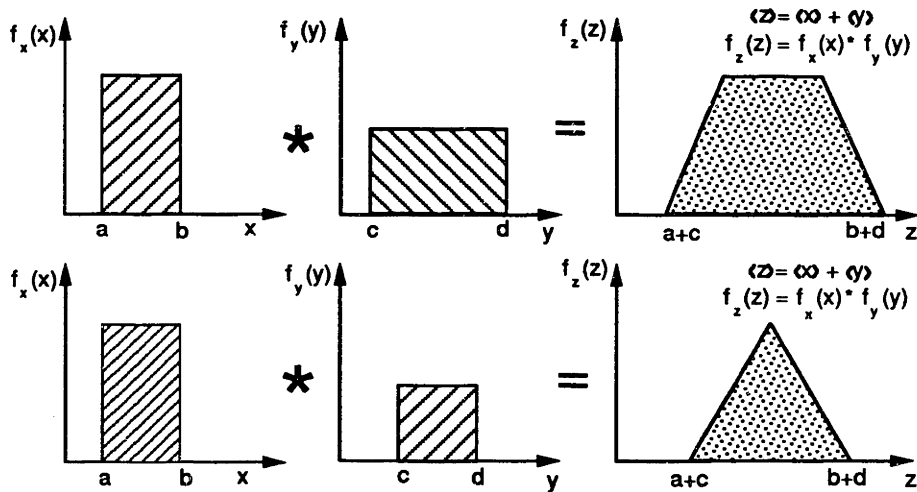


Figure D-1: Convolutions of two random variables

The density of z could be found by differentiating as per Definition D.3 or by direct methods which gives [56]

$$f_z(z) dz = \int_{-\infty}^{\infty} f(z-y, y) dy dz \tag{D.1}$$

if the RVs x and y are independent then

$$f(x, y) = f_x(x) f_y(y)$$

which modifies Equation (D.1) to;

$$f_z(z)dz = \int_{-\infty}^{\infty} f_x(z-y)f_y(y)dydz \quad (\text{D.2})$$

the above integral is the *convolution* of the functions $f_x(x)$ and $f_y(y)$, and it can be concluded that if the *RVs* are independent then the density of their sum equals the convolution of their densities.

D.2 Statistical Characteristics of Random Processes

There is no practical way of obtaining probability density functions for a physical process, hence their main use is for mathematical development, like the following *first-order* and *second-order statistics*;

Definition D.5 *Mean*;

$$\mu(t) = E \{x(t)\} = \int_{-\infty}^{\infty} x f(x) dx \quad (\text{D.3})$$

Definition D.6 *Mean Square*;

$$E \{x^2(t)\} = \int_{-\infty}^{\infty} x^2 f(x) dx \quad (\text{D.4})$$

Definition D.7 *Variance*;

$$\sigma^2(t) = E \{[x(t) - \mu(t)]^2\} = \int_{-\infty}^{\infty} [x - \mu]^2 f(x) dx \quad (\text{D.5})$$

Definition D.8 *Correlation Function*;

$$\rho(t, \tau) = E \{[x(t), x(\tau)]\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f(x_1, x_2) dx_1 dx_2 \quad (\text{D.6})$$

The first three definitions of the statistical parameters are *first-order*, since they entail only the first order probability density function. The correlation function, on the other hand is a *second-order* statistic. The symbol $E\{ \}$ denotes mathematical expectation, or the average computed by probability.

The *first-order* statistics can be defined for vector process. The generalized *correlation function* is known as *correlation matrix*;

$$R(t, \tau) = E \{ [x(t), x^T(\tau)] \} = \begin{bmatrix} E \{ [x_1(t), x_1(\tau)] \} & \cdots & E \{ [x_1(t), x_n(\tau)] \} \\ \cdots & \cdots & \cdots \\ E \{ [x_n(t), x_1(\tau)] \} & \cdots & E \{ [x_n(t), x_n(\tau)] \} \end{bmatrix} \quad (D.7)$$

from above we can conclude that;

$$R(t, \tau) = R^T(\tau, t) \quad (D.8)$$

Definition D.9 *Covariance Matrix*; A special case of the above Equation (D.8) is;

$$R(t, t) = E \{ [x(t), x^T(t)] \} = R^T(t, t) \quad (D.9)$$

Definition D.10 *Variance equation* is the following;

$$\dot{P} = AP + PA^T + FQ_vF^T \quad (D.10)$$

where P is the covariance matrix of state $x(t)$ at time t . Variance equation is used to determine how the covariance propagates with the elapse of time, without having to calculate the state transition matrix. This is extremely important in time-variant systems, where the state transition matrix is almost impossible to determine and in time-invariant systems to evaluate steady state covariance matrices.

D.3 Convolution Identities

Following are some of the important convolution identities, the notation used is;

$$\begin{aligned}
* &= \text{convolution} \\
f^{**n} &= n\text{-fold convolution of function } f \\
f_{**n} &= n\text{-fold integration of function } f \\
f_{*-n} &= n\text{-fold differentiation of function } f
\end{aligned}$$

and the identities are;

$$\begin{aligned}
f * g &= g * f \\
(f * g) * h &= f * (g * h) \\
\int (f * g) &= \left(\int f \right) * g \\
(f * g)' &= (f)' * g
\end{aligned} \tag{D.11}$$

repeated application of the last two identities from above, imply that the convolution of f with g is equivalent to the convolution of the n th integral of f with the n th derivative of g ;

$$f * g = \left(\int^n f(x) dx \right) * \left(\frac{d^n g}{dx^n} \right) = f_n * g_{-n} \tag{D.12}$$

D.4 Gaussian Distribution

A Gaussian or a normal function is

$$\mathcal{G}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{D.13}$$

and its integral is

$$\mathbf{G}(x) = \int_{-\infty}^x \mathcal{G}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{y^2}{2}} dy \tag{D.14}$$

For large x , $\mathbf{G}(x)$ is often approximated by;

$$\mathbf{G}(x) \simeq 1 - \frac{1}{x} \mathcal{G}(x) \tag{D.15}$$

$\mathbf{G}(x)$ is often expressed as an error function ε ;

$$\varepsilon(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{y^2}{2}} dy = \mathbf{G}(x) - \frac{1}{2} \tag{D.16}$$

An RV x is called a normal or gaussian if its density is the normal curve $\mathcal{G}(x)$, shifted and scaled;

$$f(x) = \frac{1}{\sigma} \mathcal{G}\left(\frac{x - \eta}{\sigma}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\eta)^2}{2\sigma^2}} \quad (\text{D.17})$$

and the corresponding distribution function is

$$F(x) = \mathbf{G}\left(\frac{x - \eta}{\sigma}\right) \quad (\text{D.18})$$

Appendix E

Control Theory : Definitions

E.1 System Dynamics : Definitions

Following are some important definitions of system dynamics

Definition E.1 *A system is an entity separable from the rest of the environment by means of physical and conceptual boundaries.*

Definition E.2 *A dynamic system is a system whose behavior is a function of time.*

Definition E.3 *A state of a dynamic system is a set of physical quantities, the specification of which (in absence of external excitation) completely determines the evolution of the system [57].*

Definition E.4 *The number of first-order differential equations in the mathematical model of the system defines the order of the system.*

Definition E.5 *The dynamic variables that appear in the system of first-order equations are called the state variables*

E.2 Control Theory : Definitions

Following are some of the important definitions of Control Theory;

Definition E.6 *A dynamic system whose state variables are the estimates of the state variables of the system under observation, then the former system is called the Observer of the system under study.*

The concept of an observer was introduced into linear system theory by D. Luenberger in 1963 [69, 70, 71]. Luenberger showed that, for any observable linear system, an observer can be designed having the property that the estimation error can be made to go to zero [57].

Several years before Luenberger's introduction of observers, R. E. Kalman with R. Bucy defined a state estimator that is optimum with respect to process noise and observation noise [67, 68]. This state estimator, called *Kalman filters* has a structure of a linear observer, hence a *Kalman filters* may be regarded as an optimum observer. Although observers are useful for estimating the state of a linear system having a known external input, their main use is in estimating the state variables that cannot be measured but are needed for implementation of feedback control. Two very important definitions for control systems are;¹

Definition E.7 (Observability) *An unforced system is said to be observable if and only if it is possible to determine any arbitrary initial state $x(t) = x_t$ by using only a finite record, $y(\tau)$ for $t \leq \tau \leq T$, of the output [57].*

Definition E.8 (Controllability) *A system is said to be controllable if and only if it is possible, by means of input, to transfer the system from any initial state $x(t) = x_t$, to any other state $x_T = x(T)$ in a finite time $T - t \leq 0$ [57].*

¹for more reading on Observability and Controllability see Friedland [57] Chapter 5.

E.3 White Noise and Linear System Response

White noise is one of those theoretical abstractions which simplifies calculations but grieves mathematicians. It is defined as following;

Definition E.9 *White noise is simply a random process with an expected value (mean) of zero and with a flat power spectrum [57]. The correlation function of white noise is;*

$$\rho(\tau) = W\delta(\tau) \quad (\text{E.1})$$

where $\delta(\tau)$ is a unit impulse at origin and W is the constant at all ω , because of the flat power spectrum ²

As the mean square value of any random process is the integral of its spectral density over all frequencies and since *white noise* has constant spectral density for all frequencies, therefore, theoretically *white noise* has infinite mean square value. A physical process that has infinite mean square value is inconceivable and therefore *white noise* is a mathematical abstraction.

A vector random process is *white noise* if its correlation matrix is of the form;

$$R(\tau) = W\delta(\tau) = E \left\{ x(t)x^T(t + \tau) \right\} \quad (\text{E.2})$$

where W is a square matrix. *white noise* is a convenient abstraction because it leads to a relatively simple expression for a correlation function and the power spectrum of the output of the linear system into which it is the input. For some examples of *white noise* as an input see Friedland [57].

²For more details on this see [57].

References

- [1] I. Sutherland. A Man-Machine Graphical Graphical Communication System, in Interactive Computer Graphics. In *1963 Spring Joint Computer Conference Proceedings*, 1963.
- [2] A Borning. *A Constraint Oriented Simulation Laboratory*. Technical Report SSL-79-3, Xerox PARC, Palo Alto, CA, 1979.
- [3] A. P. Pentland and J. Williams. Virtual Construction. *Construction*, 3(3), 1988.
- [4] D. Zeltzer, S. Pieper, and D. Sturman. An Integrated Graphical Simulation Platform. *Proc. Graphics Interface '89*, 1989.
- [5] Klaus-Jürgen Bathe and Anil Chaudhary. A Solution method for Axisymmetric Contact Problems. *International Journal for Numerical Methods in Engineering*, 21:65–88, 1985.
- [6] A. P. Pentland and J. R. Williams. Good Vibrations : Modal Dynamics for Graphics and Animation. *Computer Graphics*, 23(4):215–222, 1989.
- [7] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically Deformable Models. *Computer Graphics*, 21(4):205–214, 1987.
- [8] A. Witkin, K. Fleischer, and A. Barr. Constraints on Parameterized Models Models. *Computer Graphics*, 21(4):225–232, 1987.
- [9] Matthew Moore and Jane Wilhelms. Collision Detection and Response for Computer Animation. *Computer Graphics*, 22(4):289–298, 1988.
- [10] Ferdinand P. Beer and E. Russell Johnston Jr. *Vector Mechanics for Engineers*. McGraw-Hill, 1984.
- [11] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1988.
- [12] James M. Gere and Stephen P. Timoshenko. *Mechanics of Materials*. Brooks/Cole Engineering Division, 1984.
- [13] A. P. Pentland. Computational Complexity Versus Virtual Worlds. *Computer Graphics*, 24(2):185–192, 1990.
- [14] L. M. Taylor and D. S. Preece. Simulation of Blasting Induced Rock Motion Using Spherical Element Models. *First U.S. Discrete Element Conference Proceedings*, 1989.
- [15] C. Thornton. Applications of DEM to Process Engineering Problems. *First U.S. Discrete Element Conference Proceedings*, 1989.
- [16] J. R. Williams and A. P. Pentland. Superquadrics and Modal Dynamics for Discrete Elements in Concurrent Design. *First U.S. Discrete Element Conference Proceedings*, 1989.

- [17] J. R. Williams. Contact Analysis of Large Number of Interacting Bodies Using Discrete Modal Methods for Simulating Material Failure on the Microscopic Scale. *Engineering Computations*, 5:198–209, 1988.
- [18] Klaus-Jürgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.
- [19] Larry J. Segerlind. *Applied Finite Element Analysis*. John Wiley and Sons, 1984.
- [20] A. P. Pentland, I. A. Essa, M. Friedmann, B. Horowitz, and S. E. Sclaroff. The Thingworld Modeling System: Virtual Sculpting by Modal Forces. *Computer Graphics*, 24(2):143–144, 1990.
- [21] A. P. Pentland. *THINGWORLD Users Guide*. M.I.T. Media Lab, Vision and Modeling Group, E15-383, 20 Ames Street, Cambridge, MA 02139, 1990.
- [22] Ruzena Bajcsy and Franc Solina. Three Dimensional Object Representation Revisited. *Proceedings First International Conference on Computer Vision*, 231–240, 1987.
- [23] Alan H. Barr. Faster Calculations of Superquadric Shapes. *IEEE Computer Graphics and Applications*, 6(1):101–122, 1981.
- [24] Alan H. Barr. Superquadrics and Angle-Preserving Transforms. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [25] Alan H. Barr. Profiles, surfaces and solids : a supplement. 1981. A Supplement from a tutorial.
- [26] Alan H. Barr. Global and Local Deformations of Solid Primitives. *Computer Graphics*, 18(3):21–30, 1984.
- [27] A. P. Pentland. Perceptual Organization and Representation of Natural Forms. *Artificial Intelligence*, 28(3):293–331, 1986.
- [28] A. P. Pentland. Towards an Ideal 3-D CAD system. *SPIE Conference on machine vision and man-machine interface*, 1987.
- [29] Van-Duc Nguyen. *Shape Primitives and Deformations*. Master's thesis, M.I.T, 1985.
- [30] Franc Solina. *Shape Recovery and Segmentation with Deformable Part Models*. PhD thesis, University of Pennsylvania, 1987.
- [31] A. P. Pentland. Automatic Extraction of Part Deformable Models. *International Journal of Computer Vision*, 4:107–126, 1990.
- [32] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge, 1986.
- [33] Gilbert Strang. *Linear Algebra and its Applications*. Wellesley-Cambridge, Third edition, 1986.
- [34] James K. Hahn. Realistic Animation of Rigid Bodies. *Computer Graphics*, 22(4):299–308, 1988.

- [35] David Baroff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. *Computer Graphics*, 23(4):223–232, 1989.
- [36] Peter Schröder and David Zeltzer. The Virtual Erector Set: Dynamic Simulation with Linear Recursive Constraint Propagation. *Computer Graphics*, 24(2):23–31, 1990.
- [37] Matthew T. Mason and J. K. Salisbury. *Robot Hands and Mechanics of Manipulation*, chapter Theory of Pushing, pages 189–251. The MIT Press, 1985.
- [38] Matthew T. Mason. Mechanics and Planning of Manipulator Pushing Operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.
- [39] M. R. Cutkosky and S. H. Lee. Fixture Planning with Friction for Concurrent Product/Process Design. *NSF Engineering Design Research Conference Proceedings*, 613–628, 1989.
- [40] Per Lötstedt. Numerical Simulation of Time-dependent Contact and Friction Problems in Rigid Body Mechanics. *Society of Industrial and Applied Mechanics*, 5(2):370–393, 1984.
- [41] Andrew Glassner. Adaptive Precision in Texture Mapping. *Computer Graphics*, 20(4):297–306, 1986.
- [42] J. Blinn and M. E. Newell. Texture and Reflection in Computer Generated Images. *Communications of ACM*, 19(10), 1976.
- [43] R. Cook. Shade Trees. *Computer Graphics*, 18(3), 1984.
- [44] D. Schweitzer. Artificial Texturing: An Aid to Surface Visualization. *Computer Graphics*, 18(3), 1984.
- [45] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6(6):661–674, 1984.
- [46] B. B. Mandelbrot. *The Fractals Geometry of Nature*. W. H. Freeman and Co., 1982.
- [47] Michael Barnsley. *Fractals Everywhere*. Academic Press, 1988.
- [48] Heinz-Otto Peitgen and Dietmar Saupe. *The Science of Fractals Images*. Springer-Verlag, 1988.
- [49] James Gleick. *Chaos*. Penguin Books, 1987.
- [50] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.
- [51] Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–740, 1984.
- [52] Bruno Buchberger. Applications of Gröbner Bases in Non-Linear Computational Geometry. In Deepak Kapur and Joseph L. Mundy, editors, *Geometric Reasoning*, pages 413–446, MIT Press, 1989.

- [53] Deepak Kapur and Joseph L. Mundy, editors. *Geometric Reasoning*. MIT Press, 1989.
- [54] Jan J. Koenderink. *Solid Shape*. MIT Press, 1990.
- [55] Morris H. DeGroot. *Probability and Statistics*. Addison-Wesley, 1987.
- [56] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.
- [57] Bernard Friedland. *Control System Design*. McGraw-Hill, 1986.
- [58] Masanao Aoki. *Optimization of Stochastic Systems: Topics in Discrete-Time Dynamics*. Academic Press, second edition, 1989.
- [59] Scott N. Steketee and Norman I. Badler. Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control. *Computer Graphics*, 19(3):255–262, 1985.
- [60] Lynne S. Brotman and Arun N. Netravali. Motion Interpolation by Optimal Control. *Computer Graphics*, 22(4):309–315, 1988.
- [61] Dean Karnopp and Ronald Rosenberg. *System Dynamics: A Unified Approach*. John-Wiley and Sons, 1975.
- [62] Stephen Wolfram. *Mathematica; A System for Doing Mathematics by Computer*. Addison-Wesley, 1988.
- [63] P. S. Heckbert. Filtering by Repeated Integrations. *Computer Graphics*, 20(4):315–321, 1986.
- [64] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, 1978.
- [65] Monson H. Hayes. *Digital Wiener Filters*. Technical Report, School of Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, 1989.
- [66] A. P. Pentland. Linear Shape from Shading. *International Journal of Computer Vision*, 4:153–162, 1990.
- [67] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transaction ASME (Journal of Basic Engineering)*, 82D(1):35–45, 1960.
- [68] R. E. Kalman and R. S. Bucy. New Results in Linear Filtering and Prediction Theory. *Transaction ASME (Journal of Basic Engineering)*, 83D(1):95–108, 1961.
- [69] D. G. Luenberger. Observing the State of a Linear System. *IEEE Transactions on Military Electronics*, MIL-8:74–80, 1964.
- [70] D. G. Luenberger. Observer for Multivariable Systems. *IEEE Transactions on Automatic Control*, AC-11:190–197, 1966.
- [71] D. G. Luenberger. An Introduction to Observers. *IEEE Transactions on Military Electronics*, MIL-16:596–602, 1971.