Curtin Sarawak Research Institute

Real-time Knowledge-based Fuzzy Logic Model for Soft Tissue Deformation

Joey Tan Sing Yee

This thesis is presented for the Degree of

Master of Philosophy (Mechanical Engineering)

of

Curtin University

December 2014

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature : ……
(JOEY TAN SING YEE)

Date : December 2014

Abstract

Over the past few years, surgical simulation has emerged as an alternative medical training or pre-operation planning method in the medical field. The demands of surgical simulation increases, however, there remains a major bottleneck in the surgical simulation designation. Every patient is unique. Thus, the core part of surgical simulation which is the deformable model, plays the most crucial role in the surgical simulation designation.

In this research, we present an improved mass spring model to simulate the soft tissue deformation for surgery simulation. The human liver is modelled by the corresponding resultant Mass Spring Model (MSM) for liver deformation. The underlying MSM is redesigned where the parameters of the mass spring model are determined by using three different fuzzy knowledge-based approaches. This is to ensure that the accuracy of the simulation can be assured. The Euler integration method is adopted to calculate the position of a mass point. Next, the data in Central Processing Units (CPUs) memory which is accessed by predictable transactions is structured in such a way that coalescing is allowed according to a set of Graphical Processing Units (GPUs)-dependent alignment rules. Meanwhile, heterogeneous parallel programming is implemented for the distribution of grid threads for Computer Unified Device Architecture (CUDA) based GPU computing. The CUDA based deformable model has been tested and validated with certain dataset through coding. The novelty of this research is that for liver modelling in particular, no specific contributions in the literature exist reporting on real-time knowledge-based fuzzy mass spring model for surgical simulation.

In conclusion, the fuzzy approaches are proven to be powerful approaches in building complex and nonlinear relationship between a set of input and output data. The stiffness values determined by fuzzy approaches are in very good agreement with the benchmark result. The corresponding graphs for each of the fuzzy approaches share similar trend of displacement and velocity with the benchmark model. Among three of the fuzzy approaches, the Interval Type-2 Fuzzy Inference System has the highest graph similarity (0.8598 over 1) with the benchmark model. Besides, our results showed that when using GPU assisted hardware, our codes can handle increased structural complexity while providing significantly realistic and real-time liver deformation.

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Dr. Amandeep S.Sidhu, who has been tirelessly guiding, dedication in advising and encouraging throughout my research studies (Master of Philosophy, Mechanical Engineering). His availability for discussion on my research works has further encouraged me and I am truly grateful. Deeply appreciated for his great patience.

In addition, I would like to thanks for the opportunity in receiving fee waiver under the grant "FRGS". Also, much appreciate for the monthly stipend which have been a great financial aids for funding my research projects.

My gratitude also goes to my fellow colleagues in Curtin University Sarawak Campus namely Vijayajothi Paramasivam, Cho Chung Yik, John Alan Leong Seng Hui, Jason Tan Rong Kun and Akram Akbari for their assists on technical and non-technical support during my research studies. For anyone I may have left out, I ensured you that any help that had been given was never neglected.

All my friends and families deserve a big thank you for their precious support and encouragement throughout the past year. In particular, I thank to my parents, for their lovely tender cares, support, selfless love and direction.

Last but not least, I would like to thank God for His guidance throughout my research studies. He had led me through the hard time and had prepared me everything required for completing this thesis. All glory and honours goes on Him in the highest.

Publications

Journal Papers

1. **Tan Sing Yee**, Jason Tan Rong Kun, Sarinder K. Dhillon, Amandeep S. Sidhu, "*Virtual Reality Deformable Liver Model in CUDA"*, Virtual Reality, 2014 (Under Review, ISI indexed).
2. **Tan Sing Yee**, Sarinder K. Dhillon, Amandeep S. Sidhu, "*Mass Spring Model Parameters Selection by Using Fuzzy Knowledge-Based Approach"*, IEEE Transaction on Virtual and Computer Graphics, 2014 (Under Review, ISI indexed).
3. **Tan Sing Yee**, Sarinder . Dhillon, Amandeep S. Sidhu, "*Real-time Knowledge-based Fuzzy Modelling on Liver Deformation"*, IEEE Transaction on Fuzzy System, 2014 (Under Review, ISI indexed).
4. Vijayajothi Paramasivam, **Tan Sing Yee**, Sarinder K. Dhillon, Amandeep S. Sidhu, "*A Methodological Review of Data Mining Techniques in Predictive Medicine: An Application in Hemodynamic Prediction for Abdominal Aortic Aneurysm Disease"*, Biocybermetics and Biomedical Engineering, Vol. 34. No. 3, pp. 139-145, 2014 (Published).

Refereed Conference Papers

1. **Tan Sing Yee**, Sarinder K. Dhillon, Amandeep S. Sidhu, "*Fuzzy Estimation of Liver Stiffness in Modelling Liver Deformation"*, IEEE 14th International Conference on Bioinformatics and Bioengineering, Boca Raton, Florida, United State, 2014 (Published)

Table of Contents

List of Figures

List of Tables

Abbreviations

VR	Virtual Reality
VE	Virtual Environment
3D	Three Dimensional
CPUs	Central Processing Units
GPUs	Graphical Processing Units
CUDA	Computer Unified Device Architecture
MSM	Mass Spring Model
FEM	Finite Element Model
PDEs	Partial Differential Equations
FMPC	Knowledge-based Fuzzy MSM Parameters Controller
2D	Two Dimensional
TE	Transient Elastography
GFS	Generalized Fuzzy System
FIS	Fuzzy Inference System
COA	Centroid of Area
F0-F1	Absent/ Mild Fibrosis
F2	Significant Fibrosis
F3	Severe Cirrhosis
F4	Cirrhosis
LSM	Liver Stiffness Measurement

HCV	Hepatitis C Virus
HBV	Hepatitis B Virus
ALD	Alcoholic Liver Disease
CLD	Chronic Liver Disease
NAFLD	Non-Alcoholic Fatty Liver Disease
MFs	Membership Functions
Gauss_MF	Gaussian Membership Function

Glossary

F_N	Newton's 2nd Law of Motion
F_s	Hooke's Law
m	mass
γ	damping coefficient
k	stiffness coefficient
$\ddot{x}_{ijk}$	acceleration
$\dot{x}_{ijk}$	velocity
x_{ijk}	displacement
F_{ext}	external forces
ς	damping ratio
w	natural frequency
A	crisp set
B	crisp set
x	Fuzzy set
y	Fuzzy set
$\mu_A(x)$	crisp set of x
$\mu_B(y)$	crisp set of y
R	Relationship
$\cap$	Intersection, "AND"

$\in$	Subset
min	Fuzzy intersection
max	Fuzzy union
L_O	Lower boundary
M	Median
U_p	Upper boundary
W_k	Weighting factor
$M_{imp,k}$	the implication output membership function
F_i^s	highly non-linear elasticity behaviour of the soft tissue
K_1	constants
K_2	constants
X_C	Critical displacement of the nonlinear springs
F_i^d	Damping force
γ_0	Damping constant
γ_1	Damping constant
x_i	Position vector
x_i^0	Initial position of node i
$\mu_{A^i}(x)$	Gaussian membership function
c_i	centre of the i-th fuzzy set A^i
σ_i	width of the i-th fuzzy set A^i

A^i	Fuzzy set
m_F	Gradient of corresponding stiffness
m_B	Gradient of benchmark

Chapter 1 *Introduction*

1.1 Overview

Over the past few years, surgical simulation has emerged as an alternative medical training or pre-operation planning method in the medical field. During the early stage of surgical training, novice surgeons used to practice via animals, cadavers and real patient. Each of these methods face challenges in terms of cost, availability, ethical restriction and realistic. The animals' organs or soft tissues do not accurately represent the human anatomy, particularly the measurement of the organs' sizes. This is because the size of human liver increases as the age increases (Ahn and Kim, 2009). However, at the age of 15 years old, the human liver would reach the adult liver stage. It is approximately 21-22.5 centimetres across its widest point, 15-17.5 centimetres at its greatest vertical height, and 10-12.5 centimetres from front to back. It weighs about 1.2-1.4 kilograms for women and 1.4-1.5 kilograms for men. Therefore, the properties of living soft tissues in the cadavers are thoroughly different with the real living tissues in term of anisotropic, nonlinearity, time-dependent and viscoelasticity (Fung Y.C., 1993). Among three of the early stage surgical training methods, real patient-based training is the most impractical as it risks the safety of the patient and there exists some ethical restriction. In addition, there will be growing pressure for the novice surgeons as the responsibility is heavy when it comes across lives (Schaverien, 2010).

Apart from above mentioned early stage surgical trainings, there is another training method which is known as apprenticeship-based training. This is where a group of novice surgeons observe certified surgeon to operate on a real patient, serving a small task then gradually handling the operation alone (Schaverien, 2010). As shown in Figure 1.1, a group of novice surgeon observing without having chances to operate on the real patient. This process of learning is time consuming and impractical. Therefore, surgical simulation is introduced.

Figure 1.1 Group of novice surgeons observing the demonstration by the certified surgeon. (Source: http://www.flickr.com/photos/medicalmuseum)

Surgical simulation is proposed for the needs to complement the early stage surgical training methods. With the advances in computer graphics technology which introduces Virtual Reality (VR) into entertainment, experiments, flight simulations, car crashing demonstration and health care. VR allows the combination of convincing representation of soft tissues in computer generated visualization known as Virtual Environment (VE). In VE, interaction between the user and computer generated virtual objects are allowed. Thus, by incorporating VE into surgical simulation, it is able to present an interactively immersive surgical training for the novice surgeons. Perhaps with the aids of interactive VE, surgical simulation will help to enhance the novice surgeons learning progress and surgical skills which will boost their confidence in dealing with real surgeries (Sala et al., 2011a). Besides, surgical simulations allow repeated trial-and-error trainings and pre-operative planning for the certified surgeons

before they perform exact surgeries on the real patient (Xu et al., 2011). Figure 1.2 shows the surgical training in virtual environment.

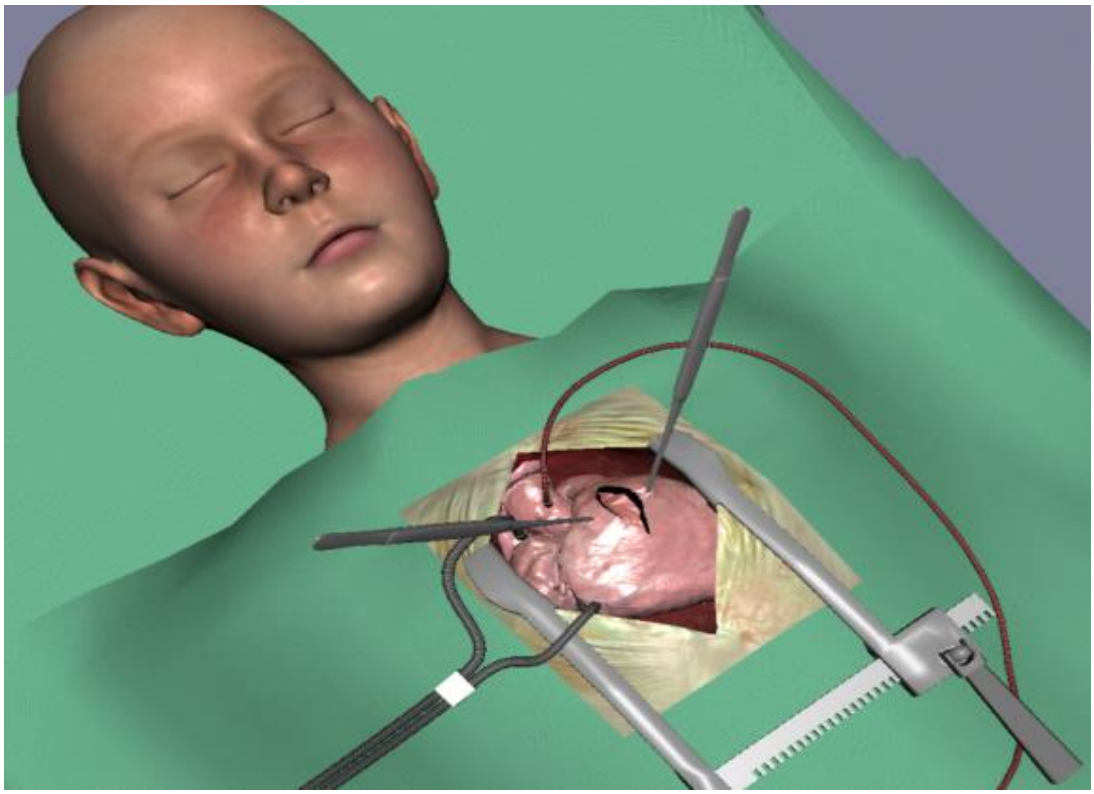


Figure 1.2 Surgical training in virtual environment. (Source: http://cg.alexandra.dk)

The first surgical simulation was introduced by Robert Mann in the ‘60s, where a rehabilitation application was developed to allow the medical surgeon to perform several surgical approaches for a given orthopaedic problem (Indelicato, 1995). Later, as stated by Rosen (2008), in 1994, a completed version of Visible Human Project is introduced for surgical simulation by the National Library of Medicine. The imaging data from real human cadavers were employed to develop anatomical models in this project. The anatomical models are represented in three dimensional (3D) views and manipulation of anatomical structures are allowed.

In addition, statistics showed that due to the risky practices on real patients, the errors in medical care caused approximately 44,000 to 98,000 deaths per year. However, the results of errors in medical care have been remarkably decreased after the innovation of surgical simulations. Surgical simulation is basically a technique but not classified as a technology and it has been practiced for years (Aebersold et al., 2012). Therefore, with the increasing demands on surgical simulation in medical field, surgical simulation is designed, improved and employed to substitute for and eventually replace cadavers and animals (De et al., 1999; Aebersold et al., 2012).

However, the design process for a state-of-the-art surgical simulation is challenging. Although sophisticated surgical simulations have been developed over the past decades, there is a remaining daunting challenge—realistic and real-time of the simulated surgery (Teschner et al., 2000). This is mainly because the existence of certain trade-offs between realistic and real-time of a surgical simulation. A realistic simulation of soft tissue is computationally intensive due to the inherent complexities of the governing mathematical models resulting from the large deformation and soft tissues properties (De et al., 1998). Therefore, the choice of a deformable model to represent the exact properties of soft tissues is essential with respect to realistic and real-time computational speed (De et al., 1999).

In this research, we present an improved mass spring model to simulate the soft tissue deformation for surgery simulation. The human liver is modelled by the corresponding resultant MSM model for liver deformation. The underlying mass spring model is redesigned where the parameters of the mass spring model are determined by using three different fuzzy knowledge-based approaches. This ensures the accuracy of the simulation can be assured. We adopt the Euler integration method to calculate the position of a mass point. Next, the data in Central Processing Units (CPUs) memory

which is accessed by predictable transactions is structured in such a way that coalescing is allowed according to a set of Graphical Processing Units (GPUs)-dependent alignment rules. Meanwhile, heterogeneous parallel programming is implemented for the distribution of grid threads for Computer Unified Device Architecture (CUDA) based GPU computing. The CUDA based deformable model will be tested and validated with certain dataset through coding. The novelty of this research is that for liver modelling in particular, no specific contributions in the literature exist reporting on real-time knowledge-based fuzzy mass spring model for surgical simulation.

1.2 Research Background

Surgical simulation is one of the promising application where deformable models are applied. It utilize these models to simulate living human liver soft tissue or anatomy in a computer generated environment (Rosen, 2008). Novice surgeons who are unfamiliar or more likely to make mistakes during surgery could practice the basic surgical skills and procedures via surgical simulation (Naksupakpong, 2010). Thus, surgical simulation provides a hazardless and stress-free learning environment for the surgical newbies. Meanwhile, certified surgeons are able to gain the opportunity to test and explore different critical surgery scenarios in a low cost environment without ethical restriction (San Vicente, 2011). It may also allow the certified surgeons to avoid undesirable outcomes by predicting the outcomes of a surgery prior to actually performing the surgery via surgical simulation. Therefore, surgical simulation plays a crucial instrumental training system for either the novice or certified surgeons.

Although surgical simulation could improve the skill of a successful surgeon, perhaps de facto surgery is complex nowadays. Therefore, the choice of a deformable model is

essential in the designation of a state-of-the-art surgical simulation. The deformable model is required to represent the deformations of the simulated deformable object in real-time and realistically. However, as mentioned in previous section, there is a trade-off between real-time and realistic deformation. This is because a realistic deformation requires a complex computational burden which delay in real-time. Among the various deformable models available concurrently, the Mass Spring Model (MSM) is potentially to represent the liver deformation in real-time surgical simulation (Xu et al, 2010). Although the MSM is doubted on its accuracy and realism as compared to the Finite Element Model (FEM), however, it is capable in representing the nonlinear and viscoelasticity for living soft tissue in real-time (Basafa et al., 2011). Even though FEM is more accurate, the creation of a quality mesh for the problem domain is the major part of the cost in surgical simulation (Basafa et al., 2011). This is because, in generating a mesh, it requires a continuum to be divided into several elements. The elements are then connected together by a topological map, which is usually called a mesh. The interpolation functions are then built on the mesh. Since FEM could be derived mathematically and have inter-relation with the stress-strain coefficient (Young's Modulus and Poisson Ratio), it could represent the liver deformation realistically but not in real-time (Xu et al., 2010). Therefore, FEM becomes a bottleneck in large-scale soft tissue deformations.

In this research, Mass Spring Model is chosen to model the living human liver. MSM is a discrete model which consists of a bunch of mass points and linked by a network of zero force springs. MSM avoids initialisation and features high refreshing frame rates, which increases the efficiency in real-time (Courtecuisse et al., 2010). Therefore, it is one of the most popular and favoured deformable model in manipulating a deformable object in terms of real-time (Vollinger et al., 2009; Zhang et al., 2010).

However, the only remaining challenge of the MSM is the selection of parameters (Er and Sun, 2001; Campos et al., 2013; Zhang et al., 2013). Many research projects have been developed to improve the underlying MSM parameter selections in terms of analytical and numerical approaches (Van Gelder, 1998; Maciel et al., 2003; Xu et al., 2010; Abulgasem et al., 2010), heuristic optimisation (Deussen et al., 1995; Bianchi et al., 2003; Pawlus et al., 2013), utilisation of Graphical Processing Unit (GPU) (Georgii et al., 2005; Tejada et al., 2005), and implementation of the Computer Unified Device Architecture (CUDA) approaches (Rasmusson et al., 2008; Leon et al., 2010; Zhang et al., 2010; Etheredge C.E, 2011; Campos et al., 2013; Zhang et al., 2013). Although the resultant models are able to simulate the deformable objects in real-time, there exists some limitations. For instance, in the work of Campos et al. (2013), the MSM is implemented to mimic the heart tissue mechanical behaviour based on cellular automation in CUDA platform. Even though the 3D simulations is performed in real time as CUDA enables the computation of Partial Differential Equations (PDEs) with much less effort, Campos urged to improve the determination of the MSM parameters. This is because the determination of the MSM parameter is based on manual tuning until a desirable simulation result is obtained.

In this research, the focus is directed to obtain the exact parameter of different stage of liver fibrosis based on fuzzy knowledge parameter selections. Aforementioned, the ultimate goal of a surgical simulation is to manipulate high accuracy 3D liver model efficiently in real-time. However, the liver is a complex model which is mechanically viscoelastic, nonlinear, anisotropy, time-dependent and non-homogeneous (Cotin et al., 1999; Marchesseau et al., 2010; Yamamoto, 2011). The mechanical behaviours of the liver tissues varies substantially depending on the age and sex of the living human

(Ahn and Kim, 2009). Thus, even for a normal healthy liver, it is a daunting task to model the exact 3D model with accurate sets of MSM parameters.

1.3 Problem Statement

Modelling the mechanical behaviour of liver soft tissues is a daunting task due to the complex tissue configuration (Delingette et al., 2004). The soft tissues are commonly featured with nonlinearity, viscoelasticity, time and rate dependency and non-homogeneous (Fung, Y.C., 1993; Cotin et al., 1999; Marchesseau et al., 2010). Several studies have been developed attempting to capture the exact mechanical properties of the liver living soft tissues via sampling across frequencies and stress-strain tests. However, almost all tests fail to capture the soft tissue characteristics (Xu et al., 2011). According to Ahn and Kim (2011), there has been insufficient investigation done in obtaining the exact mechanical properties of human living soft tissues. The already existing mathematical models or theories are incomplete and vary among the researchers in different time intervals (Halic et al., 2009). With the advance in computer graphics, physically-based deformable model is introduced in representing deformable objects such as cloth simulation (Terzopoulus, 1988). Later, in the 90', these models are used to develop 3D model to simulate surgical treatment (Delp et al., 1997; Courtecuisse et al., 2010). However, there exists certain limitations for each of the proposed deformable models due to the uncomplimentary between realism and real-time (Nesme et al., 2009).

Mass Spring Model (MSM) is the most popular physically-based deformable model because it is conceptually simple and possible to construct large deformations in real-time (Hu, 2006; Xu et al., 2010). Unfortunately, the main issue of MSM is that it is difficult to obtain the correct spring parameters, which could interrupt the accuracy of

the surgical outcomes (Vollinger et al., 2009). MSM consists of three parameters such that the spring stiffness coefficient, damping coefficient, and the mass of each mass point. Among various literature studies, there is no general physically-based or systematic method to determine the element types or parameters from physical data or a known constitutive behaviour. The parameters are typically set manually by tuning until the visual appearance is pleasing (Naksupakpong and Cavusoglu, 2008). Else, the frequently used methods in determining the parameters are empirically assumed based on the structure of the geometrical model by using a parameter optimization method to estimate the best compatible behaviour. For instance, in the work of Joukhader et al. (1997), a predefined mesh topology is used to determine the element parameters with a genetic algorithm search technique. This is because the parameters in MSM do not have direct relationship with the elastic constants—Young's Modulus and Poisson Ratio (Courtecuisse et al., 2010).

In overcoming this drawback, several optimisation techniques have been developed. Since the early works of Van Gelder (1998) presented a formula for the spring coefficient on how to map biomechanical parameters into spring coefficients for 2D membranes. However, the simulated elastic membrane is only sufficient for triangulated spring mesh. Maciel et al. (2003) suggested a generalised MSM for nonlinear springs in a hexahedral mesh where the stiffness parameter depends on the current angles between the springs. Bianchi et al. (2003) on the other hand, used a genetic algorithm based method to determine the mesh topology and the stiffness of MSM by using Finite Difference Model (FDM) as reference. Besides, Choi (2010) proposed a hybrid heuristic approach to identify the parameters by employing simulated annealing into genetic algorithm. The optimisation process is performed with reference to the FEM benchmark model in order to determine the MSM

parameters. It is obvious that there is no straightforward mathematical theories or methods in identifying the MSM parameters from previous works.

Moreover, the stiffness coefficient is one of the essential parameter which influence the stability of the mass spring system. A stiff spring exerts more force compared to a weak spring. The greater the stiffness value, the more force will be exerted onto the spring at a given angle, thus causing instability to the system. Therefore, to tackle this MSM parameter selection issue, the availability knowledge of liver stiffness shall be incorporated into the underlying MSM via fuzzy approach. As shown in the work of Pawlus et al. (2013), the fuzzy model is applied to improve FEM model fidelity. It shows that the fuzzy model's output could be adjusted to improve the model's fidelity without complicated computations.

1.4 Objectives

The primary aim of this research is to propose a novel CUDA based hybrid model for representing the human soft tissue deformation in real-time. A secondary dataset which is available from open source such as SOFA (Simulation Open Framework Architecture) is used to generate the liver model.

The key objectives of the research are as follows:

1. To improve the underlying MSM by adapting the exact spring stiffness coefficient using knowledge-based fuzzy MSM parameters controller (FMPC).
2. To implement and evaluate the proposed deformable model in CUDA based programming by closing the gap between real-time simulation and realistic modelling of soft tissue deformation.

1.5 Scope of Study

In order to achieve the above mentioned research objectives, the scope of study includes:

1. Physically-based deformable object—human liver.
2. Collecting spring stiffness coefficient data (knowledge) from previous medical research and reports.
3. Obtaining MSM parameters by knowledge-based MSM parameter controller via MATLAB r2014a.
4. Generating liver mesh via OpenGL and C++ coding.
5. Optimizing Euler Integration through Computer Unified Device Architecture (CUDA) platform.
6. Analysing the realistic and real-time deformation by comparing the proposed model with the benchmark model—Basafa's model (2011).

1.6 Outline of Dissertation

We will now provide a brief outline of this dissertation content. In Chapter 1, we discuss on the background of surgical simulation, problem statements and the objectives of this research. In chapter 2, we discuss on the already existing surgical simulation, approaches in improving the existing Mass Spring Model (MSM), fuzzy systems and parallel programming in Computer Unified Device Architecture (CUDA). In Chapter 3, we discuss on the methodology of this research. Also, we present the formulation of MSM and Fuzzy Mass Spring Model Parameters Controller (FMPC), implementation of CUDA and the benchmark model. In Chapter 4, we present the generation of rules based on fuzzy rule matrix. Here, we show the simulation result where comparison between three different fuzzy approaches are presented. We also

discuss on the graphs similarities between the three fuzzy approaches with the benchmark model. In Chapter 5, we discuss on the outcomes of simulated result from CUDA implementation. Lastly, in Chapter 6, we draw a conclusion for this research and state certain limitations in our approach and future works in overcoming the corresponding limitations are drawn.

Chapter 2 *Literature Review*

2.1 Surgical Simulation

2.2 Deformable Models

2.3 Fuzzy Systems

2.4 Graphical Hardware and Programming Optimisation

2.5 Summary

2.1 Surgical Simulation

During the early stage of surgical training, most of the surgical trainings are practiced via animals, cadavers and real patient. Each of these method faces challenges and obstacles because animals' organs or soft tissues do not represent the human anatomy accurately. Moreover, the cadaver's soft tissues properties are thoroughly different with the real living soft tissues in terms of anisotropic, nonlinearity, inhomogeneous and viscoelasticity. Whereas real patient's soft tissues have the exact properties but this will put the patient's safety at risk. Therefore, surgical simulation is introduced.

Over the past few years, surgical simulation has emerged as an alternative training method for medical education. As stated in the work of De et al. (1999), surgical simulators are designed to substitute for and eventually replace cadavers and animals which are currently used for surgical training. Surgical simulation presents a framework for interactive simulation of surgical operations which is being practiced in surgical treatment (Bielser, D. and Gross, M.H., 2000). Therefore, similar to flight simulator, surgical simulator promotes a safer and low-cost training for both novice and professional surgeons (Baur et al., 1998). In which the patient's safety is not at risk and complicated surgical procedures can be practiced repeatedly (Bosdogan et al., 2001). The acceptance of simulation-based training increases as the effectiveness of simulation is demonstrated in previous literature (Abrahamson et al., 2004; Barry-Issenberg et al., 2005; Grantcharov et al., 2004; Seymour et al., 2002). There are several deficiency in traditional surgical training which simulation-based training could directly remedy. Robert et al. stated that surgical training outperform the traditional surgical training by the extension of work hours restrictions, cost of operating room time and ethical concerns for patient safety (Roberts et al., 2006).

Most of the traditional surgical trainings rely on developing skills by practicing on real patients, thus exposing these patients to unnecessary risks. Even for recent real patients' practical surgical training, these patients are being put at risk (Barach et al., 2009; Haller et al., 2009; Ziv et al., 2003). Furthermore, practicing on real patients may limit the novice surgeons to receive consistent learning opportunities as certain crucial skills for each novice surgeon may vary depending on the exposure of different situations. Hence, instead of exposing real patients to serious risks, simulation-based training offer a safe alternative environment in which novice and professional surgeons could practice repeatedly to gain increased proficiency in complete safety (Cooke et al., 2006). Simulation-based trainings provide a unique capacity to simulate several uncommon conditions and rare surgical events. Compared to the typical surgical training, novice surgeons are allowed to facilitate self-directed learning and develop skills at their own pace rather than a simple group discussion which do not involve practical learning. Vozenilek et al. neatly emphasised the significance and potential of simulation-based surgical training by reforming the old dictum, "See one, do one, teach one" to "See one, simulate many, do one competently, and teach everyone" (Vozelinek et al. 2004).

Aforementioned, surgical simulation is introduced for the needs to complement the traditional surgical training methods. With the advances in computer technology, Virtual Reality (VR) is introduced into entertainment, experiments, flight simulators, car crashing demos and health care. VR is crucial in the surgical simulation as it combines a convincing representation of soft tissues where interaction between the end user and the virtual objects are allowed. The virtual environment (VE) increases the immersive of the learners where generated virtual object seems to be real and exist. The first surgical simulator was introduced by Robert Mann in the '60s, where a

rehabilitation application was developed to allow the medical surgeon to perform several surgical approaches for a given orthopaedic problem (Indelicato, 1995). In the ‘90s, the first generation of medical simulator applied the concept of navigation and immersion to the three-dimensional, 3D anatomical data sets, under the terminology proposed by Satava, R. (1996). The techniques based on VR are considered only on the geometrical nature of the human body. Even though with limited user interaction tools, the first generation are still fond of the interesting applications in the field of education and training.

Meanwhile, the second generation of simulators focuses on modelling the physical interaction of each anatomical structure such as the bone structures and the soft tissue. According to Delp S. and Loan J. (1995), it is crucial to model the coupling between kinematic constraints and muscle deformation for bone structures. On the other hand, necessity on modelling the soft tissue deformability under the influence of exerted forces due to surgical instruments or internal forces.

In the third generation of simulators, the functionality natures of human organs are taken into account. According to DeCarlo et al (1996), there is an interrelation between different level of simulation such as anatomy, physics or physiology as shown in Figure 2.1. In order to simulate, model and visualise the mechanism of anatomical and physiological, a reusable knowledge based with the underlying physiology—functional anatomy used to link the behaviour of the body through fundamental casual principles. For instance, suturing the liver (physical phenomena) has an influence on the blood pressure and the function of other organs or other part of the liver such as gallbladder.

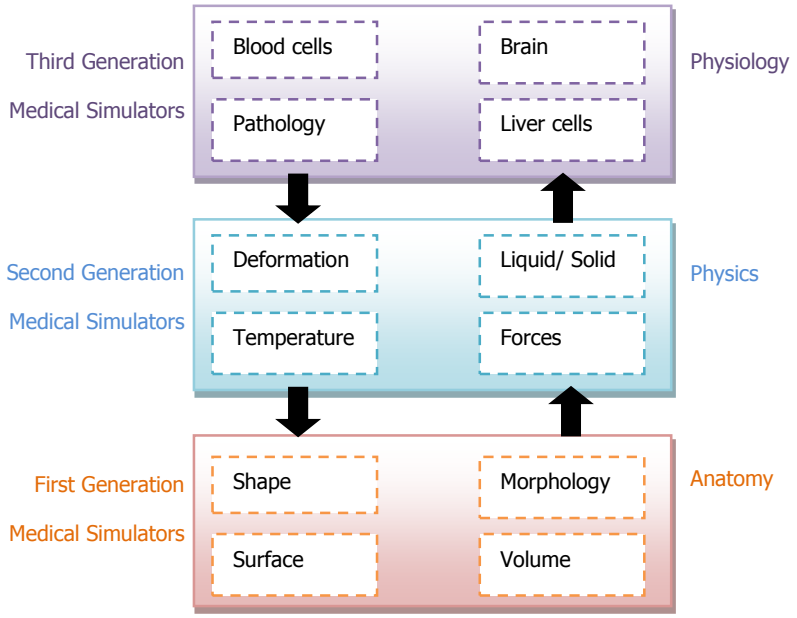


Figure 2.1 The different generations of medical simulators. (Delingette H. 1998).

The designs of a real-time surgical simulator require two major tasks: an accurate and computational efficiency deformable model to represent the soft tissue properties and a set of suitable model's parameters which is crucial in physical simulation. However, there is always a trade-off between accuracy and efficiency—the interactive computational in real-time. The existing simulators are neither fast enough for real-time application, nor can feedback for haptic rendering. The physics principles are often ignored in developing algorithms for real-time computation of deformable bodies.

As shown in Table 2-1, there are three types of applications according to the requirement. Firstly, scientific analysis in which the soft-tissue modelling focuses on validating particular hypothesis made for the design of new procedures or implants. In this case, the accuracy of deformation is essential than computational efficiency. Secondly, surgery planning which predict the outcome of surgery requires less computational efficiency (from 30s to 1h) as its allows several trials-and-error

learning. Lastly, surgery procedure training where computationally efficient took the prior (0.1s) in order to achieve real-time user interactive.

Table 2-1 The requirement of accuracy over efficiency in different simulation

Simulations	**Requirement**	
	Accuracy	**Efficiency**
Scientific analysis	✓	t >1 hour
Surgery planning	✓	*30s*< t <*1* hour
Surgery procedure training	✓	t<0.1 second

2.2 Physically-based Deformable Models

Living human soft tissues exhibit relatively high nonlinear force-feedback behaviour in which they undergo large deformations. Therefore, as a matter to design a state-of-the-art surgical simulator, realistic simulation of tissue deformation is the most crucial components (De, S. and Srinivasan, M.A., 1999). The surgical simulation involves large deformation with complex soft tissue, thus the ultimate goal of an advanced surgery simulation is to manipulate high resolution three dimensional tissue model efficiently in real-time and accurately (Yamamoto, 2011). Recent research focus is mainly concentrated on the computational models which advance in topological and geometrical representation, deformation, and three dimensional rendering of soft tissue structures. Choice of a physically realistic simulation process for such operations play a vital role in the designation of surgical simulation. Thus, an excellent deformable modelling must reflect stable forces, display realistic deformations which progress with various boundary conditions and constraint in real-time.

In general, human soft tissue properties consists of viscoelasticity, anisotropy, nonlinearity, inhomogeneous and are layered (Fung, Y.C., 1993). Although sophisticated deformable models for real-time simulation have been developed in the

past few years, integrating tissue properties into these models remain a challenge. A physically-based deformable model simulates the physical behaviour of deformable objects taking account of the internal and external forces. There are two commonly used physically-based deformable models in modelling soft tissue: the Finite Element Model (FEM) and the Mass Spring Model (MSM). Each model has its own nuances and can be applied in several different ways concerning different objectives and aims.

2.2.1 Finite Element Model (FEM)

The Finite Element Model (FEM) is a continuous linear model (continuum) which is not purely continuous since the geometrical model of the deformable object is divided into surface or volumetric elements. This discrete dividend lies in meshing step, which causes the deformation field to be only C^0 continuous across the mesh boundaries (Szabo, B. A., 1991; Zhang et al., 1997; DiMaio, S.P. and Salcudean, 2002). Often, a deformable object is represented by FEM as a finite set of elements, whereas the properties of each element are formulated, and the elements are assembled together. FEM-based simulation approximate the solutions by taking a number of simplified assumptions via numerical integration to a set of Partial Differential Equations (PDEs) in irregular grid (Nealen et al., 2005). FEM is highly accurate compared to the Mass Spring Model (MSM). It is possible to perform extensive off-line calculation where the calculation is not under the control of a central computer, particularly in simulating a small body district (Azar et al., 2001; Azar et al., 2002; Chung et al., 2008; Samani et al., 2001). However, due to the linear characteristic of FEM, it is inapplicable on nonlinear deformable object. Even though it could simulate nonlinear object accurately, it only works on a small body distinct (Srinivasan et al., 2006; Tendick et al., 2000; Webster et al., 2002). This is because FEM consists of linear model; additional linearization is embedded in the procedure to linearize the nonlinearity, thus

increase the time complexity. Therefore, high simulation frame rates are required for a relatively complex model. It is thus invalid to manipulate a real-time deformation object. (Szabo, B. A., 1991).

In fact, optimisation techniques such as condensation and pre-computation have been carried out in order to increase the time efficiency of FEM (Bro-Nielson, 1996). Throughout these techniques, some successful achievements have been obtained. In the work of Berkley et al. (2004), a virtual suturing simulation is developed by utilizing a pre-computed stiffness matrix to improve run-time performance. However, due to requirements of real-time updating time frame to overwrite the modification of stiffness matrix, the model cannot be cut or alter the mesh topology. Thus, the applicable of pre-computation on FEM is very limited and hardly play much role in improving time efficiency. Meanwhile, Wu et al. (1996) on the other hand, used a range of optimisation techniques, including some use of the GPU, to acceleration efficient shading to create a deformable soft-tissue model that can be cut.

FEM is an excellent choice when it comes to modelling the object properties accurately and realistically. In the works of Cotin et al. (1999) and Natsupakpong (2010), it showed that the FEM is stable, accurate and able to handle complex geometry. However, adapting FEM for real-time interactive cutting or deformation is non-trivial. This is because FEM could produce visually convincing static modelling but with demanding computational cost. The downside of FEM contradicted with the essential requirements of surgical simulation: realism and efficiency as FEM is a static dynamic system with demanding computational cost when soft tissues properties are taken into account.

2.2.2 Mass Spring Model (MSM)

The Mass Spring Model (MSM) is one of the most popular and favoured in modelling deformable objects (Zhang et al., 2010). It is a complete discrete model where the continuous material is discretised into mass points while the distributed interactions are discretised into springs. In other words, MSM is basically consists of a mesh of mass points link together by a network of zero mass springs (Natsupakpong, 2010). Each mass point is represented by its own position, velocity and acceleration which moves under the influence of external forces such as surgical instrument. Generally, the implementation of MSM is easy especially in representing complex models either structured or unstructured meshes (Hammer et al., 2011). It thus requires relatively small computational cost, less storage and bandwidth since the model properties are only stored at the mass point positions. MSM is able to deal with both large displacement and deformation while allowing real-time simulation. Hence, MSM is better suited for representing nonlinear soft tissue which requires large deformation and topology modification simulated in interactive speeds (Baumann et al., 1996; Meseure & Chaillou, 1997; An, 2011).

The earlier works on MSM in surgical simulation are mostly limited to two dimensional (2D) modelling and three dimensional (3D) rigid object modelling. Terzopoulos et al. (1987) are one of the first researcher, who attempted to utilize MSM to simulate the nonlinear elasticity of soft tissues. Ever since that, more researchers pursue to implement MSM in representing soft tissue deformations due to its ease of implementation (Xu et al., 2010). For instance, Cover et al. (1993) applied MSM to simulate deformation of a gallbladder that is related to liver surgery. Kuhnapfel and Neisius (1993) demonstrated a simulation of endoscopic surgery based on MSM. The simulation was driven by using a non-force-feedback motion sensor instruments. Later, Nedel and Thalmann (1998) employed MSM to simulate muscle deformation.

The muscle's shape was deformed using a MSM and angular springs were used to control the volume in the muscles during deformation. Meseure et al. (1997) simulated the dynamic behaviour of human organ based on the spring surface mesh. In their simulation, the model was filled with a virtual rigid component which did not have interaction with the environment and it provided a structure with a rigid behaviour. Meanwhile, MSMs are also applied in craniofacial surgery, Tendick et al. (2000) developed a system that is able to simulate bone cutting and bone alignment with integrated interactive collision detection. In addition, it enabled the soft tissue cutting via the haptic device. Four years later, this model was able to handle the material properties ranging from stiff to fluid-like behaviour (Teschner et al., 2004). Maciel et al. (2003), on the other hand, utilized diagonal springs in MSM to simulate soft tissues. It was the generalized MSM, which named as molecular model. The usage of diagonal springs were to avoid tetrahedral meshes. Even though it offers desired results, computing the spring coefficients remains an issue.

Limitations on simulator's ability to replicate correct tissue properties such as those described above are imposed by the requirement for very precise data on the physical characteristics of organs and tissues in vivo. Actual data of this type are very sparse. For the most part, tactile characteristics of simulated organs and tissue are based on best guesses as to these characteristics. To address this paucity of information, surgical simulator developers have used solutions to measure forces that approximate those associated with surgical manipulator by eliciting physical feedback from mannequins or benchmark models. Alternatively, developers may focus on the purely visual cues associated with tissue interactions such as the observable object deformation with instrument manoeuvres.

Due to the capabilities of MSM to model the cutting and suturing, MSM has been widely utilized in modelling soft tissue deformation compared to FEM (Hu, 2006; Xu et al., 2010). Contradictory, although MSM seems to be visually plausible, it is difficult to integrate the physical properties data such as the spring stiffness and damping coefficients. This is because the spring coefficients which describe the bio-mechanism properties of soft tissues such as Young's Modulus and Poisson Ratio do not have direct relationship with MSM (Choi, 2010). Hence, certain constraints for a continuous systems are sometimes tremendously hard to be transformed into the particular formulation of the corresponding discrete system (Etheredge, 2011). Moreover, higher stiffness coefficient value leads to numerical instabilities. Therefore, the model's parameters are typically tuned manually until the desired visual realism is obtained. It results that the selection of model's parameters becomes the fundamental drawback of MSM.

Despite of the fact that the MSM is computationally efficient and relatively simple to augment the deformation of soft tissue, several studies have been done in order to tackle the fundamental issue of MSM. Van Gelder (1998) proposed a method to choose the spring stiffness in order to minimize inhomogeneity in deformation. Radetzky et al. (2000) combined both fuzzy logic and neural networks in order to determine the stiffness and other spring parameters. Some researchers proposed optimization methods for tuning individual spring stiffness methods throughout a given mesh to approximate specific behaviour of the tissue surface (Bianchi et al., 2004; Lloyd et al., 2007).

In spite of the many reported applications of the MSM, there is little mention in the literature about how to assign values to the parameters of the springs (Jojic, N and Huang, T. S., 1997; Van Gelder, A., 1998; Bhat et al., 2003; Maciel et al., 2003). It is

well known that due to spatial discretization, the MSM can only be an approximation in some aspects to the continuum object. In case of cloth simulation for example, Van Gelder compared the MSM and the FEM and concluded that an exact simulation using the MSM is impossible (Van Gelder, A., 1998). This may explain why there is little research done to optimize parameters of the MSM. Although the MSM cannot give an exact simulation, some good parameter sets do give better simulation results than others. The criterion for better simulation appears ambiguous and hard to define. Current implementation of the MSM for a lot of applications usually chooses the parameter values on trial and error basis. That is a very tedious and time-consuming procedure. This is because without guidance, we cannot expect to obtain a good parameter set for the MSM to fully exploit its potential. So it is essential to develop some algorithms to optimize parameters for the MSM.

The first traditional attempt is analytically or numerically optimization methods. The second attempt—heuristically optimization, on the other hand, is considered as an alternative methods in optimizing the physical parameters. In the first approach, the classical optimization methods focus on deriving analytical expressions which could directly be used in optimization processes. For instance, in the works of Van Gelder, A. (1998), a formula for the spring coefficient on how to map biomechanical parameters into spring coefficients for 2D membranes is presented. The expression can be used to approximate the behaviour of linear elastic material. However, this formula is only sufficient for triangle meshes. Meanwhile, Maciel et al. (2003) suggested a generalized mass spring model for nonlinear springs in a hexahedral mesh where the stiffness parameter depends on the current angles between the springs. Each iteration even small or local changes of the model structure requires re-computation, thus this method is time consuming and applicable for certain mesh topology. Lloyd et al.

(2007) on the other hand, derived analytically expressions for the spring parameters from an isotropic linear elastic reference model such as triangle, rectangle and tetrahedron meshes. Result showed that the MSM deformations were very close to the FEM reference model. Unfortunately, the results are limited by the fact that the formulas are only valid for specific Poisson Ratios. Furthermore, classical optimization such as Gradient Method, fails to reach optimal solution as it usually trapped in local minima.

Black box/ Grey System

Due to the limitations on the analytical optimization methods, several alternative methods have been developed and applied in determining the setting of MSM parameters. Deussen et al. (1995) suggested a methods based on Simulated Annealing (SA) which allows the generation of systems with definite mechanic behaviour. This method has succeeded in determining an adequate mass distribution. Meanwhile, Andreas et al. (1999) showed that the elastodynamic shape model based on neuro-fuzzy system can be successfully used for simulation of deformable shape such as soft tissues. Besides, he mentioned that by applying artificial neural network (ANN) alone is not sufficient as incisions are unable to be made during simulation process. In fact, this is shown in the work of Petriu et al. (2007) and Zhong et al. (2007) where neural network modelling techniques are used for the real-time 3D rendering. These techniques are able to learn nonlinear behaviours from a limited set of measurement data or mimic the deformation of soft tissues. However, human intelligence in tuning of parameters or time consuming training is required. Thus, by combining the ability of learning of neural networks and the ability of interpretability in fuzzy system, the stiffness and other spring parameters could be determined automatically. Bianchi et al. (2003; 2004) on the other hand, proposed a method based on Genetic Algorithm (GA)

to identify the mesh topology of MSM. Their works inspired that MSM parameters can also include mesh topology in which the optimization are done together with the stiffness values. However, the major difficulty in this approach is the specification for the MSM parameters are not straightforward. When it is extended towards 3D approach, larger meshes increases the number of springs, the more rugged it will be therefore there is some difficulties for GA to converge. Meanwhile, in the work of Zerbato et al. (2007), GA is employed to solve search and minimization problems of spring calibration. However, time requirement of the algorithm and the difficulty in obtaining in-vivo measurement for the human organ deformation, pure GA is hard to apply and computational time is needed to improve. Apart from that, Vollinger et al. (2009) applied GA in optimizing the deformation of triangle and tetrahedral mesh models. Unfortunately, there exists some limitation to certain scope of load with conditions. Furthermore, Xu et al. (2009) presented an automatic method based on SA algorithm in determining the parameters. Kyriakos et al. (2010) shown that there is also multilevel optimization algorithm for the solution of computationally expensive optimization problem. Despite of the shortages of each optimization algorithms, combinatory optimization algorithms are introduced in order to overcome the trade-off for each shortage. For instance, in the work of Choi KS (2010), the hybrid algorithm which combines SA into GA shows advantageous over pure SA and pure GA. The virtual tissues and organs are able to deform autonomously in a way similar to that achieved mathematically more accurate FEM but at the speed of computationally efficient MSM.

2.3 Fuzzy Inference System

Fuzzy inference system (FIS) has come to ages. We have witnessed rapidly growing research interest in fuzzy inference system in recent years. There have been numbers

of successful applications (Kim C. J., 1997). There consists of four elementary components in a FIS which includes the knowledge base, fuzzification interface, reasoning mechanism, and defuzzification interface as shown in Figure 2.2 (Lee, 1990a; 1990b).

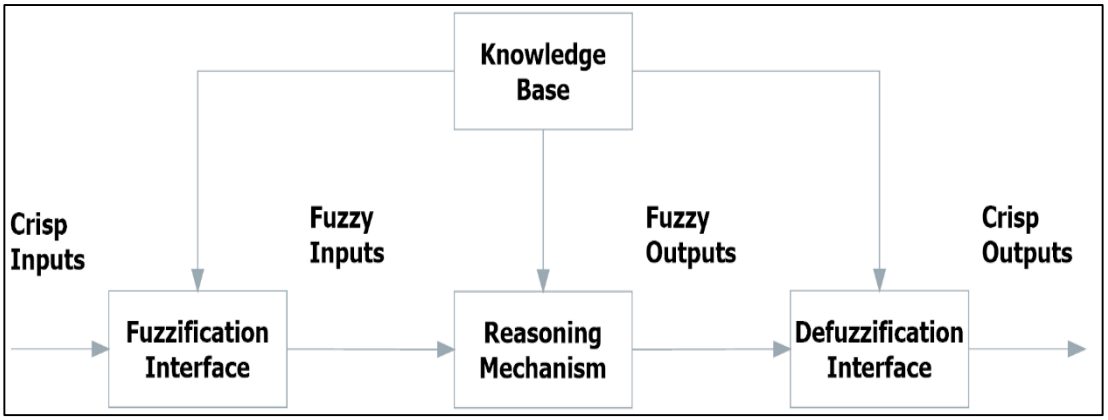


Figure 2.2 The elementary components of a fuzzy control system. (Abstracted from Feng, 2006: A survey on analysis and design of model-based fuzzy control system.)

In a fuzzy inference system, all the linguistic knowledge is contained in the knowledge base and it comprises of rules and database. The rules are made up from the available knowledge. Whereas the databases consist of a bunch of crisp sets which defines the membership functions based on the rules. The reasoning mechanism is the central part of a FIS which performs fuzzy decision making upon the rules and given facts to derive the outputs. In FIS, the inputs could either be fuzzy or crisp (exact value) inputs, however, the output shall always be crisp value in order to be implemented into the real-world system. The role of the fuzzification interface (fuzzifier) is basically mapping the exact-valued space to a fuzzy space, and conversely, the defuzzification interface (defuzzifier) defines a mapping from a fuzzy space over an output universe of discourse to an exact-valued space. In other words, the fuzzifier transforms a crisp

sets to fuzzy sets while the defuzzifier transforms the fuzzy outputs to a set of crisp outputs.

However, interestingly, fuzzy inference system shares a number of similarities with the neural network (NN) control system (Feng, 2006). For instance, both FIS and NN are generally model-free control systems in which both of them are able to store knowledge and utilize it for decision making. Moreover, both FIS and NN provide robustness of control to certain extend with respect to system variations and external disturbance. The difference between FIS and NN is the way they acquire knowledge. NN obtains knowledge through data learning or training, whereas FIS obtains qualitative and imprecise knowledge through expertise or operator's perspective. The way of NN acquiring data could be an advantage as the data is collected automatically through the iterative training progress, but it could be a disadvantage when the trained data set does not fulfil the constraint interest. Thus, by taking the advantages of both control systems, the so-called neuro-fuzzy control system is introduced. As two of these control systems complement to each other; that is NN provides data learning capabilities and efficient parallel computation whereas FIS provides a platform for expertise knowledge representation. Several researchers have been implementing neuro-fuzzy control system in enhancing the flexibility, data processing capability, and adaptability of a control system (Boroushaki et al., 2003; Da and Song, 2003; Farag et al., 1998; Frey and Kuntze, 2001; Jang, 1993; Jang and Sun, 1995; Kiguchi et al., 2004; Lazzerini et al., 1999; Li and Lee, 2003; Li et al., 2004; Lin and Hsu, 2004; Lin and Lee, 1994; Liu et al., 2004; Mar and Lin, 2001; Melin and Castillo,2001; Munasinghe et al., 2005; Tani et al., 1996; Wang and Lee, 2003; Wang et al., 2002; Wang et al., 2004; Zhang, 2005).

Even though neuro-fuzzy control system does not require any form of mathematical model in order to keep the system to be under controlled, there is a major limitation in it. The neuro-fuzzy control system is only systematically stable in the context of a closed-loop control system. Therefore, prevailing research efforts have been focused on pure fuzzy inference approaches such as Takagi-Sugeno FIS, Mamdani FIS, Type-2 FIS and Interval Type-2 FIS (Feng, 2006). Mamdani (1974; 1975) and Assilian (1975) have developed the first FIS to control a small steam engine. As predicted by Zadeh, L. A. (1996), FIS which is also known as Computing Words (CW) has evolved into a basic methodology in its own right with wide-ranging ramifications on both basic and applied levels. A great variety of applications have been discovered from previous works as shown in Table 2-2.

Table 2-2 Previous works on applications based on FIS

Application	References
Power Systems	Abdelazim and Malik, 2003; Flores et al., 2005; Guesmi et al., 2004; Ko and Niimura, 2002
Telecommunications	Aoul et al., 2004; Chen et al., 2003; Chen et al., 2003a; Kandel et al., 1999; Lee and Lim, 2001; Zhang and Phillis, 1999
Mechanical or Robotic Systems	Bai et al., 2005; Batorone et al., 2004; Boukezzoula et al., 2004; Chang and Chen, 2005; Hagras, 2004; Hong and Langari, 2000; Hwang and Kuo, 2001; Kiguchi et al., 2004; Kim, 2004; Li et al., 2004; Li et al., 2001; Mannani and Talebi, 2003; Santibanez et al., 2005; Sun and Er, 2004; Tsourdos et al., 2003; Wai and Chen, 2004; Wang and Lee, 2003; Xiao et al., 2004; Yang et al., 2004
Automobile	Bonissone et al., 1995; Hagras, 2004; Huang and Lin, 2003; Lin and Hsu, 2003; Mar and Lin, 2001; Murakami and Maeda, 1985; Niasar et al., 2003; Sugeno and Nishida, 1985
Industrial or Chemical Processes	Campello et al., 2004; Chen and Liu, 2005; Frey and Kuntze, 2001; Horiuchi and Kishimoto, 2002; Juang and Hsu, 2005; Kickert and Van Nauta Lemke, 1976; King and Mamdani, 1977; Kornblum and Tribus, 1970; Larsen, 1980; Mamdani, 1974; Ostergaard, 1977; Seker et al., 2003; Sugeno, 1985; Tani et al., 1996; Tong et al., 1980; Umbers and King, 1980
Aircrafts	Chiu et al., 1991; Farinwata et al., 1994; Kadmiry and Driankov, 2004; Larkin, 1985
Motors	Barrero et al., 2002; Guillemin, 1996; Kim and Lee, 2005), medical services (Kwok et al., 2004; Seker et al., 2003; Zheng and Zhu, 2004

Consumer Electronics	Haruki and Kikuchi, 1992; Kumar, 2005; Lee and Bien, 1994; Lee et al., 1994; Nakagaki et al., 1994; Smith, 1994; Takagi, 1992; Wu and Sung, 1994
Chaos Control	Chen et al., 2005; Lian et al., 2001
Nuclear Reactor	Bonissone et al., 1995, Munasinghe et al., 2005

Over five decades, the fuzzy inference systems (FISs) have been widely expanded as shown in Table 2-2, several fuzzy control applications have been developed. In fact, FIS has proven to be a successful control approach to many complex systems. It has been suggested as an alternative approach to conventional control techniques in many cases (Zadeh, 1996). Taylor, a research engineer who replaced the old 'threshold' method with fuzzy logic algorithm claimed that the key to fuzzy logic's success lies in its flexibility—its ability to cope with imperfect input and adapt as the situation changes (Zadeh, 1996). The fuzzy algorithm "informs" the machine how to control the system instead of learning by observing the actions of a human operator. The main recipe in fuzzy approaches is the availability of knowledge which consists of a specific domain being a field or area of expertise. Therefore, the generally well-structured domain could reduce the complication of calculations. For instance, the work of Pawlus et al. (2013) showed that the fuzzy model's output could be adjusted to improve the model's fidelity without complicated calculations. We thus could implement the fuzzy approaches into MSM in order to sort the selection of parameters based on the available liver stiffness data obtained from FibroScan®. The novelty of this research is that for liver modelling in particular, no specific contributions in the literature exist reporting on real-time knowledge-based fuzzy mass spring model for surgical simulation.

2.4 Graphical Hardware and Programming Optimisation

Due to the rapid improvement in the graphical hardware nowadays, researchers attempt to exploit the processing power of Graphical Processing Units (GPUs) and

Central Processing Units (CPUs) for MSM and FEM. The CPU is known as the brain of the PC. Architecturally, the CPU has only few cores with lots of cache memory which could only handle few threads in one cycle (NVidia CUDA, 2012). In each cycle, CPU obtains data from registers, does an operation, and send the result back (Mikes, 2013). In contrast, the GPU, as the soul of a PC which composed of hundreds of cores, is able to handle thousands of threads simultaneously (Palacios and Triska, 2011). GPU is potential to accelerate the runtime of thousands threads by 100x faster than a CPU alone (Patterson, 2009). Thus, GPU is more powerful and cost-efficient as compared to CPU.

2.4.1 Architecture of GPU and CPU

As shown in Figure 2.3, the primary difference between a GPU and CPU is that the hardware architecture of a GPU contains many Arithmetic Logic Units (ALUs) whereas a typical CPU contains fewer components for cache memory and flow control. The main reason GPU is created in the first place is to optimise one of the most crucial calculations in graphics—Matrix Manipulation. In the case of CPU implementation, the multiplication in a matrix manipulation has to be computed one-by-one in sequence. Meanwhile, the GPU is able to complete the multiplication simultaneously in one go. Thus, the characteristic of a GPU is computationally intensive, highly parallel computation which is well-suited for graphical rendering.

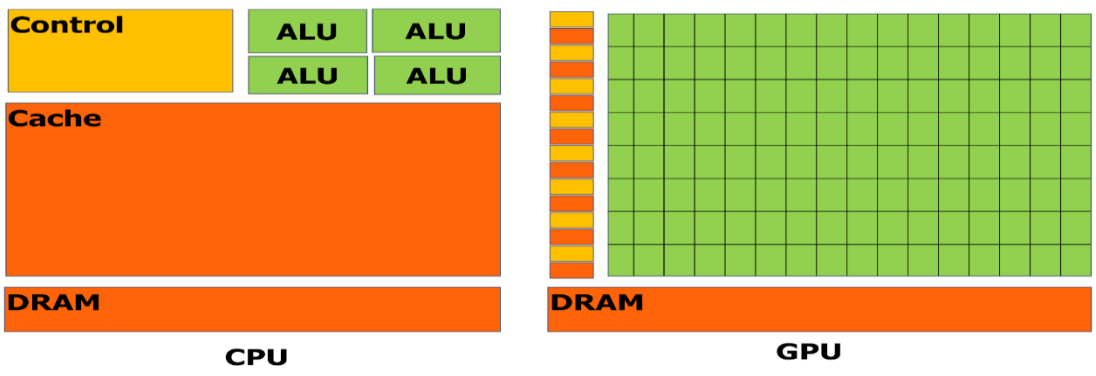


Figure 2.3: The hardware architecture of CPU and GPU where GPU devotes more transistors to data processing. (Abstracted from: CUDA C Programming Guide, 2013).

2.4.2 GPU implementation in Mass Spring Model

The parallel capability of GPU to process multiple streams of data in the same program is beneficial to the modelling of MSM which requires updating of all springs and masses iteratively in the same manner. Therefore, since the last two decades, GPUs have evolved interesting demands in optimising the graphical rendering for MSM and FEM modelling. In early '00s, Georgii and Westermann (2005) shown that the GPU implementation in dynamic MSM application comes at an additional advantages in computational efficiency and realistic visualisation. They have also exploited features of recent graphics accelerators to simulate spring elongation and compression in the GPU, saving displaced masses in graphics memory. This approach allows for interactive simulation, high complexity rendering and enables the display of internal properties of the deformed body. The deformed object is rendered directly without the need to transfer the data to the CPU and sending them again to the GPU for rendering purposes. The GPU implementation on MSM has shown significant real-time performance in the work of Tejada, E. and Ertl, T (2005).

2.4.3 Computer Unified Device Architecture (CUDA)

In year 2008, Computer Unified Device Architecture (CUDA) was created to expose the GPU's powerful parallel processing capabilities without any Graphics knowledge or experience. CUDA serves as a platform which provides a simple path for programmers to easily construct programs in CPU (host) for execution by the GPU (device). Rasmusson et al. (2008) have proven that CUDA is a very interesting new platform to compute MSM in a surgical simulator. Ever since that, CUDA is becoming the limelight in the implementation of MSM in several approaches such as tissue cutting based on GPU and CUDA in surgical training, 3D tissue deformation (Zhang

et al., 2010), CUDA-based FEM for elasticity simulation (Dick et al., 2011), laparoscopic surgical training system (Zhang et al., 2013) and so on (Etheredge, C.E., 2011; Rasmusson et al., 2008).

However, there is certain limitation such as overhead fast simulations (Zhang et al., 2010), parallelism techniques on GPU clusters (Dick et al., 2011) and realism of simulation due to the drawback of MSM (Zhang et al., 2013; Campus et al., 2013). Nevertheless, Leon et al. (2010) affirmed that the error in CUDA framework is not sufficiently large to affect the visual perception of the surgeon during the simulation and is relatively stable. According to all relevant literature studies, for liver modelling in particular, there is no specific contributions in the literature exist reporting on real-time knowledge-based fuzzy mass spring model for surgical simulation.

2.5 Summary

In this chapter, a brief history of surgical simulation is presented. Two main physically based models have been presented: Finite Element Model (FEM) and Mass Spring Model (MSM). Each of them has its own advantages and disadvantages. FEM models are realistic but it is computationally intensive thus causing delay in real-time (Sala et al., 2011a). MSM, on the other hand, is easy to be implemented due to its discrete model in nature. MSM is featured with low computational complexity thus making the model to compute efficiently. However, MSM lack of accuracy due to the selection of parameters.

Several approaches have been developed in order to obtain the accurate parameters for the MSM. However, the approaches are either limited to certain mesh topology or require manual tuning. From previous literature studies, knowledge-based fuzzy inference system has shown potential in determining the parameter of MSM. The

computational efficiency is improved with advance graphical hardware and CUDA platform.

Chapter 3 *Methodology*

3.1 Overview

In this research, the proposed research methodology comprising of the following elements:

- Mass Spring Model (MSM)—to model human liver as a collection of mass points linked by a network of springs.
- Knowledge-based Fuzzy MSM Parameters Controller (FMPC)—to obtain the corresponding stiffness coefficients regarding different liver illnesses.
- Utilization of Computer Unified Architecture (CUDA)—to utilize as a platform or framework for GPU and CPU distribution of computational works.

The interrelation between the Central Processing Unit (CPU) and Graphical Processing Unit (GPU) throughout the simulation is illustrated in Figure 3.1. The simulation begins by sending sequential matters (initialize of point cloud data and generation of mesh model) to the CPUs while the parallel matter (Euler integration) to the GPUs as shown in Section 3.4. In other words, the data is initialized on the host, CPU and then copy the data to the device, GPU for the computation of the next deformed positions. Meanwhile, the computation of stiffness coefficient based on FIS will be done in MATLAB r2014a on the host, CPU.

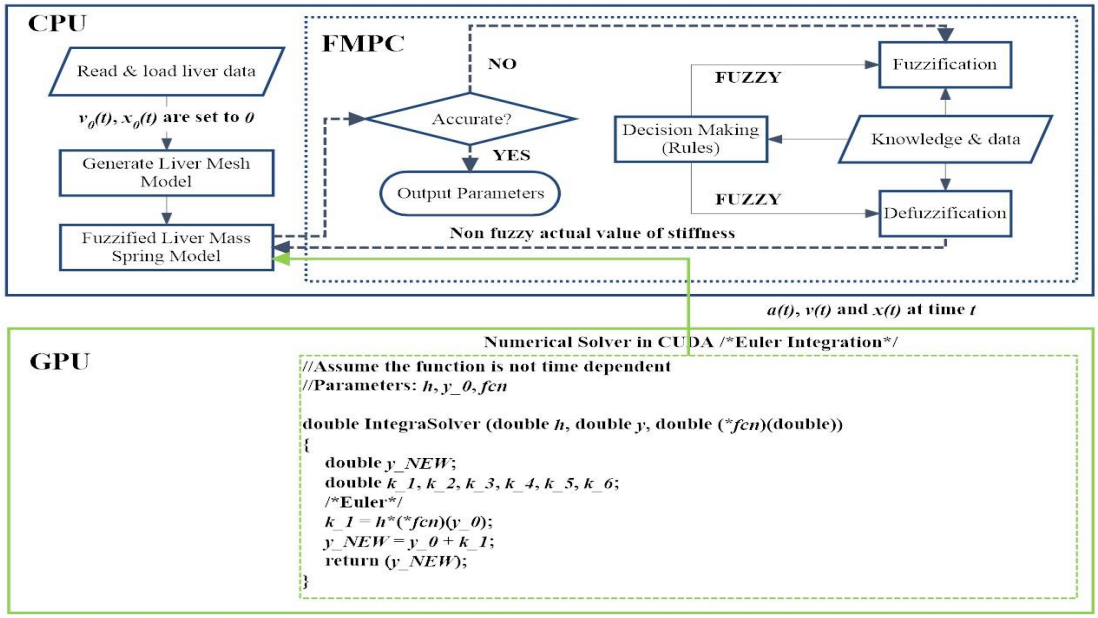


Figure 3.1 The workflow of interrelation between the Central Processing Unit (CPU) and Graphical Processing Unit (GPU) throughout the simulation.

3.2 Mass Spring Model (MSM)

3.2.1 Formulation

The 3D human liver modelling is based on the MSM as shown in Figure 3.2. The liver is modelled as a collection of mass points linked by a network of springs which include structural springs, shear springs and flexion springs. The springs linking each mass points exert forces on neighbouring points when a mass is displaced from its rest position.

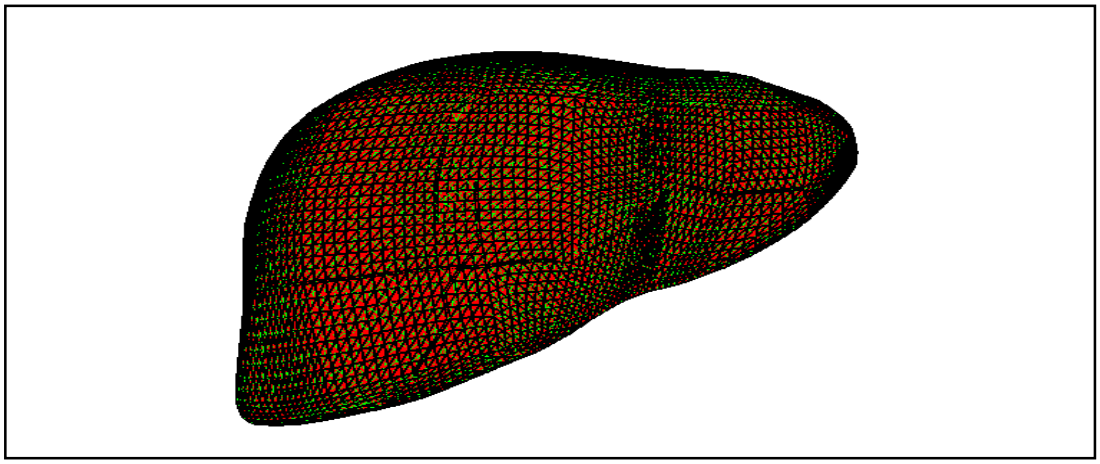


Figure 3.2 The 3D human liver modelling based on MSM.

The governed MSM equation is derived by using Newton's 2nd Law of Motion, F_N and Hooke's Law, F_s as follows:

$$F_N = m\ddot{x}_{ijk}(t) \tag{3.1}$$

$$F_s = -k \cdot x_{ijk}(t) \tag{3.2}$$

where m and k are the mass and stiffness coefficient, respectively. While $\ddot{x}_{ijk}$ and x_{ijk} denote the acceleration and displacement of a control mass point in 3D space.

The summation of the external force is obtained by adding both Eq (3.1) and Eq (3.2):

$$F_N + F_s = F_{ext} \tag{3.3}$$

$$m\ddot{x}_{ijk}(t) = -k \cdot x_{ijk}(t) + F_{ext} \tag{3.4}$$

where F_{ext} represents as the external forces. However, in actuality human liver deformation, there exists some degree of damped deformation caused by friction forces or internal forces. Therefore, in this research, the Mass Spring Modelling is considered as a damped harmonic oscillation and the damping coefficient, γ is taken account in the governing MSM equation as follows:

$$m\ddot{x}_{ijk}(t) + \gamma \cdot \dot{x}_{ijk}(t) + k \cdot x_{ijk}(t) = F_{ext} \tag{3.5}$$

where m, γ and k are the mass, damping coefficient and stiffness coefficient, respectively. While $\ddot{x}_{ijk}$, $\dot{x}_{ijk}$ and x_{ijk} denote the acceleration, velocity and displacement of a control mass point in 3D space. F_{ext} represents the external forces.

The generalized MSM is a collection of mass points fixed connectivity that are consists of springs. The corresponding pseudo code for structural and flexion springs connections of mass points are presented as Figure 3.3 and Figure 3.4. Meanwhile, the alternative form of representing the Mass Spring Model in Simulink is shown in Figure 3.5.

```
Algorithm 1: Structural Springs

//for every_point i, every_point < total_point
For all point i in total_point do
      //for every_TriXYZ, every_triangle < total_TriXYZ
      For all triangles TriXYZ in total_TriXYZ do
      //if the data in TriXYZ[1][2][3] = i
      If TriXYZ[1][2][3] = i do
            //If the structure spring is not linked yet
            If structure_spring is not linked do
                  LINK structure_spring [1][2]
                  LINK structure_spring [2][3]
                  LINK structure_spring [1][3]
            End if
      End if
      End for
End for
```

Figure 3.3 Pseudo code of structural spring.

```
Algorithm 1: Flexion Springs

//for every_triangle TriXYZ, every_triangle < total_TriXYZ
For all triangle TriXYZ in total_TriXYZ do
       //for every_triangles TriXYZ, every_triangle < total_TriXYZ
       For all triangles TriXYZ in total_TriXYZ do
       //if the first TriXYZ is not equal to second TriXYZ
       If TriXYZ_1 != TriXYZ_2 do
              //if in column A, both data are same
              If TriXYZ_1[A] = TriXYZ_2[A] do
                     //if in column C, both data are same
                     If TriXYZ_1[C] = TriXYZ_2[C] do
                            //if there is no flexion spring linked
                            If flexion_spring is not linked do
                                   LINK flexion_spring [2][5]
                            End if
                     End if
              End if
       End if
       End for
End for
```

Figure 3.4 Pseudo code of flexion spring.

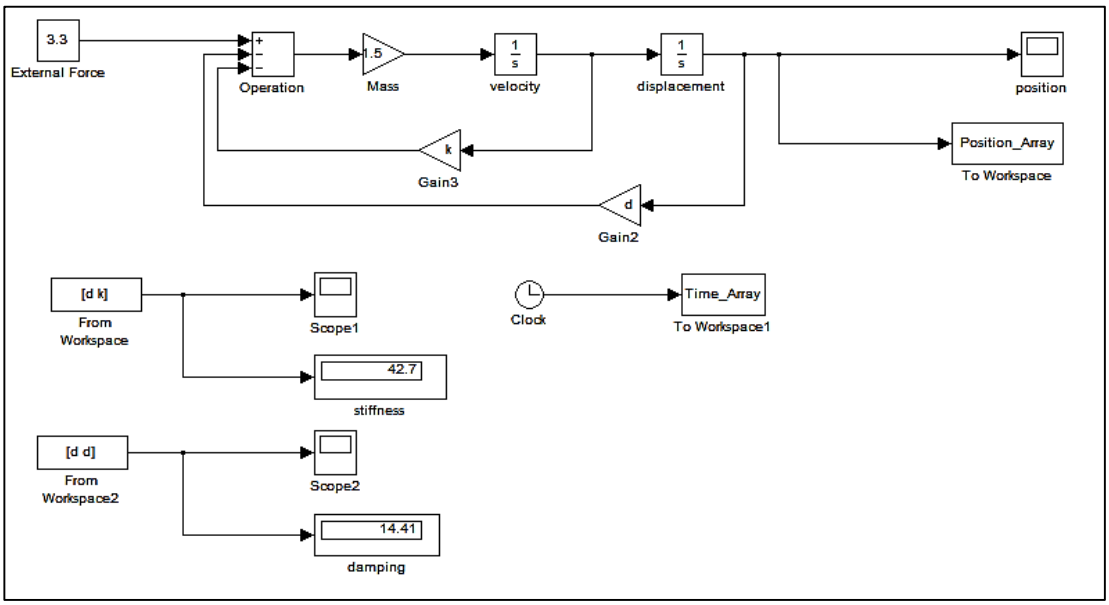


Figure 3.5 Block diagram of MSM in Simulink.

3.2.2 Determination of MSM parameters

There are two major properties for each of the springs: stiffness coefficient (k), and the damping constant (γ). In general, the measurement of resistance for a deformable elastic body towards the force with the same degree of freedom is known as stiffness coefficient while the damping coefficient is used to reduce the amplitude of vibration.

The stiffness coefficient is the resistance of a deformable object when a force is applied along a given degree of freedom and a set of loading points and boundary conditions are prescribed to the deformable object. In this research, the stiffness coefficient will be determined in FMPC by implementing the available knowledge and data from previous medical research. The detailed determination of stiffness coefficient by using the FMPC will be further discussed in the next section.

Damping constant, on the other hand, is the capacity built into a biomechanical or electrical device to prevent excessive correction and the resulting instability or oscillatory conditions. The derivation of damping coefficient is directly related to the damping ratio and natural frequency. Assume that the deformation of human liver is a damped harmonic oscillation, we could obtain the damping constant by applying Eq (3.6) as follows:

$$\gamma = \frac{2\varsigma k}{w} \tag{3.6}$$

where γ, ς, k and w denote the damping constant, damping ratio, stiffness coefficient and natural frequency, respectively. Above-mentioned, the behaviour of the system depends relatively on the natural frequency, w and the damping ratio, ς. Knowing that when $\varsigma \to 0$, the system is basically in undamped condition. Meanwhile, when $\varsigma < 1$, the system is underdamped in which the friction force is directly proportional to the velocity of the object.

When $\varsigma > 1$, the system is overdamped. Lastly, when $\varsigma = 1$, the system is critically damped in which the system return to equilibrium without any oscillation. In this research, we assumed that the human liver deformation is an underdamped system, thus the damping ratio is set to be 0.9. This is because when $\varsigma < 0.9$, the resulting damping constant does not fulfil the value in the range of benchmark damping constant

as shown in Table 3-1. Apart from that, based on the resulting dependency of damping ratio, we could not set $\varsigma = 1 \quad or \quad > 1$ as it will cause the system to become critically damped.

The ideal human liver parameters abstracted from the dissertation of Kerdok (2006) are shown as below:

Table 3-1 The range of ideal human liver parameters (Kerdok, 2006).

Parameters	**Range**
Mass (kg)	1.2-1.5
Stiffness (kPa)	3.2-8.5
Damping (N/m^2)	4.5-31.6

The provided information in Table 3-1 is the benchmark parameters for a healthy normal human's liver. Liver surgical simulation are used for either surgery procedure training or surgery planning, therefore, the context of an advanced liver surgical simulation shall extend to enable the users to tune the stiffness coefficient based on the patient's ailments, respectively.

3.3 Knowledge-based Fuzzy Mass Spring Model Parameters Controller (FMPC)

This section presents the main ideas underlying the Fuzzy MSM Parameters Controller (FMPC). The basic flow of a fuzzy inference system (FIS) is shown in Figure 3.6. Generally, a FIS is built up by three components—rules, database and reasoning mechanism. The rules consist of a collection of available linguistic knowledge. Whereas the database consist of a bunch of crisp sets which defines the membership functions based on the rules. Here, the databases are made up of the sources obtained from the past research based on FibroScan®. The reasoning mechanism performs fuzzy decision making upon the rules and given facts to derive the outputs. In FIS, the inputs could either be fuzzy or crisp (exact value) inputs. However, the output shall always be crisp value in order to be implemented into the real-world system.

Mathematically, assume that we have two crisp sets (inputs), A and B where A is a crisp set of x, denoted as $\mu_A(x)$ and B is a crisp set of y, denoted as $\mu_B(y)$. Then, by applying the IF-THEN knowledge-based rule as a decision making mechanism, such that "If x is A and y is B then z is C".

We have, the two-input single-output FIS with the intersection (AND) relationship between crisp sets A and B as follows:

$$R_z(x,y): A(x) \cap B(y) \rightarrow C(z)$$
$$\therefore \mu_{R_Z}(x,y) = f(\mu_A(x), \mu_B(y)) \tag{3.7}$$

Next, the crisp sets is fuzzified by three different fuzzy approaches. Thus, we have the following minimum operator such that

$$R = A \cap B = \min[\mu_A(x), \mu_B(y)], x \in X, y \in Y \tag{3.8}$$

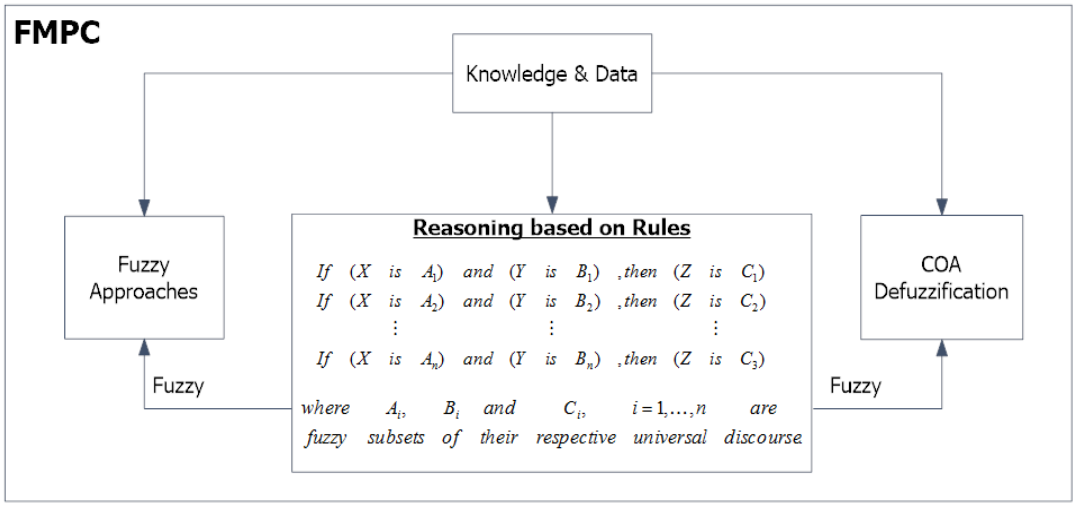


Figure 3.6 Conceptual Diagram of FMPC Fuzzy Inference System.

3.3.1 Reasoning Knowledge to Rules

In this research, the databases which are used as knowledge-based rule are obtained from previous medical research. According to the medical literature review, the corresponding values of stiffness coefficient are obtained by using Transient Elastography (TE)—FibroScan® or liver biopsy techniques. Both of these techniques

are considered as accurate and non-invasive. The FMPC consists of five inputs and one output variables. These variables are described with the corresponding membership functions (MFs) in the next chapter. The input variables include Hepatitis C Virus (HCV), Hepatitis B Virus (HBV), Alcoholic Liver Disease (ALD), Chronic Liver Disease (CLD) and Non-Alcoholic Fatty Liver Disease (NAFLD). Meanwhile, the output variable referred to the corresponding liver fibrosis staging, F0-F1: absent/ mild fibrosis; F2: significant fibrosis; F3: severe cirrhosis; F4: cirrhosis.

Thus, for simplicity, let input A be the column vector and output B be the row vector, as follows:

$$A = \{H\underline{C}V, H\underline{B}V, \underline{A}LD, C\underline{L}D, \underline{N}AFLD, B-C, A-C, L-C, A-B, L-B, N-B\}$$

$$B = \{H\underline{C}V, H\underline{B}V, \underline{A}LD, C\underline{L}D, \underline{N}AFLD\}$$

Next, by using Eq (3.8), the rules matrix as shown in Table 3-2 is built. Notice that, in Table 3-2, if there exists "AND" intersection between the input and output $\mu_A(x)$ and $\mu_B(y)$, it is denoted as 1, otherwise. After the rule matrix of the corresponding inputs has been built, we proceed to the reasoning of knowledge into rules by using the Fuzzy Logic Toolbox and the Generalized Fuzzy System (GFS) in MATLAB r2014a.

Table 3-2 The "AND" intersection between the inputs

INPUTS	**HCV**	**HBV**	**ALD**	**CLD**	**NAFLD**
HCV	1	0	0	0	0
HBV	1	1	0	0	0
ALD	1	1	1	0	0
CLD	1	1	1	1	0
NAFLD	1	1	1	1	1
B - C	0	0	0	0	0
A – C	0	1	0	0	0
L – C	0	1	1	0	0
N – C	0	1	1	1	1
A – B	0	0	0	0	1
L – B	0	0	1	0	1
N – B	0	0	0	1	0
A–C–B	0	0	0	0	0
L–C–B	0	0	1	0	0
N–C–B	0	0	1	1	0

3.3.2 Construction of Mamdani Fuzzy Inference System (M_FIS)

The architecture of Mamdani Fuzzy Inference System (M_FIS) is illustrated in Figure 3.7, it consists of five different phases: ***Fuzzification***, ***Product***, ***Implication***, ***Aggregation*** and ***Defuzzification***. M_FIS is constructed through MATLAB r2014a Fuzzy Logic Toolbox.

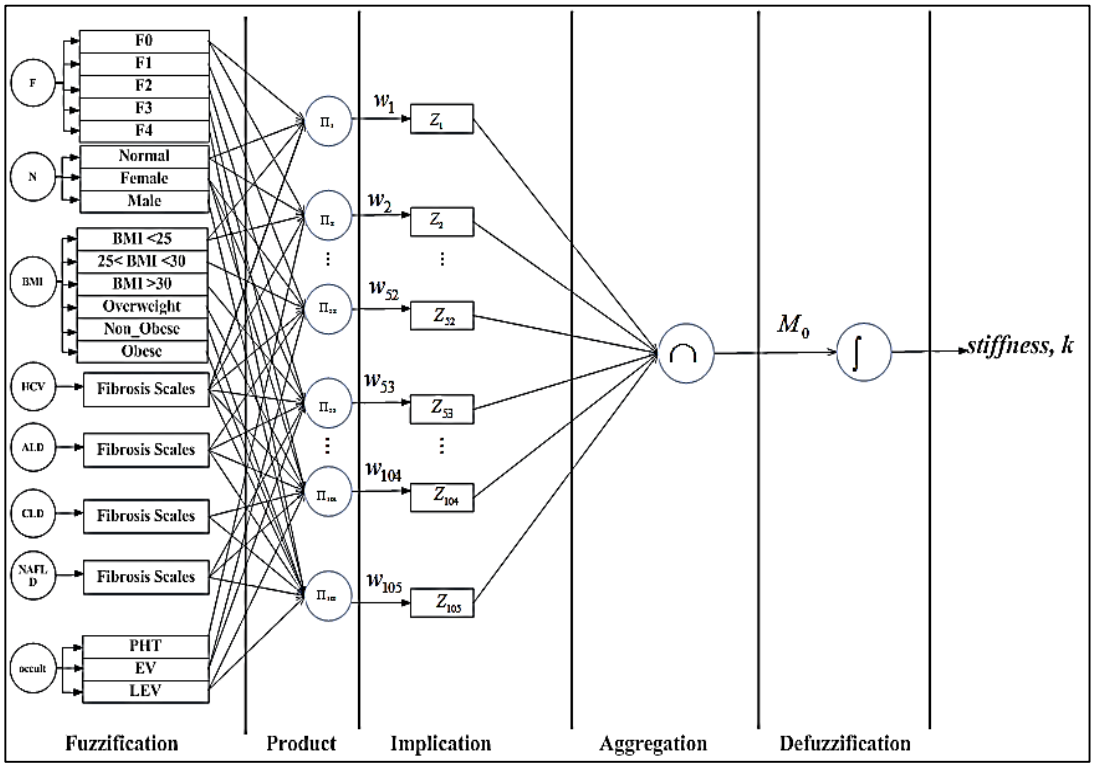


Figure 3.7 Architecture of Mamdani fuzzy inference system (M_FIS) for FMPC.

Fuzzification At first, the crisp values denoted as $[L_O \quad M \quad U_P]$ are converted to the fuzzy values by the generalized input Triangular Membership Function (trimf) as follows:

$$f(x;a,b,c) = \max\left(\left(\min\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{3.9}$$

with a, b and c being the x-coordinate of the crisp set which represent the lower boundary (L_O), median (M) and upper boundary (U_p). Note that, the lower boundary and upper boundary have the membership degree of 0, thus they are the "feet" of the

triangle while median being the peak of triangle has the membership degree equal to 1. Fig. 3.8 illustrates how the crisp inputs are being plotted into MATLAB r2012a Fuzzy Inference System.

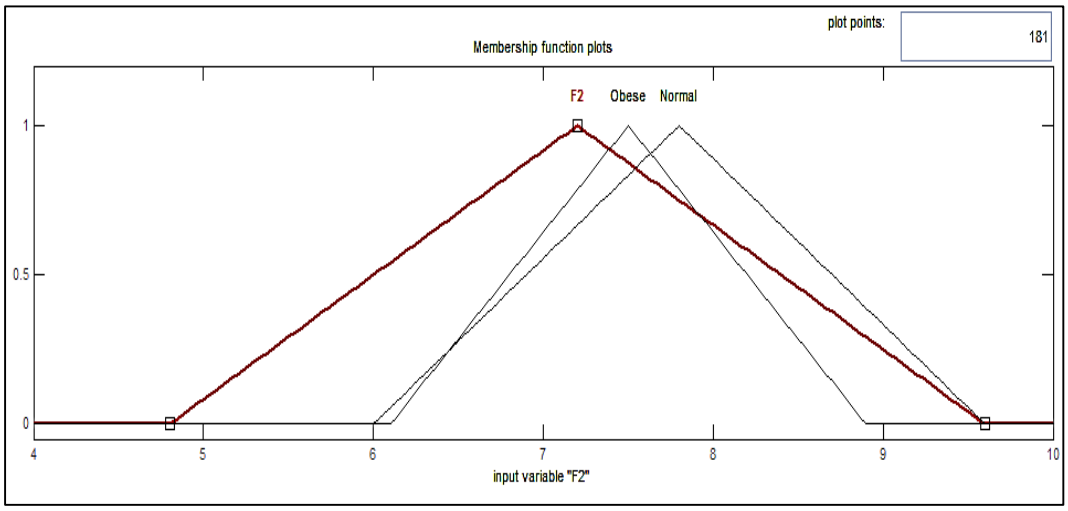


Figure 3.8 Corresponding triangular membership function for the first input-Liver Fibrosis Stage.

Product Secondly, the previously fuzzified inputs will be combined according to the fuzzy rules to establish a rule strength, which is also known as the weighting factor. The weighting factor of each rule, which is expressed as W_k, is determined by evaluating the membership expressions in the antecedent of the rule. This is accomplished by first converting the input values to fuzzy membership values by using the input MFs in the first phase and then applying the "AND" operator to these membership values. Thus, the weighting factors of the rules are computed as follows:

$$
\begin{aligned}
w_1 &= M_{11}(F)M_{21}(N)M_{31}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{81}(Occult)\\
w_2 &= M_{11}(F)M_{21}(N)M_{31}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{82}(Occult)\\
w_3 &= M_{11}(F)M_{21}(N)M_{31}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{83}(Occult)\\
w_4 &= M_{11}(F)M_{21}(N)M_{31}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{81}(Occult)\\
w_5 &= M_{15}(F)M_{21}(N)M_{31}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{82}(Occult)\\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
w_{104} &= M_{15}(F)M_{23}(N)M_{36}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{82}(Occult)\\
w_{105} &= M_{15}(F)M_{23}(N)M_{36}(BMI)M_{41}(HCV)M_{51}(ALD)M_{61}(CLD)M_{71}(NAFLD)M_{83}(Occult)
\end{aligned}
\tag{3.10}
$$

Implication Thirdly, the qualified consequent MFs based on available knowledge are calculated. In other words, the consequence of a rule, based on the available knowledge is found by combing the rule strength and the implication output membership function ($M_{imp,k}$) as follows:

$$M_{imp,k} = w_k z_k \qquad where \quad k = 1,\ldots,105 \tag{3.11}$$

Aggregation Fourthly, an overall output value will be generated according to the qualified consequent MFs. As seen in Eq (3.9), the union operator—"MAX" is implemented, such that

$$M_0(z) = \bigcup_{k=1}^{n} M_{imp,k} \tag{3.12}$$

with $k = 1,2,\cdots,n$. The type of the output MFs are Triangular MF.

Defuzzification Fifthly, the defuzzification of fuzzy output is performed by using the Centroid of Area (COA) method as seen in Eq (3.13). It is well-known as the most widely adopted defuzzification strategy, which is reminiscent of the calculation of expected values for probability distributions.

$$COA = \frac{\int_z \mu_{A\cup B}(z) \cdot z dz}{\int_z \mu_{A\cup B}(z) dz} \tag{3.13}$$

Stiffness, k Finally, this is the crisp output after the defuzzification of fuzzy output. For instance, taking 5 rules as shown below, we obtain the stiffness value as 5.6kPa. However, when rule 5 is implemented, the stiffness increase to 44.4kPa due to the intersection between the membership functions.

According to the Rules as shown in Appendix A-FMPC Rules, we have:

R1: IF (HCV is in fibrosis stage F2) and (HBV is in fibrosis stage F0-F1) THEN (Fibrosis Stage is F0-F1)

R2: IF (HCV is in fibrosis stage F2) and (HBV is in fibrosis stage F2) THEN (Fibrosis Stage is F0-F2)
R3: IF (HCV is in fibrosis stage F2) and (HBV is in fibrosis stage F3) THEN (Fibrosis Stage is F2)
R4: IF (HCV is in fibrosis stage F2) and (HBV is in fibrosis stage F4) THEN (Fibrosis Stage is F3)
R5: IF (HCV is in fibrosis stage F3) and (HBV is in fibrosis stage F0-F1) THEN (Fibrosis Stage is F0-F1)

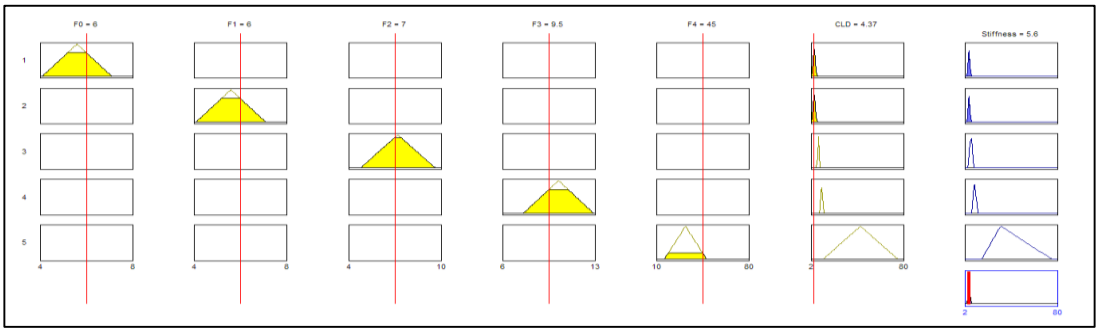


Figure 3.9 The adjustable rule viewer.

The overall implementation of Mamdani FIS (M_FIS) is illustrated in Figure 3.10. It consists of Fuzzy Inference System (FIS) editor, Membership Function Editor, Rule Editor, Surface Viewer and Rule Viewer. In this research, the Gaussian function (gaussmf) is used to represent the membership functions for each of the inputs. While Centra of Gravity (COG) defuzzification method are implemented to obtain crisp output.

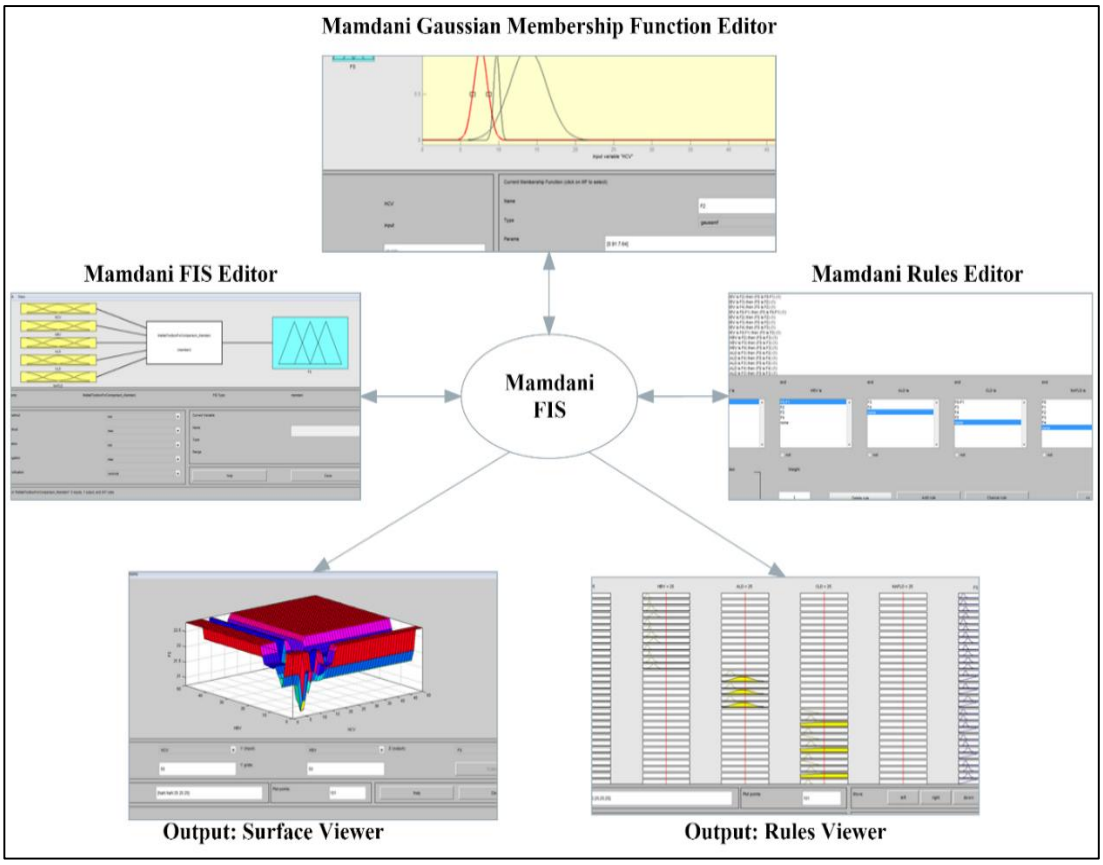


Figure 3.10 Mamdani FIS (M_FIS) with Gaussian Membership Functions.

3.3.3 Construction of Sugeno Fuzzy Inference System (S_FIS)

In this research, the Sugeno FIS is directly transformed from Mamdani FIS by implementing MATLAB r2014a function—*mam2sug*. The output membership functions of the returned Sugeno system are constants produced from the centroids of the consequent membership functions of the originality Mamdani FIS. The antecedent remains unchanged. The syntax of implementing the mam2sug is illustrated as follows:

mam_fismat = readfis('M_FIS.fis');

sug_fismat = mam2sug(mam_fis);

After the sug_fismat is generated in the MATLAB workspace, we then import it to the Fuzzy Logic Toolbox. The outcomes of the Sugeno FIS is shown in Figure 3.11.

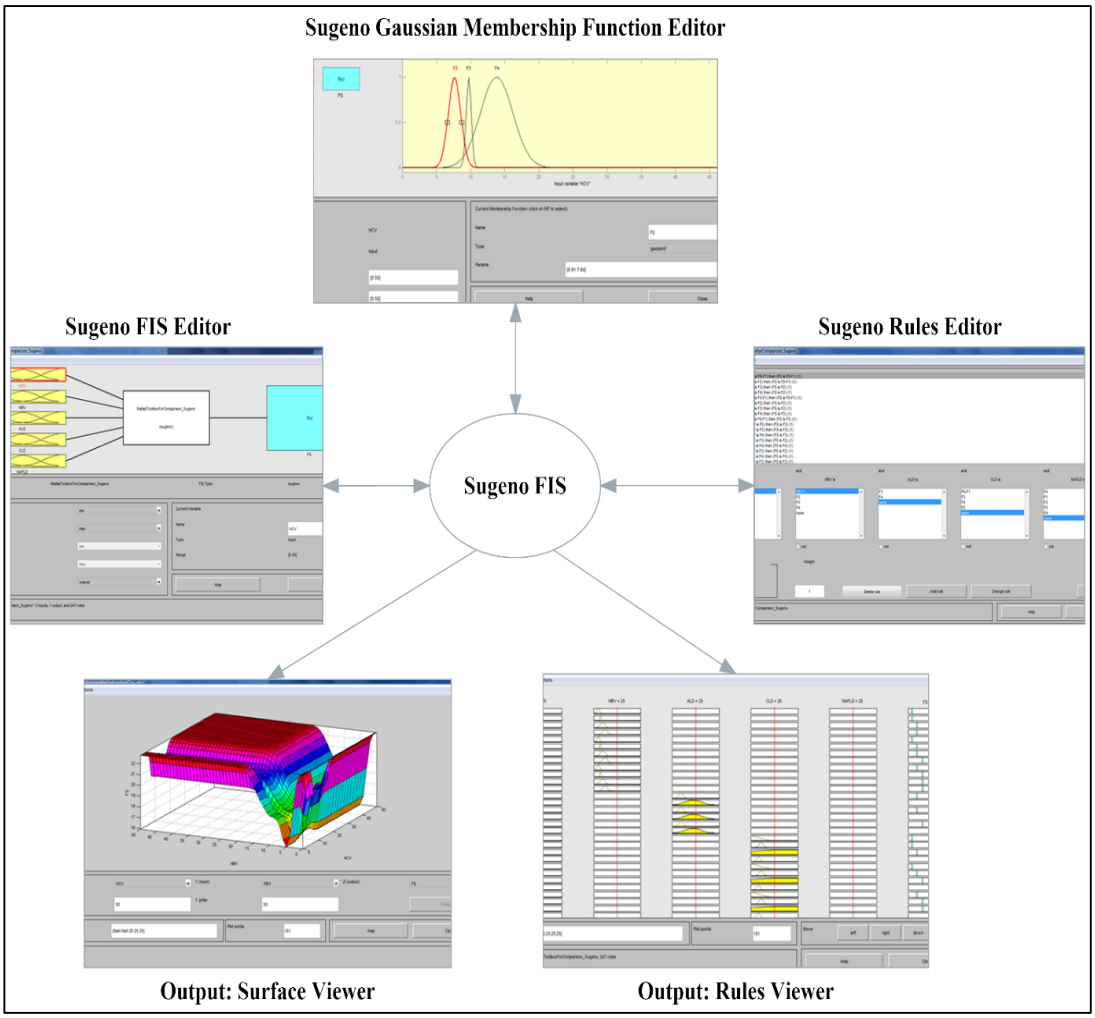


Figure 3.11 Sugeno FIS (S_FIS) with Gaussian Membership Functions.

3.3.4 Construction of Interval Type-2 Fuzzy Inference System (IT2_FIS)

The interval type-2 FIS is constructed by implementing the open source Generalized Fuzzy System (GFS) into MATLAB. The GUI of GFS is similar to the MATLAB Fuzzy Logic Toolbox. To start the GFS, we first have to change our working directory in MATLAB environment to the folder we unzipped the GFS, says, C:\GFS. Then, type 'FUZ' in the command window and press 'Enter'. A GUI for GFS will be opened as shown in Figure 3.12. The rules can simply be added to the IT2 FIS by appending the existing rules from M_FIS or S_FIS to the generated IT2 FIS MATLAB script.

The membership functions of this approach is different with the first two FISs. The shaded region is known as the Footprint of uncertainty (FOU). This region is the

interval of uncertainty. When the area is larger in the FOU, it simply means that the uncertainty is larger as well.

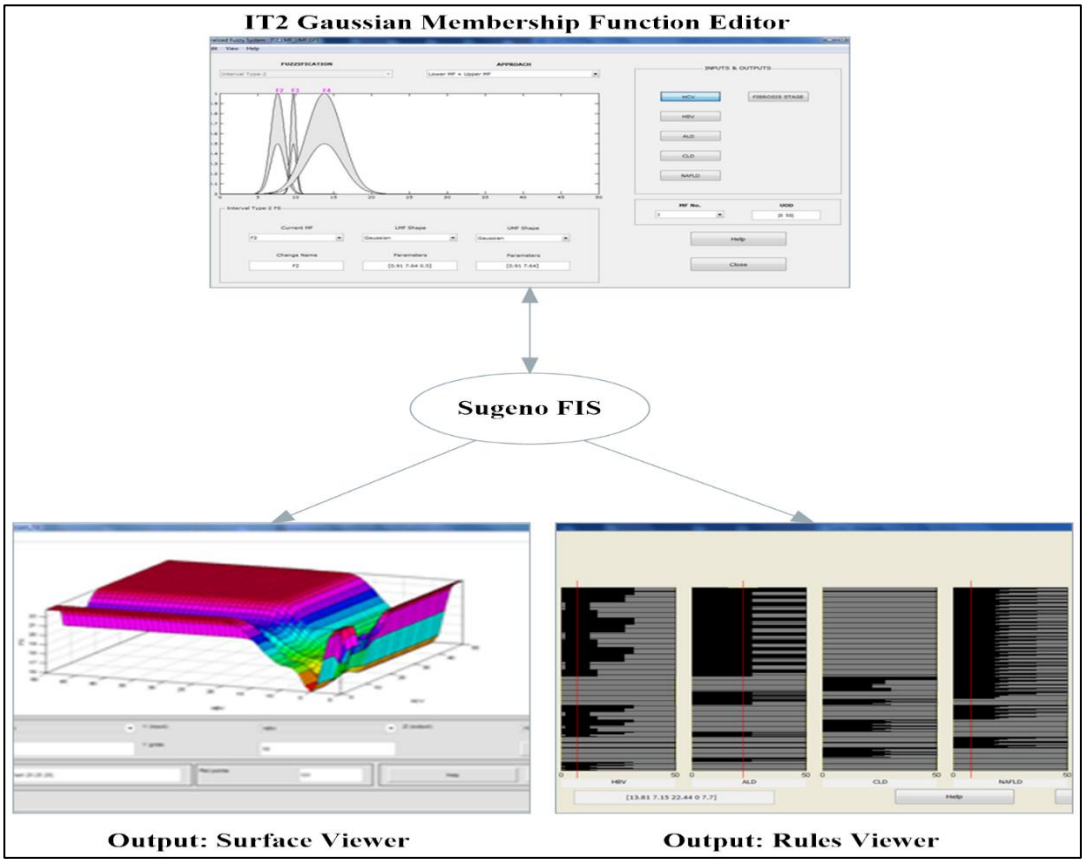


Figure 3.12 Interval Type-2 FIS (IT2_FIS) with Gaussian Membership functions.

3.4 CUDA implementation

In this research, the heterogeneous parallel programming for soft-tissue deformation is developed using the CUDA C language provided by Nvidia. Understanding the device (GPU) memory features is crucial for efficient implementation (CUDA C Programming Guide, 2013) simply because the device (GPU) has more memory than the host (CPU). By applying both host and device efficiently will provide a seamless simulation. However, it remains as a challenging issue in multi-cores application [HEB10]. The heterogeneous parallel programming is implemented in this research by sending sequential matters (initialize of point cloud data and generation of mesh model) to the CPUs while the parallel matter (integration) to the GPUs as shown in

Figure 3.13. In the implementation process, the bandwidth between the device and device memory is much higher than that between the host and device. The data is initialized on the host, CPU and then the data will be copied to the device, GPU. This is because the resulting visualization is rendered on the host, thus only the current position data of the mass point need to be transformed to the host.

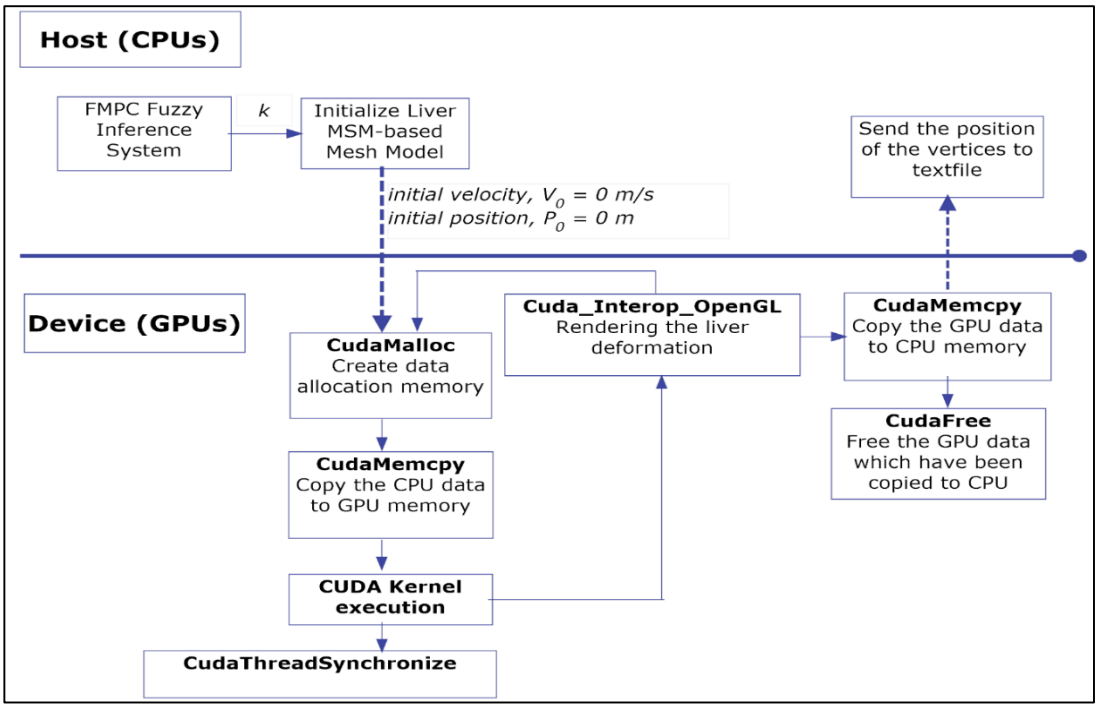


Figure 3.13 CUDA implementation.

3.5 Benchmark

In this paper, the benchmark model is abstracted from Basafa and Farahmand (2011). They has developed an improved Mass Spring Damper model to simulate the nonlinear viscoelastic behaviour of the biological soft tissues which interact with a surgical indenter. The model can be further extended to exhibit viscoelastic behaviour by adding the damping force which reacts directly to the mass point proportionally to the velocities, parallel to the spring.

The general equation is similar to Eq (3.5) such that

$$m\ddot{x}_{ijk}(t) + \gamma \cdot \dot{x}_{ijk}(t) + k \cdot x_{ijk}(t) = F_{ext}$$

$$M_i \ddot{x}_i + F_i^d + F_i^s x = F_i^{ext} \qquad (3.14)$$

where F_i^d and F_i^s are obtained using Eq (3.1) and Eq (3.2), respectively.

$$M_i\ddot{r}_i + b_0\dot{r}_i + b_1\left\|r_i - r_1^0\right\|\dot{r}_i + \sum_{j=1}^{N} F_{ij}^s \frac{r_i - r_j}{\left\|r_i - r_j\right\|} = F_i^{ext} \tag{3.15}$$

In the benchmark model, F_i^s is used to demonstrate a highly non-linear elasticity behaviour of the soft tissue. The function F_i^s is expressed in a two-step expression of the force-displacement characteristics, in the form of a third degree polynomial at low displacements, and a linear behaviour at higher displacements, such that

$$F_i^s(X) = \begin{cases} K_1 X + K_2 X_C^3, & |X \le X_C| \\ (A + B(|X| - |X_C|))\dfrac{X}{|X|}, & |X > X_C| \end{cases} \tag{3.16}$$

where K_1 and K_2 are constants, X_C is the critical displacement of the nonlinear springs and parameters A and B are defined as follows:

$$\begin{aligned} A &= K_1 X + K_2 X_C^3 \\ B &= K_1 + 3K_2 X_C^2, \end{aligned} \tag{3.17}$$

Thus, the stiffness coefficient of the spring is

$$F_s = F_{ijk}(X)\left[\frac{\vec{x}_{ijk}}{|\vec{x}_{ijk}|}\right]_0^1 \tag{3.18}$$

with $x_{ijk} = x_1(X) - x_0(X)$

Next, to achieve more realistic viscoelastic behaviour of the soft tissue deformation the nodal damping forces are further extended in the benchmark model. A displacement-velocity component and the typical velocity alone component are assumed to exist in the damping force. Thus, F_i^d is expressed as follows:

$$F_d = \gamma_0\dot{x}_i + \gamma_1\left\|x_i - x_i^0\right\|\dot{x}_i = \dot{x}\left(\gamma_0 + \gamma_1\left\|x_i - x_i^0\right\|\right) \tag{3.19}$$

where γ_0 and γ_1 are two damping constants. x_i and x_i^0 are the position vector and initial position of node i, respectively. The corresponding parameters of this benchmark model are tabulated as below:

Table 3-3 The benchmark parameters

Parameters	**Value**
K_1	0.05 N
K_2	10 N
X_c	0.2
γ_0	2 Ns/m
γ_1	1000 Ns/m2

However, it remains as a major issue in the benchmark model which is the determination of model parameters K_1, K_2, X_C, γ_0 and γ_1. Although the parameters are somehow related to the soft tissue mechanical properties, the relationship is not well defined. Therefore, these parameters do not directly determine with specific constraints. The parameters are often tuned manually by fitting the experimental data into the model.

3.6 Summary

The 3D modelling of human liver is modelled in Microsoft Visual Studio 2010 by integrating OpenGL with C++ in CUDA platform. The open source liver 3D data is obtained from http://gforge.inria.fr/frs/?group_id=690. Meanwhile, the Fuzzy Logic Toolbox and GFS are implemented in MATLAB r2014a to model the FIS. As presented in the designation of the FMPC in Section 3.3, three types of fuzzy approaches are utilized in obtaining the corresponding stiffness coefficients. The fuzzy approaches include the Mamdani FIS, Sugeno FIS and Interval Type-2 FIS. The later fuzzy approach is adopted from GFS which is an open source toolbox for visualizing the fuzzification using all types of fuzzy sets. It could be reached at http://sourceforge.net/projects/gfs. Lastly, the comparison between the benchmark

model and FMPC outputs were presented in Microsoft Excel 2013. Throughout this research, the simulations are carried out on a personal laptop equipped with Intel® Core™ i5-2410M CPU @ 2.30GHz (2 Cores); 8GB RAM; 64-bit Windows OS; NVidia GeForce GT 540M with MSVS 2010 and CUDA 5.5 installed.

Chapter 4 *Fuzzy Inference System*

4.1 Overview

Liver fibrosis is the natural wound-healing response to parenchymal injury in chronic liver diseases. It may eventually results in liver Cirrhosis (F4) and its various complications (F0-F1, F2, F3). The assessment of liver fibrosis staging is essential for people suffering from chronic liver diseases. The degree of fibrosis plays a crucial indication for the severity of the underlying liver disease. Besides, it may have prognosis significance. In liver surgical training correspond to different stages of liver fibrosis, the liver stiffness is a novel parameter for the diagnosis. Two of the most commonly used liver stiffness measurement (LSM) methods are the invasive liver biopsy and non-invasive Transient Elastography (TE). By measuring the liver stiffness of one patient, it allows prognosis significance in predicting long-term outcome and monitoring disease progression which then decide on specific-treatment needed.

Interpretation of LSM results should always be done by expert clinicians according to clinical context, imaging and laboratory findings. This is because the liver stiffness (LS) is an excellent surrogate marker of advanced fibrosis (F3) and Cirrhosis (F4) outscoring all previous non-invasive approaches to detect Cirrhosis. It helps to have a better understand on the molecular mechanisms underlying liver fibrosis. Many other factors may increase LS such as hepatic infiltration with tumour cells, mast cells (mactocytosis), inflammatory cells (all form of hepatitis) or amyloidosis. However, the major factors which liver stiffness depends on are extracellular matrix, applied constraints, internal pressure and viscous effect of the organ. Other factors which influence the liver stiffness are hepatitis, mechanic cholestasis, liver congestion, cellular infiltrations and deposition of amyloid irrespective of fibrosis stages.

Aforementioned, there are two most commonly used measuring methods—Liver Biopsy and Transient Elastography (TE). Liver biopsy; an invasive procedure in assessing liver fibrosis, however it remains as gold standard in measuring till date. However, due to the potential adverse effects associated with the invasive procedure, patients may be unwillingly to undergo a biopsy procedure. Apart from adverse effects, the accuracy of liver biopsy depend on the quality, length, volume of the specimen obtained from patient. This is because biopsies require only a tiny sampling portions of the liver, therefore, they may subjected to sampling errors. Thus, an alternative non-invasive method—Transient Elastography (TE) is developed. TE works as an ultrasound transducer probe which is mounted on the axis of a vibrator. The vibrations of mild amplitude with low frequency of 50Hz is transmitted by the transducer, inducing a plastic shear wave that propagates the underlying tissues. The stiffer the tissue, the faster the shear wave propagates. Thus, it is able to measure liver stiffness in patients who suffered from a range of chronic liver diseases. Previous works have shown TE to be satisfactory accurate and reproducibility to estimate liver fibrosis.

Therefore, by integrating the available knowledge and dataset obtained from previous medical research through liver biopsy and TE, the stiffness coefficient of the MSM parameter could be determined in correspond to the disease-specific.

4.2 Illustration of Problem

This chapter solves the problem in selecting the stiffness coefficient of the MSM spring by using Knowledge-based Fuzzy Inference System (FIS). Generally, a fuzzy inference system (FIS) is built up by three components—rules, database and reasoning mechanism. The rules consist of a collection of available linguistics knowledge and the databases consist of a bunch of crisp sets which defines the membership functions

based on the rules. The reasoning mechanism performs fuzzy decision making upon the rules and given facts to derive the outputs. In FIS, the inputs could either be fuzzy or crisp (exact value) inputs, however, the output shall always be crisp value in order to be implemented into the real-world system.

The designed knowledge-based fuzzy inference system is based on the available knowledge and dataset abstracted from previous LSM medical research outcomes (Ziol et al., 2005; Castera et al., 2005; Corpechot et al., 2006; de Ledinghen et al., 2006; Roulot et al., 2008; Yoneda et al., 2008; Fung et al., 2009; Wong et al, 2010; Jeremy et al., 2008; Jeremy Cobbold et al., 2008; Marcellin et al., 2009; Fung et al., 2010; Mueller et al., 2010; Kim et al, 2010; Myers et al., 2011; Umut Ozcan et al., 2011; Robic et al., 2011; Castera, 2011; de Ledinghen et al., 2012; Myers et al., 2012; Castera et al., 2012; Arora and Sharma, 2012; Kim et al., 2013; Redd et al., 2013; Kumar et al., 2013; Machado et al., 2013; Frulio and Trillaud, 2013). This knowledge-based fuzzy inference system allows the so-called existing 'intelligence' to manipulate the machine. This system consists of five input variables and one output variable for result. The input variables include Hepatitis C Virus (HCV), Hepatitis B Virus (HBV), Alcoholic Liver Disease (ALD), Chronic Liver Disease (CLD) and Non-Alcoholic Fatty Liver Disease (NAFLD). Meanwhile, the output variable refers to the staging of liver fibrosis in the patient.

In this study, the Fuzzy Logic Toolbox and Generalized Fuzzy System (GFS) are implemented in MATLAB r2014a to model the Fuzzy Inference System (FIS). Aforementioned in previous chapter, Section 3.3, three types of fuzzy approaches are utilized in obtaining the corresponding stiffness coefficients. The fuzzy approaches include the Mamdani FIS, Sugeno FIS and Interval Type-2 FIS. The later fuzzy approach is adopted from GFS which is an open source toolbox for visualizing the

fuzzification using all types of fuzzy sets. It could be reached at http://sourceforge.net/projects/gfs. Lastly, the comparison between the benchmark model and FMPC outputs were presented in the following sections based on Microsoft Excel 2013. Throughout this research, the simulations are carried out on a personal laptop equipped with Intel® Core™ i5-2410M CPU @ 2.30GHz (2 Cores); 8GB RAM; 64-bit Windows OS; NVidia GeForce GT 540M with MSVS 2010 and CUDA 5.5 installed.

4.3 Designation of Fuzzy MSM Parameters Controller (FMPC)

In this section, the designation of FMPC associated with the respective input and output variables. Later on, the determination of the corresponding membership functions (MFs) for each of the input and output variables are shown.

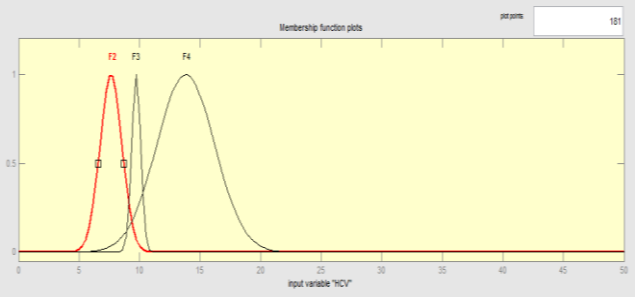
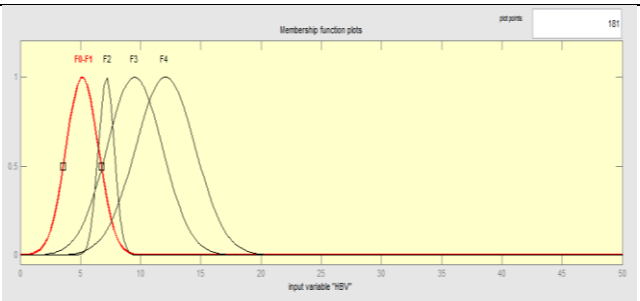
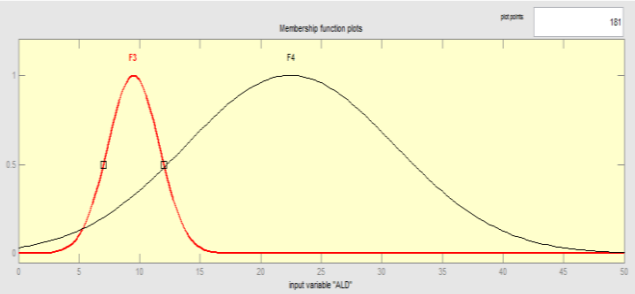
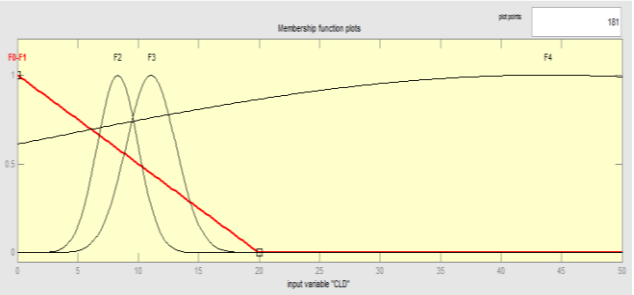
4.3.1 Determination of Membership Functions (MFs)

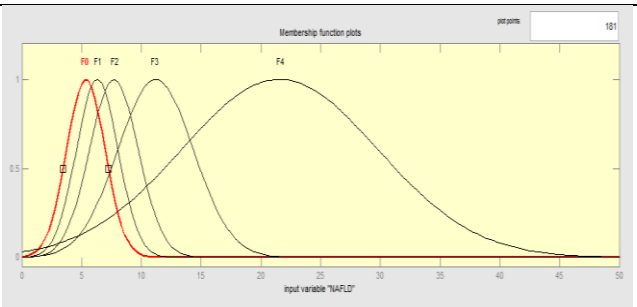
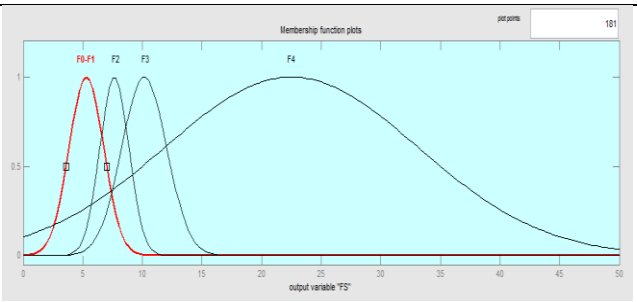
Fuzzy MSM Parameter Controller (FMPC) consists of five input and one output variables. These variables are described with corresponding membership functions (MFs). These MFs are used to determine the membership of crisp (exact range) sets to fuzzy sets. Aforementioned in previous section, the input variables include Hepatitis C Virus (HCV), Hepatitis B Virus (HBV), Alcoholic Liver Disease (ALD), Chronic Liver Disease (CLD) and Non-Alcoholic Fatty Liver Disease (NAFLD). Meanwhile, the output variable referred to the corresponding liver fibrosis staging, F0-F1: absent/ mild fibrosis; F2: significant fibrosis; F3: severe cirrhosis; F4: cirrhosis. Each of the respective MFs are expressed in terms of Gaussian Membership Function (Gauss_MF) as below:

$$\mu_{A^i}(x) = \exp\left(-\frac{(c_i - x)^2}{2\sigma_i^2}\right) \tag{4.1}$$

where c_i and σ_i denote the centre and width of the i-th fuzzy set A^i, respectively.

The description in terms of Gauss_MFs for each of the variables are presented as follows:

Inputs & Output	Fuzzy Sets $[c_i \quad \sigma_i]$	Illustration
HCV	F2: [0.91 7.64] F3: [0.38 9.72] F4: [2.38 13.81]	
HBV	F0-F1: [1.34 5.15] F2: [0.66 7.15] F3: [2.3 9.5] F4: [2.47 12.06]	
ALD	F3: [2.12 9.5] F4: [8.6 22.44]	
CLD	F0-F1: [-20 0 20] F2: [2.05 11.05] F3: [44.12 43.8] F4: [1.69 8.3]	

NAFLD	F0: [1.61 5.35] F1: [1.74 6.3] F2: [2.04 7.7] F3: [3.15 11.19] F4: [8.22 21.57]	
FS	F0-F1:[1.45 5.3] F2: [1.21 7.65] F3: [1.88 10.15] F4: [10.62 22.5]	

4.3.2 FMPC Rule Base

In this research, FMPC Rule Base consists of 247 rules which are generated based on the rule matrix as shown in Table 3.2 from previous chapter with the corresponding membership functions for each of the inputs. The respective rules are tabulated in Appendix A. The output intersection in terms of surface plot and pseudocolor for each of the fuzzy approaches are shown in the respective sections as follows. Meanwhile, the comparison between the benchmark model and fuzzy approaches in terms of displacement vs time, velocity vs time and velocity vs displacement (phase plane plot) are illustrated in the respective sections.

4.4 FMPC Result and Analysis

4.4.1 Obtaining the Stiffness Coefficient from FMPC

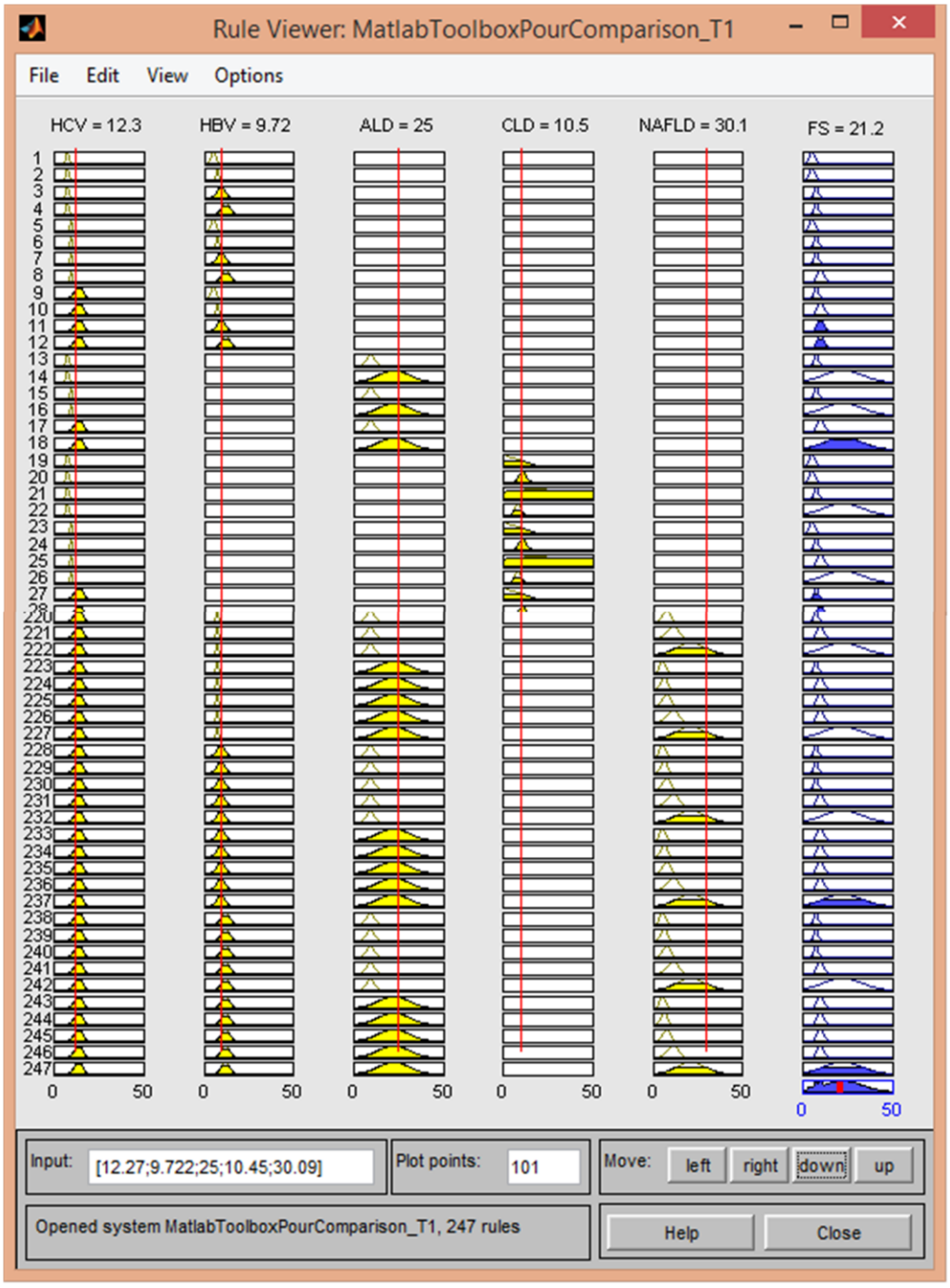


Figure 4.1 The adjustable rule viewer of FMPC.

4.4.2 Comparison between Mamdani FIS, Sugeno FIS and Interval Type-2 FIS

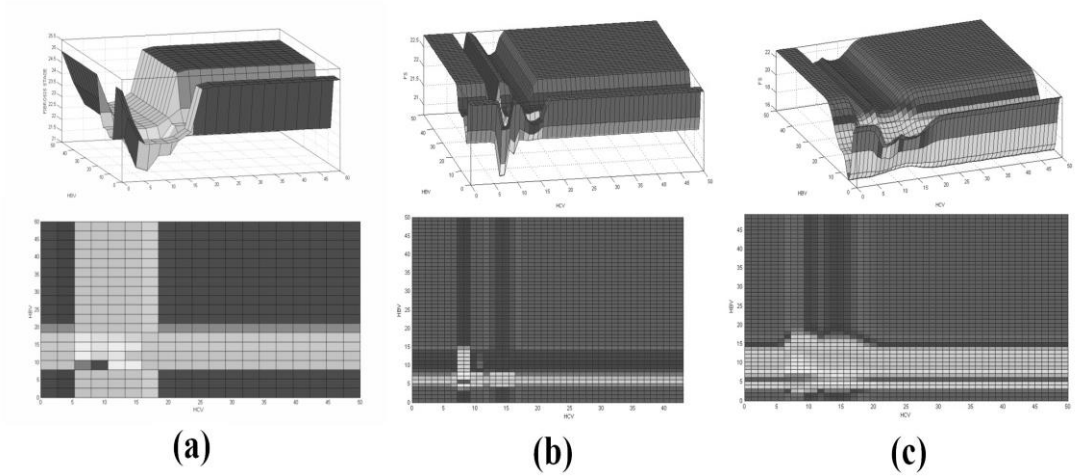


Figure 4.2 Surface view of three different fuzzy approaches.

4.4.3 Comparison between the Benchmark Model with the Fuzzy Approaches

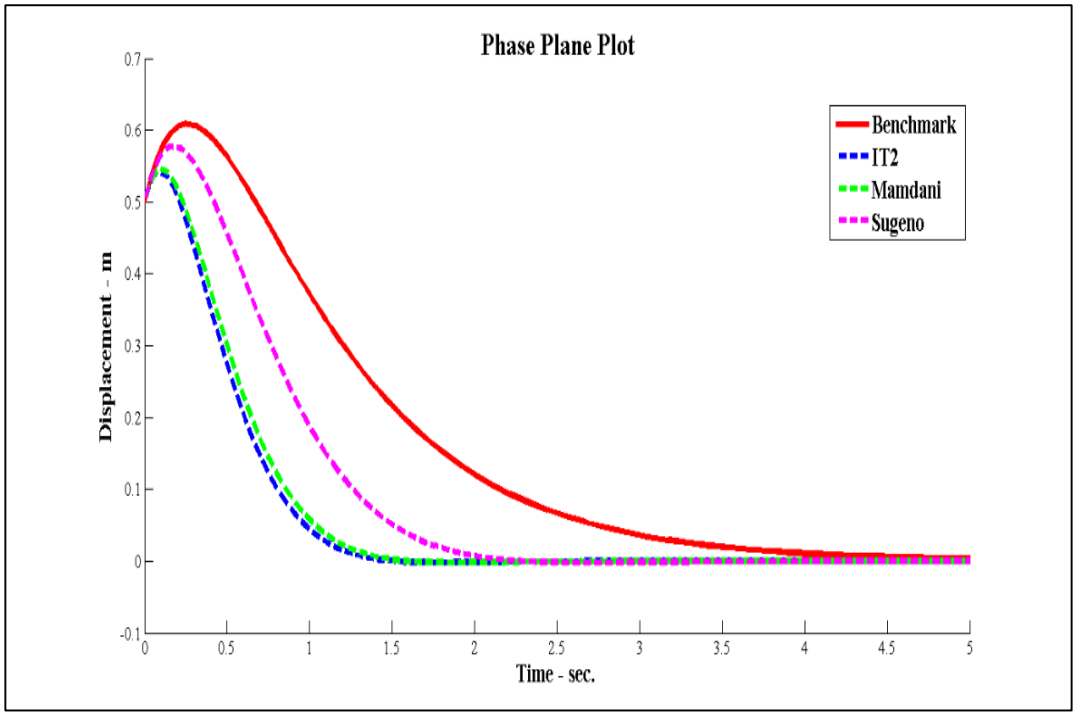


Figure 4.3 Comparison of benchmark model with the respective fuzzy approaches in terms of phase plane plot.

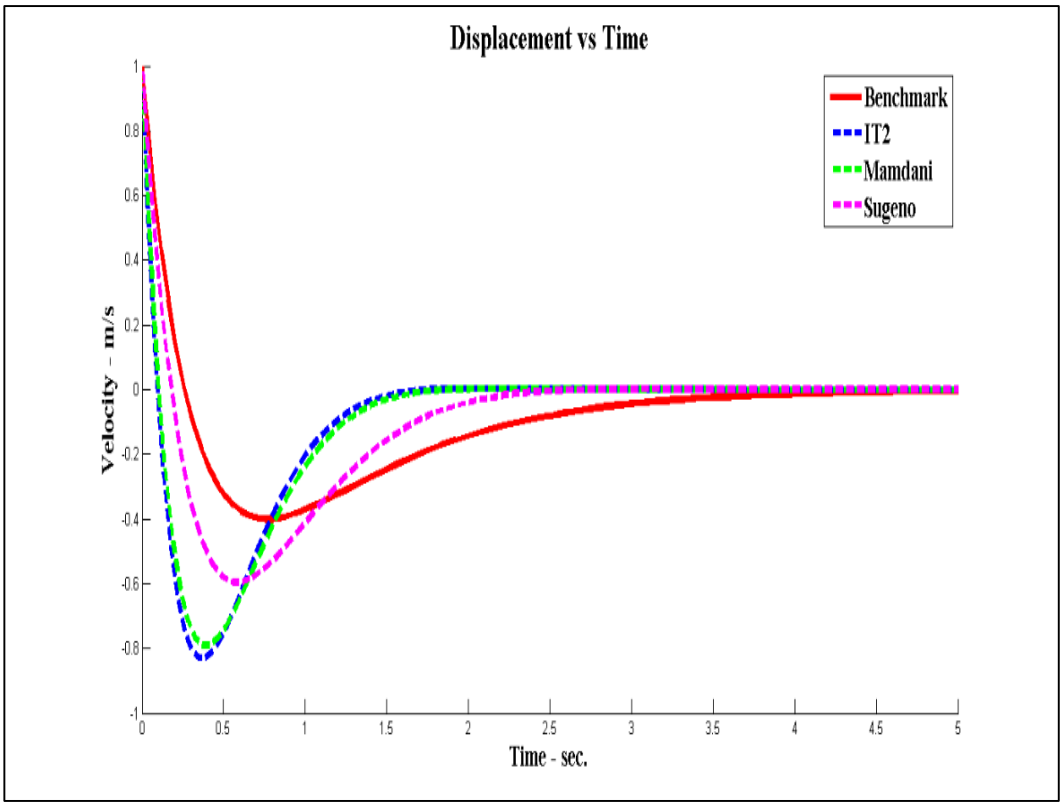


Figure 4.4 Comparison of benchmark model with the respective fuzzy approaches in terms of displacement versus time.

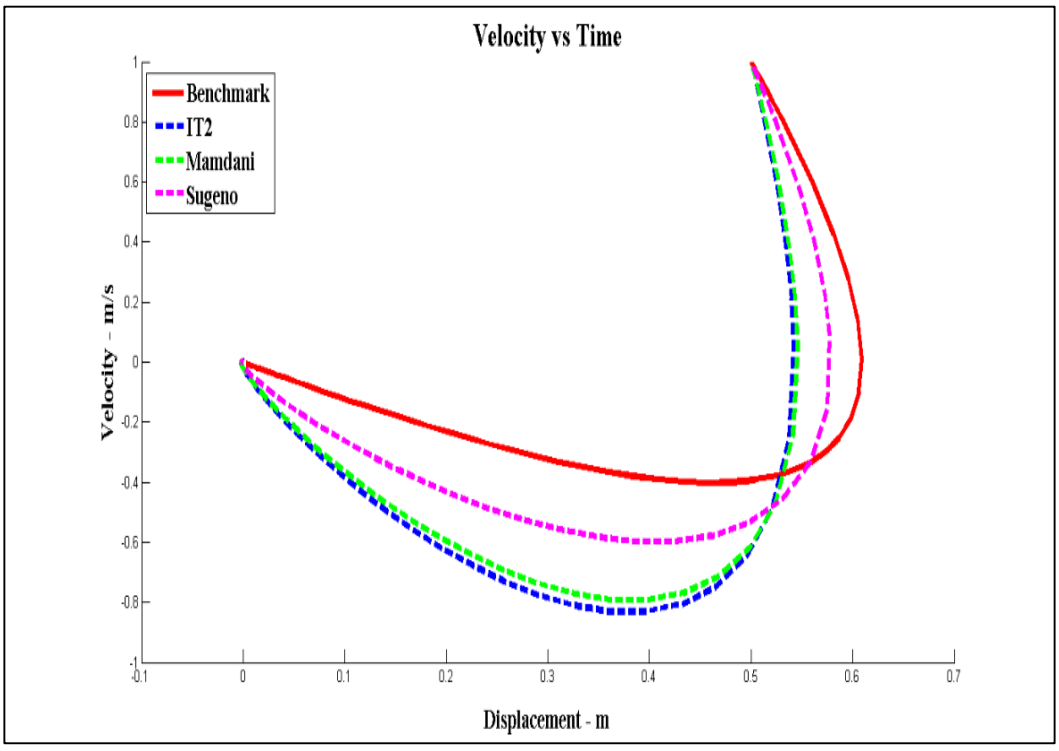


Figure 4.5 Comparison of benchmark model with the respective fuzzy approaches in terms of velocity versus time.

4.4.4 Graphs Similarity

Table 4-1 Graphs similarity between benchmark model with fuzzy approaches

Graphs	**Similarity [0 1]**
IT2	**0.8598**
Mamdani	0.8580
Sugeno	0.8549

4.5 Discussions

4.5.1 Obtaining the Stiffness Coefficient from FMPC

In this research, the respective stiffness coefficients, k, are obtained via tuning the inputs in Rule Viewer which is illustrated in Figure 4.1. The inputs are directly related to the correlation between several liver fibrosis stages which might be occult in one patient. The defuzzified value (crisp) of stiffness coefficient is shown in the last column. The resultant outputs of the corresponding stiffness coefficient based on the respective rules are shown in Appendix A.

Next, by substituting the corresponding value of stiffness coefficient, k into the MSM block diagram, we first obtain the corresponding damping coefficient accordingly. Then the simulation began in order to obtain the position_array and time_array data which are saved in the MATLAB workspace. These data are then used to plot the following graphs as shown in Figure 4.2, 4.3, 4.4 and 4.5.

Table 4-2 Outputs of the corresponding stiffness coefficient based on selected rules

RULES	Stiffness coefficient, K (kPa)	
	Mamdani FIS (M_FIS)	Interval Type-2 FIS (IT2_FIS)
25	19.4	22.2
50	19.3	21.7
100	18.7	20.9
125	22.8	22.8
150	19.0	20.9
200	21.1	18.9
225	20.7	20.8

Aforementioned, the underlying MSM is redesigned where the parameters are determined by using a knowledge-based fuzzy logic controller. The concept of knowledge-based control is to capture and implement experience and knowledge available from the experts. Here, the stiffness coefficient which is obtained from FMPC is not generated automatically. However, based on the available expert knowledge, the user of the surgical simulation could simply obtain the damping coefficient of the liver. Moreover, the damping coefficient is directly dependent on the stiffness value as seen in Eq. 3.6.

Meanwhile, in Fig. 4.6, the comparison of the displacement outcome between the upper bound (mass=1.5kg) benchmark model with FMPC is illustrated. It shows that the graphs seem to share a similar trend. As the force is exerted on the soft tissue, a similar trend of displacement occurs. This can be seen more clearly in Fig. 4.7 which compare the trend lines of benchmark with some selected rules. Notice that, when the graphs are shown in linear form, the gradient of the corresponding stiffness, m_F is directly proportional to the gradient of benchmark, m_B such that

$$m_F \propto m_B \tag{4.1}$$

As shown in the results, the linguistic nature of fuzzy knowledge-based control makes it possible to express process knowledge concerning how the stiffness should be controlled or how the liver deforms based on the liver diseases. The outputs from FMPC are able to mimic the results of benchmark model and the FMPC has simplified the underlying MSM by avoiding the complex calculation of the spring coefficients which do not have a direct relationship with the Young's Modulus and Poisson Ratio.

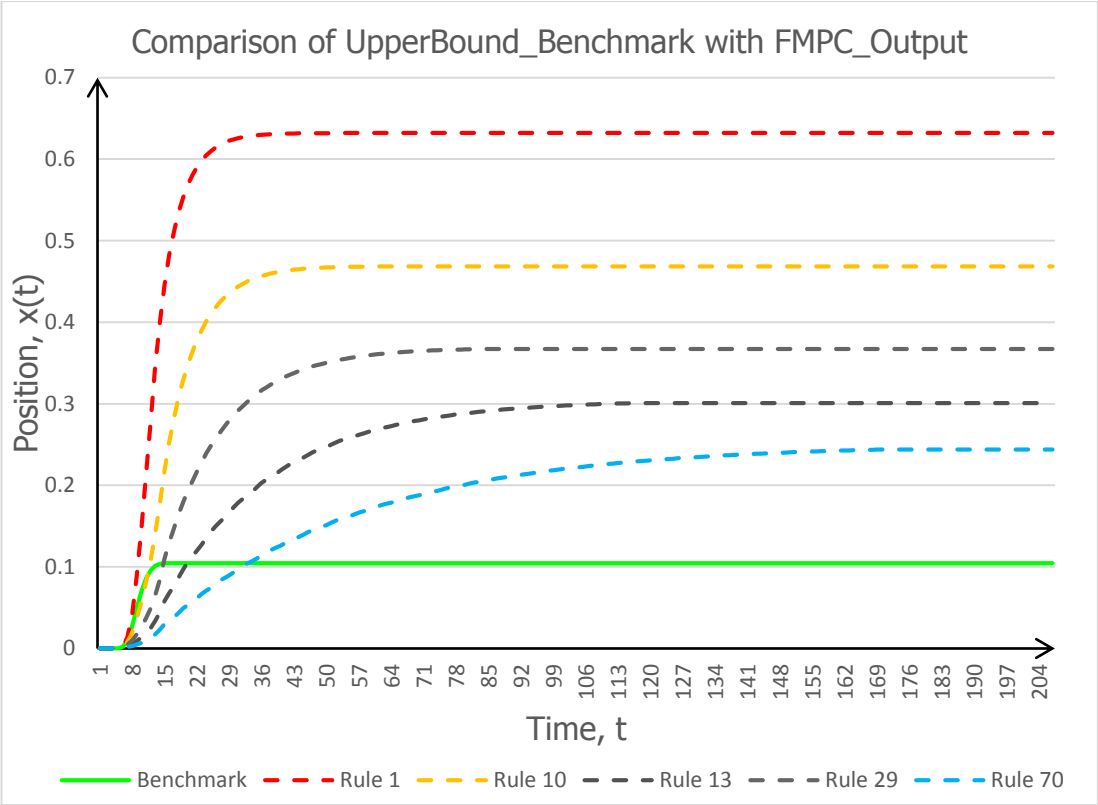


Figure 4.6 Comparison of the Displacement corresponding to each of the stiffness value.

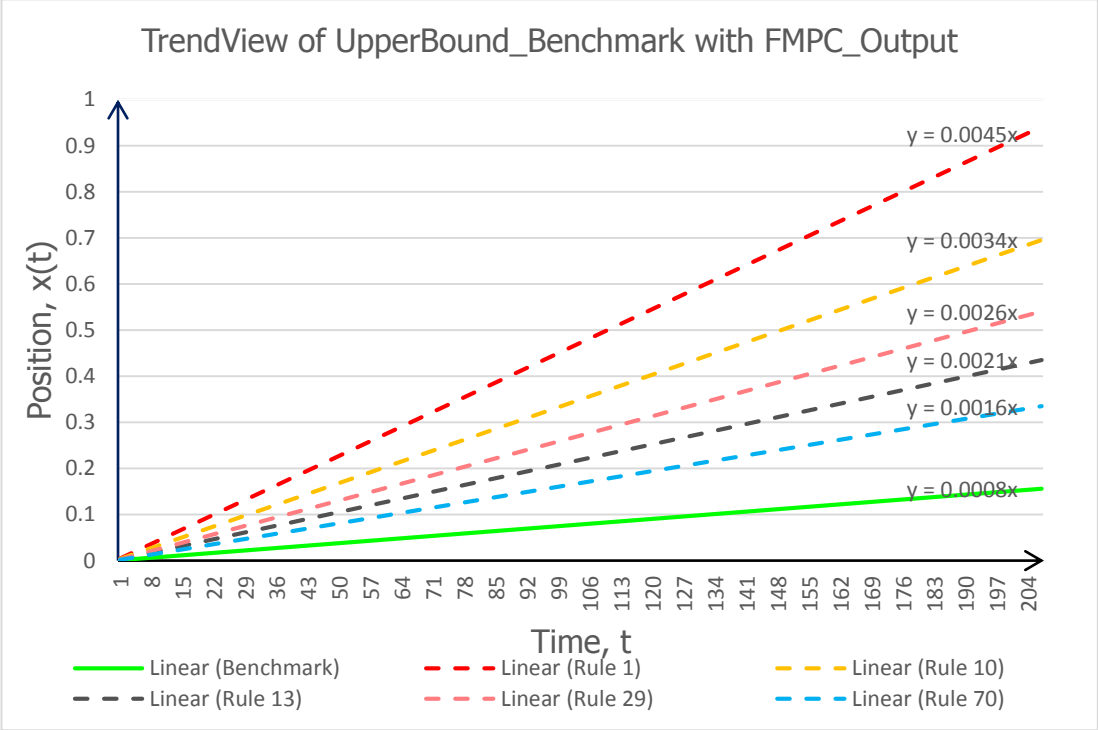


Figure 4.7 Trend view of benchmark with selected rules.

Notice that, as illustrated in Figure 4.2, the surface plot and pseufo-color of the IT2 FIS allows more intersection compared to the Mamdani and Sugeno FIS. This is

because IT2 concists of type-2 membership function with certain level of uncertainty interval while the membership functions for the other two are made of crisp sets. Meanwhile, in Figure 4.3, it showed that the graphs between the IT2 FIS, Mamdani FIS and Sugeno FIS with the benchmark model seems to share the similar trend. Therefore, the fuzzy approaches in this study show that the stiffness value predicted by FIS are in very good agreement with the benchmark result. However, how accurate is the stiffness value between three of these fuzzy approaches?

4.5.2 Graphs Similarity

In order to answer the above mentioned research question, it is crucial to evaluate the graphs similarity between each of the fuzzy approach graphs with the benchmark graph. Taking each of the graph vertices, the graphs similarity is measured. The range of similarity is [0 1]. The closer the fuzzy approaches graph similarity value to 1, the more similar it would be with the benchmark graph. Each of the graphs similarity are shown in Table 4-1, notice that among the three fuzzy approaches, IT2 has the highest similarity (0.8598 over 1) with the benchmark model. Thus, the stiffness coefficient value obtained from IT2 is the most reliable and accurate among the three.

4.6 Summary

In this study, the fuzzy approaches are proved to be powerful approaches in building complex and nonlinear relationship between a set of input and output data. The stiffness values estimated by fuzzy approaches are in very good agreement with the benchmark result. The corresponding graphs for each of the fuzzy approaches share the similar trend of displacement and velocity with the benchmark model. Among three of the fuzzy approaches, the Interval Type-2 FIS has the highest similarity with the benchmark model.

However, the outputs of FMPC is directly dependent upon the interpretation of the available knowledge. Therefore, extra care should be taken while selecting the membership function ranges as well as the reasoning mechanism for each rule. Furthermore, due to the limitation of liver stiffness data, there are only 247 rules could be generated.

For future development, the Interval Type-2 FIS shall be improved in terms of real-time efficiency. More professional knowledge shall be added to further improve the accuracy of the fuzzy inference system. At the same time, the FMPC scheme shall be extended into real-time invasive liver surgical training simulations in order to validate its practicality.

Chapter 5 *CUDA Implementation*

5.1 Overview

In this research, the heterogeneous parallel programming for soft-tissue deformation is developed using the CUDA C language provided by Nvidia. Understanding the device (GPU) memory features is crucial for efficient implementation. This is because device has more memory than the host (CPU). By applying both host and device efficiently will provide a seamless simulation. However, it remains a challenging issue in multi-cores application (Hermann et al., 2010). The heterogeneous parallel programming is implemented in this research by sending sequential matters (initialize of point cloud data and generation of mesh model) to the CPUs while the parallel matter (integration) to the GPUs as shown in Figure 3.12. In the implementation process, the bandwidth between the device and device memory is much higher than that between the host and device. The data is initialized on the host, CPU and then copy the data to the device, GPU. This is because the resulting visualization is rendered on the host, thus only the current position data of the mass point need to be transformed to the host.

5.2 Simulation Results

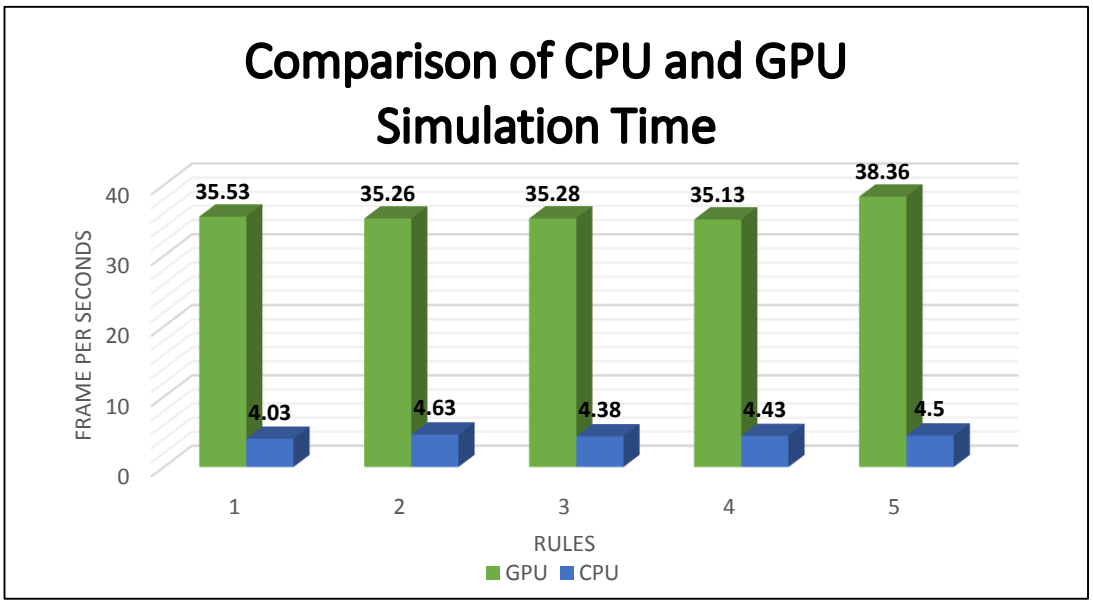


Figure 5.1 Comparison of CPU and GPU simulation runtime.

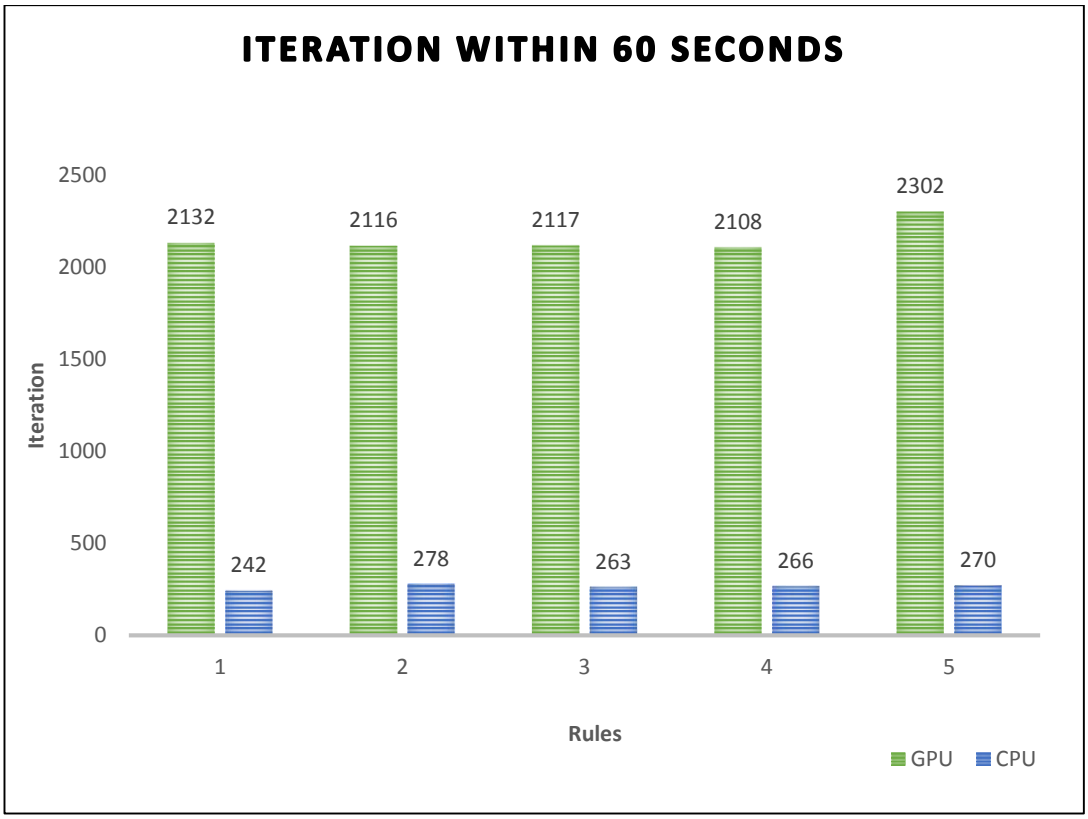


Figure 5.2 Comparison of CPU and GPU on the iteration within 60 seconds.

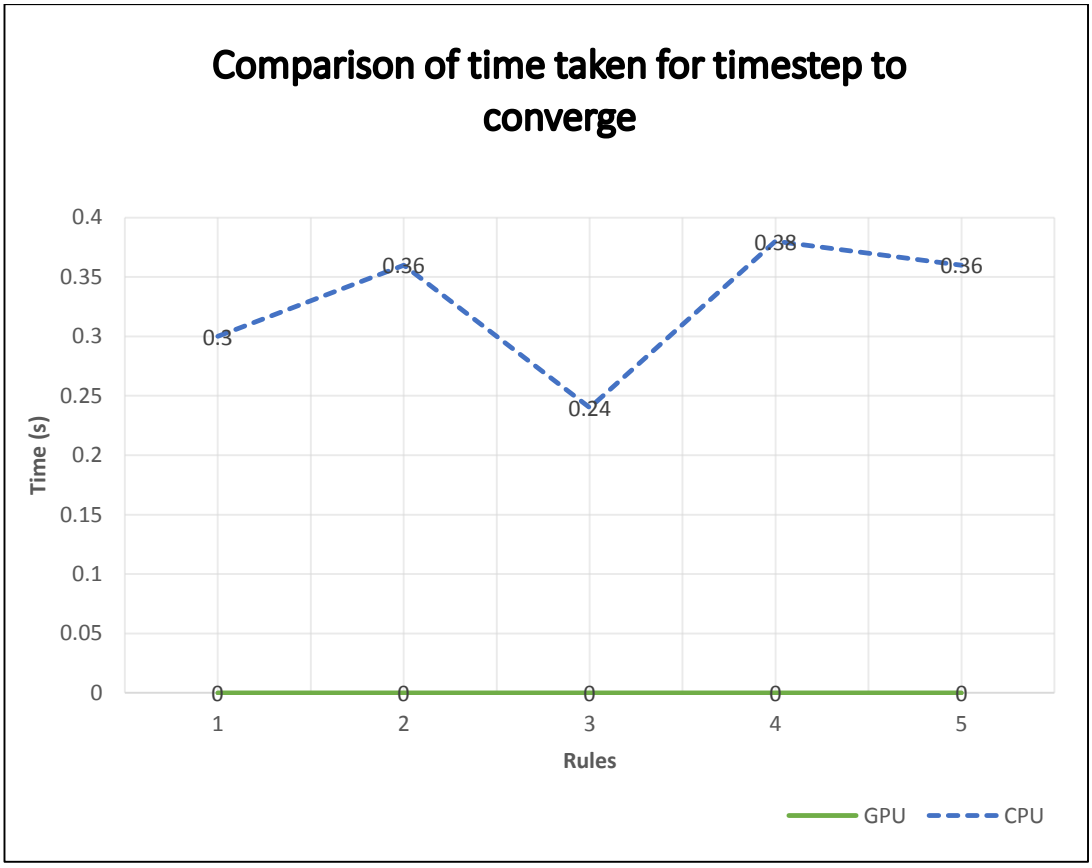


Figure 5.3 Timestep convergency between CPU and GPU.

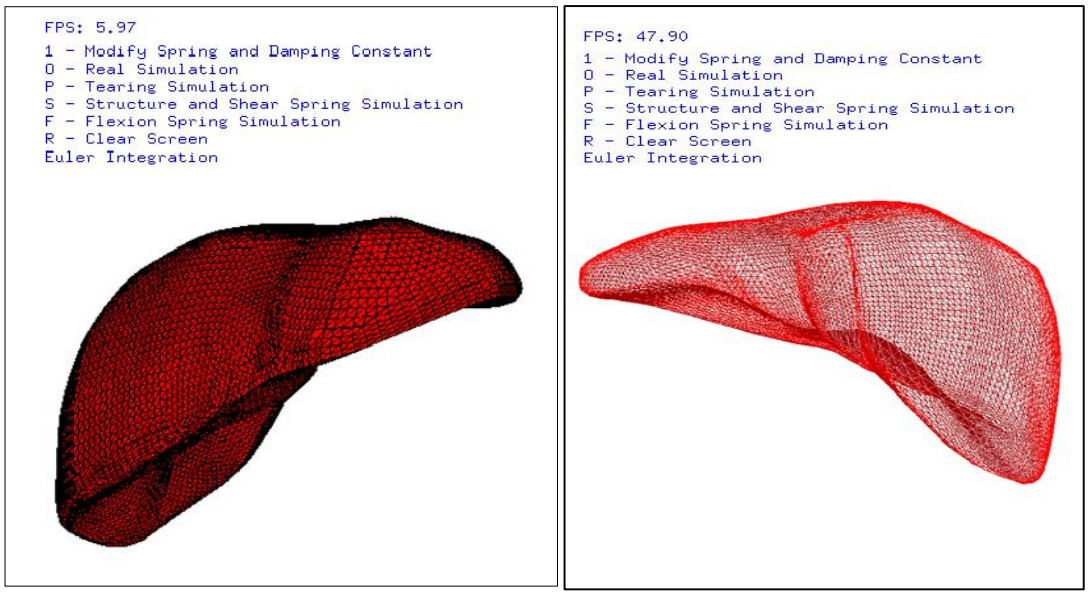


Figure 5.4 CUDA-based Simulation of Liver Deformation.

5.3 CUDA Findings and Discussion

In this research, the human liver which consists of 8466 vertices is modelled by using Mass Spring Model (MSM) as shown in Figure 5.4. The liver stiffness are obtained from the Interval Type-2 Fuzzy Inference System (IT2 FIS) based on the five randomly selected rules. The liver stiffness coefficients are then plugged into the source codes to render the liver deformation.

During the simulation, the CUDA based GPU renders nine time faster than the CPU based modern OpenGL. As illustrated in Figure 5.1, the frame per seconds (fps) in CUDA rendering is in the range of 35 to 40 fps, while the CPU based modern OpenGL rendering is lower than 5 fps. The liver deformation rendered in the CUDA is much smoother, real-time and realistic than the modern OpenGL. Within every 60 seconds, the CUDA based simulation could iterate up to 2000 frames as shown in Figure 5.2. Whereas the CPU based modern OpenGL could only iterate up to 300 frames within 60 seconds.

Here, the adaptive timestep is implemented in the Euler integration method to compute the next positions. As shown in Figure 5.3, the CUDA based GPU simulation obtained the stable timestep within 0 seconds per more than 2000 iterations. In contrast, the CPU based modern OpenGL obtained the stable timestep in the range of 24 to 36 seconds per less than 300 iterations.

Basically, the modern OpenGL is used to develop graphics applications. However, due to the memory allocation in the CPU, the OpenGL is more constrained than the CUDA. Especially, the real-time complex biomedical computations of the liver deformation. CUDA based GPU simulation, on the other hand, is designed from the ground-up for efficient general computation on GPUs.

This enables our algorithm to process liver deformation efficiently using CUDA while maintaining significant realism. Also OpenGL stores data in textures, which requires packing long arrays into 2D textures. This is cumbersome and imposes extra computational burden on the CPU whereas CUDA can instantaneously compute the liver deformation in seconds.

Generally, the CUDA programming involves running code on two different platforms simultaneously, such that a host system with CPUs and CUDA-enabled NVIDIA GPU devices. CUDA exposes several hardware features which are lacking in the development of graphics application via modern OpenGL. One of the most fundamental feature is by using blocks of threads, CUDA can access to the shared memory through parallel programming. This shared memory allows caching of frequently used data, thus the simulation is speedups compared to the modern OpenGL which retrieve data from textures for each iteration repeatedly. Meanwhile, the thread synchronization primitive blended in CUDA allows cooperative parallel processing of

on-chip data. Therefore, the expensive off-chip bandwidth requirements of many parallel algorithms has tremendously reduced.

Execution pipelines on host systems can uphold a limited number of simultaneous threads. Hosts that have four hex-core processors today can execute only 24 threads simultaneously or 48 threads if the CPUs support Hyper-Threading. By comparing the smallest executable unit of correspondence on a CUDA device comprises 32 threads, threads on a CPU are generally heavyweight entities. In order to perform multithreading capability, the Operating System (OS) is required to swap threads on and off in the CPU execution channels. These context switches are simply time-consuming and computationally expensive. As shown in the previous section, the modern OpenGL running on the CPU is much slower than CUDA based GPU simulation. This is because in executing the modern OpenGL simulation, there are thousands of threads queued up for work, but it has to wait until the previous thread is solved only then it can proceed to the next queue.

On the contrary, threads on GPUs are extremely lightweight. Even though there are thousands of threads queuing up for work, it simply begins executing work on another without waiting the previous thread to be solved. This is because there is no swapping of registers occur as the separate registers have been allocated to all active GPU threads. The resource will de-allocate from each thread once it finished its implementation. Therefore, regardless on the complexity of the liver model, the speed of numerical computation in CUDA based GPU simulation is significantly faster than the modern OpenGL.

5.4 Summary

Graphics processing is parallel in nature which is not handled efficiently by CPUs that process data serially. In this research, we present a case study with improved Mass Spring Model (MSM) parameter selection using Interval Type-2 Fuzzy Inference System (IT2 FIS) is presented. We utilized the stiffness values obtained from IT2 FIS to the resultant model in CUDA to generate VR models for liver deformation. Our results show that when using GPU assisted hardware, the VR model can handle increased structural complexity while providing significantly realistic and real-time liver deformation.

CUDA enables real-time liver deformation as the corresponding system iterates more than 2000 frames per 60 seconds which means it took 35 to 40 frame per seconds (fps) during simulation. Meanwhile, the CPU based modern OpenGL could only iterates 300 frames per 60 seconds, which is less than 5fps. Therefore, the liver deformation running on CPU is not real-time and unrealistic. Besides, the CPU based modern OpenGL required 24 to 36 seconds to achieve a converging timestep in Euler integration. Whereas the CUDA based GPU required 0.00 (correct to 2 d.p) seconds to achieve a converging timestep.

Other than CUDA, there are two other useful parallel computing frameworks which are known as OpenCL (Computing Language) and OpenCV (Computer Vision). Similar to CUDA, OpenCL allows parallel programming on the host to launch kernels on GPUs and managing the corresponding memory allocation. Although it is still new, it provides a multi-vendor framework for doing parallel calculations. Further read up on OpenCL is encouraged if CUDA does not meet your needs. OpenCV, on the other

hand, is a host-level application programming interface (API) which is designed for the ease of use and does not require any knowledge of CUDA.

In future, we will compare our CUDA based liver model on OPENCL and OPENCV for an augmented reality model for liver deformation.

Chapter 6 *Conclusion*

6.1 Conclusion

6.2 Limitation

6.3 Future Works

6.1 Conclusion

In this study, the fuzzy approaches are proved to be powerful approaches in building complex and nonlinear relationship between a set of input and output data. The stiffness values estimated by fuzzy approaches are in very good agreement with the benchmark result. The corresponding graphs for each of the fuzzy approaches share the similar trend of displacement and velocity with the benchmark model. Among three of the fuzzy approaches, the Interval Type-2 FIS has the highest similarity with the benchmark model.

Graphics processing is parallel in nature which is not handled efficiently by CPUs that process data serially. In this research, we present a case study with improved Mass Spring Model (MSM) parameter selection using Interval Type-2 Fuzzy Inference System (IT2 FIS). We utilized the stiffness values obtained from IT2 FIS to the resultant model in CUDA to generate VR models for liver deformation. Our results show that when using GPU assisted hardware, the VR model can handle increased structural complexity while providing significantly realistic and real-time liver deformation.

CUDA enables real-time liver deformation as the corresponding system iterates more than 2000 frames per 60 seconds which means it took 35 to 40 frame per seconds (fps) during simulation. Meanwhile, the CPU based modern OpenGL could only iterates 300 frames per 60 seconds, which is less than 5fps. Therefore, the liver deformation running on CPU is not real-time and unrealistic. Besides, the CPU based modern OpenGL required 24 to 36 seconds to achieve a converging timestep in Euler integration. Whereas the CUDA based GPU required 0.00 (correct to 2 d.p) seconds to achieve a converging timestep.

6.2 Limitation

The outputs of FMPC is directly dependent upon the interpretation of the available knowledge. Therefore, extra care should be taken while selecting the membership function ranges as well as the reasoning mechanism for each rule. Furthermore, due to the limitation of liver stiffness data, there are only 247 rules could be generated.

6.3 Future Works

For future development, the Interval Type-2 FIS shall be improved in terms of real-time efficiency. More professional knowledge shall be added to further improve the accuracy of the fuzzy inference system. At the same time, the FMPC scheme shall be extended into real-time invasive liver surgical training simulations in order to validate its practicality.

Other than CUDA, there are two other useful parallel computing frameworks which are known as OpenCL (Computing Language) and OpenCV (Computer Vision). Similar to CUDA, OpenCL allows parallel programming on the host to launch kernels on GPUs and managing the corresponding memory allocation. Although it is still new, it provides a multi-vendor framework for doing parallel calculations. Further read up on OpenCL is encouraged if CUDA does not meet your needs. OpenCV, on the other hand, is a host-level application programming interface (API) which is designed for the ease of use and does not require any knowledge of CUDA.

In future, we will compare our CUDA based liver model on OPENCL and OPENCV for an augmented reality model for liver deformation.

References

Abdelazim, T. and Malik, O. P. An adaptive power system stabilizer using on-line self-learning fuzzy systems. In *Proc. IEEE Power Engineering Society General Meeting*, Toronto, ON, Canada, 2003, pp. 1715–1720.

Abrahamson, S., Denson, J., and Wolf, R., Effectiveness of a Simulator in Training Anesthesiology Residents. 2004, Quality and Safety in Health Care. P. 395-397.

Aebersold, M., Tschannen, D. & Bathish, M. 2012. Innovative simulation strategies in education. *Nursing Research and Practice* : 7.

Ahn, B. & Kim, J. 2010. Measurement and characterization of soft tissue behavior with surface deformation and force response under large deformations. *Medical Image Analysis*, **14**(2): 138–148.

An, D.S. 2011. *Designing deformable models of soft tissue for virtual surgery planning and simulation using the Mass-Spring Model*. PhD. Disertation. University of Navarra.

Aoul, Y. H., Nafaa, A., Negru, D. and Mehaoua, A. FAFC: Fast adaptive fuzzyAQMcontroller for TCP/IP networks. In *Proc. IEEE Global Telecom. Conf.*, Dallas, TX, 2004, pp. 1319–1323.

Arora, A. and Sharma, P. Non-invasive diagnosis of fibrosis in non-alcoholic fatty liver disease. Journal of Clinical and Experimental Hepatology. Vol. 2, No. 2, pp. 145-156. 2012

Azar, F.S., Metaxas, D.N., Schnall, M.D. A deformable finite element model of the breast for predicting mechanical deformations under external perturbations. Acad Radiol 2001;8:965–75. (2001)

Azar, F.S., Metaxas, D.N., Schnall, M.D. Methods for modeling and predicting mechanical deformations of the breast under external perturbations. Med Image Anal 2002;6:1–27. (2002)

Babuska, R. *Fuzzy Modeling for Control*. Boston, MA: Kluwer, 1998.

Bai, Y., Zhuang, H. Q. and Roth, Z. S. Fuzzy logic control to suppress noises and coupling effects in a laser tracking system. *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 1, pp. 113–121, Jan. 2005.

Barach, P. and Johnson, J.K., Reducing Variation in Adverse Events during the Academic Year. British Medical Journal, 2009. 339(1): p. 3949.

Barrero, F., Gonzalez, A., Torralba, A., Galvan, E. and Franquelo, L. G. Speed control of induction motors using a novel fuzzy sliding-mode structure. *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 3, pp. 375–383, Jun. 2002.

Barry-Issenberg, S., McGaghie, W., Petrusa, E., et al., Features and Uses of High-fidelity Medical Simulations that Lead to Effective Learning: A BEME Systematic Review. Medical Teacher, 2005. 27(1): p. 10-28.

Basafa, E. & Farahmand, F. 2011. Real-time simulation of the nonlinear visco-elastic deformations of soft tissues. *International Journal of Computer Asisted Radiology and Surgery*, **6**: 297–307.

based deformable models in computer graphics. *Eurographics State of the Art Report (2005)*, pp. 71-94.

Baturone, I., Moreno-Velo, F. J., Sanchez-Solano, S. and Ollero, A. Automatic design of fuzzy controllers for car-like autonomous robots. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 447–465, Aug. 2004.

Baumann, R. & Glauser, D., Tappy, D., Baur, C. & Clavel, R. 1996. Force feedback for virtual reality based minimally invasive surgery simulator. *Health Technology and Informatics*, **29**: 564-579.

Baur, C., Guzzoni, D. and Georg, O. "Virgy: A virtual reality surgical trainer and force feedback based endoscopic surgery simulator", Medicine Meets Virtual Reality (MMVR 6), 1998.

Bellman, R. E. and Zadeh, L. A. Decision making in a fuzzy environment. *Manage. Sci.*, vol. 17, pp. 141–164, 1970.

Berkley, J., Turkiyyah, G., Berg, D. Real-time Finite Element Modelling for Surgery Simulation: An application to virtual suturing. IEEE Transactions on Visualization and Computer Graphics, 2004. 10(3): p. 314-325.

Bezdek, J. C., Keller, J. M., Krishnapuram, R. and Pal, N. R. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Boston, MA: Kluwer, 1999.

Bhat, K. S., Twigg, C. D., Hodgins, J. K., Khosla, P. K., Popovic, Z. and Seitz, S. M. Estimating cloth simulation parameters from video. In: Proceedings ACM SIGGRAPH/ Eurographics Symposium on Computer Animation, D. Breen and M. C. Lin, Eds., 2003, pp. 37-51.

Bianchi, G., Harders, M., Szekely, G. Mesh topology identification for mass-spring models. In: MICCAI 2003, Vol. 1, pp. 50-58. 200

Bianchi, G., Solenthaler, B., Szekely, G. and Harders, M. 2004. Simultaneous Topology and Stiffness Identification for Mass-Spring Models Based On FEM Reference Deformations. *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2004)*, **2**:293-301.

Bianchi, G., Solenthaler, B., Szekely, G., Harders, M. Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. In: MICCAI 2004, Vol. 2, pp. 293-301. 2004

Bielser, D. and Gross, M.H., "Interactive simulation of surgical cuts." Proceedings of Pacific Graphics 2000, IEEE Computer Society Press, pp. 116-125, 2000.

Bonissone, P. P., Badami, V., Chiang, K. H., Khedkar, P. S., Marcelle, K. W. and Schutten, M. J. Industrial applications of fuzzy logic at General Electric. *Proc. IEEE*, vol. 38, no. 3, pp. 450–465, Mar. 1995.

Boroushaki, M., Ghofrani, M. B., Lucas, C. and Yazdanpanah, M. J. Identification and control of a nuclear reactor core (VVER) using recurrent neural networks and fuzzy systems. *IEEE Trans. Nucl. Sci.*, vol. 50, no. 1, pp. 159–174, Feb. 2003.

Bosdogan, C., Ho, C. and Srinivasan, M.A. "Virtual Environments for Medical Training: Graphical and haptic simulation of laparoscopic common bile duct exploration", IEEE/ ASME Transactions on Mechatronics, vol. 6, pp. 269-285, 2001.

Boukezzoula, R., Galichet, S. and Foulloy, L. Observer-based fuzzy adaptive control for a class of nonlinear systems: Real-time implementation for a robot wrist. *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 340–351, May 2004.

Bro-Nielson, M. 1996. Surgery simulation using fast finite elements. In: VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing. London, UK: Springer-Verlag, 1996, pp. 529-534.

Bro-Nielson, M. 1997. Fast finite elements for surgery simulation. In: Proceedings 5th Medicine Meets Virtual Reality, 1997.

Buck, I., Foley, T., Horn, D., et al., Brook for GPUs: Stream Computing on Graphics Hardware. ACM Transactions on Graphics, 2004. 23(3): p. 777-786.

Campello, R. J. G. B., Meleiro, L. A. C. and Amaral, W. C. Control of a bioprocess using orthonormal basis function fuzzy models. in *Proc. IEEE Int. Conf. Fuzzy Systems*, Budapest, Hungary, 2004, pp. 801–806.

Castera, L., Pinzani, M. and Bosch, J. Non invasive evaluation of portal hypertension using transient elastography. Journal of Hepatology, Vol. 56, pp. 696-703. 2012

Castera, L., Sebastiani, G., Le Bail, B., de Ledinghen, V., Couzigou, P. and Alberti, A. Prospective comparison of two algorithms combining non-invasive methods for staging liver fibrosis in chronic hepatitis C. Journal of Hepatology, Vol. 52, pp. 191-198, 2010.

Castero, L. Non-invasive assessment of liver fibrosis in chronic hepatitis C. Hepatology International, Vol. 5, pp 625-634. 2011

Chang Y. C. and Chen, B. S. Intelligent robust tracking controls for holonomic and nonholonomic mechanical systems using only position measurements. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 491–507,Aug. 2005.

Chen B. and Liu, X. Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by backstepping and application to chemical processes. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 832–847, Dec.2005.

Chen, B. S., Tsai, C. L. and Chen, D. S. Robust H and mixed H =H filters for equalization designs of nonlinear communication systems: Fuzzy interpolation approach," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 3, pp. 384–398, Jun. 2003.

Chen, B. S., Yang, Y. S., Lee, B. K. and Lee, T. H. Fuzzy adaptive predictive flow control of ATM network traffic," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 568–581, Aug. 2003.

Chen, C. L., Feng, G., Sun, D. and Zhu, Y. H-infinity output feedback control of discrete-time fuzzy systems with application to chaos control. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 531–543, Aug. 2005.

Chiu, S., Chand, S., Moore, D. and Chaudhary, A. Fuzzy logic for control of roll and moment for a flexible wing aircraft. *IEEE Control Syst. Mag.*, vol. 11, no. 1, pp. 42–48, Jan. 1991.

Choi, K.S., 2010. Toward Realistic Virtual Surgical Simulation: Using Heuristically Parametrized Anisotropic Mass-Spring Model to Simulate Tissue Mechanical Responses. *Education Technology and Computer (ICETC), 2010 2nd International Conference*, 22-24 June 2010, pp. V1-446- V1-450.

Chung, J.H., Rajagopal, V., Nielsen, P.M.E., Nash, M.P. A biomechanical model of mammographic compressions. Biomech Model Mech 2008;7:43–52. (2008)

Cooke, M., Irby, D., Sullivan, W., et al., American Medical Education 100 Years after the Flexner Report. New England Journal of Medicine, 2006. 355: p. 1339-44.

Corpechot, C., El Naggar, A., Poujol-Robert, A., Ziol, M., Wendum, D., Chazouilleres, O., de Ledingher, V., Dhumeaux, D., Marcelling, P., Beaugrand, M. and Poupon, R. Assessment of biliary fibrosis by transient elastography in patients with PBC and PSC. Journal of Hepatology, Vol. 43, No. 5, pp. 1118-1124. 2006

Cotin, S., Delingette, H., and Ayache, N. 1999. Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation. IEEE Transactions on Visualization and Computer Graphics, 5:62-73. (1999)

Courtecuisse, H. & Jung, H. 2010. GPU-based real-time soft tissue deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology*, **3**: 159–168.

Courtecuisse, H. & Jung, H. 2010. GPU-based real-time soft tissue deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology*, **3**: 159–168.

Cover, S. A., Ezquerra, N. F., O'Brien, J. F., Rowe, R., Gadacz, T. and Palm, E. 1993. Interactively Deformable Models for Surgery Simulation. IEEE Computer Graphics and Application, 13: 6: 68-75. (1993)

De Ledinghen, V., Douvin, C., Kettaneh, A., Ziol, M., Roulot, D., Marcellin, P., Dhumeaux, D. and Beaugrand, M. Diagnosis of hepatic fibrosis and cirrhosis by transient elastography in HIV/ Hepatitis C Virus—Coinfected patients. Journal

of Acquire Immune Deficient Syndrome, Vol. 41, No. 2, pp. 175-179. February 2006.

De Ledinghen, V., Wong, W.S., Vergniol, J., Wong, L.H., Foucher, J., Chu, H.T., Le Bai, B., Choi, C.L., Chermak, F., Yiu, K.L., Merrouche, W. and Chan, L.Y. Diagnosis of liver fibrosis and cirrhosis using liver stiffness measurement: comparison between M and XL probe of FibroScan®. Journal of Hepatology, Vol. 56, pp. 833-839. 2012

De, S. and Srinivasan, M.A. 1999. Thin walled models for haptic and graphical rendering of soft tissues in surgical simulations. *Medicine Meets Virtual Reality, J.D. Westwood et al. (Eds) IOS Press, 1999*, 94-99.

Delingette, H. & Ayache, N. 2004. Soft Tissue Modeling for Surgery Simulation. *Computational Models for the Human Body*, **12**: 453-550.

Delingette, H., Cotin, S. & Ayache, N. 1999. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *Computer animation:* 1-12.

Delp, S., Loan, P., Basdogan, C. & Rosen, J. 1997. Surgical simulation: an emerging technology for training in emergency medicine. *Teleoperators and Virtual Environments*, 6(2): 147-159.

Deussen, O., Kobbelt, L. and Tücke, P. 1995. Using Simulated Annealing to Obtain Good Approximations of Deformable Bodies. *Proceedings of the EuroGraphics Workshop Computer Animation and Simulation.* D. Terzopoulost and D.Thalmann, Eds. New York: Springer-Verlag.

DiMaio, S. P. and Salcudean, S. E. 2002. Simulated interactive needle insertion. In: Proceeding 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002, pp. 344-351.

Dutta, S. Fuzzy logic applications: Technological and strategic issues. *IEEE Trans. Eng. Manage.*, vol. 40, no. 3, pp. 237–254, Aug. 1993.

Er, M.J. and Sun, Y.L. Hybrid fuzzy proportional integral plus conventional derivative control of linear and nonlinear system, IEEE Transactions on Industrial Electronics 2001;48(6):1109-1117.

Etheredge, C.E. A Parallel Mass-Spring Model for Soft Tissue Simulation with Haptic Rendering in CUDA. 15th Twente Student Conference on IT 15 (2011).

Farinwata, S. S., Pirovolou, D. and Vachtsevanos, G. J. On input-output stability analysis of a fuzzy controller for a missile autopilot's yaw axis. In *Proc. 3rd IEEE Int. Conf. Fuzzy Systems*, Orlando, FL, 1994, pp. 930–935.

Feng, G. 2006. A survey on analysis and design of model-based fuzzy control systems. IEEE Transactions on Fuzzy Systems, Vol. 14, no. 5, October 2006.

Flores, A., Saez, D., Araya, J., Berenguel, M. and Cipriano, A. Fuzzy predictive control of a solar power plant. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 58–68, Feb. 2005.

Frey, C. W. and Kuntze, H. B. A neuro-fuzzy supervisory control system for industrial batch processes. *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 570–577, Aug. 2001.

Frulio, N. and Trillaud, H. Ultrasound elastography in liver. Diagnostic and Interventional Imaging, Vol. 94, pp, 515-534. 2013

Fung, J., Lai, C.L., Chan, S.C., But, D., Seto, W.K., Cheng, C. et al. Correlation of liver stiffness and histological features in healthy persons and in patients with occult hepatitis B, Chronic Active Hepatitis B, or Hepatitis B Cirrhosis. The American Journal of Gastroenterology, Vol. 105, pp. 1116-1122. May 2010.

Fung, Y. C. 1993. Biomechanics: mechanical properties of living tissues. New York, *NY, USA:* Springer-Verlag.

Fung, Y.Y., Lai, C.L. and Yuen, M. F. Clinical application of transient elastography (Fibroscan®) in liver diseases. The Hong Kong Medical Diary Medical Bulletin, Vol. 14, No. 11, pp. 22-25, Nov 2009.

Garland, M., Le Grand, S., Nickolls, J., et al., Parallel Computing Experiences with CUDA. IEEE Micro, 2008. 28(4): p. 13-27.

Georgii, Joachim, and Rüdiger Westermann. Mass-Spring Systems on the GPU. Simulation Modelling Practice and Theory 13, no. 8 (11// 2005): 693-702. (2005)

Grantcharov, T., Kristiansen, V., Bendix, J., et al., Randomized Clinical Trial of Virtual Reality Simulation for Laparoscopic Skills Training. British Journal of Surgery, 2004. 91(2): p. 146-150.

Guesmi, T., Adballah, H. H. and Toumi, A. Transient stability fuzzy control approach for power systems. In *Proc. IEEE Int. Conf. Industrial Technology*, Hammamet, Tunisia, 2004, pp. 1676–1681.

Guillemin, P. Fuzzy logic applied to motor control. *IEEE Trans. Ind. Appl.*, vol. 32, no. 1, pp. 51–56, Jan. 1996.

Hagras, H. A. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, Aug. 2004

Halic, T., Kockara, S., Bayrak, C., Rowe, R. & Chen, B. 2009. Soft tissue deformation and optimized data structures for mass spring methods. *International Conference on Bioinformatics and BioEngineering* : 45–52.

Haller, G., Myles, P.S., Taffe, P., et al., Rate of Undesirable Events at Beginning of Academic Year: Retrospective Cohort Study. British Medical Journal, 2009. 339(1): p. 3974.

Hammer, P.E., Sacks, M.S., Nido, P.J. & Howe, R.D. 2011. Mass-spring model for simulation of heart valve tissue mechanical behavior. *Annals of biomedical engineering*, **39**(6): 1668–79.

Haruki, T. and Kikuchi, K. Video camera system using fuzzy logic. *IEEE Trans. Consumer Electron.*, vol. 38, no. 3, pp. 624–634, Aug. 1992.

Hong, S. K. and Langari, R. Robust fuzzy control of a magnetic bearing system subject to harmonic disturbances. *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 2, pp. 366–371, Mar. 2000

Horiuchi, J. I. and Kishimoto, M. Application of fuzzy control to industrial bioprocesses in Japan. *Fuzzy Sets Syst.*, vol. 128, no. 1, pp. 117–124, May 2002.

Hu, T. 2006. Reality-based Soft Tissue Probing : Experiments and Computational Model for Application to Minimally Invasive Surgery. Dissertation for Degree of Doctor of Philosophy. Drexel University.

Huang, S. J. and Lin, W. C. Adaptive fuzzy controller with sliding surface for vehicle suspension control," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp. 550–559, Aug. 2003.

Hwang, C. L. and Kuo, C. Y. A stable adaptive fuzzy sliding-mode control for affine nonlinear systems with application to four-bar linkage systems. *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 2, pp. 238–252, Apr. 2001.

Indelicato, D. 1995. Virtual Reality in Surgical Training, 21–24.

Jeremy Cobbold, F.L. and Simon Taylor-Robinson, D. Liver stiffness values in healthy subjects: implications for clinical practice. Journal of Hepatology, Vol. 48, pp. 529-531, 2008.

Jojic, N. and Huang, T. S. Estimating cloth draping parameters from range data. In: Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, 1997, pp. 73-76.

Juang, C. F. and Hsu, C. H. Temperature control by chip-implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm. *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 11, pp. 2376–2384, Nov. 2005.

Kadmiry B. and Driankov, D. A fuzzy gain-scheduler for the attitude control of an unmanned helicopter. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 502–515, Aug. 2004.

Kandel, A., Manor, O., Klein, Y. and Fluss, S. ATMtraffic management and congestion control using fuzzy logic. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 3, pp. 474–480, Aug. 1999.

Kerdok, A. E. 2006. Characterizing the nonlinear mechanical response of liver to surgical manipulation. Dissertation for Degree of Doctor of Philosophy. Harvard University, *Massachusetts.*

Kickert, W. J. M. and Van Nauta Lemke, H. R. Application of a fuzzy logic controller in a warm water plant," *Automatica*, vol. 12, pp. 301–308, 1976.

Kiguchi, K., Tanaka, T. and Fukuda, T. Neuro-fuzzy control of a robotic exoskeleton with EMG signals. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 481–490, Aug. 2004.

Kim C. J. 1997. An algorithmic approach for fuzzy inference. IEEE Transactions on Fuzzy Systems, Vol. 5, No. 4, November 1997.

Kim, B.K., Fung, J., Yuen, M.F. and Kim, S.U. Clinical application of liver stiffness measurement using transient elastography in chronic liver disease from longitudinal perspectives. World Journal of Gastroenterology. Vol. 19, No. 12, pp. 1890-1900. 2013

Kim, E. and Lee, S. Output feedback tracking control of MIMO systems using a fuzzy disturbance observer and its application to the speed control of a PM synchronous motor. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 725–741, Dec. 2005.

Kim, E. Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 3, pp. 368–378, Jun. 2004.

Kim, S.U., Choi, G.H., Han, W.K., Kim, B.K., Park, J.Y., Kim, D.Y., Choi, J.S., Yang, S.C., Choi, E.H., Ahn, S.H., Han, K.H. and Chon, C.Y. What are 'true normal' liver stiffness values using Fibroscan®? : a prospective study in healthy living liver and kidney donors in South Korea. Liver International, Vol. 30, Issue: 2, pp. 268-274, Feb 2010.

King, P. J. and Mamdani, E. H. The application of fuzzy control systems to industrial process. *Automatica*, vol. 13, pp. 235–242, 1977

Ko, H. S. and Niimura, T. Power system stabilization using fuzzy neural hybrid intelligent control. In *Proc. IEEE Int. Symp. Intelligent Control*, Vancouver, BC, Canada, 2002, pp. 879–884.

Kornblum R. J. and Tribus, M. The use of Bayesian inference in the design of an endpoint control system for the basic oxygen steel furnace," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 2, pp. 339–348, Mar./Apr. 1970.

Koska, B. *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

Kuhnapfel, U. G. & Neisius, B. 1993. CAD-based graphical computer simulation in endoscopic surgery. *Endoscopic Surgery*, **1**: 369-378.

Kumar, R., Rastogi, A., Sharma, M.K., Bhatia, V., Tyagi, P., Sharma, P., Garg, H., Kumar, K.N.C., Bihari, C. and Sarin, S.K. Liver stiffness measurements in patients with different stages of nonalcoholic fatty liver disease: diagnosis performance and clinicopathological correlation. Dig. Dis. Sci. Vol. 58, pp. 265-274. 2013

Kumar, S. A review of smart volume controllers for consumer electronics. *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp. 600–605, May 2005.

Kwok, H. F., Linkens, D. A., Mahfouf, M. and Mills, G. H. SIVA: A hybrid knowledge-and-model-based advisory system for intensive care ventilators," *IEEE Trans. Inform. Technol. Biomed.*, vol. 8, no. 2, pp. 161–172, Jun. 2004.

Larkin, L. I. A fuzzy logic controller for aircraft flight control. In *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam, The Netherlands: North-Holland, 1985, pp. 87–104.

Larsen, P. M. Industrial applications of fuzzy logic control. *Int. J. Man Mach. Stud.*, vol. 12, pp. 3–10, 1980.

Lee, C. C. 1990a. Fuzzy logic in control systems: Fuzzy logic controller—Part I. IEEE Trans. Syst., Man, Cybern, vol. 20, no. 2, pp. 404-418, Mar/ Apr. 1990.

Lee, C. C. 1990b. Fuzzy logic in control systems: Fuzzy logic controller—Part II. IEEE Trans. Syst, Man, Cybern, vol. 20, no. 2, pp. 419-435, Mar/ Apr. 1990.

Lee, S. H. and Bien, Z. Design of expandable fuzzy inference processor. *IEEE Trans. Consumer Electron.*, vol. 40, no. 2, pp. 171–175, May 1994.

Lee, S. H. and Lim, J. T. Multicast ABR service in ATM networks using a fuzzy-logic-based consolidation algorithm," *Proc. Inst. Elect. Eng.—Commun.*, vol. 148, pp. 8–13, 2001.

Lee, W., The Acquisition of Clinical Ward Skills during Undergraduate Medical Training. Journal of Medical Education, 1980. 55(12): p. 1029-31.

Lee, Y. M., Jang, S. I., Chung, K. W., Lee, D. Y., Kim, W. C. and Lee, C. W. A fuzzy-control processor for automatic focusing. *IEEE Trans. Consumer Electron.*, vol. 40, no. 2, pp. 138–144, May 1994.

Leon, C. A. D., S. Eliuk, and H. T. Gomez. Simulating Soft Tissues Using a GPU Approach of the Mass-Spring Model. Paper presented at the Virtual Reality Conference (VR), 2010 IEEE, 20-24 March 2010 (2010)

Li, T. H. S., Chang, S. J. and Tong, W. Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 491–501, Aug. 2004.

Li, W., Chang, X. G., Farrell, J. and Wahl, F. M. Design of an enhanced hybrid fuzzy P+ID controller for a mechanical manipulator. *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 31, no. 6, pp. 938–945, Dec. 2001.

Lian, K. Y., Chiu, C. S., Chiang, T. S. and Liu, P. LMI-based fuzzy chaotic synchronization and communications. *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 539–553, Aug. 2001.

Lin, C. M. and Hsu, C. F. Self-learning fuzzy sliding-mode control for antilock braking systems. *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 2, pp. 273–278, Mar. 2003.

Lloyd, B., Székely, G. & Harders, M. 2007. Identification of spring parameters for deformable object simulation. *Transactions on Visualization and Computer Graphics*, **13**(5): 1081–94.

Machado, M. and Pinto. H.C. Non-invasive diagnosis of non-alcoholic fatty liver disease. A critical appraisal. Journal of Hepatology. Vol. 58, pp. 1007-1019. 2013

Maciel, A., Boulic, R. and Thalmann, D. 2003. Deformable Tissue Parameterized by Properties of Real Biological Tissue. In: *Proceedings International Symposium on Surgery Simulation and Soft Tissue Modelling*, 2003, pp. 74-87.

Mamdani, E. H. and Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Stud.*, vol. 7, pp. 1–13, 1975.

Mamdani, E. H. Application of fuzzy algorithms for simple dynamic plant. *Proc. Inst. Elect. Eng.*, vol. 121, pp. 1585–1588, 1974.

Mannani, A. and Talebi, H. A. A fuzzy Lyapunov-based control strategy for a macro-micro manipulator. In *Proc. IEEE Conf. Control Applications*, Istanbul, Turkey, 2003, pp. 368–373.

Mar J. and Lin, F. J. An ANFIS controller for the car-following collision prevention system. *IEEE Trans. Veh. Technol.*, vol. 50, no. 4, pp. 1106–1113, Jul. 2001.

Marcellin, P., Ziol, M., Bedossa, P., Douvin, C., Poupon, R., de Ledinghen, V. and Beaugrand, M. Non-invasive assessment of liver fibrosis by stiffness measurement in patients with chronic hepatitis B. Liver International, Vol. 29, Issue 2, pp. 242-247, 2009.

Marchesseau, S., Heimann, T., Chatelin, S., Willinger, R. & Delingette, H. 2010. Fast porous visco-hyperelastic soft tissue model for surgery simulation : Application to liver surgery. *Progress in Biophysics and Molecular Biology*, **103**(2-3): 185–196.

Mendel, J. M. Fuzzy logic systems for engineering: A tutorial. *Proc. IEEE*, vol. 83, no. 2, pp. 345–377, Mar. 1995.

Meseure, P. & Chaillou, C. 1997. Deformable body simulation with adaptative subdivision and cuttings. *Proceedings of the WSCG'97 Conference* : 361-370.

Mikes, G. people.maths.ox.ac.uk/gilesm/hpc.html

Mueller S. and Sandrin, L. Liver stiffness: a novel parameter for the diagnosis of liver disease. Hepatic Medicine Evidence and Research, pp. 49-67, May 2010.

Munasinghe, S. R., Kim, M. S. and Lee, J. J. Adaptive neurofuzzy controller to regulate UTSG water level in nuclear power plants. *IEEE Trans. Nucl. Sci.*, vol. 52, no. 1, pp. 421–429, Feb. 2005.

Murakami, S. and Maeda, M. Application of fuzzy controller to automobile speed control system. In *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam, The Netherlands: North-Holland, 1985, pp. 105–124.

Myers, R.P., Pomier-Layrargues, G., Duarte-Rojo, A., Wong, D.K., Beaton, M.D., Levstik, M.A., Kirsch, R., Pollett, A., Elkashab, M. Performance of the fibroscan XL probe for liver stiffness measurement in obese patients: a multicenter validation study. Journal of Hepatology, Vol. 54, pp. S136-S137. 2011

Myers, R.P., Pomier-Layrargues, G., Kirsch, R., Pollett, A., Beaton, M., Levstik, M., Duarte-Rojo, A., Wong, D., Crotty, P. and Elkashab, M. Discordance in fibrosis staging between liver biopsy and transient elastography using the FibroScan XL probe. Journal of Hepatology, Vol. 56, pp. 564-570. 2012

Nakagaki, N., Bando, Y., Mori, T., Torikoshi, S. and Suzuki, S. Wide aspect TV receiver with aspect detection and non-linear control for picture quality. *IEEE Trans. Consumer Electron.*, vol. 40, no. 3, pp. 743–752, Aug. 1994.

Natsupakpong, S. & Cavusoglu, M. C. 2008. Comparison of numerical integration methods for simulation of physically-based deformable object models in surgical simulation. *National Symposium on Computational Science and Engineering* : 27-29.

Natsupakpong, S. 2010. Physically based modeling and simulation for virtual environment. Dissertation for Degree of Doctor of Philosophy. Case Western Reserve University.

Nealen, A., Muller, M., Keiser, R., Boxerman, E., and Carlson, M. 2005. Physically

Nedel, L. P. & Thalmann, D. 1998. Real time muscle deformations using mass-spring systems. *Computer Graphics International* :156- 165.

Nesme, M., Faure, F. & Yohan, F. 2009. Accurate interactive animation of deformable models at arbitrary resolution. *International Journal of Image and Graphics*, **10**(13): 1-20.

Niasar, A. H., Moghbeli, H. and Kazemi, R. Yaw moment control via emotional adaptive neuro-fuzzy controller for independent rear wheel drives of an electric vehicle. In *Proc. IEEE Conf. Control Applications*, Istanbul, Turkey, 2003, pp. 380–385.

Nürnberger, A., Radetzky, A., Kruse, R. Determination of Elastodynamic Model Parameters using a Recurrent Neuro-Fuzzy System. In Proceedings of the Seventh European Congress on Intelligent Techniques and Soft Computing 1999 (EUFIT ’99).

Nvidia, Computer Unified Device Architecture Programming Guide 2.2. 2008, NVIDIA: Santa Clara, CA.

Ostergaard, J. J. Fuzzy logic control of a heat exchanger process. In *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis, and B. R. Gaines, Eds. Amsterdam, The Netherlands: North-Holland, 1977, pp. 285–320.

Pedrycs, W. *Fuzzy Control and Fuzzy Systems*. Somerset, U.K.: Research Studies Press, Ltd., 1993

Petriu, E.M., Cretu, A., Payeur, P. Neural network modelling techniques for the real-time rendering Of the geometry and elasticity of 3D objects. IEEE International Workshop on Soft Computing Applications 11-16. 2007

Radetzky, A., Nurnberger, A. & Pretschner, P. 2000. Elastodynamic shape modeler: A tool for defining the deformation behavior of virtual tissues. *RadioGraphics*, **20**(1): 865-881.

Rasmusson, A., Mosegaard, J., Sangild, T. Exploring Parallel Algorithms for Volumetric Mass-Spring-Damper Models in CUDA. In Proceedings of the 4th international symposium on Biomedical Simulation, 49-58. London, UK: Springer-Verlag. (2008)

Redd, A.D., Wendel, S.K., Grabowski, M.K., Ocama, P., Kiggundu, P., Bbosa, F., Boaz, I., Balagopal, A., Reynolds, S.J., Gray, R.H., Serwadda, D., Kirk, G.D., Quinn, T.C., Stabinski, L. Liver stiffness is associated with monocyte activation in HIV-infected Ugandans without viral hepatitis. Journal of AIDS Research and Human Retroviruses. Vol. 29, pp. 1026-1030. 2013

Roberts, K., Bell, R., and Duffy, A., Evolution of Surgical Skills Training. World Journal of Gastroenterology, 2006. 12(20): p. 3219.

Robic, M.A., Procopet, B., Metivier, S., Peron, J.M., Selves, J., Vinel, J.P. and Bureau, C. Liver stiffness accurately predicts portal hypertension related complications in patients with chronic liver disease: a prospective study. Journal of Hepatology, Vol.: 55, pp. 1017-1024. 2011

Rosen, R. & Kathleen. 2008. The history of medical simulation. *Journal of Critical Care*, **23**: 157-166.

Roulot, D., Czernichow, S., Le Clesiau, H., Costes, J., Vergnaud, A. and Beaugrand, M. Liver stiffness values in apparently healthy subjects: Influence of gender and metabolic syndrome. Journal of Hepatology, Vol. 48, pp. 606-613, 2008.

Ryoo, S., Rodrigues, C., Baghsorkhi, S., et al. Optimization Principles and Application Performance Evaluation of a Multithreaded GPU using CUDA. In the 13th ACM Symposium on Principles and Practice of Parallel Programming. 2008. Salt Lake City, UT: ACM.

Sala, A., Turini, G., Ferrari, M., Mosca, F. & Ferrari, V. 2011a. Integration of biomechanical parameters in tetrahedral mass-spring models for virtual surgery simulation. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, **4550**(4).

Samani, A., Bishop, J., Yaffe, M.J., Plewes, D.B. Biomechanical 3D finite element modeling of the human breast using MRI data. IEEE Trans Med Imaging 2001;20:271–9. (2001)

San Vincent, O. 2011. Designing deformable models of soft tissue for virtual surgery planning and simulation using the Mass-Spring Model. Dissertation for Degree of Doctor of Philosophy. University of Navarra.

Santibanez, V., Kelly, R. and Llama, M. A. A novel global asymptotic stable set-point fuzzy controller with bounded torques for robot manipulators. *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 362–372, Jun. 2005.

Satava, R. (1998) Medical virtual reality: The current status of the future. In: Proc. 4th Conf. Medicine Meets Virtual Reality (MMRV IV), San Diego, CA, 1996, pp. 100-106.

Schaverien, M.V. 2010. Development of expertise in surgical training. *Journal of surgical education*, **67**(1): 37–43.

Seker, H., Odetayo, M. O., Petrovic, D. and Naguib, R. N. G. A fuzzy logic based-method for prognostic decision making in breast and prostate cancers. *IEEE Trans. Inform. Technol. Biomed.*, vol. 7, no. 2, pp. 114–122, Jun. 2003.

Seymour, N., Gallagher, A., Roman, S., et al., Virtual Reality Training Improves Operating Room Performance. Annals of Surgery, 2002. 236(4): p. 458-464.

Shen, J. and Lipasti, M., Modern Processor Design: Fundamentals of Superscalar Processors. 2003: McGraw-Hill.

Smith, M. L. Sensors, appliance control, and fuzzy logic. *IEEE Trans. Ind. Appl.*, vol. 30, no. 2, pp. 305–310, Mar./Apr. 1994.

Srinivasan, S., Mital, D.P., Haque, S. A quantitative analysis of the effectiveness of laparascopy and endoscopy virtual reality simulators. Comput Electr Eng 2006:283–98 (2006)

Stone, J., Phillips, J., Freddolino, P., et al., Accelerating Molecular Modelling Applications with Graphics Processors. Journal of Computational Chemistry, 2007. 28(16): p. 2618.

Sugeno M. and Nishida, M. Fuzzy control of model car. *Fuzzy Sets Syst.*, vol. 16, pp. 103–113, 1985.

Sugeno, M. and Yasukawa, T. A fuzzy-logic-based approach to qualitative modelling. *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.

Sugeno, M. *Industrial Applications of Fuzzy Control*. New York: Elsevier, 1985.

Sun, Y. L. and Er, M. J. Hybrid fuzzy control of robotics systems. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 755–765, Dec. 2004.

Szabo, B. A. 1991. Finite Element Analysis. New York: Wiley, 1991.

Takagi, H. Application of neural networks and fuzzy logic to consumer products. In *Proc. Int. Conf. on Industrial Electronics, Control, Instrumentation, and Automation*, San Diego, CA, Nov. 1992, pp. 1629–1633.

Tani, T., Murakoshi, S. and Umano, M. Neuro-fuzzy hybrid control system of tank level in petroleum plant. *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 360–368, Aug. 1996.

Tejada, E. and Ertl, T. Large Steps in GPU-Based Deformable Bodies Simulation. In Simulation Modelling Practice and Theory, 703–15: Elsevier. (2005)

Tendick, F., Downes, M., Goktekin, T., Cavusoglu, M.C., Feygin, D., Wu, X.L. A virtual environment testbed for training laparoscopic surgical skills. Presence 2000;9(3):236–55. (2000)

Teodorescu, H. N., Jain, L. C. and Kandel, A. *Fuzzy and Neuro-Fuzzy Systems in Medicine*. Boca Raton, FL: CRC, 1998.

Terzopoulus, D. and Witkin, A.P. 1988. Deformable Models. *IEEE CGA (8),* **6**:41-51.

Terzopoulus, D., Platt, J., Barr, A., and Fleischer, K. 1987. Elastically deformable models. SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 205-214. (1987)

Terzopoulus, D., Platt, J., Barr, A., and Fleischer, K. 1987. Elastically deformable models. *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 205-214

Teschner, M., Heidelberger, B., Muller, M. & Gross, M. 2004. A versatile and robust model for geometrically complex deformable solids. *Proceedings of the Computer Graphics International* : 312-319.

The LaparoscopyVR Virtual-Reality System. Immersion. <http://www.immersion.com/markets/medical/products/laparoscopy/index.html >.

Tong, R. M., Beck, M. B. and Latten, A. Fuzzy control of the activated sludge wastewater treatment process. *Automatica*, vol. 6, pp. 695–701, 1980.

Tsourdos, A., Economou, J. T., White, A. B. and Luk, P. C. K. Control design for a mobile robot: A fuzzy LPV approach. in *Proc. IEEE Conf. Control Applications*, Istanbul, Turkey, 2003, pp. 552–557.

Umbers, I. G. and King, P. J. An analysis of human-decision making in cement kiln control and the implications for automation. *Int. J. Man Mach. Stud.*, vol. 12, pp. 11–23, 1980.

Umut Ozcan, M., Ocal, S., Basdogan, C., Dogusoy, G. and Tokat, Y. Characterization of frequent-dependent material properties of human liver and its pathologies using an impact hammer. Medical Image Analysis, Vol.: 15, pp. 45-52. 2011

Uragami, M., Mizumoto, M. and Tanaka, K. Fuzzy robot controls. *Cybern.*, vol. 6, pp. 39–64, 1976.

Van Gelder, A. 1998. Approximate Simulation Of Elastic Membranes By Triangulated Spring Meshes. *J. Graphics tools*, **3**:21-42.

Vollinger, U., Setier, H., Priesnitz, J. & Krause, F. L. 2009. Evolutionary optimization of mass-spring models. *Journal of Manufacturing Science and Technology*, **1**(3): 137-141.

Vozelinek, J., Huff, J., and Reznek, M., See One, Do One, Teach One: Advanced Technology in Medical Education. Academic Emergency Medicine, 2004. 11(11): p. 1149-1154.

Wai, R. J. and Chen, P. C. Intelligent tracking control for robot manipulator including actuator dynamics via TSK-type fuzzy neural network. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 552–560, Aug. 2004.

Wang, J. S. and Lee, C. S. G. Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle. *IEEE Trans. Robot. Automat.*, vol. 19, no. 2, pp. 283–295, Apr. 2003

Webster, R., Haluck, R.S., Ravenscroft, R., Mohler, B., Crouthamel, E., Frack, T. Elastically deformable 3D organs for haptic surgical simulation. Med Meets Virtual Real 2002:570–2. (2002)

Weller, J., Robinson, B., Larsen, P., et al., Simulation-Based Training to improve Acute Care Skills in Medical Undergraduates. Journal of the New Zealand Medical Association, 2004. 117: p. 1204.

Wong, W.S., Verginol, J., Wong, L.H., Foucher, J., Chan, L.Y., Le Bail, B., Choi, C.L., Kowo, M., Chan, W.H., Merrouche, W., Sung, J.Y., de Ledinghen, V. Diagnosis of fibrosis and cirrhosis using liver stiffness measurement in nonalcoholic fatty liver disease. Journal of Hepatology, Vol. 51, No. 2, pp. 454-462, 2010.

Wu, C. J. and Sung, A. H. The application of fuzzy logic to JPEG. *IEEE Trans. Consumer Electron.*, vol. 40, no. 4, pp. 976–984, Nov. 1994.

Wu, J.C. and Liu, T.S. A sliding-mode approach to fuzzy control design. IEEE Transactions on Control Systems Technology 1996;4(2):141-151.

Xiao, J., Xiao, J. Z., Xi, N., Tummala, R. L. and Mukherjee, R. Fuzzy controller for wall-climbing microrobots. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 466–480, Aug. 2004.

Xu, S., Liu, X. P., Member, S., Zhang, H., Member, S. & Hu, L. 2010. An improved realistic mass-spring model for surgery simulation. *International Conference on Automation and Logistics* : 1-6.

Xu, S., Liu, X.P., Member, S., Zhang, H. & Hu, L. 2011. *A Nonlinear Viscoelastic Tensor Mass Visual Model for Surgery Simulation*, **60**(1): 14–20.

Yamamoto, T. 2011. Applying tissue models in Teleoperated Robot-Assisted surgery. Dissertation for Degree of Doctor of Philosophy. Johns Hopkins University.

Yang, S. X., Li, H., Meng, M.Q. H. and Liu, P. X. An embedded fuzzy controller for a behavior-based mobile robot with guaranteed performance. *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 436–446, Aug. 2004.

Yoneda, M., Yoneda, M., Mawatari, H., Fujita, K., Endo, H., lida, H., Nozaki, Y., Yonemitsu, K., Higurashi, T., Takahashi, H., Kobayashi, N., Kirikoshi, H., Abe, Y., Inamori, M., Kubota, K., Saito, S., Tamano, M., Hiraishi, H., Maeyama, S., Yamaguchi, N., Togo, S. and Nakajima, A. Noninvasive assessment of liver fibrosis by measurement of stiffness in patients with nonalcoholic fatty liver disease (NAFLD). Journal of Digestive and Liver Disease, Vol. 40, pp. 371-378, 2008.

Yu, L. X. and Zhang, Ya. Qi. Evolutionary fuzzy neural networks for hybrid financial prediction. *IEEE Trans. Syst., Man, Cybern., C, App. Rev.*, vol. 35, no. 2, pp. 244–249, May 2005.

Zadeh, L. A. Fuzzy Logic = Computing with words. IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, May 1996.

Zadeh, L. A. Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, Jan. 1973.

Zerbato, D., Galvan, S., Fiorini, P. Calibration of mass spring models for organ simulations. In: Proceedings of the 2007 IEEE RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 370-375 (2007)

Zhang J, Zhou J, Huang W, Qin J, Yang T, Liu J, Su Y, Chui CK, and Chang S. GPU-Friendly Gallbladder Modeling in Laparoscopic Cholecystectomy Surgical Training System. Computers & Electrical Engineering 39, no. 1 (1// 2013): 122-29. 2MSM LT 4-7 (2013)

Zhang R. T. and Phillis, Y. A. Fuzzy control of queueing systems with heterogeneous servers," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 1, pp. 17–26, Feb. 1999.

Zhang, Y., Zhao, J., Yuan, Z., Ding, Y., Long, C. & Xiong, L. 2010. CUDA Based GPU Programming to Simulate 3D Tissue Deformation. *International Conference on Biomedical Engineering and Computer Science*, (1): 1–5.

Zheng H. and Zhu, K. Y. A fuzzy controller-based multiple-model adaptive control system for blood pressure control," in *Proc. 8th Conf. Control, Automation, Robotics and Vision*, Kunming, China, 2004, pp. 1353–1358.

Zhong, Y., Shirinzadeh, B., Smith, J. Soft tissue deformation with neural dynamics for surgery simulation. International Journal of Robotics and Automation. 2007

Zimmermann, H. J. *Fuzzy Set Theory and Its Application*, 2nd ed. Boston, MA: Kluwer, 1991.

Ziol, M., Handra-Luca, A., Kettaneh, A., Christidis, C., Mal, F., Kazemi, F., de Ledinghen, V., Marcellin, P., Dhumeaux, D., Trinchet, J. and Beaugrand, M.

Noninvasive assessment of liver fibrosis by measurement of stiffness in patients with Chronic Hepatitis C. Hepatology, Vol. 41, No. 1, 2005.

Ziv, A., Wolpe, P., Small, S., et al., Simulation-Based Medical Education: An Ethical Imperative. Academic Medicine, 2003. 78: p. 783-788.

Appendices

Appendix A—FMPC Rules

RULES	HCV	HBV	ALD	CLD	NAFLD	FIBROSIS STAGE
1	F2	F0-F1	none	none	none	F0-F1
2	F2	F2	none	none	none	F0-F2
3	F2	F3	none	none	none	F2
4	F2	F4	none	none	none	F3
5	F3	F0-F1	none	none	none	F0-F1
6	F3	F2	none	none	none	F2
7	F3	F3	none	none	none	F3
8	F3	F4	none	none	none	F3
9	F4	F0-F1	none	none	none	F2
10	F4	F2	none	none	none	F3
11	F4	F3	none	none	none	F3
12	F4	F4	none	none	none	F3
13	F2	none	F3	none	none	F2
14	F2	none	F4	none	none	F4
15	F3	none	F3	none	none	F2
16	F3	none	F4	none	none	F4
17	F4	none	F3	none	none	F3
18	F4	none	F4	none	none	F4
19	F2	none	none	F0-F1	none	F0-F1
20	F2	none	none	F2	none	F0-F1
21	F2	none	none	F3	none	F2
22	F2	none	none	F4	none	F4
23	F3	none	none	F0-F1	none	F0-F1
24	F3	none	none	F2	none	F2
25	F3	none	none	F3	none	F3
26	F3	none	none	F4	none	F4
27	F4	none	none	F0-F1	none	F2
28	F4	none	none	F2	none	F3
29	F4	none	none	F3	none	F3
30	F4	none	none	F4	none	F4
31	F2	none	none	none	F0	F0-F1
32	F2	none	none	none	F1	F0-F1
33	F2	none	none	none	F2	F2
34	F2	none	none	none	F3	F2
35	F2	none	none	none	F4	F4
36	F3	none	none	none	F0	F0-F1
37	F3	none	none	none	F1	F2
38	F3	none	none	none	F2	F2
39	F3	none	none	none	F3	F3
40	F3	none	none	none	F4	F4
41	F4	none	none	none	F0	F2

42	F4	none	none	none	F1	F2
43	F4	none	none	none	F2	F3
44	F4	none	none	none	F3	F3
45	F4	none	none	none	F4	F4
46	none	F0-F1	F3	none	none	F0-F1
47	none	F0-F1	F4	none	none	F3
48	none	F2	F3	none	none	F2
49	none	F2	F4	none	none	F4
50	none	F3	F3	none	none	F2
51	none	F3	F4	none	none	F3
52	none	F4	F3	none	none	F3
53	none	F4	F4	none	none	F4
54	none	F0-F1	none	F0-F1	none	F0-F1
55	none	F0-F1	none	F2	none	F0-F1
56	none	F0-F1	none	F3	none	F2
57	none	F0-F1	none	F4	none	F4
58	none	F2	none	F0-F1	none	F0-F1
59	none	F2	none	F2	none	F2
60	none	F2	none	F3	none	F2
61	none	F2	none	F4	none	F4
62	none	F3	none	F0-F1	none	F0-F1
63	none	F3	none	F2	none	F2
64	none	F3	none	F3	none	F3
65	none	F3	none	F4	none	F4
66	none	F4	none	F0-F1	none	F2
67	none	F4	none	F2	none	F3
68	none	F4	none	F3	none	F3
69	none	F4	none	F4	none	F4
70	none	F0-F1	none	none	F0	F0-F1
71	none	F0-F1	none	none	F1	F0-F1
72	none	F0-F1	none	none	F2	F0-F1
73	none	F0-F1	none	none	F3	F2
74	none	F0-F1	none	none	F4	F3
75	none	F2	none	none	F0	F0-F1
76	none	F2	none	none	F1	F0-F1
77	none	F2	none	none	F2	F0-F1
78	none	F2	none	none	F3	F2
79	none	F2	none	none	F4	F4
80	none	F3	none	none	F0	F0-F1
81	none	F3	none	none	F1	F2
82	none	F3	none	none	F2	F2
83	none	F3	none	none	F3	F3
84	none	F3	none	none	F4	F4
85	none	F4	none	none	F0	F2
86	none	F4	none	none	F1	F2

87	none	F4	none	none	F2	F2
88	none	F4	none	none	F3	F3
89	none	F4	none	none	F4	F4
90	none	none	F3	F0-F1	none	F0-F1
91	none	none	F3	F2	none	F2
92	none	none	F3	F3	none	F3
93	none	none	F3	F4	none	F4
94	none	none	F4	F0-F1	none	F3
95	none	none	F4	F2	none	F4
96	none	none	F4	F3	none	F4
97	none	none	F4	F4	none	F4
98	none	none	F3	none	F0	F0-F1
99	none	none	F4	none	F0	F3
100	none	none	F3	none	F1	F2
101	none	none	F4	none	F1	F4
102	none	none	F3	none	F2	F2
103	none	none	F4	none	F2	F4
104	none	none	F3	none	F3	F3
105	none	none	F4	none	F3	F4
106	none	none	F3	none	F4	F4
107	none	none	F4	none	F4	F4
108	none	none	none	F0-F1	F0	F0-F1
109	none	none	none	F0-F1	F1	F0-F1
110	none	none	none	F0-F1	F2	F0-F1
111	none	none	none	F0-F1	F3	F2
112	none	none	none	F0-F1	F4	F3
113	none	none	none	F2	F0	F0-F1
114	none	none	none	F2	F1	F0-F1
115	none	none	none	F2	F2	F2
116	none	none	none	F2	F3	F2
117	none	none	none	F2	F4	F4
118	none	none	none	F3	F0	F2
119	none	none	none	F3	F1	F2
120	none	none	none	F3	F2	F2
121	none	none	none	F3	F3	F3
122	none	none	none	F3	F4	F4
123	none	none	none	F4	F0	F4
124	none	none	none	F4	F1	F4
125	none	none	none	F4	F2	F4
126	none	none	none	F4	F3	F4
127	none	none	none	F4	F4	F4
128	F2	F0-F1	F3	none	F0	F0-F1
129	F2	F0-F1	F3	none	F1	F0-F1
130	F2	F0-F1	F3	none	F2	F0-F1
131	F2	F0-F1	F3	none	F3	F2

132	F2	F0-F1	F3	none	F4	F4
133	F2	F0-F1	F4	none	F0	F2
134	F2	F0-F1	F4	none	F1	F2
135	F2	F0-F1	F4	none	F2	F2
136	F2	F0-F1	F4	none	F3	F3
137	F2	F0-F1	F4	none	F4	F4
138	F2	F2	F3	none	F0	F0-F1
139	F2	F2	F3	none	F1	F0-F1
140	F2	F2	F3	none	F2	F2
141	F2	F2	F3	none	F3	F2
142	F2	F2	F3	none	F4	F4
143	F2	F2	F4	none	F0	F2
144	F2	F2	F4	none	F1	F2
145	F2	F2	F4	none	F2	F2
146	F2	F2	F4	none	F3	F3
147	F2	F2	F4	none	F4	F4
148	F2	F3	F3	none	F0	F0-F1
149	F2	F3	F3	none	F1	F0-F1
150	F2	F3	F3	none	F2	F2
151	F2	F3	F3	none	F3	F2
152	F2	F3	F3	none	F4	F4
153	F2	F3	F4	none	F0	F2
154	F2	F3	F4	none	F1	F2
155	F2	F3	F4	none	F2	F2
156	F2	F3	F4	none	F3	F3
157	F2	F3	F4	none	F4	F4
158	F2	F4	F3	none	F0	F2
159	F2	F4	F3	none	F1	F2
160	F2	F4	F3	none	F2	F3
161	F2	F4	F3	none	F3	F3
162	F2	F4	F3	none	F4	F4
163	F2	F4	F4	none	F0	F2
164	F2	F4	F4	none	F1	F3
165	F2	F4	F4	none	F2	F3
166	F2	F4	F4	none	F3	F3
167	F2	F4	F4	none	F4	F4
168	F3	F0-F1	F3	none	F0	F0-F1
169	F3	F0-F1	F3	none	F1	F0-F1
170	F3	F0-F1	F3	none	F2	F2
171	F3	F0-F1	F3	none	F3	F2
172	F3	F0-F1	F3	none	F4	F4
173	F3	F0-F1	F4	none	F0	F2
174	F3	F0-F1	F4	none	F1	F2
175	F3	F0-F1	F4	none	F2	F3
176	F3	F0-F1	F4	none	F3	F3

177	F3	F0-F1	F4	none	F4	F4
178	F3	F2	F3	none	F0	F0-F1
179	F3	F2	F3	none	F1	F0-F1
180	F3	F2	F3	none	F2	F2
181	F3	F2	F3	none	F3	F2
182	F3	F2	F3	none	F4	F4
183	F3	F2	F4	none	F0	F2
184	F3	F2	F4	none	F1	F2
185	F3	F2	F4	none	F2	F3
186	F3	F2	F4	none	F3	F3
187	F3	F2	F4	none	F4	F4
188	F3	F3	F3	none	F0	F0-F1
189	F3	F3	F3	none	F1	F2
190	F3	F3	F3	none	F2	F2
191	F3	F3	F3	none	F3	F3
192	F3	F3	F3	none	F4	F4
193	F3	F3	F4	none	F0	F2
194	F3	F3	F4	none	F1	F3
195	F3	F3	F4	none	F2	F3
196	F3	F3	F4	none	F3	F3
197	F3	F3	F4	none	F4	F4
198	F3	F4	F3	none	F0	F2
199	F3	F4	F3	none	F1	F2
200	F3	F4	F3	none	F2	F2
201	F3	F4	F3	none	F3	F3
202	F3	F4	F3	none	F4	F4
203	F3	F4	F4	none	F0	F3
204	F3	F4	F4	none	F1	F3
205	F3	F4	F4	none	F2	F3
206	F3	F4	F4	none	F3	F3
207	F3	F4	F4	none	F4	F4
208	F4	F0-F1	F3	none	F0	F0-F1
209	F4	F0-F1	F3	none	F1	F2
210	F4	F0-F1	F3	none	F2	F2
211	F4	F0-F1	F3	none	F3	F3
212	F4	F0-F1	F3	none	F4	F4
213	F4	F0-F1	F4	none	F0	F2
214	F4	F0-F1	F4	none	F1	F2
215	F4	F0-F1	F4	none	F2	F3
216	F4	F0-F1	F4	none	F3	F3
217	F4	F0-F1	F4	none	F4	F4
218	F4	F2	F3	none	F0	F2
219	F4	F2	F3	none	F1	F2
220	F4	F2	F3	none	F2	F2
221	F4	F2	F3	none	F3	F3

222	F4	F2	F3	none	F4	F4
223	F4	F2	F4	none	F0	F2
224	F4	F2	F4	none	F1	F3
225	F4	F2	F4	none	F2	F3
226	F4	F2	F4	none	F3	F3
227	F4	F2	F4	none	F4	F4
228	F4	F3	F3	none	F0	F2
229	F4	F3	F3	none	F1	F2
230	F4	F3	F3	none	F2	F2
231	F4	F3	F3	none	F3	F3
232	F4	F3	F3	none	F4	F4
233	F4	F3	F4	none	F0	F3
234	F4	F3	F4	none	F1	F3
235	F4	F3	F4	none	F2	F3
236	F4	F3	F4	none	F3	F3
237	F4	F3	F4	none	F4	F4
238	F4	F4	F3	none	F0	F2
239	F4	F4	F3	none	F1	F2
240	F4	F4	F3	none	F2	F2
241	F4	F4	F3	none	F3	F3
242	F4	F4	F3	none	F4	F4
243	F4	F4	F4	none	F0	F3
244	F4	F4	F4	none	F1	F3
245	F4	F4	F4	none	F2	F3
246	F4	F4	F4	none	F3	F3
247	F4	F4	F4	none	F4	F4

Appendix B—CUDA Kernel

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "utils_math.h"

typedef struct {
  int from;
  int to;
  double springconstant;
  double dampingconstant;
  double restlength;
} PARTICLESPRING;

typedef struct {
  double x;
  double y;
  double z;
} vec3d;

__device__ vec3d closestpoint (
double start_x,double start_y,double start_z,
double end_x,double end_y,double end_z,
double pos_x,double pos_y,double pos_z,
double *t)
{
        vec3d AB;
        AB.x=end_x-start_x;
        AB.y=end_y-start_y;
        AB.z=end_z-start_z;

        double ab_square=AB.x*AB.x+AB.y*AB.y+AB.z*AB.z;
        vec3d AP;
        AP.x=pos_x-start_x;
        AP.y=pos_y-start_y;
        AP.z=pos_z-start_z;

        double ap_dot_ab=AB.x*AP.x+AB.y*AP.y+AB.z*AP.z;
        *t=ap_dot_ab/ab_square;

        vec3d Q;
        Q.x=start_x+(start_x*end_x)* (*t);
        Q.y=start_y+(start_y*end_y)* (*t);
        Q.z=start_z+(start_z*end_z)* (*t);
        return Q;
}
```

```
__device__ double distance (
        double pos_x,double pos_y,double pos_z,
        double pos2_x,double pos2_y,double pos2_z)
{
        double xd = pos2_x-pos_x;
        double yd =  pos2_y-pos_y;
        double zd = pos2_z-pos_z;
        double len = sqrt(xd*xd + yd*yd + zd*zd);
   return len;
}
__device__ bool pointraytest (
        double start_x,double start_y,double start_z,
        double end_x,double end_y,double end_z,
        double pos_x,double pos_y,double pos_z,
        vec3d *pt,
        double  *radius,
        double *t,double epsilon,int i){
        *pt = closestpoint (
                                start_x,start_y,start_z,
                                end_x,end_y,end_z,
                                pos_x,pos_y,pos_z,
                                t);
        double len = distance((*pt).x,(*pt).y,(*pt).z,pos_x,pos_y,pos_z);
        return len < (radius[i]+epsilon);
}
__global__
void calcDerivatives (
        double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double *dev_dpdt_x,double *dev_dpdt_y,double *dev_dpdt_z,
        double   *dev_dvdt_x,double *dev_dvdt_y,double *dev_dvdt_z,
        double *dev_force_x,double *dev_force_y,double *dev_force_z,
        double  *dev_mass,int np)
{
   //@@ Insert code to implement vector addition here
   int i = blockIdx.x * blockDim.x + threadIdx.x;
   if (i < np) {
        dev_dpdt_x[i] =dev_vel_x[i];
        dev_dpdt_y[i] =dev_vel_y[i];
        dev_dpdt_z[i] =dev_vel_z[i];
        dev_dvdt_x[i] = dev_force_x[i] /dev_mass[i];
        dev_dvdt_y[i] = dev_force_y[i] /dev_mass[i];
        dev_dvdt_z[i] = dev_force_z[i] /dev_mass[i];
        }
}
__global__
void updateParticles (
        double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double *dev_nex_vel_x,double *dev_nex_vel_y,double *dev_nex_vel_z,
        double *dev_pos_x,double *dev_pos_y,double *dev_pos_z,
        double *dev_nex_pos_x,double *dev_nex_pos_y,double *dev_nex_pos_z,
```

```
        double *dev_prev_pos_x,double *dev_prev_pos_y,double *dev_prev_pos_z,
        double *dev_dpdt_x,double *dev_dpdt_y,double *dev_dpdt_z,
        double   *dev_dvdt_x,double *dev_dvdt_y,double *dev_dvdt_z,
        int np,double dt)
{
   //@@ Insert code to implement vector addition here
   int i = blockIdx.x * blockDim.x + threadIdx.x;
   if (i < np) {
                dev_nex_pos_x[i] = dev_pos_x[i] + dev_dpdt_x[i] * dt;
                dev_nex_pos_y[i] = dev_pos_y[i] + dev_dpdt_y[i] * dt;
                dev_nex_pos_z[i] = dev_pos_z[i] + dev_dpdt_z[i] * dt;
                dev_nex_vel_x[i] =dev_vel_x[i]+ dev_dvdt_x[i] * dt;
                dev_nex_vel_y[i] =dev_vel_y[i]+ dev_dvdt_y[i] * dt;
                dev_nex_vel_z[i] =dev_vel_z[i] + dev_dvdt_z[i] * dt;

                dev_prev_pos_x[i] = dev_pos_x[i];
                dev_prev_pos_y[i] = dev_pos_y[i];
                dev_prev_pos_z[i] = dev_pos_z[i];
                dev_pos_x[i] =dev_nex_pos_x[i];
                dev_pos_y[i] =dev_nex_pos_y[i];
                dev_pos_z[i] =dev_nex_pos_z[i];

                dev_vel_x[i]=dev_nex_vel_x[i];
                dev_vel_y[i]=dev_nex_vel_y[i];
                dev_vel_z[i] =dev_nex_vel_z[i];
        }
}

__global__
        void raytest (
        double *start_x,double *start_y,double *start_z,
        double *end_x,double *end_y,double *end_z,
        double *pos_x,double *pos_y,double *pos_z,
        double *t,double *minDistToStart,double *radius,
        bool *foundCollision,int *pointID,double epsilon,int np)
{
        int i = blockIdx.x * blockDim.x + threadIdx.x;
        vec3d pt;
        if (i < np) {
        if (pointraytest(
                start_x[0],start_y[0],start_z[0],
                end_x[0],end_y[0],end_z[0],
                pos_x[i],pos_y[i],pos_z[i],
                &pt,
                radius,
                t,epsilon,i))
                {
                        double dst = distance(
                                        start_x[0],start_y[0],start_z[0],
                                        pt.x, pt.y, pt.z);
```

```
                        if (dst < minDistToStart[0])
                        {
                                minDistToStart[0] = dst;
                                pointID[0] = i;
                                foundCollision[0] =true;
                        }
                }
                }

}
__global__
void calculateforcesfirst  (
        double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double *dev_force_x,double *dev_force_y,double *dev_force_z,
        int *dev_fixed2,int np,double viscousdrag)
{
   //@@ Insert code to implement vector addition here
   int i = blockIdx.x * blockDim.x + threadIdx.x;
   if (i < np) {
            dev_force_x[i]=0;
            dev_force_y[i]=0;
            dev_force_z[i]=0;

            if ( dev_fixed2[i]==0){
             // Viscous drag
                    dev_force_x[i] -= viscousdrag *dev_vel_x[i];
                    dev_force_y[i] -= viscousdrag *dev_vel_y[i];
                    dev_force_z[i] -= viscousdrag *dev_vel_z[i];
            }
        }
}
__global__
void calculateforcessecond (
        double * dev_pos_x,double * dev_pos_y,double * dev_pos_z,
        double*  dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double * dev_force_x,double * dev_force_y,double * dev_force_z,
        PARTICLESPRING* dev_s, int ns,bool * dev_breaklink,int* dev_fixed2)
{
   //@@ Insert code to implement vector addition here
   int i = blockIdx.x * blockDim.x + threadIdx.x;
         double x,y,z=0;
   if (i < ns) {
        int p1 = dev_s[i].from;
        int p2 = dev_s[i].to;
        double dx =dev_pos_x[p1] -dev_pos_x[p2];
        double dy =dev_pos_y[p1] -dev_pos_y[p2];
        double dz =dev_pos_z[p1] -dev_pos_z[p2];
        double len = sqrt(dx*dx + dy*dy + dz*dz);
          if(len > 10*(dev_s[i].restlength))
          {
```

```
                dev_breaklink[p1] = true;
                 dev_breaklink[p2] = true;
            }
    x  = dev_s[i].springconstant  * (len - dev_s[i].restlength);
    x += dev_s[i].dampingconstant * (dev_vel_x[p1] - dev_vel_x[p2]) * dx / len;
    x *= - dx / len;
    y  = dev_s[i].springconstant  * (len - dev_s[i].restlength);
    y += dev_s[i].dampingconstant * (dev_vel_y[p1] - dev_vel_y[p2]) * dy / len;
    y *= - dy / len;
    z  = dev_s[i].springconstant  * (len - dev_s[i].restlength);
    z += dev_s[i].dampingconstant * (dev_vel_z[p1] - dev_vel_z[p2]) * dz / len;
    z *= - dz / len;

    if (!dev_fixed2[p1])
          {
                dev_force_x[p1] += x;
                dev_force_y[p1] += y;
                dev_force_z[p1] += z;
    }
    if (!dev_fixed2[p2])
          {
                dev_force_x[p2] -= x;
                dev_force_y[p2] -= y;
                dev_force_z[p2] -= z;
    }
        }
}

extern "C" void
calculate_derivatives(
                dim3 grid,
                dim3 block,
                double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
                double *dev_dpdt_x,double *dev_dpdt_y,double *dev_dpdt_z,
                double   *dev_dvdt_x,double *dev_dvdt_y,double *dev_dvdt_z,
                double *dev_force_x,double *dev_force_y,double *dev_force_z,
                double  *dev_mass,int np)
{
        calcDerivatives<<<grid,block>>>(
                        dev_vel_x,dev_vel_y,dev_vel_z,
                        dev_dpdt_x,dev_dpdt_y,dev_dpdt_z,
                        dev_dvdt_x,dev_dvdt_y,dev_dvdt_z,
                        dev_force_x,dev_force_y,dev_force_z,
                        dev_mass,np);
 //    cudaProcess<<< grid, block, sbytes >>> (g_data, g_odata, imgw, imgh,
block.x+(2*radius), radius, 0.8f, 4.0f);
}

extern "C" void
calculate_forces_first(
```

```
                dim3 grid,
                dim3 block,
                double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
                double *dev_force_x,double *dev_force_y,double *dev_force_z,
                int *dev_fixed2,int np,double viscousdrag)
{
        calculateforcesfirst<<<grid,block>>>(
                                 dev_vel_x,dev_vel_y,dev_vel_z,
                                dev_force_x,dev_force_y,dev_force_z,
                                dev_fixed2, np, viscousdrag);
 //    cudaProcess<<< grid, block, sbytes >>> (g_data, g_odata, imgw, imgh,
block.x+(2*radius), radius, 0.8f, 4.0f);
}

extern "C" void
 calculate_forces_second(
        dim3 grid,
        dim3 block,
        double * dev_pos_x,double * dev_pos_y,double * dev_pos_z,
        double*  dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double * dev_force_x,double * dev_force_y,double * dev_force_z,
        PARTICLESPRING* dev_s,int ns,bool * dev_breaklink,int* dev_fixed2)
{
        calculateforcessecond<<<grid,block>>>(
                                dev_pos_x,dev_pos_y,dev_pos_z,
                                 dev_vel_x,dev_vel_y,dev_vel_z,
                                dev_force_x,dev_force_y,dev_force_z,
                                dev_s, ns,dev_breaklink,dev_fixed2);
 //    cudaProcess<<< grid, block, sbytes >>> (g_data, g_odata, imgw, imgh,
block.x+(2*radius), radius, 0.8f, 4.0f);
}

extern "C" void
        update_particles(
        dim3 grid,
        dim3 block,
        double *dev_vel_x,double *dev_vel_y,double *dev_vel_z,
        double *dev_nex_vel_x,double *dev_nex_vel_y,double *dev_nex_vel_z,
        double *dev_pos_x,double *dev_pos_y,double *dev_pos_z,
        double *dev_nex_pos_x,double *dev_nex_pos_y,double *dev_nex_pos_z,
        double *dev_prev_pos_x,double *dev_prev_pos_y,double *dev_prev_pos_z,
        double *dev_dpdt_x,double *dev_dpdt_y,double *dev_dpdt_z,
        double   *dev_dvdt_x,double *dev_dvdt_y,double *dev_dvdt_z,
        int np,double dt)
{
        updateParticles<<<grid,block>>>(
                                dev_vel_x,dev_vel_y,dev_vel_z,
                                dev_nex_vel_x,dev_nex_vel_y,dev_nex_vel_z,
                                dev_pos_x,dev_pos_y,dev_pos_z,
                                dev_nex_pos_x,dev_nex_pos_y,dev_nex_pos_z,
```

```
                                dev_prev_pos_x,dev_prev_pos_y,dev_prev_pos_z,
                                dev_dpdt_x,dev_dpdt_y,dev_dpdt_z,
                                dev_dvdt_x,dev_dvdt_y,dev_dvdt_z,
                                np,dt);
}

extern "C" void
 raytest(
        dim3 grid,
        dim3 block,
        double *start_x,double *start_y,double *start_z,
        double *end_x,double *end_y,double *end_z,
        double *pos_x,double *pos_y,double *pos_z,
        double *t,double *minDistToStart,double *radius,
        bool *foundCollision,int *pointID,double epsilon,int np)
{
        raytest<<<grid,block>>>(
                                start_x,start_y,start_z,
                                end_x,end_y,end_z,
                                pos_x,pos_y,pos_z,
                                t,minDistToStart,radius,
                                foundCollision,pointID,epsilon,np);
 //     cudaProcess<<< grid, block, sbytes >>> (g_data, g_odata, imgw, imgh,
block.x+(2*radius), radius, 0.8f, 4.0f);

}
```