DEVELOPMENT OF A BIOMECHANICAL TESTING PLATFORM FOR THE STUDY OF THE HUMAN KNEE JOINT

by

Jorge Enrique Gil

BS, University of Pittsburgh, 1998

Submitted to the Graduate Faculty of

the School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Jorge Enrique Gil

It was defended on

November 11, 2003

and approved by

Richard E. Debski, Ph.D., Assistant Professor, Department of Bioengineering

Patrick McMahon, M.D., Assistant Professor, Department of Orthopaedic Surgery

Thesis Advisor: Savio L-Y. Woo, Ph.D., D.Sc., Professor, Dept. of Mechanical Engineering

DEVELOPMENT OF A BIOMECHANICAL TESTING PLATFORM

FOR THE STUDY OF THE HUMAN KNEE JOINT

Jorge E. Gil, M.S.

University of Pittsburgh, 2003

The knee joint is a sophisticated biological mechanism involved in locomotion at the lower extremity. Despite its apparently simple motion during gait, the knee actually features complex 6-DOF kinematic patterns and 3D force distributions that stem from the biomechanical interdependence and balance of its component tissues. Following joint injury, such balance is upset and is difficult to restore with existing clinical treatments. In the interest of studying and characterizing the mechanics of the knee, a robotic/UFS testing system capable of recording the complexity of joint kinematics and of the forces transmitted by the soft-tissues in response to meaningful loading conditions, has been used by various laboratories to obtain quantitative data with which to evaluate injury mechanisms, prevention, treatment and rehabilitation. This system has been successfully used to quantify the mechanical behavior of knee ligaments and their reconstruction grafts, menisci and cartilage, in response to a variety of experimental conditions. The effort of this work is to modernize the robotic/UFS testing system by upgrading its software control to manage more general and realistic loading conditions. The resulting software system, named the biomechanical testing platform, is expected to ultimately integrate the operation of the robotic/UFS

testing system with that of other valuable experimental and computational approaches aimed at the study of the human knee joint.

The biomechanical testing platform is designed with the use of state-of-the-art development technologies and comprises the mathematical formulations, control algorithms, and data abstractions specialized to a clinically relevant description of the kinematics and kinetics of the human knee. The system accommodates logical choices of hardware, motion description, iterative algorithms, as well as the use of automatic regression verifications. The biomechanical testing platform is demonstrated with a homologous experiment to that of the robotic/UFS testing system: the measurement of in situ forces in the ACL of a cadaver specimen, in response to anterior-posterior (translation) and varus-valgus (rotation) tibial loads. Furthermore, an application with concurrent interoperability between the robotic/UFS testing system and a computational analysis method is proposed.

DESCRIPTORS

Biomechanics	Hybrid Control
In-situ Force	Kinematics
Knee	Object-Oriented Programming
Robotics	Testing Platform

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Anatomy:

ACL	Anterior Cruciate Ligament
LCL	Lateral Collateral Ligament
MCL	Medial Collateral Ligament
PCL	Posterior Cruciate Ligament

Motion:

AP	Anterior-Posterior
DOF	Degree-of-Freedom
FE	Flexion-Extension
IE	Internal-External
JMD	Joint Motion Description
ML	Medial-Lateral
PD	Proximal-Distal
VV	Varus-Valgus

Coordinate Systems:

E	End-Effector Coordinates System
F	Femur Coordinate System

S	Sensor Coordinate System
T	Tibial Coordinate System
TCP	Tool Center Point
U	Global (User) Coordinate System
W	World Coordinate System

Software:

API	Application Programming Interface
ATL	Active Template Library
COM	Component Object Model
GUI	Graphical User Interface
MFC	Microsoft Foundation Classes
OpenGL	Graphics Library
STL	Standard Template Library

Miscellaneous:

MSRC	Musculoskeletal Research Center
N	Newtons
RBSM	Rigid Body/Spring Model
SD	Standard deviation
T (superscript)	Matrix Transpose
UFS	Universal Force-Moment Sensor

PREFACE

This work concludes an effort that recruited the support and talents of many people:

I want to thank my wife, Line Gil, for understanding what I needed to complete this work and finding a way to provide it. She sheltered my courage and inspired me to imagine my best and see it realized.

I want to thank my team, especially Shon Darcy and Diane Budzik, because together we were fearless at integrating new and untested technology to this work. Shon was my most trusted advocate and did everything in his power to see that our execution had the fair chance it needed to succeed. I'll always be grateful to him.

I want to thank Dr. Richard Debski, because he saw promise in our work and gave me his support. He offered me the crucial chance to take our experiments to Detroit, giving a fresh breath of possibility to the platform. It proved important to me.

I want to thank my old team, made up by Drs Yukihisa Fukuda and John Loh, who are two great clinicians and equally great engineers: they taught me how to program. I also want to thank John Fischer and Dan Hubbard. Their experimental data helped me navigate some obscure places I encountered during development.

Ted Rudy was also an important source of support to this work because he understood the history and details of the legacy robotic/UFS testing system and offered them to me as guidance during development of the biomechanical testing platform.

I want to thank Dr. Patrick McMahon for evaluating this work and helping me frame its clinical significance.

Finally, I want to thank my advisor, Dr. Savio L-Y. Woo, because he was present at two critical crossroads in my life. At each of them, his trust in my ability to make mature and responsible decisions, and his timely guidance and oversight, gave me the assurance I needed to embark on the best path, complete it, and learn.

1.0 MOTIVATION

The human knee joint is a sophisticated biological mechanism involved in locomotion at the lower extremity charged with negotiating the high mobility of the femur and the tibia with the transmission of loads from the ground to the body[1, 2]. Its function is often sustained for the lifetime of a subject, at a rate of over 1 million cycles per year[3, 4], without signs of damage.

Endurance of the knee joint results from the balance of loads occurring throughout the constituent structures: the articular cartilage, the ligaments and menisci[5]. Such a balance is delicate, however. Despite the apparent simplicity of joint movement (flexion and extension) the knee actually features a nuanced sequence of coupled motions in all degrees of freedom[6-8] reflecting the biomechanical interdependence of its constituent structures[9-13]. When any one of the structures suffers acute injury the sequelae to the joint are often systemic and chronic. For example, patient history of knee injury has been identified as a strong predictor of unilateral osteoarthritis[14]. Osteoarthritis of the knee has further been correlated with significant functional impairments, with implications reaching as far as the patient's psychological well-being[15-17]. With an incidence of 240 per 100,000 people per year, or over 1 million new clinical cases per year in the U.S. alone, the potential cascade of costs in health care and productivity loss is in the order of billions of dollars.

For acute injury to the ligaments and meniscus, incidence is in the same order of magnitude, at 60-100 for every 100,000 people per year[18]. For the anterior cruciate ligament

(ACL) alone, incidence during popular sport activities such as skiing and football is high, at over 50 per 100,000 player-days[19-21]. More alarming yet is the mounting epidemiological evidence suggesting the failure of common clinical treatments, surgical or not, to restore normal joint function[22-24]. In view of this, significant attention has been dedicated to the prevention, characterization, treatment and rehabilitation of knee joint injuries. Of particular interest is the formulation of quantitative criteria with which to evaluate ensuing joint performance in settings that simulate activities of daily living[25-29] in terms of the kinematics of the joint and the forces borne by the constituent structures[10, 11, 30-33].

Making accurate estimations of forces that occur *in vivo* in the component tissues of the human locomotive apparatus remains one of the most significant challenges in orthopedic biomechanics, and the knee joint is no exception[34-38]. In fact, direct measurement of *in vivo* forces in the human knee is currently limited in applicability. Existing strain transducers require some form of contact with the subject tissue in order to convert the mechanical energy stored in its deformation into a measurable signal, rendering them impractical[25, 26, 28, 39]. Part of the challenge lies on identifying a non-invasive and non-destructive methodology to access and yield the data of interest.

The rest of the challenge derives from anatomic and functional variability[40]. At the very least, both the mechanical properties and geometry of the tissues in the knee are sufficiently different among subjects, thus limiting the power of inference attainable with direct measurements[41, 42]. Also, different subjects can perform any single activity of daily living, such as walking, running or stair climbing, quite differently [8, 29, 43-45]. In response, computational approaches are emerging that enable simulation of the constitutive behavior, geometry and mechanical environment of the knee using 3D computer models[46-50]. With this kind of

technology, the bioengineer could expose a variety of subject-specific models of the knee to a wider range of loading conditions and "what-if" scenarios, eventually at a lower cost, than is possible experimentally. Such models must be grounded on a wealth of experimental data, of course, for their formulation (they must match at least one experimentally measured response) and validation (they must be able to predict a range of others). Therefore, the data from powerful experimental methods are imperative to the successful study of the human knee joint[51-54].

This work is motivated by an effort to modernize an existing experimental approach to study the knee joint biomechanics: the robotic/UFS testing system[6, 38]. This testing system combined robotic and force-moment sensor technology to the task of recording the 6-DOF kinematics of the knee and measuring the 3D in situ forces and moments in the soft tissues in response to loading conditions of interest. It has yielded valuable data on the loads borne by ligaments, cartilage and menisci, following injury and surgical repair, in response to loading conditions that simulate clinical examinations[10, 32, 55]. Given the generality of the robotic/UFS testing system, its integration with other experimental approaches, and more importantly, computational methods, is of foreseeable interest and benefit. Part of the modernization effort consisted in upgrading the robotic equipment to allow higher payloads to simulate more realistic loading conditions. Even more of the effort consisted in upgrading the software algorithms and architecture to achieve system integration.

This work proposes a biomechanical testing platform, a state of the art software system, which combines the mathematical descriptions, software control and methods necessary to make experimental measurement comprising a high-payload robotic/UFS testing system[56-58]. The system design abstracts the algorithms and data flow that are common so that the study of any subject knee, real or virtual, becomes indistinguishable. This way, experimentalists (and

eventually computational analysts) will have at their disposal the same flow of biomechanical data: forces, moments, displacement and rotations, in any variety of coordinate systems and clinical motion descriptions, applied as loading conditions and measured in the mechanical response of the knee.

1.1 THE FEATURES

The biomechanical testing platform operates on the representation of the human knee as a 6-DOF mechanism. Its premise is that to each DOF there are a displacement and a force (rotation and moment, for the angular counterpart) one of which is prescribed, the other recorded in the response of the joint during a study. Aside from this, the details of prescribing or recording such quantities are irrelevant: the platform is independent of the choice of clinical motion description and ensuing DOF, of iterative algorithms, of position control device (robot or otherwise) and force-moment sensor manufacturer, and whether the knee is an actual specimen or is simulated by a computational model. In fact, it allows users to select from existing choices in each of these areas of functionality, or incorporate one of their own, without modifying its core functionality.

The platform is also equipped with a multi-purpose positioning system to track the motion of an arbitrary number of objects in 3D space. It also allows post-processing relevant positions, orientations, forces and moments in clinical motion descriptions[56-58]. Such a positioning system is the centerpiece to any system intended to study the mechanics of the knee joint; in this work, it has been designed to be reusable and testable.

The platform crystallizes expertise in robotic control, mathematical modeling, software engineering and bioengineering necessary to modernize the approach to study the mechanics of the human knee.

1.2 THE DEVELOPMENT PROCESS

The biomechanical testing platform is sufficiently complex to benefit from industrial strength development practices to provide its functionality and maintain quality. Testing has been pursued with the help of a unit-testing framework[59, 60]. A unit-test is a small program verifying the integrity of a module in the platform software. A battery of unit-tests that can be executed automatically and on a regular basis has been assembled to provide quantitative evidence that the individual, as well as integrated software components, behave according to specification.

1.3 THE APPLICATION

In this work, the biomechanical testing platform will be demonstrated by measuring the kinematics and in situ forces in the ACL of a porcine cadaver specimen. Tests are also performed using two different robotic arms to demonstrate platform versatility. Finally, a possible scenario for interoperability between experiments and computations is entertained to suggest the applicability of the platform to estimate the forces that occur in the structures of the knee in vivo.

2.0 BACKGROUND

2.1 ANATOMY OF THE HUMAN KNEE JOINT

The human knee joint is a sophisticated biological mechanism involved in locomotion at the lower extremity. It must negotiate the high mobility between very large bones with their function of transmitting loads from the ground to the body.

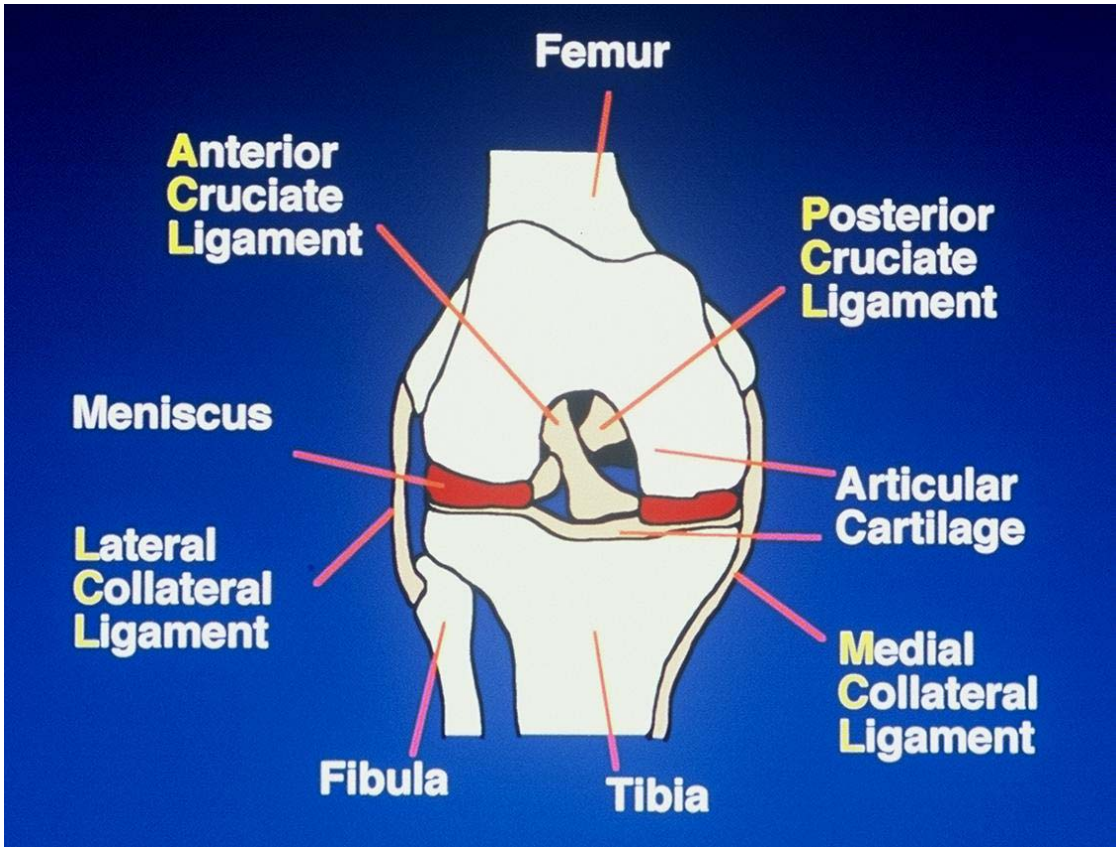


Figure 1 Anatomy of the tibio-femoral joint.

It is composed of several bones: the femur, tibia, fibula and patella. As a result, the knee joint is really a collection of several joints, the major one being the tibio-femoral joint, but including patello-femoral, and tibio-fibular joints as well. This work concentrates on the tibio-femoral joint, and the term "knee joint" alludes to such unless stated otherwise.

2.2 BIOMECHANICAL STUDY OF THE HUMAN KNEE JOINT

Many authors have tried to study the knee joint mechanism. This entails recording the rigid body motion of the bones, and measuring the forces carried by the individual soft tissues.

Studies have been performed in vivo or in vitro (on cadaver specimens), usually depending on the invasiveness of the measurement technique, using a vast array of creative experimental and computational setups.

The force balance that maintains knee stability in vivo entails the muscles together with the ligaments, menisci and articular cartilage. The lack of muscle activity does not preclude the relevance of cadaver measurements, however. In fact, significant knowledge of the knee mechanics can be extracted from the biomechanical analysis of the passive structures alone[7, 13, 61], i.e., without considering the muscle activity which is difficult to recreate realistically[28]. The ligaments, cartilage and menisci can be thought to guide joint motion while responding to the effective loading condition, part of which is determined by the muscles, and excellent cadaver models can be established on the basis of good loading condition data[62].

2.2.1 Measurement of Joint Kinematics

Although the term kinematics could generally include the motion and deformation of the constituent soft-tissues of the knee, in this study we are especially interested in the 6-DOF rigid body kinematics of the tibio-femoral joint. Measurement of kinematics thus entails the assignment of coordinate systems to the tibia and the femur and recording their relative position and orientation as they move through the workspace. Existing techniques differ on whether the motion of the bones is tracked directly, or indirectly (by inferring it from the motion of the surrounding skin). Unfortunately, studies also differ significantly on the definition of the ensuing loading conditions and specific motion descriptions used to report the data to the extent that they can become difficult to compare[44].

Some of the earlier studies able to physically record the process of locomotion in humans and animals depended on the quick acquisition of successive photographs of the subjects *in vivo* using ordinary cameras[63]. In fact, the first successful serial images of fast motion ever assembled (by Eadweard Muybridge in 1878) were intended for a gait analysis of the horse (i.e., to prove that the animal's trot involved an airborne phase, cinema happened to be born). With these techniques, it was difficult to resolve small motions, particularly translations occurring perpendicular to the plane of view, as well as rotations about axes parallel to it (often M-L translations and V-V and I-E rotations for the knee)[8], so their utility was initially limited to gross goniometry of limb segment motion. More recently, the tracking of skin markers, and the automated synchronization and spatial registration of cameras with intersecting viewing volumes, have provided a more comprehensive and accessible measurement of the 6-DOF motion of the limbs. The current challenge consists in correctly estimating the rigid body motion of the bones from the inevitably more complicated trajectory of the markers on the surrounding skin. New techniques, such as the point cluster method, are being implemented that reduce the kinematic artifacts attributed to skin motion, and faithfully predict the kinematics of the tibio-femoral joint in response to selected loading conditions[29, 44]. They remain to be proven in circumstances that induce mass displacements or vibrations of the skin, affecting the configuration of the surface markers relative to the bone[64].

Several attempts have been made to circumvent the difficulty in compensating for skin motion by tracking the motion of the bones directly. Both exoskeletal linkages[1] and intracortical pins[8] have been used, but they are often invasive and sufficiently cumbersome on the natural gait performance that they have limited applicability to accurate measurements *in vivo*. A promising new technique consists in using cine phase-contrast MRI from which the bone motion can be

integrated from velocity fields in the view volume of the radiographic equipment. The technique is entirely non-contact, non-invasive and avoids the skin marker dependence altogether[65].

2.2.2 Measurement of soft tissue forces

Measurement of soft tissue forces is more advanced for ligaments and cartilage than for the menisci. Numerous researchers have achieved tensile testing of the structural and mechanical properties of ligaments. Similarly, cartilage compression studies have yielded information of the static and dynamic response of the tissue in response to static and dynamic loads[21, 66, 67]. Tests have been performed with the tissues prepared as tensile or compressive specimens, although more recently, the emphasis is on measurements performed in situ, i.e., performing in the mechanical environment native to their insertions. Of these, some have been achieved in vitro, and others in vivo.

Towards the experimental measurement of the in situ forces of ligaments there are several approaches that leverage the use of strain transducers. Among them are the use of buckle transducers[68], implantable force transducers (IFT)[69], liquid metal strain gauges[70, 71] and Hall-Effect transducers, among others. These transducers must be installed directly on the surface of the tissue, or clamp it directly to transform the mechanical energy stored in tissue deformation into a measurable signal. A different variation is that of using in-line external force transducers that are attached to the ligament insertions, to yield the data of interest.

These approaches, although sometimes used, have limited applications to the purpose of measuring the forces occurring in the ligaments in vivo, because they are simply too invasive.

Some experimental methods have used them, in fact, in daring setups where the patient is sedated and have had the transducers Arthroscopically installed.

There is a myriad of other drawbacks involved in using such direct contact methods. Among them:

1) It is difficult to use the transducers to measure force variations and distributions within in the soft tissue, for example within bundles of the ACL. Doing so would require an array of transducers deployed on the surface of the ligament, increasing the trauma inflicted.

2) It is difficult to measure forces in hard tissue such as subchondral bone. In the case of DVRTs and buckles, multiple transducers would be required to measure forces in 3D if attempted in structures like the cartilage or menisci. Such transducers cannot measure forces that are unaligned with the primary direction of operation.

3) It is also difficult to measure effects over time, such as those related to ligament healing. Doing so would require multiple surgical interventions on a given subject.

3.0 OUR APPROACH: THE ROBOTIC/UFS TESTING SYSTEM

Our approach is to combine robotic technology with a force moment sensor, to the purpose of recording both the kinematics of the knee, and the in situ forces in the soft-tissues. Experiments with the robotic/UFS testing system are performed using cadaver specimens installed so that the femur is rigidly attached to the manipulator base and the tibia rigidly attached to the sensor[54].

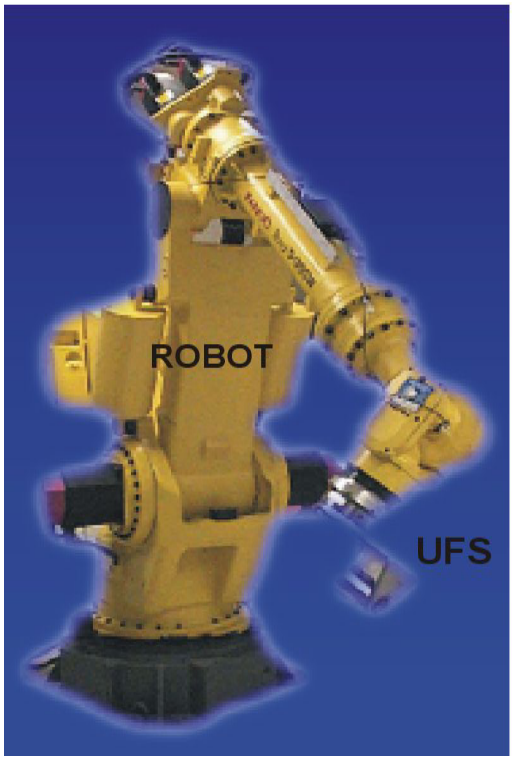


Figure 2 Robotic/UFS testing system diagram. The UFS is installed at the end-effector

The robotic/UFS testing system is capable of measuring the kinematics of the knee and the in-situ forces in the ligaments by operating in two different control modes: force control and position control.

When the robotic/UFS testing system operates in force control, the robot is guided by force-moment feedback from the UFS, allowing it to find and record the kinematics of the knee joint that occur in response to a loading condition of interest. The knee kinematics are nothing more than the locus of positions and orientations in 6-DOF that satisfy certain 3D force-moment target. Because the robot is, by default, a position control device it is capable of reproducing the kinematics of the knee previously recorded, by operating in position control mode.

By combining the two modes of operation, force control and position control, the in situ forces in the various soft-tissues can be measured using a multi-step approach. To measure the in situ forces in the ACL in response to an anterior load, for example, the first step consists in operating the robotic/UFS testing system in force control mode to record the resulting kinematics of the knee. The second step consists in transecting the ACL; immediately prior to and following ligament transection, the robotic/UFS testing system is operated in position control mode to reproduce the previously recorded kinematics while measuring the total forces with the UFS. By the principle of superposition, the in situ forces of the ligament is manifest as a change in the total forces recorded at the UFS following ligament transection [10, 32, 38, 55].

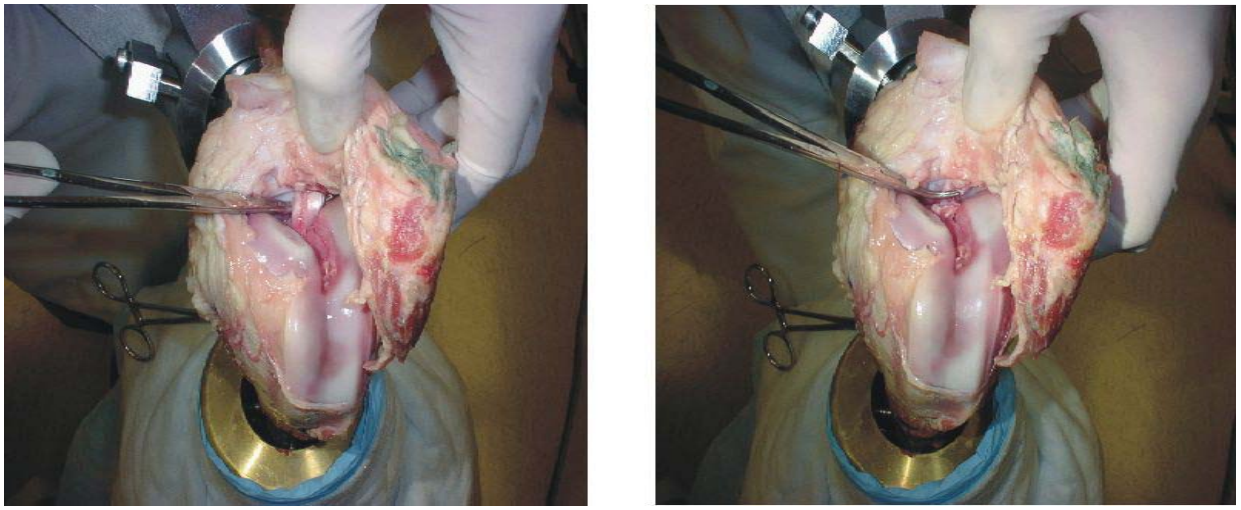


Figure 3 Transection of the ACL of a porcine specimen (before and after) during a test to measure the ligament in situ forces

Although the robotic/UFS testing system requires transection of the structure of interest, it is considered a non-contact method because, barring elastic bony deflection, it reproduces the loads and boundary conditions of the structure in the intact and transected configuration, from which the measurement is inferred.

The motivation for upgrading the robotic/UFS testing system was based on the system's hardware and software limitations. A higher payload and faster operation was sought for hardware improvement. With regard to software, the system needed to furnish certain choices, in particular the choice of motion descriptions and iterative control algorithms while providing interoperability with computational resources.

4.0 OBJECTIVES

The objective of this study is to develop a biomechanical testing platform consisting of a software design that integrates hardware interfaces, control algorithms, data processing and transformation specialized to the motion description of the knee. The platform uses industrial quality programming languages and development processes and is intended to remain interoperable with arbitrary choices of experimental equipment and eventually with computational formulations.

The experimental approach comprises a high-payload robotic/UFS testing system able to apply clinically relevant loads to the knee and measure resulting 6-DOF kinematics and 3D *in situ* forces throughout the joint. Ultimately, computational simulations are meant to operate concurrently to experimental testing, to provide the bioengineer with the greatest possibilities in his inquiry into the mechanics of the knee joint.

4.1 SPECIFIC AIM 1 – MOTION TRACKING

To formulate a software component whereby the motion of multiple rigid bodies in 3D space can be generally represented and managed. This component would encapsulate all algorithms relevant to tracking the position and orientation of coordinate systems within the

workspace of the testing platform. Furthermore, it could be verified independently of the biomechanical testing platform and be reused identically to manage the motion of bodies in an experiment or in a computer simulation.

4.2 SPECIFIC AIM 2 – KINEMATICS AND KINETICS

To formulate the mathematical relationships necessary to represent all relevant kinematic and kinetic variables of the knee in a clinically relevant motion description in the biomechanical testing platform.

4.3 SPECIFIC AIM 3 – CONTROL ALGORITHMS

To implement a hybrid and iterative control algorithm to find the positions and orientations of the knee that satisfy static equilibrium with a given load and boundary conditions.

4.4 SPECIFIC AIM 4 – SOFTWARE ARCHITECTURE

To propose a modular architecture for the platform software that accommodates the choice of hardware, clinical motion description and control algorithms, and that is accessible to regression testing and software reuse.

5.0 TRACKING MOTION OF MULTIPLE BODIES

Inherent to recording the kinematics of bodies in space is the deployment and management of coordinate systems. These are necessary to record the position and orientation of motion, forces and moments, with respect to landmarks of interest. Typically, experiments involving the biomechanical testing platform include a number of coordinate systems for the subject bones (tibia and femur) as well as the testing equipment (robot base and end-effector, force-moment sensor and spatial digitizer).

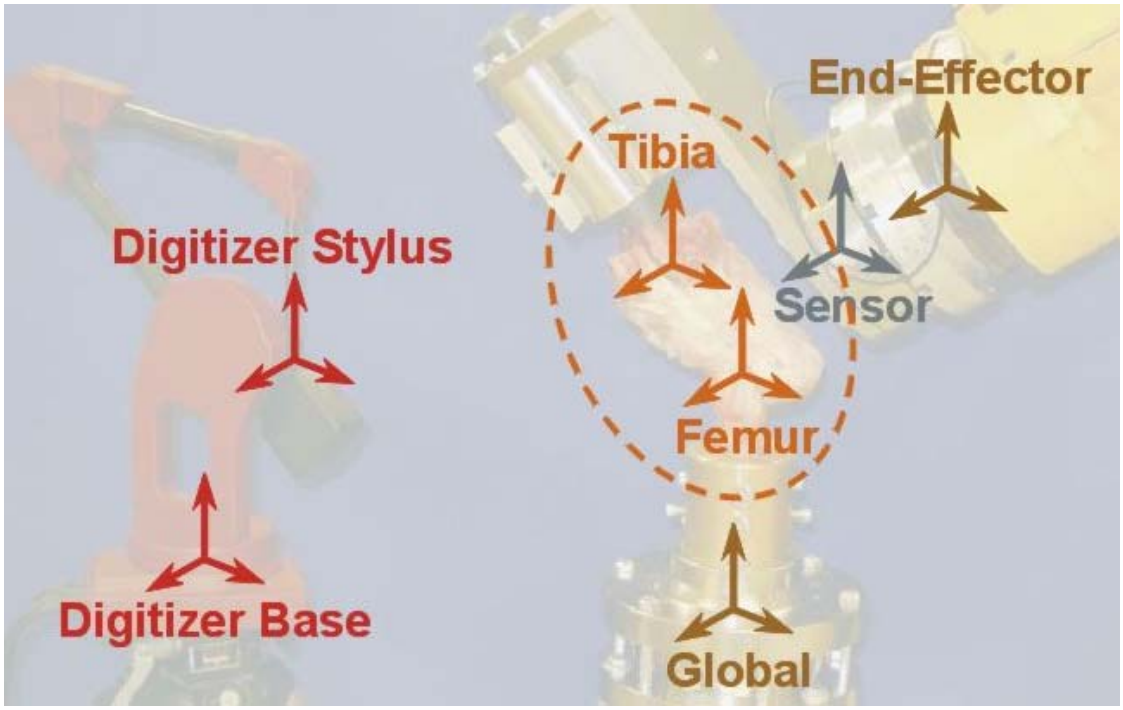


Figure 4 Typical coordinate systems in the workspace of the biomechanical testing platform

Every coordinate system can move about the workspace following the object with which it associates. For example, there is a coordinate system that follows the stylus as it moves to record spatial data during an experiment. Some constraints may also apply to a group of coordinate systems and govern their concerted motion. For example, motion of the end-effector also corresponds with motion at both the sensor and tibia by virtue of their rigid attachment. In view of the motion of one or more objects in the workspace, a simple and general question will invariably be asked: What is the position and orientation of any coordinate system X with respect to another coordinate system Y? The purpose of this section is to provide a programmatic answer.

The problem of tracking objects in the platform workspace reoccurs during the development of control algorithms for both experimental testing equipment and computational methods in biomechanical analyses. It would be of significant advantage to formulate a solution that is widely applicable, reusable and testable.

5.1 MULTI-PURPOSE POSITIONING SYSTEM FOR TRACKING RIGID MOTION OF MULTIPLE BODIES

In this work, a multi-purpose positioning system (MPS) for tracking the motion of multiple bodies is proposed. It consists of a software component encapsulating the mathematical expertise, developed using the C++ programming language and deployed as a static-link library for reuse[72, 73]. The MPS is initialized by the sequential input of the name of each subject coordinate system, together with its position and orientation relative to another, and whether such relation should remain constant due to a constraint. Henceforth, the MPS is able to manage

the spatial relations between all coordinate systems as a function of motion by correctly updating a minimum subset of spatial relations.

Relevant to the design of the MPS are three aspects:

1) Selecting the minimal and sufficient subset of mathematical relations that fully determine the position and orientation of all coordinate systems.
2) Formulating a mechanism whereby all possible inter-relations can be calculated.
3) Selecting the mathematical operations to update all applicable relations in response to coordinate system movement and constraint.

5.1.1 Minimal and Sufficient Set of Mathematical Relations

For a set of N=5 coordinate systems (Femur, Tibia, Sensor, Global, End-Effector), N-1 = 4 independent relations are necessary for the position and orientation of every coordinate system to be known relative to at least one other. On the other hand, the number of possible relations for the set is larger and equivalent to the number of distinct pairs that can be chosen from N coordinate systems, totaling $N \times (N - 1)/2 = 10$. Therefore, the N-1 independent relations constitute a minimal and sufficient subset to store so that, by design, the MPS grows linearly with the size of the problem at hand, rather than parabolically.

As coordinate systems begin to move, the notion of a minimum and sufficient subset of relations remains intact and the MPS is able to determine all possible $N \times (N - 1)/2$ relations by selectively updating those among the N-1 that are stored. When a single body moves, bearing one or more coordinate systems, this is possible to imagine. The general case when multiple

systems move simultaneously and in an arbitrary fashion, can be reduced into commuting operations, each with the net effect of single body movement.

The initialization process for the MPS also reflects the minimal and sufficient set of relations: the user/client must provide N-1 distinct relations. Doing so ensures the integrity of the data, because any additional relations invite the possibility for conflicts or inconsistencies. During initialization, the relations are registered serially with both the names of the coordinate systems involved, and the relation between them is a constraint, i.e. constant:

Table 1 Example initialization of the MPS with coordinate systems typically found in the biomechanical testing platform workspace

Serial Entry #	Name 1	Name 2	Constant?	Value
1	Global	EndEffector	No	${}^{Global}_{EndEffector}R$
2	EndEffector	Sensor	Yes	${}^{EndEffector}_{Sensor}R$
3	Sensor	Tibia	Yes	${}^{Sensor}_{Tibia}R$
4	Global	Femur	No	${}^{Global}_{Femur}R$

In selecting the N-1 inputs for the MPS, two conditions apply:

- The coordinate systems must be governed by a minimal and sufficient number of relations to satisfy a valid MPS at every stage of initialization. In other words, the number of relations must always be n-1, where n is the number of coordinate systems declared at any given stage. For example, upon declaring the pair {Global, End-Effector}, the pair {Sensor, Tibia} would constitute an invalid

choice because it yields n = 4 coordinate systems for only 2 relations. Either {End-Effector, Sensor} or {Global, Femur} are pre-requisite.

- Each constrained subset of coordinate systems must be governed by a minimal and sufficient set of constant relations itself by the end of initialization. For example, should a group of N=10 coordinate systems {A, B, C, D, E, F, G, H, J, K} be subdivided into subsets bound by constraints W={A, B}, X={C, D, E}, Y={F, G, H, J}, Z={K}, where N = W + X + Y + Z, the subsets of relations must also be subdivided accordingly to remain minimal and sufficient. Set W={A, B} requires 1 constant relation, X={C, D, E} requires 2, Y={F, G, H, J} requires 3, while Z={K} requires none. Therefore of the N-1 = 9 possible relations, at least 6 must be declared constant, leaving only three to vary.

	Femur	Tibia	Sensor	EndEff	Global
Femur		*Femur Tibia* **R**	*Femur Sensor* **R**	*Femur EndEff* **R**	*Femur Global* **R**
Tibia	*Tibia Femur* **R**		*Tibia Sensor* **R**	*Tibia EndEff* **R**	*Tibia Global* **R**
Sensor	*Sensor Femur* **R**	*Sensor Tibia* **R**		*Sensor EndEff* **R**	*Sensor Global* **R**
EndEff	*EndEff Femur* **R**	*EndEff Tibia* **R**	*EndEff Sensor* **R**		*EndEff Global* **R**
Global	*Global Femur* **R**	*Global Tibia* **R**	*Global Sensor* **R**	*Global EndEff* **R**	

Figure 5 MPS relations. The highlighted subset constitutes the MPS, as initialized. However, any relation in this table can be queried and obtained. Relationships highlighted in blue are constant and the one highlighted in yellow is variable

5.1.2 Calculating the Full Set of Mathematical Relations

In memory, the relations are arranged into associative arrays or maps for fast query[72, 74]. Any coordinate system provided first in an entry pair during initialization is arbitrarily labeled as a "base" while any system provided second is labeled as "local", and there are as many maps as there are distinct base coordinate systems. Revisiting our earlier example of N=10 coordinate systems, one possible arrangement (considered in the table below) would yield 5 different maps, namely for base systems A, C, E, F and J. The map for A would contain 4 entries for the local coordinate systems B, C, F and K, referring to ${}^{A}_{B}R$, ${}^{A}_{C}R$, ${}^{A}_{F}R$ and ${}^{A}_{K}R$, respectively.

Table 2 Example initialization of the MPS for a large number of coordinate systems

Serial Entry #	Name 1	Name 2	Constant?	Value
1	A	B	Yes	${}^{A}_{B}R$
2	A	C	No	${}^{A}_{C}R$
3	C	D	Yes	${}^{C}_{D}R$
4	E	C	Yes	${}^{E}_{C}R$
5	A	F	No	${}^{A}_{F}R$
6	F	G	Yes	${}^{F}_{G}R$
7	F	H	Yes	${}^{F}_{H}R$
8	J	F	Yes	${}^{J}_{F}R$
9	A	K	No	${}^{A}_{K}R$

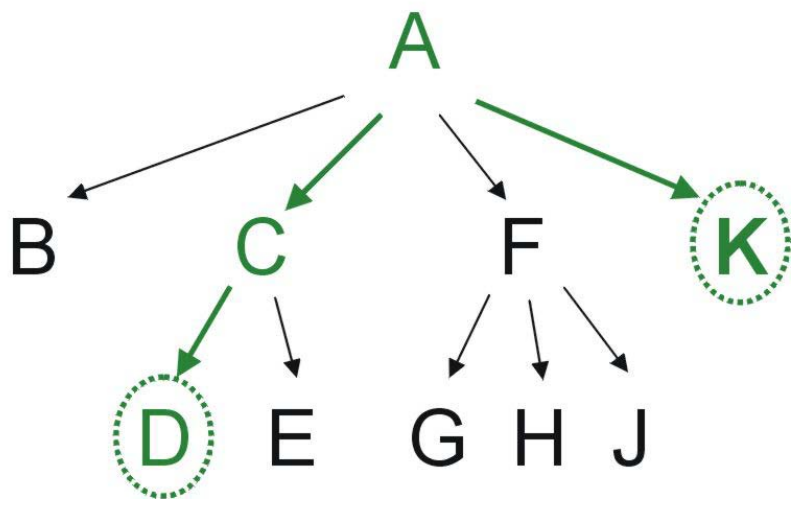


Figure 6 Hierarchy of coordinate system connectivity. The relation between D and K, among the N x (N – 1)/2, is inferred from the relationships {C, D} {A, C} and {A, K}, among the N – 1 that were registered

All such maps can be arranged into a hierarchy that is traversed to calculate the relation between any two coordinate systems.

The hierarchical arrangement is arbitrary. For the MPS, coordinate systems with greater number of relations are arranged at higher levels. For example, in the figure, the positions of coordinate systems A and F (both directly connected to 4 other systems) could be interchanged, varying only the shape of the hierarchy, yet not its connectivity. The hierarchy guarantees that there will always exist a unique upward path connecting any two coordinate systems in a set. Such an arrangement of relationships is referred to as a singly connected network.

Given that a unique path can always be identified to connect two coordinate systems, two fundamental properties of the relations involved are exercised: transitivity and the existence of an inverse. Both are used to compose the N-1 relations into the full set of N x (N –1)/2

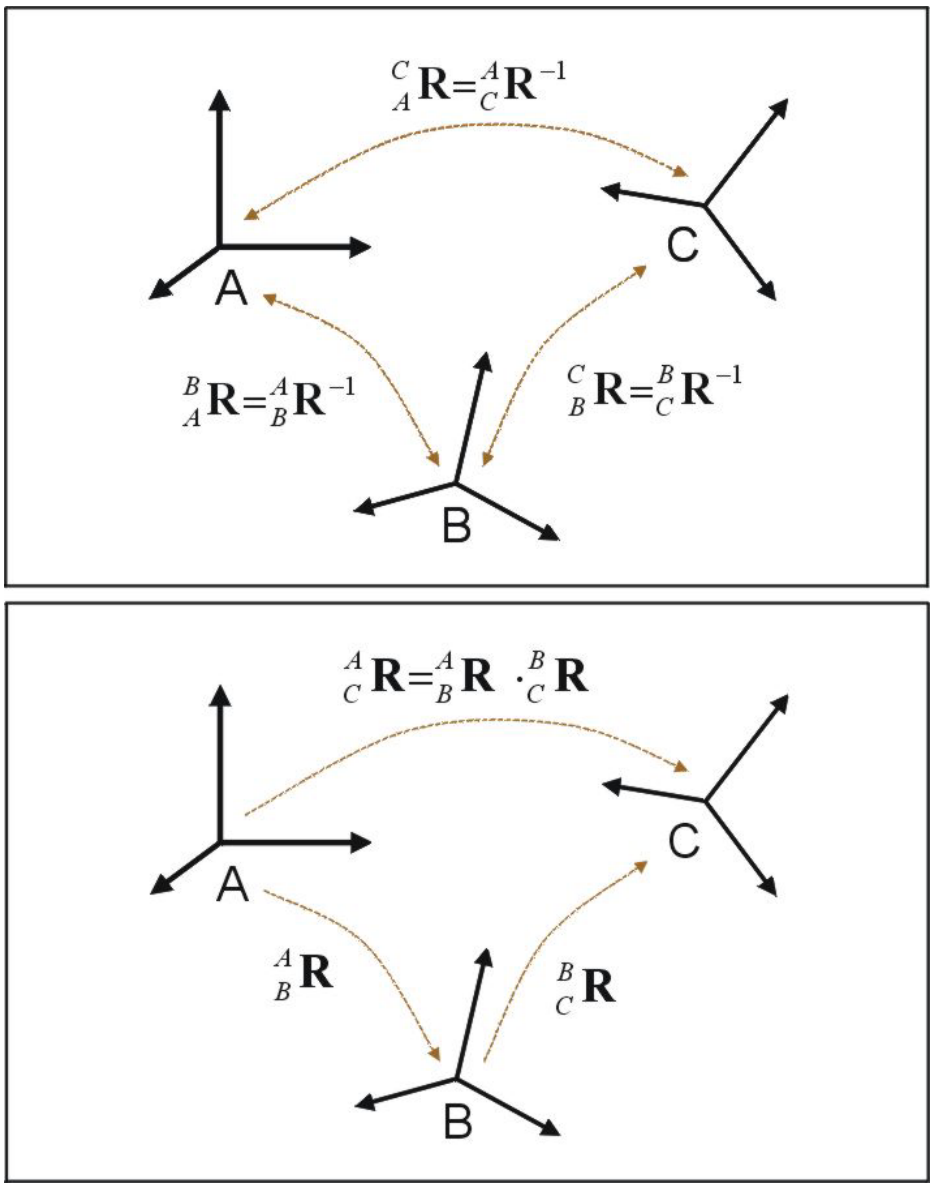


Figure 7 Two elementary properties of relations among coordinate systems: existence of inverse relations (top) and transitive relations (bottom)

For example, given ${}^{C}_{D}R$, ${}^{A}_{C}R$ and ${}^{A}_{K}R$:

$$ {}^{D}_{K}R = {}^{D}_{A}R \cdot {}^{A}_{K}R = {}^{D}_{C}R \cdot {}^{C}_{A}R \cdot {}^{A}_{K}R = \left({}^{C}_{D}R\right)^{-1} \cdot \left({}^{A}_{C}R\right)^{-1} \cdot {}^{A}_{K}R \qquad (5\text{-}1) $$

5.1.3 Motion and Constraint

Although not mentioned due to the generality of the claims made thus far, the relations between the coordinate systems are actually 4x4 homogenous orthonormal matrices. In the previous sections, such matrices have provided the description of one object relative to another, i.e., the position and orientation of coordinate system 2 with respect to coordinate system 1, or ${}^{1}_{2}R$. Alternatively, such matrices can also be interpreted as transforming a single object from an initial position and orientation to a final one by a translation and/or rotation. Using both interpretations, motion is represented in the MPS by the application of matrices encoding transformation to the appropriate matrices that describe the corresponding coordinate systems.

When a motion is applied to a coordinate system K (K is known relative to a base system A), in the absence of constraints affecting K, the following expression is used:

$$ {}^{A}_{K}R_{final} = {}^{A}_{K}R_{initial}\,{}^{K}R_{motion} \tag{5-2} $$

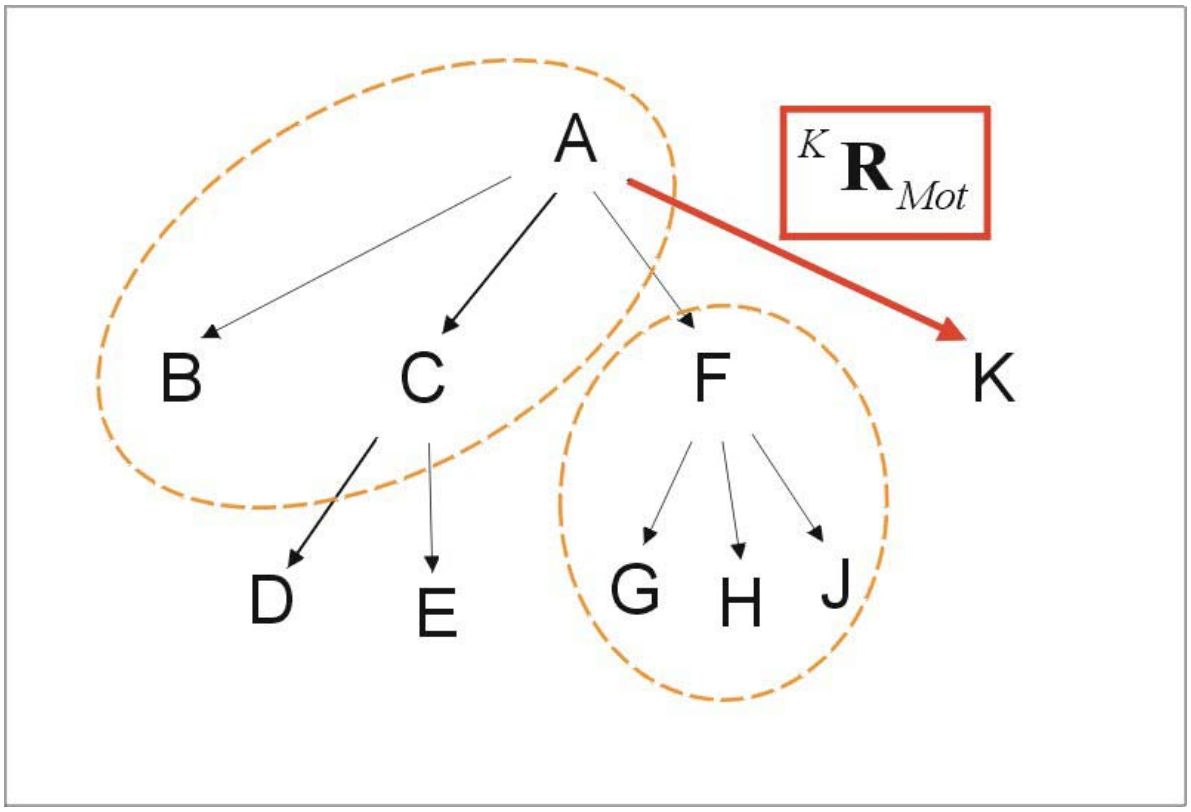


Figure 8 Simple motion: Systems A, B and C are constrained to each other; so are F, G, H and J. Notice that the motion that is applied to coordinate system K. It affects the single variable relation between A and K

If motion occurs at a coordinate system B that is constrained to systems A and C, motion of B must also be applied to A and C. Before doing so, the motion must be appropriately transformed to the target corresponding systems. For example, for coordinate system A:

$$\begin{aligned} {}^{A}_{K}R_{final} &= {}^{A}R_{motion} \cdot {}^{A}_{K}R_{initial} \\ {}^{A}_{F}R_{final} &= {}^{A}R_{motion} \cdot {}^{A}_{F}R_{initial} \end{aligned} \tag{5-3}$$

where ${}^{A}R_{motion} = \left({}^{B}_{A}R_{const}\right)^{-1} \cdot {}^{B}R_{motion} \cdot \left({}^{B}_{A}R_{const}\right)$. The same concept applies to coordinate system C, with variable relations to D and E.

$$
\begin{aligned}
{}^{C}_{D}R_{final} &= {}^{C}R_{motion} \cdot {}^{C}_{D}R_{initial} \\
{}^{C}_{E}R_{final} &= {}^{C}R_{motion} \cdot {}^{C}_{E}R_{initial}
\end{aligned}
\tag{5-4}
$$

where ${}^{C}R_{motion} = \left({}^{B}_{C}R_{const}\right)^{-1} \cdot {}^{B}R_{motion} \cdot {}^{B}_{C}R_{const}$

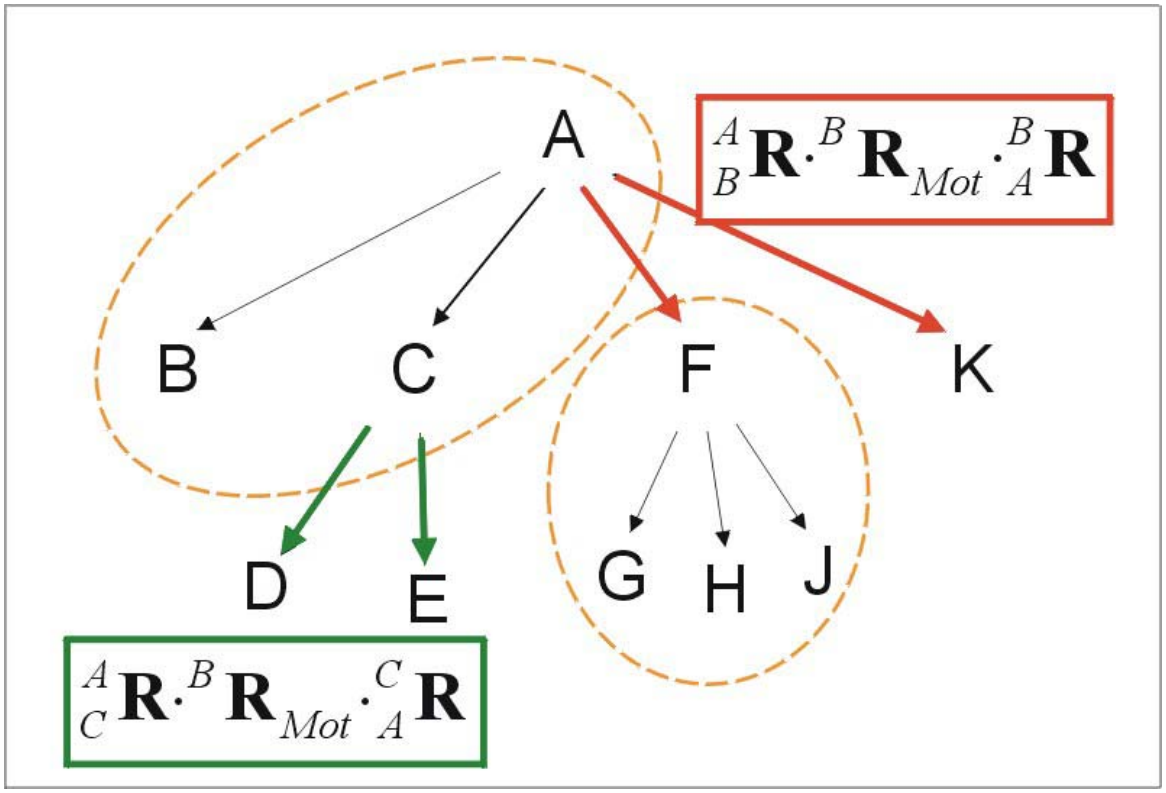


Figure 9 Motion in the presence of constraints: Systems A, B and C are constrained to each other; so are F, G, H and J. Notice that the motion applied is that of system B, which has a single constant relationship. The motion of B, however, produces motion at A and C that must be correctly accounted for by the appropriate transformations in red and green, respectively

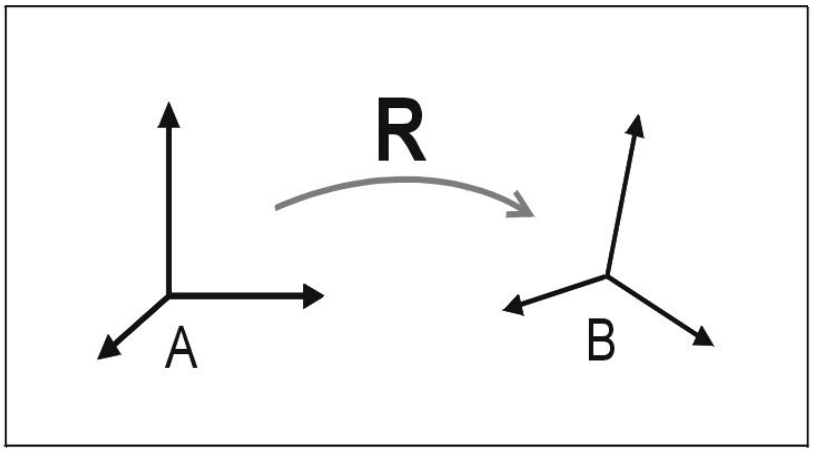
Generally, to all coordinate systems 'i', constrained to a coordinate system D to which a motion has been applied, the following relation are applied for all such i:

$$
{}^{G}_{D}R_{final} = {}^{G}_{D}R_{initial} \cdot {}^{D}R_{motion} \tag{5-5}
$$

$$
{}^{G}_{i}R_{final} = {}^{G}_{i}R_{initial} \cdot \left({}^{D}_{i}R_{const}\right)^{-1} \cdot {}^{D}R_{motion} \cdot {}^{D}_{i}R_{const} \tag{5-6}
$$

5.2 DISCUSSION

The present section outlines a general method for representing the kinematics of multiple objects that are in mutual relative motion. The mathematics are later assembled into software components with the capacity to encapsulate this expertise and make it available for not only the robotic/UFS testing system, but any other deserving application within the biomechanical testing platform. Such an encapsulated software component is not only reusable, but also testable. This is a significant innovation and departure from the development of the legacy control system.



$$\mathbf{R}={}^{A}_{B}\mathbf{R}=\left[\begin{array}{ccc|c} {}^{A}_{B}\mathbf{x} & {}^{A}_{B}\mathbf{y} & {}^{A}_{B}\mathbf{z} & {}^{A}_{B}\mathbf{p} \\ 0 & 0 & 0 & 1 \end{array}\right] \qquad \mathbf{R}={}^{A}\mathbf{R}_{Mot}=\left[\begin{array}{ccc|c} & Rotation & & Translation \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Figure 10 Two possible interpretations. (Bottom left) R is the expression of the coordinate system B in terms of coordinate system A; the matrix features the axes vectors x, y, z and the position vector p, of the B coordinate system relative to A. (Bottom right) R the motion applied to a single coordinate system that takes it from configuration A to configuration B. The motion involves a rotational and translation parts.

6.0 THE JOINT MOTION DESCRIPTION

The joint motion description is designed to conveniently describe the 6-DOF motion of the knee, all translations and rotations, in clinically relevant terms. It is convention that the three translations are named medial-lateral (ML), anterior-posterior (AP), and proximal-distal (PD), and the three rotations flexion-extension (FE), varus-valgus (VV) and internal-external (IE). Although the primary movement of the knee joint is that of flexion-extension, movements do occur in all 6-DOF. For example, small disturbances in the normal path of motion of the knee in the other DOFs, as the increased AP laxity associated with ACL deficiency, are important indicatives of knee injury, and impact the functional performance of the patient.

The joint motion description serves the same function of reporting the motion, forces and moments at the knee, as ordinary coordinate systems. However, its construction is different. It consists of three axes, not mutually perpendicular or intersecting along which translations and rotations occur: ML translation and FE rotations share an axis, so do AP and VV, and finally PD and IE. The arrangement of the axes can be visualized as three segments in a serial linkage system consisting of three cylindrical joints. As the knee moves, each of these cylindrical joints can be thought to slide and rotate, to provide the appropriate configuration of values for ML, AP, PD, FE, VV and IE at any point in time. From now on, a configuration in the JMD (the current position and orientation along and about each of the 6-DOFs) will be denoted as $(d_{ML}, d_{AP}, d_{PD}, \theta_{FE}, \theta_{VV}, \theta_{IE})$.

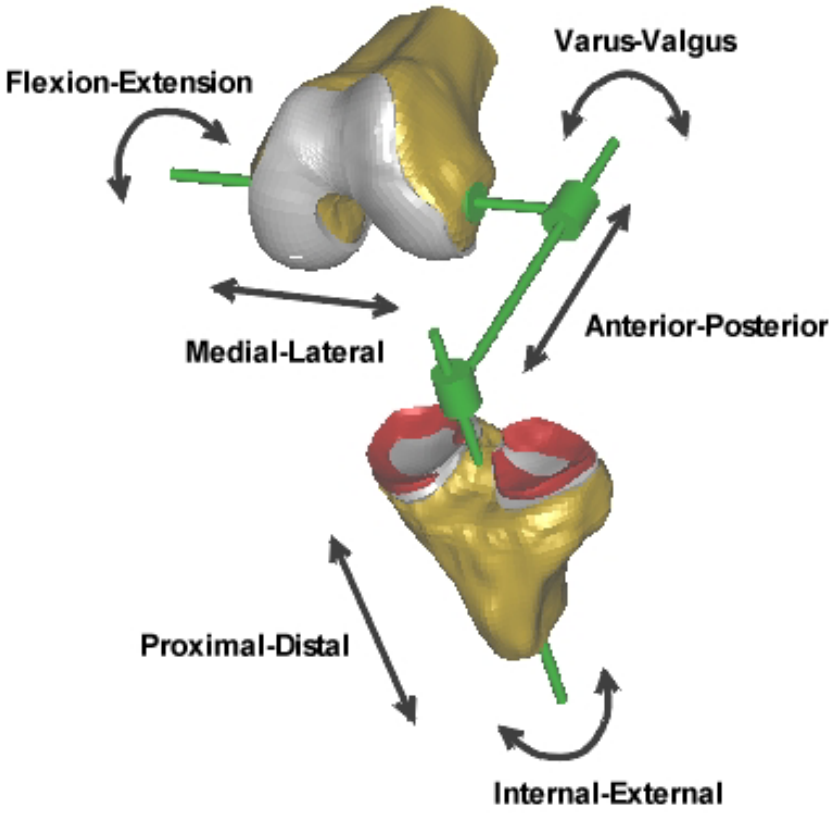


Figure 11 Joint Motion Description with all 6-DOFs identified

There are two important tasks at hand. First is that of inter-converting kinematics between the JMD coordinates and the orthogonal coordinate systems that are actually used to track positions and orientations in the MPS. The other is that of inter-converting the kinetics, i.e., forces and moments.

6.1 KINEMATICS

The following work adapts the definition of the JMD, as first proposed by Grood, Suntay and Chao[56, 57], to the specific coordinate system assignments and conventions adopted for the MPS in preceding sections.

6.1.1 From JMD to Orthogonal Coordinate Systems

The configuration of the JMD is uniquely determined by the relative position and orientation between the femur and the tibia, ${}^{F}_{T}R$, and vice-versa. In order to obtain ${}^{F}_{T}R$ from a given configuration of ML, AP, PD, FE, VV and IE, i.e., $(d_{ML}, d_{AP}, d_{PD}, \theta_{FE}, \theta_{VV}, \theta_{IE})$ a sequence of intermediate coordinate systems (L_o through L_6) can be visualized at strategic points along the JMD linkage, and whose axes are conveniently aligned with the links. By construction, adjacent systems are related by a simple translation or rotation that coincides with one of the 6 motions in the JMD. Because L_o is the same as F and L_6 the same as T, the overall transformation required to move from the beginning to the end of the sequence, is the same as ${}^{F}_{T}R$.

The detailed mathematical analysis is included. For the case of a left knee, positive translations, namely medial, anterior and proximal, occur along the negative directions for x of L_1, z of L_3 and y of L_5, respectively. On the other hand, positive rotations, flexion, varus and

internal, occur about positive direction for x of L_0, z of L_2 and y of L_4. These conventions are for the motion of the tibia relative to a fixed femur. From these facts, it is concluded that[75]:

$$ {}^{F}_{T}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0) \cdot R_z(\theta_{VV}) \cdot T(0,0,-d_{AP}) \cdot R_y(\theta_{IE}) \cdot T(0,-d_{PD},0) \qquad (6\text{-}1) $$

$$ {}^{F}_{T}R = \begin{bmatrix} C_{VV}C_{IE} & -S_{VV} & C_{VV}S_{IE} & -d_{ML} + d_{PD}S_{VV} \\ C_{FE}S_{VV}C_{IE} + S_{FE}S_{IE} & C_{FE}C_{VV} & C_{FE}S_{VV}S_{IE} - S_{FE}C_{IE} & -d_{PD}C_{FE}C_{VV} + d_{AP}S_{FE} \\ S_{FE}S_{VV}C_{IE} - C_{FE}S_{IE} & S_{FE}C_{VV} & S_{FE}S_{VV}S_{IE} + C_{FE}C_{IE} & -d_{AP}C_{FE} - d_{PD}S_{FE}C_{VV} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6\text{-}2) $$

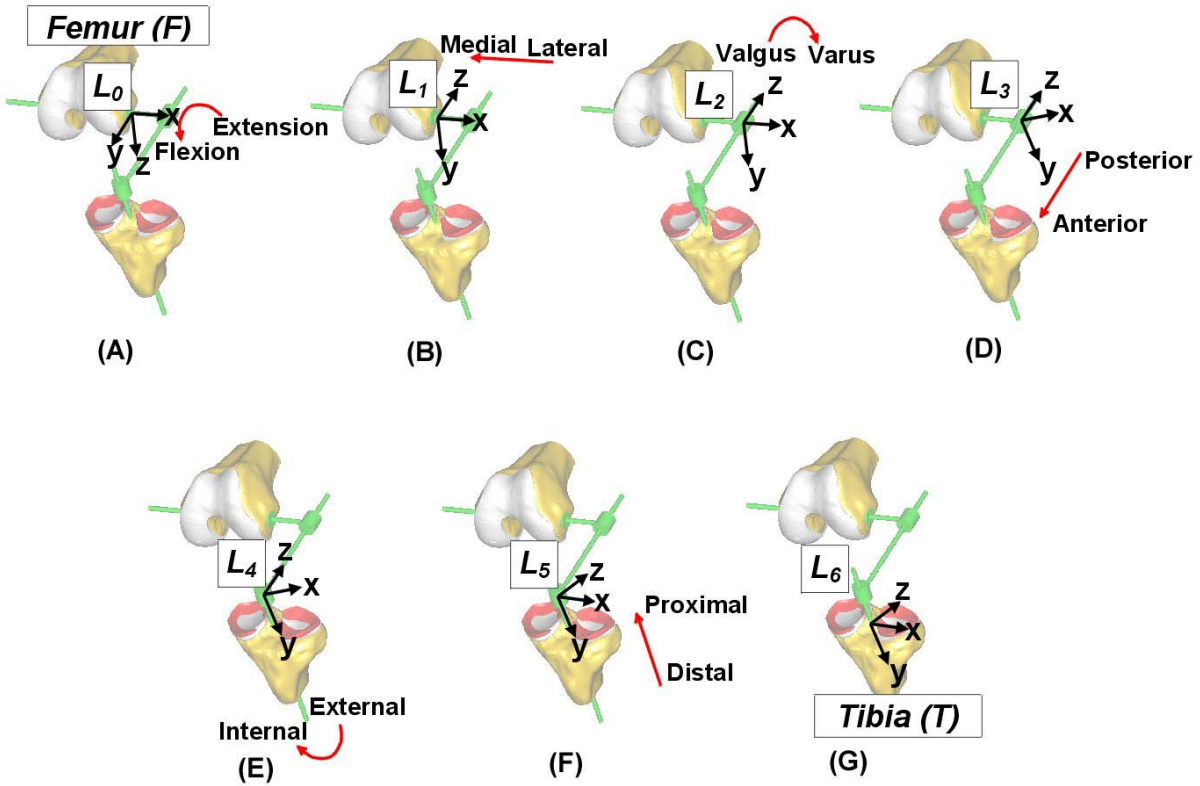


Figure 12 Using a sequence of orthogonal coordinate systems to produce equivalent motion to that in the JMD for a left knee (positive directions of motion in the JMD are displayed by red arrows for all DOFs). The knee joint depicted is flexed (90°), lateral (20 mm), valgus(5°), anterior (40 mm), internal (20°), and distal (10 mm)

Table 3 Relating the configuration of the JMD to orthogonal transformations

Motion	Increment	Overall Transformation
Flexion - Extension	$R_x(\theta_{FE})$	${}^{L0}_{L1}R = R_x(\theta_{FE})$
Medial - Lateral	$T(-d_{ML},0,0)$	${}^{L0}_{L2}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0)$
Varus - Valgus	$R_z(\theta_{VV})$	${}^{L0}_{L3}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0) \cdot R_z(\theta_{FE})$
Anterior - Posterior	$T(0,0,-d_{AP})$	${}^{L0}_{L4}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0) \cdot R_z(\theta_{VV}) \cdot T(0,0,-d_{AP})$
Internal - External	$R_y(\theta_{IE})$	${}^{L0}_{L5}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0) \cdot R_z(\theta_{VV}) \cdot T(0,0,-d_{AP}) \cdot R_y(\theta_{IE})$
Proximal - Distal	$T(0,-d_{PD},0)$	${}^{L0}_{L6}R = R_x(\theta_{FE}) \cdot T(-d_{ML},0,0) \cdot R_z(\theta_{VV}) \cdot T(0,0,-d_{AP}) \cdot R_y(\theta_{IE}) \cdot T(0,-d_{PD},0)$

6.1.2 From Orthogonal Coordinate Systems to JMD

The next problem is to extract the configuration of the JMD linkage $(d_{ML}, d_{AP}, d_{PD}, \theta_{FE}, \theta_{VV}, \theta_{IE})$ from the relative position and orientation between the femur and tibia. This amounts to decomposing the overall ${}^{F}_{T}R$ transformation into individual rotations and translations.

In order to deduce the necessary mathematical expressions, it is helpful to recognize that after several individual rotations and translations are multiplied into an overall transformation matrix, the rotations can be treated independently of translations.

This can be illustrated using partitioned matrix operations. For example, the canonical 4x4 matrix for rotation about the x-axis is partitioned as:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{ccc:c} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c:c} R_x^{3x3} & \mathbf{O}^{3x1} \\ \hdashline \mathbf{O}^{1x3} & 1 \end{array}\right] \tag{6-3}$$

The y- and z-axes rotation matrix can be partitioned into the same form. Similarly, the translation matrix can be partitioned:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{ccc:c} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c:c} \mathbf{I}^{3x3} & T_{xyz} \\ \hdashline \mathbf{O}^{1x3} & 1 \end{array}\right] \tag{6-4}$$

Upon multiplying an arbitrary number and sequence of rotation and translation matrices, the resultant transformation can also be partitioned, as follows:

$$R_{RESULTANT} = R_1(\theta_1)\cdot T_2(d_{x2},d_{y2},d_{z2})\cdot R_3(\theta_3)\cdot R_4(\theta_4)\cdot T_5(d_{x5},d_{y5},d_{z5})$$
$$= \left[\begin{array}{c:c} R_1^{3x3}(\theta_1)\cdot R_3^{3x3}(\theta_3)\cdot R_4^{3x3}(\theta_4) & T_{xyz}^{3x1} \\ \hdashline \mathbf{O}^{1x3} & 1 \end{array}\right] = \left[\begin{array}{c:c} R_{RESULTANT}^{3x3} & T_{xyz}^{3x1} \\ \hdashline \mathbf{O}^{1x3} & 1 \end{array}\right] \quad (6\text{-}5)$$

By equating each of the submatrix pairs, the rotational component in the transformation $R_{RESULTANT}^{3x3}$ is found to depend on the sequence of rotation matrices R_1, R_3 and R_4 involved in the multiplication and not on the translations T_2 and T_5. The translation component T_{xyz}, however, has a more complicated relationship to the sequence of translation matrices T_2 and T_5. Each component will be treated separately

Using this principle, the rotational component of the ${}_T^F R$ transformation, ${}_T^F R^{3x3}$ is decomposed into:

$${}_T^F R^{3x3} = R_x^{3x3}(\theta_{FE})\cdot R_z^{3x3}(\theta_{VV})\cdot R_y^{3x3}(\theta_{IE}) \quad (6\text{-}6)$$

Because ${}_T^F R^{3x3}$ is given, the calculation could be pursued as stated, solving for the individual angles. Instead, it is preferable to rearrange it into the following form in the interest of numerical accuracy[76]:

$$(R_x^{3x3}(\theta_{FE}))^{-1}\cdot {}_T^F R^{3x3} = R_z^{3x3}(\theta_{VV})\cdot R_y^{3x3}(\theta_{IE}) \quad (6\text{-}7)$$

Substituting the form of each of the matrices involved yields:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{FE} & S_{FE} \\ 0 & -S_{FE} & C_{FE} \end{bmatrix}\cdot\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_{VV} & -S_{VV} & 0 \\ S_{VV} & C_{VV} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} C_{IE} & 0 & S_{IE} \\ 0 & 1 & 0 \\ -S_{IE} & 0 & C_{IE} \end{bmatrix} \quad (6\text{-}8)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ C_{FE}r_{21}+S_{FE}r_{31} & C_{FE}r_{22}+S_{FE}r_{32} & C_{FE}r_{23}+S_{FE}r_{33} \\ -S_{FE}r_{21}+C_{FE}r_{31} & -S_{FE}r_{22}+C_{FE}r_{32} & -S_{FE}r_{23}+C_{FE}r_{33} \end{bmatrix} = \begin{bmatrix} C_{VV}C_{IE} & -S_{VV} & C_{VV}S_{IE} \\ S_{VV}C_{IE} & C_{VV} & S_{VV}S_{IE} \\ -S_{IE} & 0 & C_{IE} \end{bmatrix} \quad (6\text{-}9)$$

Finally, special selections are made from the above 9 equations to be solved with the inverse tangent function. The first yields θ_{FE}:

$$\begin{aligned} -S_{FE}r_{22} + C_{FE}r_{32} &= 0 \\ \frac{S_{FE}}{C_{FE}} &= \frac{r_{32}}{r_{22}} \\ \theta_{FE} &= \tan^{-1}\left(\frac{r_{32}}{r_{22}}\right) \end{aligned} \tag{6-10}$$

The second yields θ_{VV}:

$$\begin{aligned} r_{12} &= -S_{VV} \\ C_{FE}r_{22} + S_{FE}r_{32} &= C_{VV} \\ \frac{S_{VV}}{C_{VV}} &= \frac{-r_{12}}{C_{FE}r_{22} + S_{FE}r_{32}} \\ \theta_{VV} &= \tan^{-1}\left(\frac{-r_{12}}{C_{FE}r_{22} + S_{FE}r_{32}}\right) \end{aligned} \tag{6-11}$$

The final yields θ_{IE}:

$$\begin{aligned} -S_{FE}r_{21} + C_{FE}r_{31} &= -S_{IE} \\ -S_{FE}r_{23} + C_{FE}r_{33} &= C_{IE} \\ \frac{S_{IE}}{C_{IE}} &= \frac{S_{FE}r_{21} - C_{FE}r_{31}}{-S_{FE}r_{23} + C_{FE}r_{33}} \\ \theta_{IE} &= \tan^{-1}\left(\frac{S_{FE}r_{21} - C_{FE}r_{31}}{-S_{FE}r_{23} + C_{FE}r_{33}}\right) \end{aligned} \tag{6-12}$$

Taking advantage of the fact that the translations are confined to the T_{xyz}^{3x3} sub-matrix, the translation component can be calculated:

$$ {}^{F}_{T}R = \left[\begin{array}{c|c} {}^{F}_{T}R^{3x3} & \begin{matrix} d_x \\ d_y \\ d_z \end{matrix} \\ \hline \mathbf{O}^{1x3} & 1 \end{array}\right] = \left[\begin{array}{ccc|c} C_{VV}C_{IE} & -S_{VV} & C_{VV}S_{IE} & -d_{ML}+d_{PD}S_{VV} \\ C_{FE}S_{VV}C_{IE}+S_{VV}S_{IE} & C_{FE}C_{VV} & C_{FE}S_{VV}S_{IE}-S_{FE}C_{IE} & -d_{PD}C_{FE}C_{VV}+d_{AP}S_{FE} \\ S_{FE}S_{VV}C_{IE}-C_{FE}S_{IE} & S_{FE}C_{VV} & S_{FE}S_{VV}S_{IE}+C_{FE}C_{IE} & -d_{AP}C_{FE}-d_{PD}S_{FE}C_{VV} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{6-14} $$

This last expression yields:

$$ \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} -1 & 0 & S_{VV} \\ 0 & S_{FE} & -C_{FE}C_{VV} \\ 0 & -C_{FE} & -S_{FE}C_{VV} \end{bmatrix} \begin{bmatrix} d_{ML} \\ d_{AP} \\ d_{PD} \end{bmatrix} \tag{6-13} $$

And upon solving the equation, the translation components of the JMD configuration (d_{ML}, d_{AP}, and d_{PD}) are finally obtained:

$$ \begin{bmatrix} d_{ML} \\ d_{AP} \\ d_{PD} \end{bmatrix} = \frac{1}{C_{VV}} \begin{bmatrix} -C_{VV} & -C_{FE}S_{VV} & -S_{FE}S_{VV} \\ 0 & S_{FE}C_{VV} & -C_{FE}C_{VV} \\ 0 & -C_{FE} & -S_{FE} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{6-15} $$

6.1.3 Prescribing Motion to the Knee

Suppose the knee joint must move from an initial configuration $(d^o_{ML}, d^o_{AP}, d^o_{PD}, \theta^o_{FE}, \theta^o_{VV}, \theta^o_{IE})$ to a final configuration $(d^f_{ML}, d^f_{AP}, d^f_{PD}, \theta^f_{FE}, \theta^f_{VV}, \theta^f_{IE})$. The increment of motion in the JMD is simply:

$$ \begin{aligned} &(\Delta d_{ML}, \Delta d_{AP}, \Delta d_{PD}, \Delta\theta_{FE}, \Delta\theta_{VV}, \Delta\theta_{IE}) = \\ &\quad (d^f_{ML}, d^f_{AP}, d^f_{PD}, \theta^f_{FE}, \theta^f_{VV}, \theta^f_{IE}) - (d^o_{ML}, d^o_{AP}, d^o_{PD}, \theta^o_{FE}, \theta^o_{VV}, \theta^o_{IE}) \end{aligned} \tag{6-16} $$

$$ {}^{F}_{T}R_f = R_x(\theta^f_{FE}) \cdot T(-d^f_{ML},0,0) \cdot R_z(\theta^f_{VV}) \cdot T(0,0,-d^f_{AP}) \cdot R_y(\theta^f_{IE}) \cdot T(0,-d^f_{PD},0) \tag{6-17} $$

The appropriate increment of motion that must be applied to the *T* system is can be inferred. The relationship between the femur and tibia before and after the movement must be: Therefore, the relative motion of the tibia is given by:

$$ {}^{T}R_{Mot} = {}^{T}_{F}R_{o} \cdot {}^{F}_{T}R_{f} = ({}^{F}_{T}R_{o})^{-1} \cdot {}^{F}_{T}R_{f} \tag{6-18} $$

6.2 KINETICS

The robotic/UFS testing system is capable of applying loads to the knee joint that can realistically simulate the mechanics of a clinical examination, physical activity or rehabilitation. In order to do so, it requires force and moment feedback from the UFS. The system is designed to operate using the JMD as a reference not only for kinematics, but for forces and moments as well. Several steps are required to translate the force and moment readings that are obtained from the UFS into forces and moments in the JMD:

1) Forces and moments are initially measured in device specific units (counts)
2) The measurement in counts is translated into physical units of N and N-mm
3) The weight of clamping devices attached to the UFS are tared from the force and moment measurement, yielding a net values at the knee
4) The net force and moments at the knee is finally transformed into the JMD

Inherent to recording the kinematics of bodies in space is the management of coordinate system. Two types of coordinate systems are adopted in the robotic/UFS testing system. The first consists of ordinary right-handed and orthogonal coordinate systems that track the position and orientation of landmarks throughout the testing system and the specimen. They are necessary

to record the direction of all motions, forces and moments, with respect to a global frame of reference. The second type corresponds to a non-orthogonal system specialized for the study of the human knee joint. This Joint Motion Description (JMD) distills the motion, forces and moments, into a clinically relevant format, convenient for reporting the biomechanical behavior of the specimen.

The following sections adapt the calculation of Jacobians matrices to the transformations of forces and moments between orthogonal coordinates and JMD. The calculations constitute a standard problem in robotics[75, 77] (the calculation of forces and moments that must be applied the joints of a manipulator in order to achieve a given payload), that was first given an application to problem at hand by Fujie[58]. This work details the calculations to the conventions and assignments of the MPS.

6.2.1 Forces and Moments in the JMD

Once the net forces and moments are known at the sensor, it is necessary to transform them into the JMD. This entails transforming the forces and moments from an orthogonal coordinate system *S*, into the non-orthogonal JMD, and amounts to identifying a matrix **J** that satisfies:

$$^{JMD}F = \mathbf{J} \cdot {}^{S}F \tag{6-19}$$

where ^{JMD}F and ^{S}F are 6-DOF vectors containing both the forces and moments measured, distributed along the relevant axes of each coordinate system.

$$ {}^{JMD}F = \begin{bmatrix} F_{ML} \\ F_{AP} \\ F_{PD} \\ M_{FE} \\ M_{VV} \\ M_{IE} \end{bmatrix} \quad \text{and} \quad {}^{S}F = \begin{bmatrix} {}^{S}F_x \\ {}^{S}F_y \\ {}^{S}F_z \\ {}^{S}M_x \\ {}^{S}M_y \\ {}^{S}M_z \end{bmatrix} \tag{6-20} $$

The overall transformation can be thought to occur in two stages. The first involves a change in the center of measurement of the force and moment from the sensor *S* system, to a selected point in the knee, which is chosen to be the tibia *T*. The second transformation involves a deformation of the coordinate axes, from the orthogonal *T* system, to the non-orthogonal JMD:

$$ \begin{aligned} {}^{T}F &= \mathbf{J}_1 \cdot {}^{S}F \\ {}^{JMD}F &= \mathbf{J}_2 \cdot {}^{T}F \\ {}^{JMD}F &= \mathbf{J}_2 \cdot \mathbf{J}_1 \cdot {}^{S}F \end{aligned} \tag{6-21} $$

Comparing equations 6-19 and 6-21:

$$ \mathbf{J} = \mathbf{J}_2 \cdot \mathbf{J}_1 \tag{6-22} $$

The matrices $\mathbf{J}$, $\mathbf{J}_1$ and $\mathbf{J}_2$ are called the Jacobian transformation matrices.

A convenient method to analytically derive the Jacobian matrices is to invoke the principle of virtual work[77]. Using this principle, $\mathbf{J}_1$ and $\mathbf{J}_2$ can be found and multiplied to yield $\mathbf{J}$.

The principle of virtual work states that the amount of work that would be needed to move the point of application of a force through a virtual differential displacement must be independent of the choice of coordinate system[78]. For example, applying this to the orthogonal *T* system and the non-orthogonal JMD, we get:

$$
\begin{aligned}
&\delta W_T = \delta W_{JMD} \\
&\left({}^{T}F\right)^{\mathbf{T}} \cdot {}^{T}\delta x = \left({}^{JMD}F\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x \\
&\begin{bmatrix} {}^{T}\delta F_x & {}^{T}\delta F_y & {}^{T}\delta F_z & {}^{T}\delta M_x & {}^{T}\delta M_y & {}^{T}\delta M_z \end{bmatrix} \cdot \begin{bmatrix} {}^{T}\delta d_x \\ {}^{T}\delta d_y \\ {}^{T}\delta d_z \\ {}^{T}\delta \theta_x \\ {}^{T}\delta \theta_y \\ {}^{T}\delta \theta_z \end{bmatrix} = \begin{bmatrix} \delta F_{ML} & \delta F_{AP} & \delta F_{PD} & \delta M_{FE} & \delta M_{VE} & \delta M_{IE} \end{bmatrix} \cdot \begin{bmatrix} \delta d_{ML} \\ \delta d_{AP} \\ \delta d_{PD} \\ \delta \theta_{FE} \\ \delta \theta_{VV} \\ \delta \theta_{IE} \end{bmatrix}
\end{aligned}
\tag{6-23}
$$

In other words, displacing the force ${}^{JMD}F$ by an amount ${}^{JMD}\delta x$ should require the same addition of work to the system, as displacing the force ${}^{T}F$ by ${}^{T}\delta x$. Thus, the transformation that deforms the coordinate system *T* into JMD, and therefore deforms its differentials ${}^{T}\delta x$ into differentials ${}^{JMD}\delta x$, is related to that which transforms forces and moments:

$$
\begin{aligned}
&\left({}^{T}F\right)^{\mathbf{T}} \cdot {}^{T}\delta x = \left({}^{JMD}F\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x \\
&\left({}^{T}F\right)^{\mathbf{T}} \cdot {}^{T}\delta x = \left(\mathbf{J}_2 \cdot {}^{T}F\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x \\
&\left({}^{T}F\right)^{\mathbf{T}} \cdot {}^{T}\delta x = \left({}^{T}F\right)^{\mathbf{T}} \cdot \left(\mathbf{J}_2\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x
\end{aligned}
\tag{6-24}
$$

Simplifying affairs:

$$
\begin{aligned}
&\left({}^{T}F\right)^{\mathbf{T}} \cdot {}^{T}\delta x = \left({}^{T}F\right)^{\mathbf{T}} \cdot \left(\mathbf{J}_2\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x \\
&{}^{T}\delta x = \left(\mathbf{J}_2\right)^{\mathbf{T}} \cdot {}^{JMD}\delta x \\
&{}^{T}\delta x = {}^{T}_{JMD}\mathbf{J} \cdot {}^{JMD}\delta x
\end{aligned}
\tag{6-25}
$$

Noticing that from now on, $\mathbf{J}_2$ will be renamed to $\left({}^{T}_{JMD}\mathbf{J}\right)^{\mathbf{T}}$, the two important relationships between *T* and JMD are:

$$
\begin{aligned}
&{}^{T}\delta x = {}^{T}_{JMD}\mathbf{J} \cdot {}^{JMD}\delta x \\
&{}^{JMD}F = \left({}^{T}_{JMD}\mathbf{J}\right)^{\mathbf{T}} \cdot {}^{T}F
\end{aligned}
\tag{6-26}
$$

A similar analysis yields, after renaming $\mathbf{J}_1$ to $\left({}^{S}_{T}\mathbf{J}\right)^{\mathbf{T}}$, the following two relationships:

$$
\begin{aligned}
{}^{S}\delta x &= {}^{S}_{T}\mathbf{J} \cdot {}^{T}\delta x \\
{}^{T}F &= \left({}^{S}_{T}\mathbf{J}\right)^{\mathbf{T}} \cdot {}^{S}F
\end{aligned}
\tag{6-27}
$$

From the last two equations, it can be inferred that deriving the analytical form of the Jacobian matrices can be done by studying the properties of the transformation of the forces and moments or, equivalently, of the increments of position in the relevant coordinate systems.

It is clear that the immediate task is the derivation of $\left({}^{T}_{JMD}\mathbf{J}\right)^{\mathbf{T}}$. When interpreted as

$$
{}^{T}\delta x = {}^{T}_{JMD}\mathbf{J} \cdot {}^{JMD}\delta x \tag{6-28}
$$

the Jacobian matrix ${}^{T}_{JMD}\mathbf{J}$ is responsible for transforming an increment, as it occurs in the non-orthogonal JMD, into its form in the orthogonal tibial coordinate system *T*. ${}^{T}\delta x$ and ${}^{JMD}\delta x$ are differential increments[77] in each of the coordinate systems JMD and *T*. One interpretation of Equation 6-28 is that the elements in the vector ${}^{T}\delta x$ are linear transformations of the elements in ${}^{JMD}\delta x$. In other words:

$$
\begin{bmatrix}
{}^{T}\delta d_x \\ {}^{T}\delta d_y \\ {}^{T}\delta d_z \\ {}^{T}\delta \theta_x \\ {}^{T}\delta \theta_y \\ {}^{T}\delta \theta_z
\end{bmatrix}
=
\begin{bmatrix}
{}^{T}\delta d_x\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right) \\
{}^{T}\delta d_y\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right) \\
{}^{T}\delta d_z\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right) \\
{}^{T}\delta \theta_x\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right) \\
{}^{T}\delta \theta_y\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right) \\
{}^{T}\delta \theta_z\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta \theta_{FE}, \delta \theta_{VV}, \delta \theta_{IE}\right)
\end{bmatrix}
\tag{6-29}
$$

Each of the six sub-equations in Equation 6-29 can be recognized as the statement of a total differential. For example, the total differential in the first row, has the form

$$
\begin{aligned}
{}^{T}\delta d_x &= {}^{T}\delta d_x\left(\delta d_{ML}, \delta d_{AP}, \delta d_{PD}, \delta\theta_{FE}, \delta\theta_{VV}, \delta\theta_{IE}\right) \\
&= \left(\frac{\partial d_x}{\partial d_{ML}}\right)\delta d_{ML} + \left(\frac{\partial d_x}{\partial d_{AP}}\right)\delta d_{AP} + \left(\frac{\partial d_x}{\partial d_{PD}}\right)\delta d_{PD} + \\
&\left(\frac{\partial d_x}{\partial \theta_{FE}}\right)\partial\theta_{FE} + \left(\frac{\partial d_x}{\partial \theta_{VV}}\right)\partial\theta_{VV} + \left(\frac{\partial d_x}{\partial \theta_{IE}}\right)\partial\theta_{IE}
\end{aligned}
\tag{6-30}
$$

When assembling them together, the six total differentials yield the form of the Jacobian matrix:

$$
\begin{bmatrix} {}^{T}\delta d_x \\ {}^{T}\delta d_y \\ {}^{T}\delta d_z \\ {}^{T}\delta\theta_x \\ {}^{T}\delta\theta_y \\ {}^{T}\delta\theta_z \end{bmatrix} = \begin{bmatrix}
\left(\frac{\partial d_x}{\partial d_{ML}}\right) & \left(\frac{\partial d_x}{\partial d_{AP}}\right) & \left(\frac{\partial d_x}{\partial d_{PD}}\right) & \left(\frac{\partial d_x}{\partial \theta_{FE}}\right) & \left(\frac{\partial d_x}{\partial \theta_{VV}}\right) & \left(\frac{\partial d_x}{\partial \theta_{IE}}\right) \\
\left(\frac{\partial d_y}{\partial d_{ML}}\right) & \left(\frac{\partial d_y}{\partial d_{AP}}\right) & \left(\frac{\partial d_y}{\partial d_{PD}}\right) & \left(\frac{\partial d_y}{\partial \theta_{FE}}\right) & \left(\frac{\partial d_y}{\partial \theta_{VV}}\right) & \left(\frac{\partial d_y}{\partial \theta_{IE}}\right) \\
\left(\frac{\partial d_z}{\partial d_{ML}}\right) & \left(\frac{\partial d_z}{\partial d_{AP}}\right) & \left(\frac{\partial d_z}{\partial d_{PD}}\right) & \left(\frac{\partial d_z}{\partial \theta_{FE}}\right) & \left(\frac{\partial d_z}{\partial \theta_{VV}}\right) & \left(\frac{\partial d_z}{\partial \theta_{IE}}\right) \\
\left(\frac{\partial \theta_x}{\partial d_{ML}}\right) & \left(\frac{\partial \theta_x}{\partial d_{AP}}\right) & \left(\frac{\partial \theta_x}{\partial d_{PD}}\right) & \left(\frac{\partial \theta_x}{\partial \theta_{FE}}\right) & \left(\frac{\partial \theta_x}{\partial \theta_{VV}}\right) & \left(\frac{\partial \theta_x}{\partial \theta_{IE}}\right) \\
\left(\frac{\partial \theta_y}{\partial d_{ML}}\right) & \left(\frac{\partial \theta_y}{\partial d_{AP}}\right) & \left(\frac{\partial \theta_y}{\partial d_{PD}}\right) & \left(\frac{\partial \theta_y}{\partial \theta_{FE}}\right) & \left(\frac{\partial \theta_y}{\partial \theta_{VV}}\right) & \left(\frac{\partial \theta_y}{\partial \theta_{IE}}\right) \\
\left(\frac{\partial \theta_z}{\partial d_{ML}}\right) & \left(\frac{\partial \theta_z}{\partial d_{AP}}\right) & \left(\frac{\partial \theta_z}{\partial d_{PD}}\right) & \left(\frac{\partial \theta_z}{\partial \theta_{FE}}\right) & \left(\frac{\partial \theta_z}{\partial \theta_{VV}}\right) & \left(\frac{\partial \theta_z}{\partial \theta_{IE}}\right)
\end{bmatrix} \begin{bmatrix} \delta d_{ML} \\ \delta d_{AP} \\ \delta d_{PD} \\ \delta\theta_{FE} \\ \delta\theta_{VV} \\ \delta\theta_{IE} \end{bmatrix}
\tag{6-31}
$$

Because Equation 6-31 is a statement of several total differentials, it makes sense to apply controlled increments along each of the 6 JMD directions and calculate their equivalent effect in the orthogonal coordinate system *T*. In fact, this is the approach used in the following calculations, with the increments treated in order of complexity. Starting at the tibia, the simplest is the PD increment δd_{PD}, which occurs along the negative y-axis of the L_6 coordinate system.

$$
{}^{L6}\delta R(0,0,0,0,\delta d_{PD},0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{PD} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\tag{6-32}
$$

Therefore, the increment itself is expressed as:

It is of interest what this increment in L_6 is equivalent to in the coordinate system *T*. Therefore, the mutual inverses of the transformation ${}^{T}_{L6}R$ are used:

$$ {}^{T}_{L6}R = {}^{L6}_{T}R = I \tag{6-33} $$

The actual transformation is given as follows:

$$ {}^{T}\delta R_{PD} = {}^{T}_{L6}R \cdot {}^{L6}\delta R(0,0,0,0,-\delta d_{PD},0) \cdot {}^{L6}_{T}R $$

$$ {}^{T}\delta R_{PD} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{PD} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-34} $$

The next controlled increment is of IE rotation, and because itoccurs along the y-axis of coordinate L_5.

$$ {}^{T}\delta R_{IE} = {}^{T}_{L5}R \cdot {}^{L5}\delta R(0,\delta\theta_{IE},0,0,0,0) \cdot {}^{L5}_{T}R $$

$$ {}^{T}\delta R_{IE} = {}^{T}_{L6}R \cdot {}^{L6}_{L5}R \cdot {}^{L5}\delta R(0,\delta\theta_{IE},0,0,0,0) \cdot {}^{L5}_{L6}R \cdot {}^{L6}_{T}R $$

$$ {}^{T}\delta R_{IE} = \begin{bmatrix} 0 & 0 & \delta\theta_{IE} & 0 \\ 0 & 0 & 0 & 0 \\ -\delta\theta_{IE} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-35} $$

Continuing with the remaining relations, all possible increments can be inferred from ${}^{L_i}\delta R$ (for the translation or rotation along and about a simple axis x, y or z) and ${}^{T}_{Li}R$ calculated in the section on kinematics. Once calculated, all the increments can be added to yield the necessary Jacobian matrix

Table 4 Increments of motion in the intermediate L systems and tibial coordinate systems

Motion	${}^{L_i}\delta R$	${}^{T}\delta R_{L_i}$
PD	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{PD} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{PD} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
IE	$\begin{bmatrix} 0 & 0 & \delta\theta_{IE} & 0 \\ 0 & 0 & 0 & 0 \\ -\delta\theta_{IE} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & \delta\theta_{IE} & 0 \\ 0 & 0 & 0 & 0 \\ -\delta\theta_{IE} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
AP	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{AP} \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \delta d_{AP} S_{IE} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\delta d_{AP} C_{IE} \\ 0 & 0 & 0 & 0 \end{bmatrix}$
VV	$\begin{bmatrix} 0 & -\delta\theta_{VV} & 0 & 0 \\ \delta\theta_{VV} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -\delta\theta_{VV} C_{IE} & 0 & \delta\theta_{VV} d_{PD} C_{IE} \\ \delta\theta_{VV} C_{IE} & 0 & \delta\theta_{VV} S_{IE} & 0 \\ 0 & -\delta\theta_{VV} S_{IE} & 0 & \delta\theta_{VV} d_{PD} S_{IE} \\ 0 & 0 & 0 & 0 \end{bmatrix}$
ML	$\begin{bmatrix} 0 & 0 & 0 & -\delta d_{AP} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & \delta d_{ML} C_{IE} C_{VV} \\ 0 & 0 & 0 & \delta d_{ML} S_{VV} \\ 0 & 0 & 0 & -\delta d_{ML} S_{IE} C_{VV} \\ 0 & 0 & 0 & 0 \end{bmatrix}$
FE	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\delta\theta_{FE} & 0 \\ 0 & \delta\theta_{FE} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -\delta\theta_{FE} S_{IE} C_{VV} & -\delta\theta_{FE} S_{VV} & \delta\theta_{FE}(d_{PD} S_{IE} C_{VV} + d_{AP} C_{IE} S_{VV}) \\ \delta\theta_{FE} S_{IE} C_{VV} & 0 & -\delta\theta_{FE} C_{IE} C_{VV} & \delta\theta_{FE} d_{AP} C_{VV} \\ \delta\theta_{FE} S_{VV} & \delta\theta_{FE} C_{IE} C_{VV} & 0 & \delta\theta_{FE}(d_{AP} S_{IE} S_{VV} - d_{PD} C_{IE} C_{VV}) \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Table 5 The sum of all the increments performed about the specially oriented L axes, broken down into rotational component (top) and translation component (bottom) yields the overall increment in the tibial system:

$$
{}^{T}\delta x =
$$

$$
\begin{bmatrix}
0 & -\delta\theta_{FE}S_{IE}C_{VV} - \delta\theta_{VV}C_{IE} & -\delta\theta_{FE}S_{VV} + \delta\theta_{IE} & 0 \\
\delta\theta_{FE}S_{IE}C_{VV} + \delta\theta_{VV}C_{IE} & 0 & -\delta\theta_{FE}C_{IE}C_{VV} + \delta\theta_{VV}S_{IE} & 0 \\
\delta\theta_{FE}S_{VV} - \delta\theta_{IE} & \delta\theta_{FE}C_{IE}C_{VV} - \delta\theta_{VV}S_{IE} & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} +
$$

$$
\begin{bmatrix}
0 & 0 & 0 & \delta d_{ML}C_{IE}C_{VV} + \delta d_{AP}S_{IE} + \delta\theta_{FE}(d_{PD}S_{IE}C_{VV} + d_{AP}C_{IE}S_{VV}) + \delta\theta_{VV}d_{PD}C_{IE} \\
0 & 0 & 0 & \delta d_{ML}S_{VV} + \delta d_{PD} + \delta\theta_{FE}d_{AP}C_{VV} \\
0 & 0 & 0 & -\delta d_{ML}S_{IE}C_{VV} - \delta d_{AP}C_{IE} + \delta\theta_{FE}(d_{AP}S_{IE}S_{VV} - d_{PD}C_{IE}C_{VV}) + \delta\theta_{VV}d_{PD}S_{IE} \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

By factoring the previous equations from matrix form to 6-DOF vector form, the Jacobian is readily inferred:

$$\begin{bmatrix} {}^{T}\delta d_x \\ {}^{T}\delta d_y \\ {}^{T}\delta d_z \\ {}^{T}\delta \theta_x \\ {}^{T}\delta \theta_y \\ {}^{T}\delta \theta_z \end{bmatrix} = \begin{bmatrix} -C_{IE}C_{VV} & S_{IE} & 0 & d_{PD}C_{VV}S_{IE} + d_{AP}C_{IE}S_{VV} & d_{PD}C_{IE} & 0 \\ S_{VV} & 0 & -1 & d_{AP}C_{VV} & 0 & 0 \\ -C_{VV}S_{IE} & -C_{IE} & 0 & -d_{PD}C_{IE}C_{VV} + d_{AP}S_{IE}S_{VV} & d_{PD}S_{IE} & 0 \\ 0 & 0 & 0 & C_{IE}C_{VV} & -S_{IE} & 0 \\ 0 & 0 & 0 & -S_{VV} & 0 & 1 \\ 0 & 0 & 0 & C_{VV}S_{IE} & C_{IE} & 0 \end{bmatrix} \begin{bmatrix} {}^{JMD}\delta d_{ML} \\ {}^{JMD}\delta d_{AP} \\ {}^{JMD}\delta d_{PD} \\ {}^{JMD}\delta \theta_{FE} \\ {}^{JMD}\delta \theta_{VV} \\ {}^{JMD}\delta \theta_{IE} \end{bmatrix} \tag{6-36}$$

Finally, the desired Jacobian:

$$\begin{bmatrix} {}^{JMD}F_{ML} \\ {}^{JMD}F_{AP} \\ {}^{JMD}F_{PD} \\ {}^{JMD}F_{FE} \\ {}^{JMD}F_{VV} \\ {}^{JMD}F_{IE} \end{bmatrix} = \begin{bmatrix} -C_{IE}C_{VV} & S_{VV} & -C_{VV}S_{IE} & 0 & 0 & 0 \\ S_{IE} & 0 & -C_{IE} & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ d_{PD}C_{VV}S_{IE}+d_{AP}C_{IE}S_{VV} & d_{AP}C_{VV} & -d_{PD}C_{IE}C_{VV}+d_{AP}S_{IE}S_{VV} & C_{IE}C_{VV} & -S_{VV} & C_{VV}S_{IE} \\ d_{PD}C_{IE} & 0 & d_{PD}S_{IE} & -S_{IE} & 0 & C_{IE} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{T}F_x \\ {}^{T}F_y \\ {}^{T}F_z \\ {}^{T}M_x \\ {}^{T}M_y \\ {}^{T}M_z \end{bmatrix} \tag{6-37}$$

6.3 DISCUSSION

This section details the transformations required to express all physically relevant data, kinematics and kinetics, in a single and consistent motion description dedicated to the human knee joint. As a result, the positions and orientation of the bones, as well as the forces and moments applied, can be readily obtained from the information recorded with the experimental equipment, namely, the robotic manipulator and the force-moment sensor, at any given time.

The kinematic data, i.e., the positions and orientations in 6-DOF, $(d_{ML}, d_{AP}, d_{PD}, \theta_{FE}, \theta_{VV}, \theta_{IE})$ and their applicable constraints, together with the kinetic data, $(F_{ML}, F_{AP}, F_{PD}, M_{FE}, M_{VV}, M_{IE})$, fully describe the boundary condition or loading

environment to the knee. Combined, they can be used to report the state of the knee joint mechanism at any given time in clinically relevant terms. The purpose of subsequent sections is to use this information in the reporting of biomechanical data and formulation of control algorithms to study the knee joint.

7.0 CONTROL ALGORITHMS

An important function of the robotic/UFS testing system is that of finding the position and orientation of the knee that occurs in response to an externally applied load and constraint. For example, the testing system could be used to study the response of the knee to a simulated anterior tibial drawer (combining an anterior tibial force with a flexion angle constraint). The results of this clinical diagnostic test could be gathered for intact, ligament deficient and surgically reconstructed knees to provide a quantitative basis to understand the changes in joint motion that follow simulated injury and operative repair.

The purpose of this section is to implement an algorithm that can handle flexion angle constraints and achieve force-moment control of the knee in the remaining degrees of freedom. The algorithm is an iterative, hybrid force-control algorithm, using an estimate of joint stiffness to minimize the objective function, and borrows from previously tested algorithms of the robotic/UFS testing system[38, 54, 58, 79].

In the previous sections, the kinematic and kinetic representations of the knee joint were detailed. The position and orientation of the knee $\mathbf{x} = (d_{ML}, d_{AP}, d_{PD}, \theta_{FE}, \theta_{VV}, \theta_{IE})$, its forces and moments $\mathbf{F} = (F_{ML}, F_{AP}, F_{PD}, M_{FE}, M_{VV}, M_{IE})$, and the joint stiffness $\mathbf{K}$, are assembled into a state $\{\mathbf{x}, \mathbf{F}, \mathbf{K}\}$ that is relevant to describing the control algorithm structure. One important innovation of the platform from the robotic/UFS testing system is that all such state variables are within a single and consistent motion description: the JMD.

Every iteration starts by collecting or *updating* the state $\{\mathbf{x}_i, \mathbf{F}_i, \mathbf{K}_i\}$ and checks whether it satisfies the *objective* function. As long as this criterion remains unmet, a new position is *predicted*, given the amount of deviation from the target force and moment. Such a position is often *adjusted* to prevent artificially large movements from being commanded to the robotic arm.

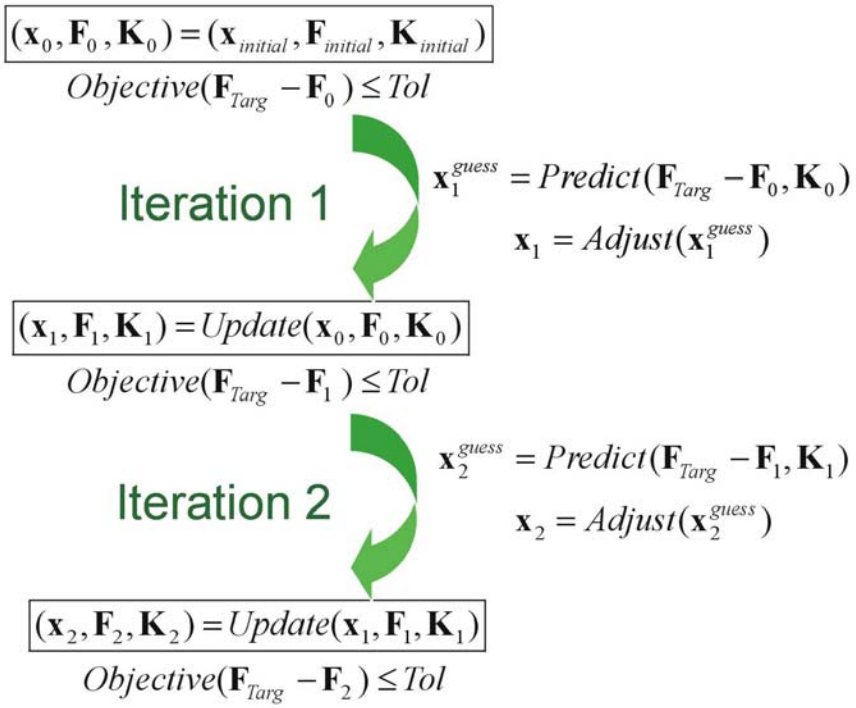


Figure 13 Flow-chart of iterative control algorithm. Boxed are the states of the knee as they converge towards the objective function.

7.1 THE ALGORITHM

7.1.1 Objective Function

The objective function is a scalar measure of the overall deviation of forces and moments away from the target. To non-dimensionalize forces and moments into a single scalar expression, a factor of 10 is used to account for the relative size of moment quantities relative to

$$Objective(\mathbf{F}_{Targ} - \mathbf{F}_{Current}) = \sqrt{\sum_{i=AllForces} \left(F_i^{Targ} - F_i^{Current}\right)^2 + 10 \cdot \sum_{j=AllMoments}^{ExceptFE} \left(M_j^{Targ} - M_j^{Current}\right)^2} \tag{7-1}$$

force quantities in experiments involving soft tissues, when such are expressed in N-m and N, respectively. Flexion extension moments are excluded because a flexion constraint precludes their being controlled in the algorithm.

7.1.2 Predicting Subsequent Positions and Orientations

Hooke's Law, governing the force-elongation relationship for springs, is used to predict the position and orientation of the knee that satisfies static equilibrium with its applied loads. It states that the change in applied force ΔF required for an elongation Δl of a spring is given by,

$$\Delta F = K \cdot \Delta l \tag{7-2}$$

where k is the stiffness constant. Assuming that one end of the spring is rigidly fixed and the other is at a coordinate x_i subject to a force F_i, Hooke's law can be used to predict the position that satisfies equilibrium with the desired uniaxial load F_{Targ}:

$$\begin{aligned} F_{Targ} - F_i &= K \cdot (x_{Targ} - x_i) \\ x_{i+1} &= K^{-1} \cdot (F_{Targ} - F_i) + x_i \end{aligned} \tag{7-3}$$

It is assumed that for a sufficiently small increment, this relationship holds for a constant stiffness value K. This relationship is used for all 6 DOFs in an uncoupled manner.

7.1.3 Adjusting Preliminary Calculated Movements

The force-displacement relation for each DOF of the knee is nonlinear. This means that the stiffness can vary, and it does so widely from one state to the next during the iteration process. Such variation is undesirable, particularly when the stiffness increases during a given iteration, because the extent of force-moment change resulting from the prescribed position and orientation increment can be underestimated. Chronic underestimations would cause the algorithm to repeatedly overshoot its translation or rotation along and about desirable directions of movement making convergence slow. Acute underestimation could result in damage to the specimen or testing equipment.

To prevent chronic underestimations of joint stiffness from insidiously throwing iteration convergence off-target, the increments of motion are halved any time the deviation from force-moment target changes sign. In other words, when the current force and moment goes from below target to above target (or vice-versa) a scale factor goes into effect to make the fraction of

the predicted increment to apply to the current iteration. On the other hand, to prevent acute underestimations of stiffness from producing motion increments that could destroy the setup, motion in all DOFs is capped to a value, typically of not more than 1 mm translation and 1 deg rotation about each of the JMD axes.

The scale factors are simple mechanisms to attenuate the initially predicted increment of motion as the iteration algorithm approaches equilibrium, i.e., they guide the process of iteration according to its history of convergence. For any given DOF (take a translational DOF consisting of a force and a translation, as an example) scale factors begin at 1 (100% of the calculated increment) at the outset, and are halved (50%, 25%, and so on) each time the algorithm detects that the direction of translation must be reversed to achieve convergence. The direction of translation must be reversed whenever the deviation from the target force changes sign (the algorithm went from having a force or moment smaller than desired to larger than desired, or vice-versa). This is similar to the procession of a bisection method during nonlinear root-finding.

$$F_i^{Dev} = F_i - F_{Targ}$$
$$if\,(F_i^{Dev} \cdot F_{i-1}^{Dev} < 0) \longrightarrow c_i = \frac{1}{2} c_{i-1} \tag{7-4}$$

As shown in the above expression, if the deviation from target changes sign, the scale factor for that DOF, c, is halved. The scale factor is then applied to the increment of motion that is first calculated from the prediction function (previous section).

$$x_{i+1}^{guess} = K^{-1} \cdot (F_{Targ} - F_i) + x_i$$
$$x_{i+1} = c_{i+1} \cdot (x_{i+1}^{guess} - x_i) + x_i \tag{7-5}$$

Scale factors are equally applicable to any DOF consisting of moments and rotations. The current approach is to use a separate scale factor for each DOF, although a scale factor applied to the effective helical axis motion (a single axis along which a translation and rotation take place, each requiring a scale factor) has also been pursued with limited success.

The increment of motion, once modified by the application of scale factors, is then capped to a limit of 1 mm and 1 deg in every JMD DOF to prevent large movements. The resulting increment is the actual increment of motion seen at the robot.

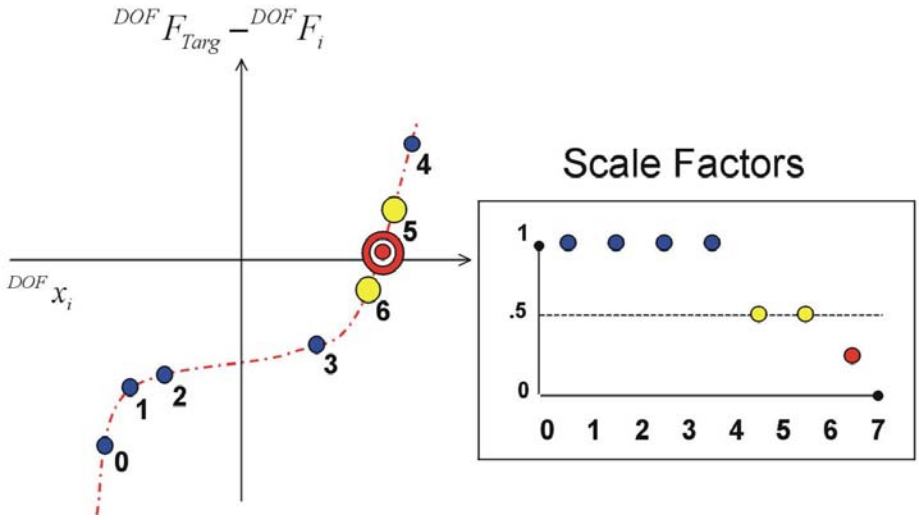


Figure 14 Progress of iterative control algorithm on the force-displacement curve for a given DOF. The initial 4 iterations (changing from state 0 to 1, 1 to 2, 2 to 3 and 3 to 4) occur with increasing value of x. At state 4, an overshoot is detected, causing displacements along the DOF to be reversed. At state 6, an additional overshoot occurs and the displacements are once again reversed. At state 7, the objective function is satisfied and iteration concludes. Scale factors are halved with each reversal

7.1.4 Updating the State

Updating the state consists in going from $(\mathbf{x}_i, \mathbf{F}_i, \mathbf{K}_i)$ to the subsequent $(\mathbf{x}_{i+1}, \mathbf{F}_{i+1}, \mathbf{K}_{i+1})$ once the increment of motion has been decided upon. Position and orientation are updated first by issuing the matching command to the robotic manipulator. Force and moment can be read as soon as the manipulator reaches its destination. Finally, stiffness is updated using the following relation:

$$K_{i+1} = \frac{1}{4} K_i + \frac{3}{4} \frac{\cdot (F_{i+1} - F_i)}{x_{i+1} - x_i} \tag{7-6}$$

7.2 APPLICATIONS

The control algorithm produces the sequence of position and orientation iterations necessary to converge the force and moments resulting on the specimen to desired targets. These equilibrium iterations can be composed into sequences that simulate that yield knee configurations that simulate diagnostic exams. Two applications are exemplified: finding the path of passive flexion and extension, and applying single DOF loads at a fixed flexion angle.

7.2.1 Path of Passive Flexion and Extension

The path of passive flexion and extension consists in the locus of positions at which the knee is under little to no applied extern loads, at every possible flexion angle. To record this path, the equilibrium algorithm is used with a zero-load target in the ML, AP, and VV DOFs. In

the PD direction, a force target in the order of 5-10 N is applied, to provide a small compressive force that maintain joint contact throughout the iteration process. The other two DOFs, FE and IE operate under position target. The FE angle is determined to be the current flexion angle for iteration, and is the independent variable for the path of passive flexion and extension, sampled at intervals of 1°. Because the envelope of joint laxity for IE rotations can span several degrees at any flexion angle (angles that produce moments below a threshold ~0.5 N-m), IE rotation could defer by several degrees from one flexion angle to the next if the DOF were allowed to operate under load control. Therefore, the IE orientation is selected to be the mid-point of the laxity envelope for any flexion angle. One favored approach is to calculate the IE envelope for flexion angles that are sparsely distributed along the passive path (every 15 degrees) in a preliminary "rough path" and selecting IE angles that are interpolated as the final IE rotation for a second "fine" path of passive flexion and extension

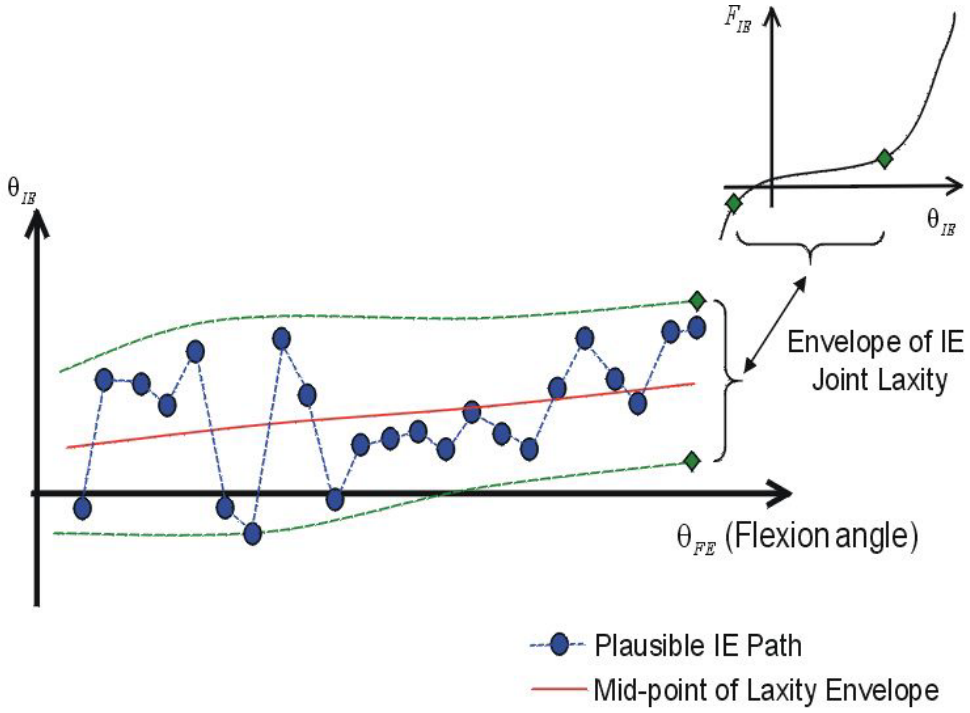


Figure 15 Selecting IE rotations cannot be done with IE operating under a moment target. Otherwise visibly noisy paths of IE rotation vs flexion angle could be found that satisfy the moment target (within the laxity envelope). It is preferable to measure the laxity envelope at selected intervals and interpolate the IE rotations.

7.2.2 Loading Conditions

A simple application is the anterior tibial drawer test, or a varus valgus test. The loading algorithm is implemented with zero-load targets in all DOF, except FE, which is controlled by position (current flexion angle), and the DOF of interest, AP or VV respectively, along or about which a force or moment is applied.

8.0 SOFTWARE ARCHITECTURE AND DEVELOPMENT PROCESS

The most significant aspect of the modernization of the robotic/UFS testing system is the upgrade of its software control. An obvious disadvantage with the legacy system was the anchoring of its control algorithms to the development technology of the robot manufacturer, namely the UNIMATE VAL language. Such a development language is neither commonly accessible nor well known. It interfaces poorly with third party software solutions rendering the development effort for custom applications, such as that of the robotic/UFS testing system, effort intensive. Furthermore, replacing the robotic arm with that of a different manufacturer incurs the cost of a software redevelopment.

The biomechanical testing platform is grounded on the C++ language for core software development, used for the development of static-link libraries and COM components. Developers in C++ are much more ubiquitous, as are third-party software solutions written in this language. This would help in the recruiting of users and developers for the platform, and guarantee that the new software control is more maintainable. Ultimately, the technology selection will help in establishing the interoperability of the robotic/UFS testing system with computational analyses under the umbrella of a biomechanical testing platform.

8.1 TECHNOLOGIES

The C++ programming language is selected because it lends itself well to the representation of data and their algorithms in a modular form, specifically in the form of objects that have a well-defined state, interface and persistence[72]. For example, the mathematical operations, motion descriptions and control algorithms can be represented by modules (called classes) each formulated encapsulating its own expertise. If well designed, the modules can be reused in a variety of contexts and tested automatically and independently[59].

Writing software in C++, as in other object-oriented languages, involves the design of the individual classes, each aimed at solving a logical subsection of the problem. Finally, an overarching client is formulated to put together or orchestrate the classes into the final application. To achieve different effects, the client can reuse the same classes, while orchestrating them differently.

Classes are very sophisticated building blocks for software reuse, and their design and composition affects the ways they can be assembled into meaningful applications[80]. In our case, many classes were designed in house, because of their level of specialization, although many were also recruited from libraries to reduce the software development effort, for example by reusing components for hardware interfacing, GUI development[81] and the management of standard containers and operations[74].

The C++ classes were packaged into two binary forms: static-link libraries and COM components. Static link libraries are simple mechanisms for reuse of commonly used technology such as mathematical operations within the C++ language[73]. COM components are compiled versions of the classes that comply with a binary standard to make them interoperable with other

programming languages, as well, within the Windows operating system. This opens the possibility of making the software accessible from languages such as VisualBasic, Java, etc, that are better suited for the creation of a GUI. The software could also be accessible from scientific software such as LabView or Matlab. Packaging classes into COM components was achieved using the Active Template Library (ATL)[82, 83].

For the current version of the platform, a GUI that leverages the Microsoft Foundation Classes (MFC) has been adopted[81, 84]. The current GUI model uses the Document/View architecture and multiple threads to separate a worker thread from the GUI thread so that data can stream to the diagnostic windows while the biomechanical testing platform interacts with the various components and hardware.

Automatic verification of the software comprising the biomechanical testing platform is accomplished using a C++ port of the JUnit framework for unit testing[59].

Finally, OpenGL is also used to provide visualization of the various coordinate systems and eventually specimen or computational model geometry[85].

8.2 ARCHITECTURE

Architecture consists of the set of rules that are selected by the developer to govern interrelations between data and algorithms throughout the software. It is a level of design concerned with issues beyond the individual algorithms and data structures, and emerges in response to the problem of specifying overall system structure[86, 87]. In object-oriented languages such as C++, architecture is the result of settling questions like “Should class A be a part of class hierarchy H or J?” or “Should class B have knowledge of class C?” or “Does class E belong in

module X or Y?" or "Should algorithm M be applied to an instance of class F or be its method?" Resolving such issues greatly affects the ability of the system to accommodate choice[88, 89].

To the platform's users, choice represents the freedom to select among different types of position control device (robot or otherwise) and force-moment sensor manufacturer, to select the effective clinical motion description, the type of iterative algorithm for force-moment convergence, etc. The user must be able to select not only from existing choices but, more importantly, among those of his/her making. In other words, the platform must accommodate user contributions and customizations.

Any software can be customized or reused; the question is the expense. When a user undertakes customization of a piece of software he/she is faced with acquiring detailed knowledge of its design. The amount of knowledge required is often called the "conceptual weight". If the conceptual weight of the software structures is too high (e.g. the user must learn the contents and interrelations between too many classes and algorithms to proceed with confidence), customizations run the risk of exceeding estimated costs or introducing substantial errors. Under those circumstances, the user will eventually question the maintainability of the software and consider replacement[90].

To the platform's developer, choice is furnished by strategically parameterizing the software throughout the architectural design. For example, to furnish the choice of robot, the developer must determine what is characteristic about robot function and, in particular, distinguish what is common to all possible robots in relation to the software (declaring the 'robot' class) vs. what is peculiar to each (defining a given instance of such a class). In doing so, the software can be made to operate correctly as a whole in response to any valid selection.

Choice requires significant effort in mapping the logical relationships within the problem/solution that is being modeled, to the software data structures and algorithms that model it. It is the hallmark of an architecture that naturally represents the problem/solution at hand. Architectures that accommodate choice are the result of a drafting process throughout which the developer gains the experience necessary to distinguish parameters that are important, their scope and interactions; this is often not obvious at the outset. Therefore, the architecture can reveal the philosophy and state-of-the-art thinking in modeling a solution to a given problem. It can also reveal the true value of the software when expressed in terms of cost of reusability and maintenance.

The following sections will detail architectural designs used to support specific choices in the software.

8.2.1 Choice of Robot

The choice of robot is supported by polymorphism[89]. In other words, it depends on design patterns that leverage the concept of class hierarchies[80]. A class hierarchy is an arrangement in which a group of classes are said to derive or inherit from a more general class, often called a base or parent.

Class hierarchies are established when it becomes clear that a variety of objects share commonality. Take the current example in which we are faced with formulating software to control communication between the platform and a variety of robot types (ABB, FANUC, KUKA, etc). Suppose we decide to dedicate a class to manage communications with each robot type (ABBInterface, FANUCInterface, KUKAInterface). We would soon determine that there

are methods that repeat themselves in their signature and content across the classes. We would always require a method to move the robot end-effector to a given position and orientation, for example. This method would have a simple signature: the input of a position and orientation, and the output of a code indicating success or failure of the operation. Because the method is common to all robots, it should be factored into a general base class to represent robot communications (for example, called RobotInterface) from which all the manufacturer specific kinds can derive.

Grouping classes into hierarchies can potentially achieve two things: implementation inheritance and interface inheritance[89]. Implementation inheritance occurs when methods are factored into a base class to make them accessible to derived classes. In such case, the derived classes can reuse algorithms directly from the base class without having to duplicate them. Although this is often valuable, algorithm reuse rarely motivates inheritance in the biomechanical testing platform's design. The primary motivation is the testing platform is class polymorphism; interface inheritance is really what we are after. With interface inheritance, we factor the algorithm signatures (inputs and outputs)

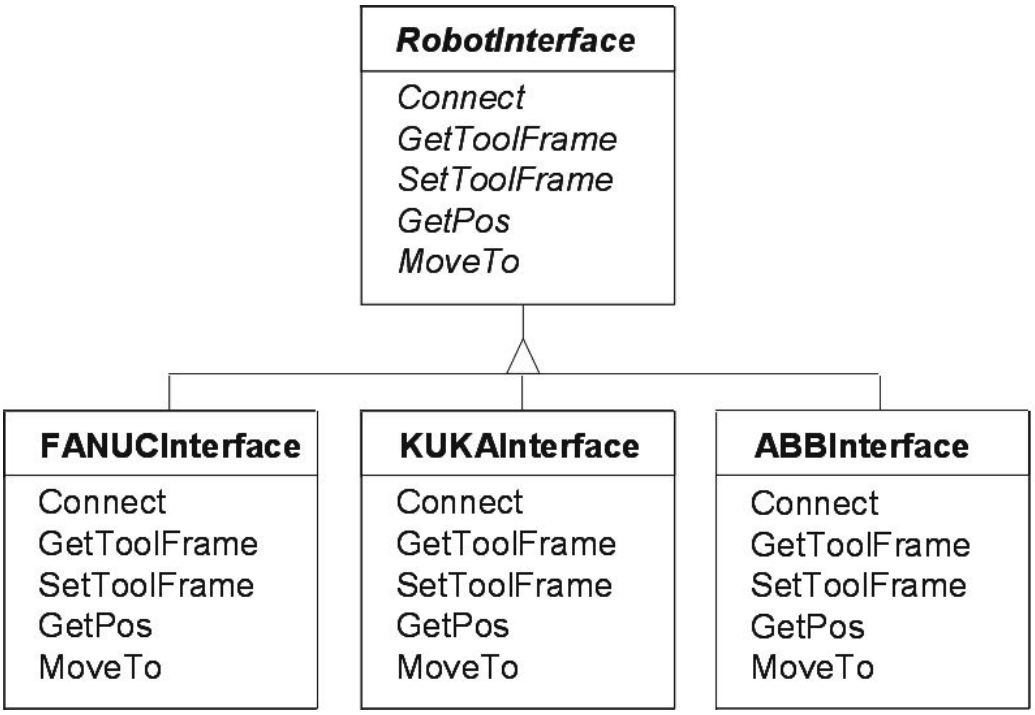


Figure 16 Robot Interface class hierarchy represented in Object Modeling Technique (OMT). The abstract base class RobotInterface is the parametric class that is substituted by one of the concrete derived classes in the diagram

The RobotInterface is an example of polymorphism used to furnish robot choice. Any concrete object that derives from this interface is guaranteed to interact correctly with the rest of the software in the biomechanical testing platform as a valid robot type. The concrete classes displayed before, ABBInterface, FANUCInterface and KUKAInterface contain software that connects to some COM component that performs the actual hardware communications. RobotInterface does not perform any tracking or coordinate system book keeping. That task is assigned to the parts of the software that operate the MPS. This separation between managing hardware communications and motion tracking is what makes it effortless to implement the code required to support interoperability with the robot from a new manufacturer.

The interface and implementation reside within the core program. The implementation should eventually form its own library.

There are other example of class hierarchies that supplement the provision of robot choice. Different robot manufacturers often have specialized Euler angle sequences or Denavit-Hartenberg parameters to specify or query motion from the hardware. The interfaces of BaseXYZWPR and BaseJointPos are intended to parameterize this possibility. Use of such hardware-specific is rare in the code (only for the purpose of recording already measured paths). The RobotInterface itself is formulated around a more general variable type, named the MatrixRot, which fulfills the purpose of encoding the data and operations of orthogonal coordinate system. MatrixRot is, in fact, the common currency throughout the software for exchanging position and orientation data between components without regard to robot manufacturer. It constitutes the logical link between the RobotInterface and the MPS.

The classes are provided within the static-link libraries for mathematical operations.

8.2.2 Choice of Force Moment Sensor

This choice is achieved with the use of an abstract base class to represent interactions with the sensor. As in the case of the RobotInterface, the FMSenorInterface is implemented, preferably by instantiating a COM component that performs the hardware communications with the sensor. In the case of the force-moment sensor utilized in the experiments, the manufacturer already supplied such a component.

Functions of interest are to query the forces and moments in counts (raw units of the sensor), and in a standard unit system. The unit system in the platform corresponds to N and N-

mm, so the interface must hide the details of providing such a converted magnitude for forces and moments.

The interface and implementation reside within the core program. The implementation should eventually form its own library.

8.2.3 Choice of Joint Motion Description

For the moment, the Joint Motion Description can be replaced by any arrangement of axes that results in three translations and three rotations labeled as ML, AP, PD, FE, VV, IE. Two interfaces, BaseJMD and BaseJMDFM assume this order. The translation components are assumed in mm while the rotations are assumed in degrees.

The classes are provided within the static-link libraries for mathematical operations.

8.2.4 Choice of Iterative Algorithm

The choice of iterative algorithm is provided by another parametric interface called EquilibriumIterationData. It is currently limited in applicability, because the functions that must be common to all algorithms have not matured to independence. With the use of a template method or a bridge design pattern[80], this could be achieved. However, EquilibriumIterationData uses the interfaces of BaseJMD and BaseJMDFM to establish the control for the biomechanical testing platform.

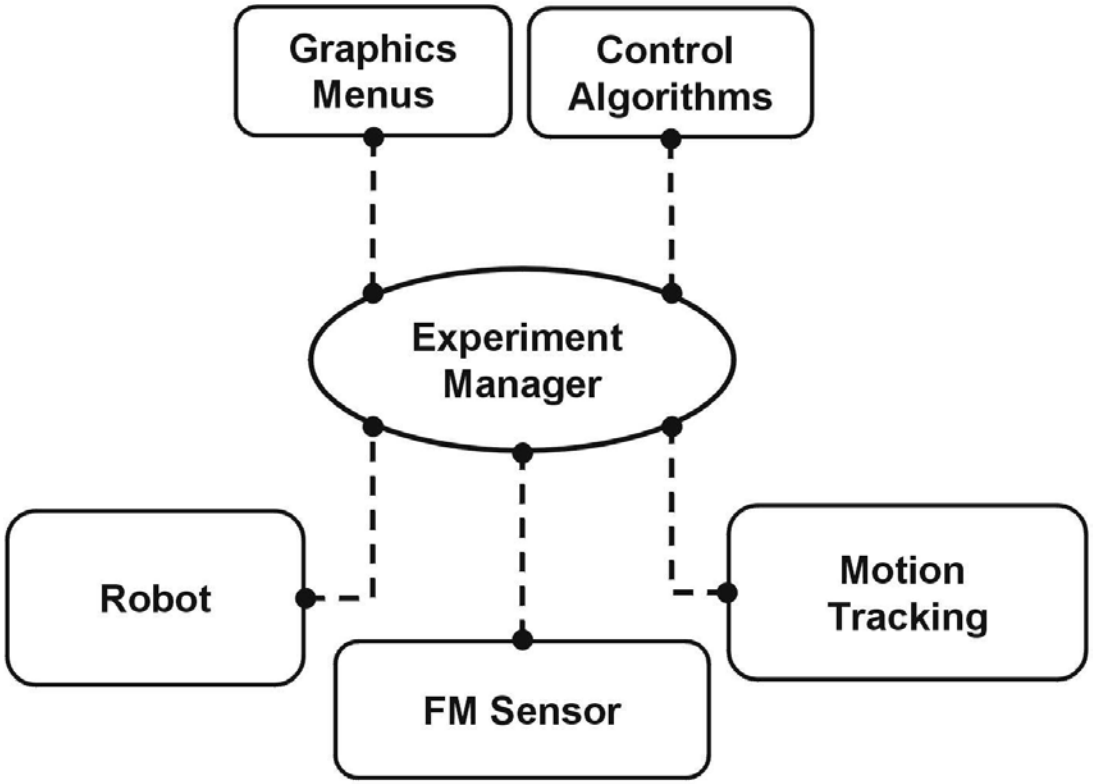


Figure 17 Architecture of the biomechanical testing platform. The experiment manager, at the core, integrates the behavior of the 5 components displayed. Each of these components has no direct knowledge of the others and can be independently replaced, furnishing functional choice

8.3 MODULES

The overall architectural design of the testing platform consists of a central component called an ExperimentManager. The ExperimentManager is knowledgeable of all the components and has the role of client that orchestrates overall system functionality. The components, namely the GUI, control algorithms, Robot, FM Sensor and MPS have no direct knowledge of each

other. They interact by the ExperimentManager intermediate. In this way, any of the components can be replaced with little or no effort, providing versatility in the platform.

The physical layout of the software includes an MFC Document/View executable that resides in a Microsoft Visual C++ 6 project called Biomech. This project contains all the meaningful interfaces for class parameterizations. Undesirably, the implementations of the components for the robot, force-moment sensor and control algorithms reside in this project. These should be removed to projects of their own. However, their interfaces are fairly distinct and their interdependencies are minimal, which will guarantee that such an effort will be minor. The MPS and the basic mathematical operations, including those implemented for derived classes of BaseXYZWPR, BaseJointPos, BaseJMD and BaseJMDFM, together with the MatrixRot and all supporting classes are in their own static-link library projects (BiomechMathLib and BiomechGPS). Of all the software, the amount packaged in static link libraries has received the best automatic test coverage.

9.0 EXPERIMENTAL STUDY

The following two experiments demonstrate the functionality of the biomechanical testing platform and its applicability to the study of the knee joint. The first experiment consists in measuring the *in situ* forces occurring in the ACL in response to translational (AP) and rotatory (VV) loads of 100 N and 3 N-m, respectively. This experiment features the platform operating in combinations of force and position control following the paradigm of the robotic/UFS testing system to measure forces in the component structures of the knee. The second experiment uses the biomechanical testing platform to control different robot types to record the kinematics of the knee in response to a simple loading condition, thereby proving its versatility to interoperate with hardware choices of interest.

9.1 PORCINE MODEL

The studies were performed using cadaveric porcine knee specimens obtained from two local farming companies, where the animals were grown and sacrificed. To harvest the knees, the hind limbs were disarticulated at the hip and heel, and the musculature was removed being careful to leave all ligaments intact. The joints consisted of the exposed femur, tibia and fibula,

surrounded by the capsule. The specimens were preserved in two-ply airtight plastic bags at a temperature of –20°C.

Porcine specimens have the advantage of being relatively uniform in age and size, and can be obtained in large quantities from local suppliers. The porcine is an established animal model for the study of ACL biomechanics. Although the range of flexion of the porcine knee is different (full extension occurs near 30°), the size of the knee joint is comparable to that of humans, and its ACL can be readily identified.

9.2 SPECIMEN MOUNTING

Specimens were thawed at room temperature 24 hours prior to testing. The bones were cut to a length of 20 cm from the joint-line and were potted into a cylindrical shape using an epoxy compound. The fibula was secured to the tibia at its anatomic position using two cortical screws.

The femur was rigidly fixed relative to the base of the robotic manipulator, while the tibia was attached to the end-effector through the UFS. A custom-built pedestal and clamp were used respectively. Under this configuration, the UFS can measure the forces in the joint and enable the robotic manipulator to apply specific loading conditions and record the resulting movements.

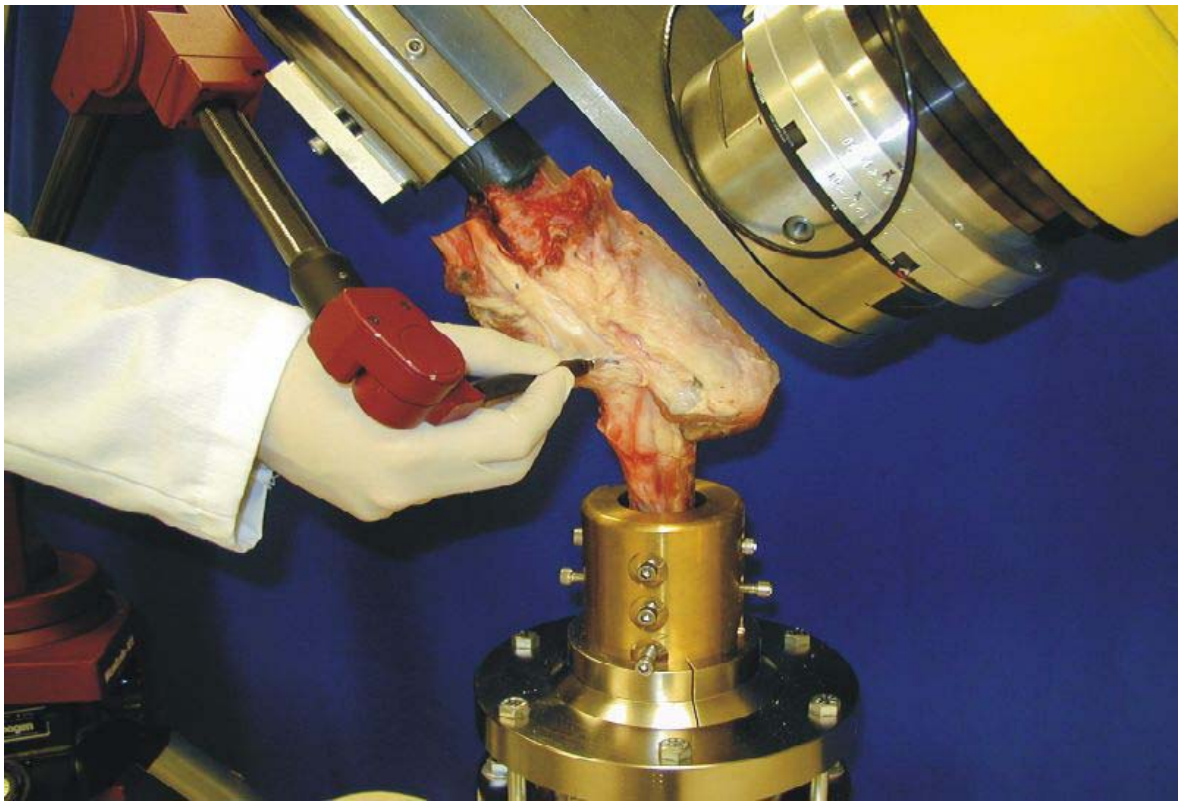


Figure 18 The tibia is attached to the sensor via the tibial clamp while the femur is fixed to the ground via the femoral pedestal (emerging from the bottom of the picture). Spatial digitization is performed with a Microscribe system

9.3 INITIALIZATIONS

9.3.1 Coordinate Systems

In order to record the kinematics of the knee with the testing system, four Cartesian coordinate systems are used in the MPS. Naturally, two of them are the tibia (T) and femur (F) bone systems and are used to track the rigid-body motion of the joint; their initializations will be

specified later. A sensor (S) system is used to track the motion of the UFS and is coincident with the inherent coordinate system used by the sensor for reporting forces and moments. Finally, a global (U) coordinate system fixed to the laboratory floor is selected as a common frame of reference. These 4 coordinate systems are named in accordance with Appendix A.

To fully establish interrelations between 4 coordinate systems, 3 relations are necessary, of which 2 are recognized as constants. Because the femur is rigidly attached to the floor via the femoral pedestal, ${}^{U}_{F}R$ is constant. Similarly, the tibia is rigidly attached to the UFS via the tibial clamp, so ${}^{S}_{T}R$ is constant as well. The variable relation ${}^{G}_{S}R$ is the position of the sensor relative to global and is directly controlled with the robot.

The two constant transformations involve the position and orientation of the bones in the testing system and are established by measuring the initial configuration of the knee relative to the sensor, ${}^{S}_{F}R_{o}$ and ${}^{S}_{T}R$, from a series of spatial locations. These, together with the transformation representing the initial configuration of the robot, ${}^{U}_{S}R$ allow the MPS to become fully specified:

Once the bones are installed into their rigid fixtures, and with the robot motionless, a set of 11 landmarks is selected and digitized on the surface of each bone. A number of predefined points on the surface of the sensor are digitized along with them. Together, these points are processed to yield ${}^{S}_{F}R_{o}$ and ${}^{S}_{T}R$. A serial linkage system with position accuracy of 0.1 mm for translations and 0.1° for rotations (Microscribe 3DX, Immersion Corporation) is used in this process.

The 11 landmarks on each bone are subdivided into two groups (for detailed locations, see appendix). The first group consists of 8 points used to infer the axis of the bone diaphysis.

The remaining three points, located medial, lateral and anterior, are used to establish an epiphyseal plane parallel to the tibio-femoral articular contact. The point where the diaphyseal axis and epiphyseal plane intersect is designated as the origin of the bone coordinate system. As far as the general orientation of the coordinate systems, the x-axis points left, the y-axis points distally and the z-axis posteriorly; this much is common to the coordinate systems of the femur and tibia. The way in which these coordinate system measurements differ is in the detailed inference of the axes from the digitized points.

For the femur, two of the landmarks used to establish the epiphyseal plane are digitized on the condyles near the respective insertions of the collateral ligaments. The axis running exactly through these two points is designated as the x-axis of the femur (F) coordinate system. The y and z-axes are selected to complete the right-handed orthogonal coordinate system, with the y-axis running as closely as possible to the diaphyseal axis. For the tibia, on the other hand, the priority is for the y-axis to run exactly through the diaphyseal axis, with the x-axis running as closely as possible through the medial-lateral landmarks.

The rationale for these choices is that the JMD was designed in previous sections so that the x-axis of the femur system corresponds to the axis of flexion-extension (FE) and medial lateral (ML) translation. Similarly the y-axis of the tibial system corresponds to the axis of internal-external (IE) rotation and proximal-distal translation (PD). With this arrangement, the JMD configuration of the knee can be easily inferred from the observation of digitized bony landmarks.

9.3.2 Path of passive flexion and extension

Once the specimen is mounted, the robotic/UFS testing system is used to record the path of passive flexion and extension at 1° increments between 35° and 90°. Because the resulting forces and moments in the joint are minimized throughout this path, the knee is effectively unloaded. Therefore, each position along this path serves as a reference from which loading conditions are prescribed.

9.3.3 External Loads

Two external loads are applied to each specimen. In order to demonstrate the functioning of the ACL a 100 N A-P load was applied to the tibia. This loading condition has been previously studied with the robotic/UFS testing system. It simulates clinical examinations in which the tibia is drawn anteriorly; an ACL that is functioning normally would restrain the knee from excessive motion. The second loading condition consisted of a 3 N-m V-V load, and is intended to demonstrate the ability of the high-payload system to apply moments, and record the corresponding kinematics of the knee.

Both loading conditions were applied to the knee at two flexion angles: 35° and 90°. As was mentioned earlier, 35° for a porcine specimen corresponds roughly with full extension in a human knee.

The kinematics of the intact knee are measured in response to both loading conditions at 35° and 90° of flexion. The kinematics of the ACL-deficient knee were also measured at 35° of flexion and compared to those of the intact knee. This comparison demonstrates the ability of

the robotic/UFS testing systems to resolve differences in the kinematics of a specimen that has been subject to an experimental treatment.

The *in situ* forces in the ACL specimen are measured in response to both loading conditions, 100 N A-P and 3 N-mm V-V, at both flexion angles studied.

10.0 RESULTS

10.1 ANTERIOR LOAD

In response to a 100 N anterior tibial load at 35° of flexion, AP translation reached 6.3 ± 1.4 mm in the intact knee. As for coupled translations, ML reached –2.1 ± 0.8 mm and PD reached 1.8 ± 0.6 mm. Coupled rotations were –5.5 ± 4.1° and 1.8 ± 1.4 about the IE and VV axes, respectively.

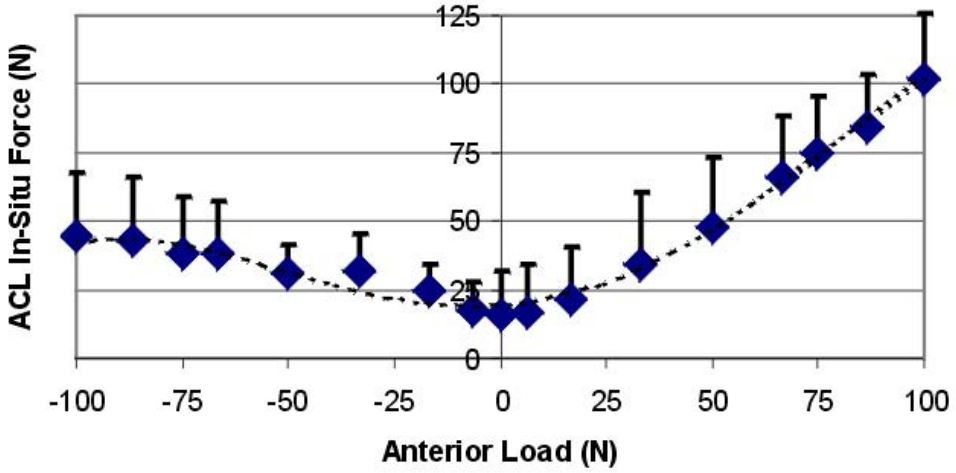


Figure 19 ACL *in situ* force in response to AP loads up to 100 N

After transecting the ACL, AP translation increased to 19.2 ± 0.4 mm. As for coupled translations, 4.5 ± 2.8 mm and 5.7 ± 1.1 mm were recorded in the ML and PD directions, respectively. IE rotations increased to 13.0 ± 4.2°, whereas VV rotations reached 5.4 ± 2.9°. The increased in AP translation and IE rotations were statistically significant for a paired sample t-test ($p < 0.05$).

At 90° of flexion, AP translation reached 5.0 ± 1.4 mm. Coupled translations were close to zero, at 0.2 ± 0.6 mm and 1.3 ± 0.6 mm in the ML and PD directions, respectively. Coupled rotations were also found to be near zero, at 2.6 ± 2.7° and –0.5 ± 1.7°.

The ACL in-situ force was measured to increase monotonically from 16.3 ± 15.8 N to 101 ± 23.8 N as the applied load ramped to 100 N anterior. At a fully reversed posterior load of 100 N, the *in situ* force was measured at 44.3 ± 23 N. In-situ forces in this study were approximated using the JMD components of force (see Appendix D).

Anterior tibial loads were also applied using a KUKA robotic manipulator. The resulting displacements were measured at 50° of flexion on a single specimen to reach 4.2 mm at 100 N. Displacements are shown for comparison. Data for the Unimate PUMA 762 robot was collected using its legacy robotic/UFS testing system control software, whereas data for the FANUC S900 W and KUKA 2000 Series-210 were obtained using the control software for the biomechanical testing platform. A notable fact involved the development time for the KUKA robot. Having once accomplished operation of the platform on the FANUC S900-W robot, adaptations on the software to integrate the KUKA hardware were accomplished in a few days, with a mixture of development done locally (Pittsburgh, PA) and on-site, at a remote facility (Detroit, MI). This is in contrast to the development of the UNIMATE robot, which involved an effort spanning several years.

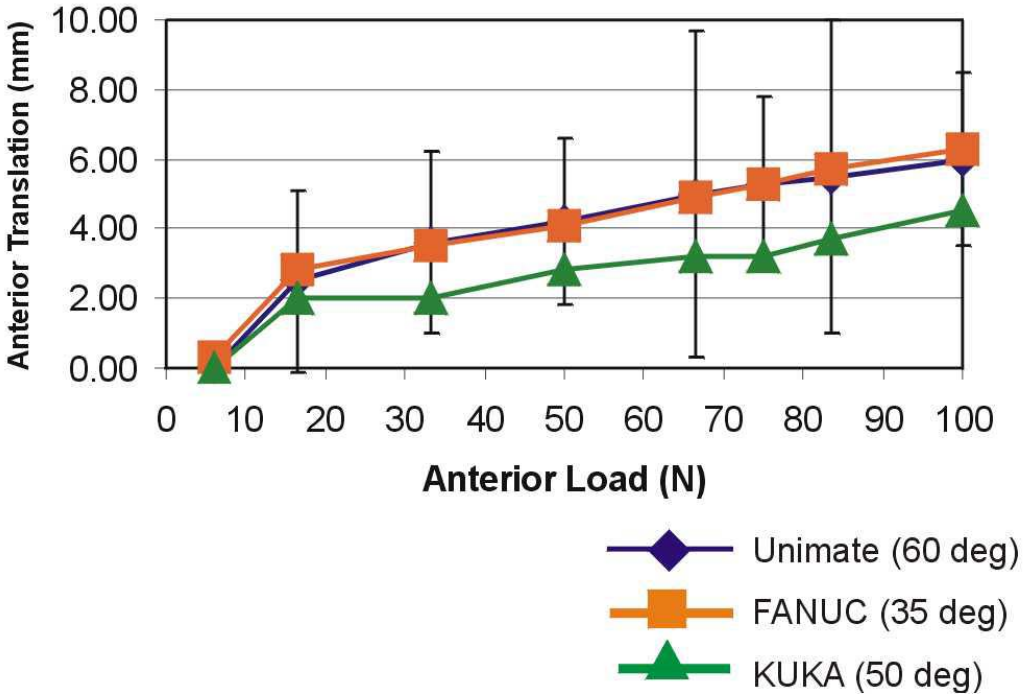


Figure 20 Anterior tibial translation in response to anterior tibial loads up to 100 N for 3 robot types. Data obtained with the Unimate robot is displayed (with error bar) for comparison. The FANUC and KUKA robots were controlled using the biomechanical testing platform software.

10.2 VARUS-VALGUS LOAD

In response to a 3 N-m varus tibial load at 35° of flexion, VV rotation reached 5.2 ± 2.9° in the intact knee. Coupled translations were close to zero at -2.7 ± 1.9 mm, 2.5 ± 2.0 mm and 2.0 ± 0.3 mm in the ML, AP and PD directions, respectively. Coupled IE rotations were –3.5 ± 2.7°.

After transecting the ACL, VV rotation increased to 10.3 ± 2.6°. Coupled translations were –3.5 ± 1.6 mm, 8.6 ± 4.0 mm and 1.1 ± 1.9 mm in the ML, AP and PD directions

respectively. IE rotations remained close to zero at –2.8 ± 2.1°. The increased in VV rotation were statistically significant for a paired sample t-test ($p < 0.05$).

At 90° of flexion, VV rotation reached 4.4 ± 2.6°. Coupled translations were close to zero, at -2.0 ± 1.0 mm, 1.5 ± 2.0 mm and –0.3 ± 0.8 mm, in the ML, AP and PD directions, respectively. Coupled IE rotations were also near zero, at –0.5 ± 1.7°.

The ACL in-situ force was measured to increase monotonically from 10.4 ± 4.8 N to 26.6 ± 6.6 N as the applied load ramped to 3 N-m anterior. When the load was fully reversed to a 3 N-m valgus moment, ACL *in situ* force reached 14 ± 5 N

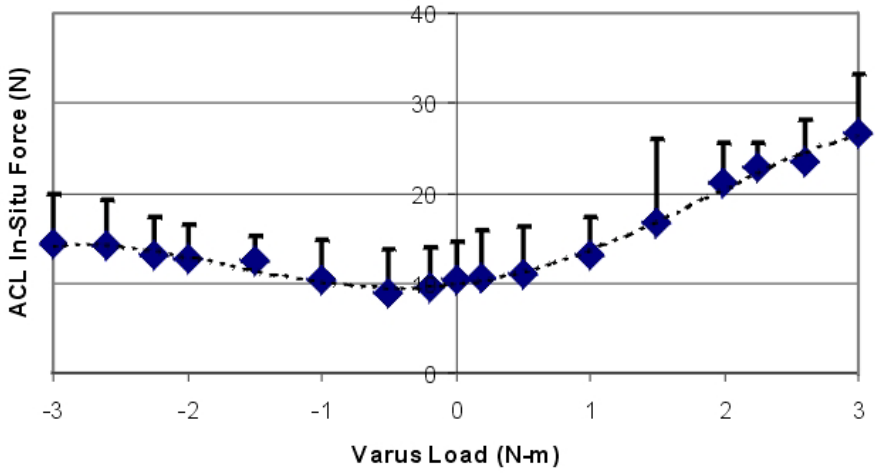


Figure 21 ACL *in situ* force in response to VV loads up to 3 N-m

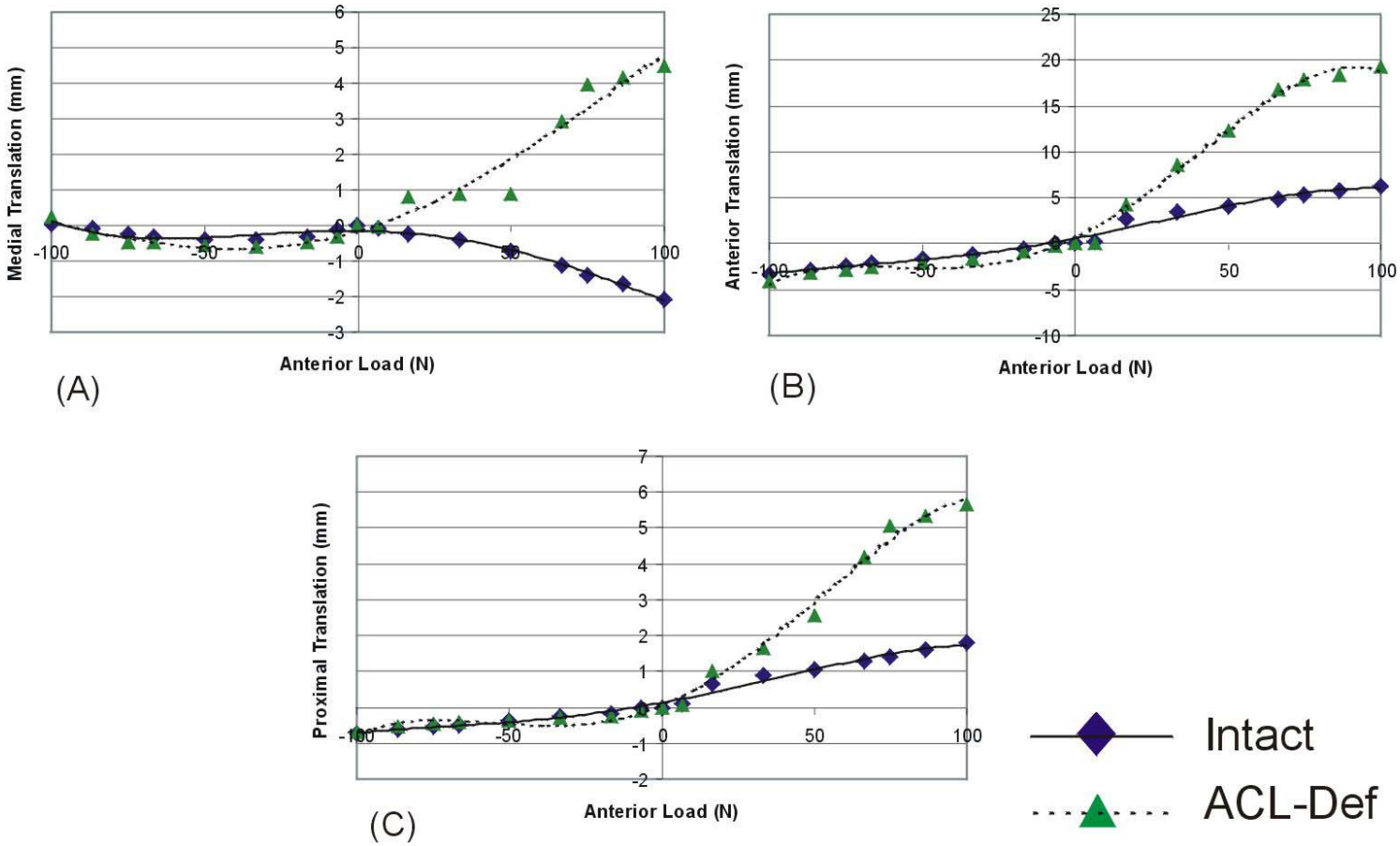


Figure 22 Translation in the medial (A), anterior (B) and posterior (C) directions in response to AP loads up to 100 N

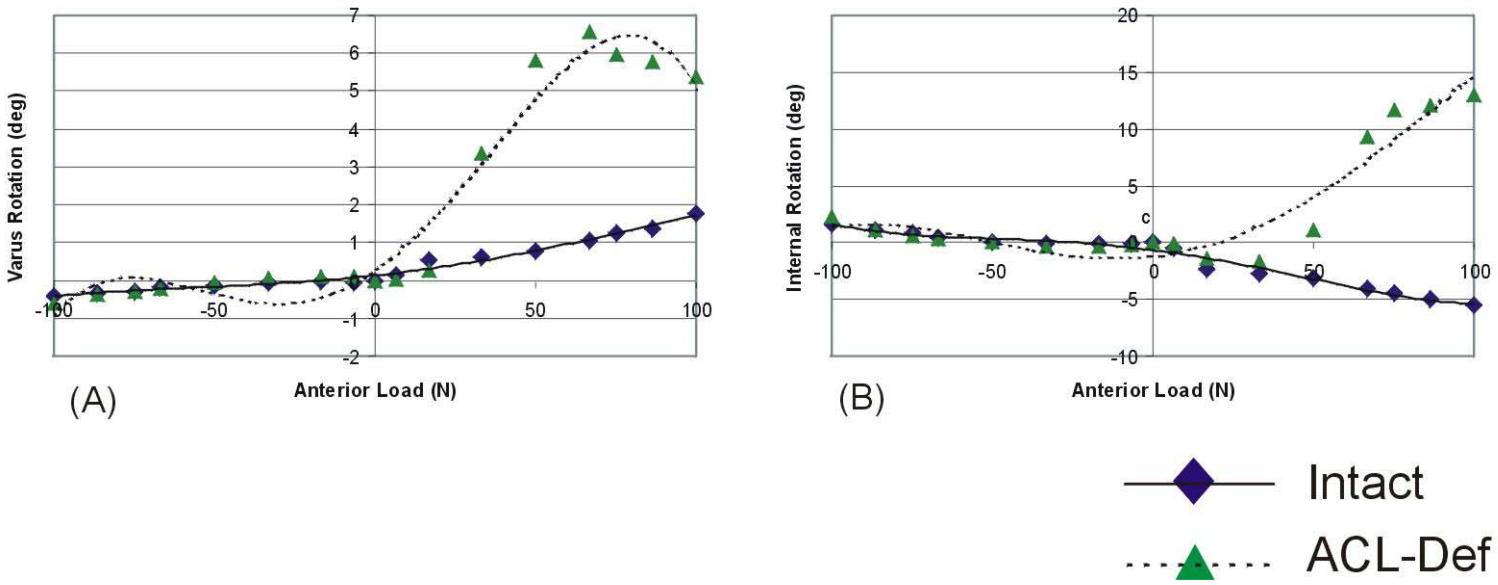


Figure 23 Rotation about the Varus (A), Internal (B) axes in response to AP loads up to 100 N

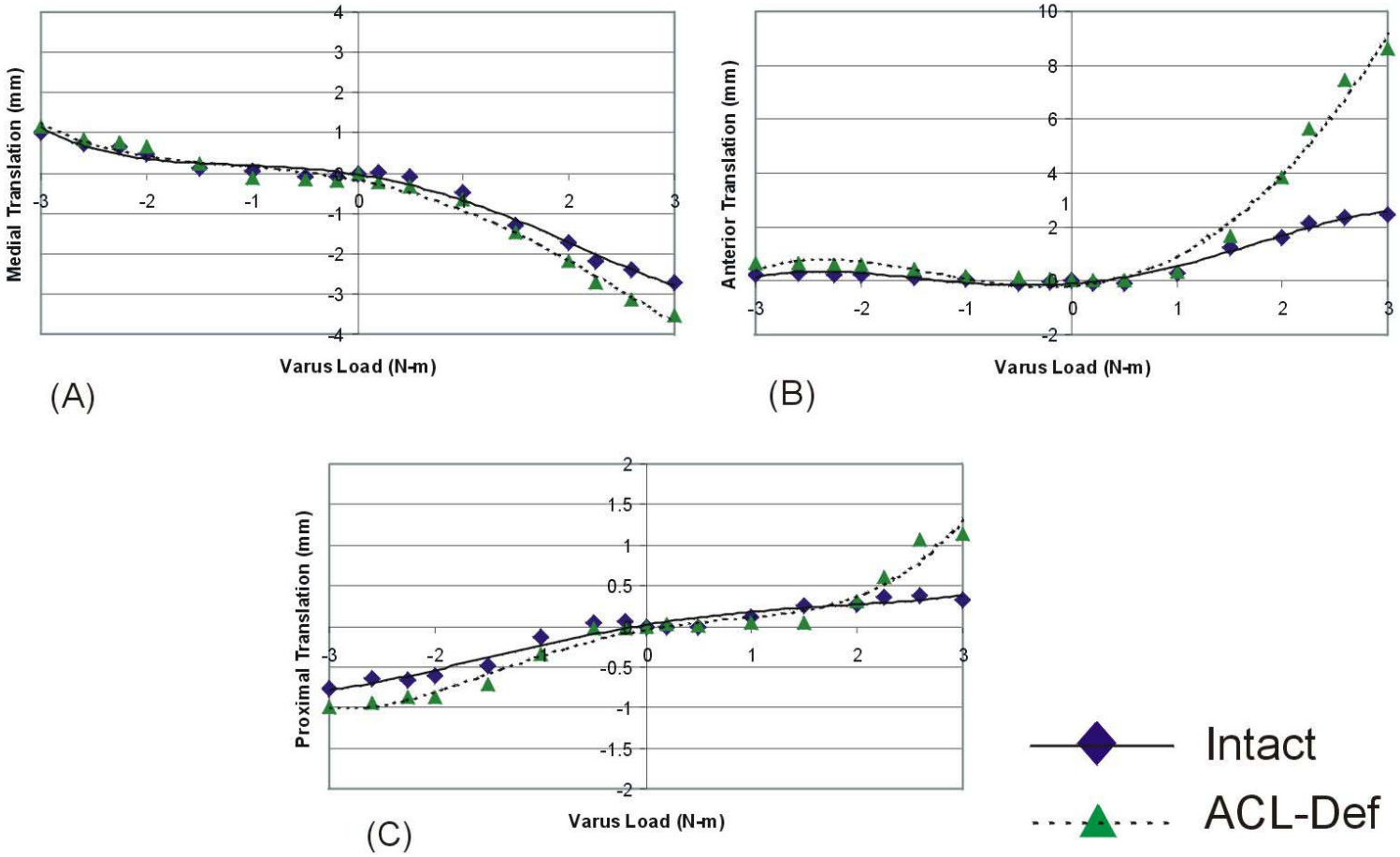


Figure 24 Translation in the medial (A), anterior (B) and posterior (C) directions in response to VV loads up to 3 N-m

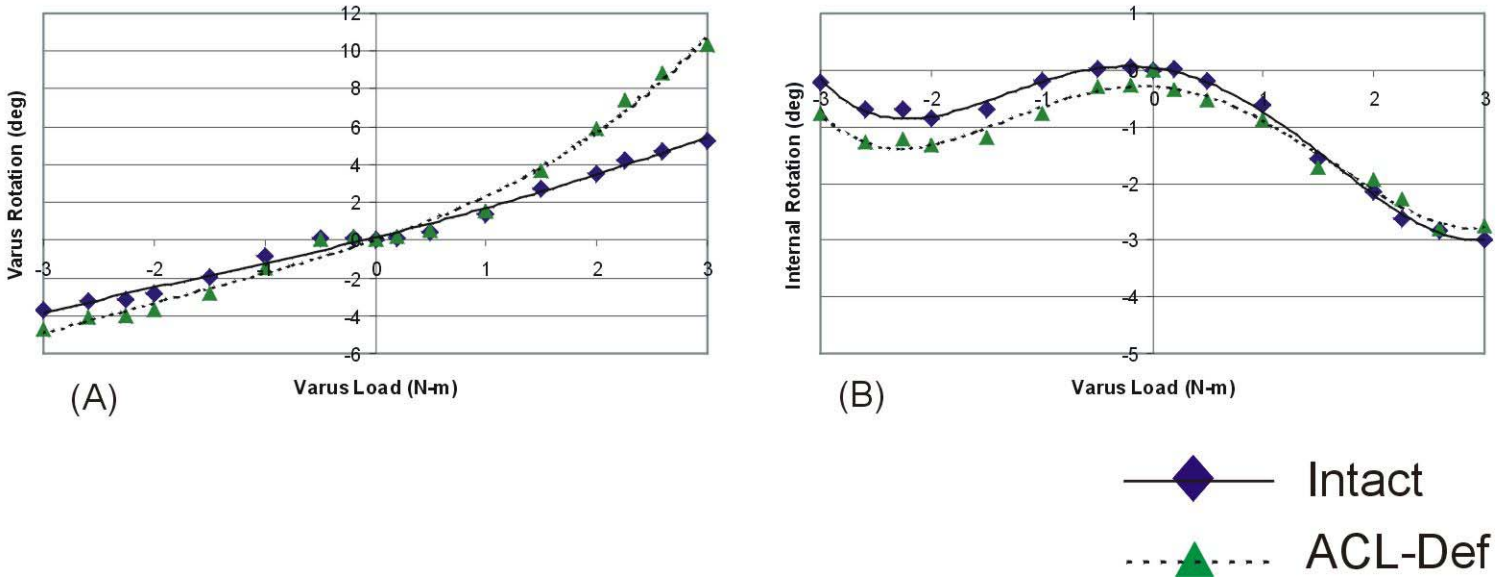


Figure 25 Rotation about the Varus (A), Internal (B) axes in response to VV loads up to 3 N-m

11.0 DISCUSSION

In this study, a biomechanical testing platform was developed to study the human knee joint. The platform consists of a software architecture that harbors the control algorithms necessary to examine the mechanical behavior of the knee joint in response to loading conditions of interest using a consistent joint motion description to represent all kinematics and kinetics. It was initially conceived as an update to the legacy robotic/UFS testing system but now innovates on its design in important ways, the most notable of which is its ability to accommodate necessary choices: choice of hardware, control algorithms and motion descriptions. Achievements in the software architecture followed the recruitment of state of the art programming technology towards the pursuit of three cardinal design principles: modularity, testability and versatility. It is proposed that the platform, as currently formulated, strategically positions the robotic/UFS testing methodology to be integrated with computational analyses in order to achieve concurrent multi-disciplinary studies of the mechanics of the human knee.

In relation to the specific aims of this work the platform features the following accomplishments:

Specific Aim 1 – Motion Tracking: The multi-purpose positioning system, or MPS, was formulated to generally represent the motion of arbitrary numbers of coordinate systems within the workspace. The MPS constitutes an expert system that manages a compact selection of

mathematical entities and operations to fully encode the position and orientation of coordinate systems as a function of time.

Specific Aim 2 – Kinematics/Kinetics: The coordinate system conventions and assignments made within the MPS were used to adapt an existing clinically relevant joint motion description, or JMD, within which all positions, orientations, forces and moments, and joint stiffness are consistently expressed.

Specific Aim 3 – Control Algorithms: Control algorithms were crafted after those of the legacy robotic/UFS testing system to provide force-moment feedback from the sensor and achieve a force control mode of operation for the biomechanical testing platform.

Specific Aim 4 – Software Architecture: All mathematical operations, control algorithms and data transformations and flow were integrated into an architecture that furnishes choices of interest to the bioengineer: choice of hardware, control algorithms and motion descriptions.

The platform was used to measure the *in situ* forces in the ACL for a sample of porcine specimens in response to anterior tibial drawer and varus-valgus tests, thereby demonstrating its ability to perform experimental studies homologous to those using the robotic/UFS testing system. More importantly, a variety of robot types by different manufacturers were operated by the platform, demonstrating its capacity to accommodate, among others, choices of hardware. The most poignant indication of success in the selection of technology and architecture for the platform is the reduction in development time required to integrate robots by different manufacturers: The development of control software for the legacy robotic/UFS testing system was estimated to involve several years for the UNIMATE PUMA 762 robot. The development of the platform itself required several years, but the modifications required to adapt it to three different robotic manipulators, FANUC S900-W, KUKA 2000 Series-210 and an ABB model

IRB6400, was accomplished in a matter of weeks. By selecting development technologies that were independent of robot manufacturer and by emphasizing a modular, testable and versatile architecture, robot choice, among others, was conveniently achieved.

There are limitations to note regarding the biomechanical testing platform. The system, for one, features an architecture that will need to continue to evolve to meet the inevitable emergence of new demands. One demand is to keep the software current and interoperable with the most recent software development advances: new languages, such as C# and its libraries could be used to supplement and simplify the current offering of technology (C++, STL, COM, MFC, OpenGL) and further ensure its maintainability. Fortunately migration is neither forced, and if undertaken can be achieved incrementally. Another important demand is the clarification of the software's mechanisms for concurrency: the architecture should mature further in regard to the type and timing of the interactions between its various components. More widespread threading and event-driven options should be furnished for each of the components to provide richer mechanism to push and pull their data (for bi-directional communication). For example, the model of interaction between the 'ExperimentManager' and the force-moment sensor is one where the former pulls data from the latter (unidirectional), without any concurrency. It is advisable to replace this with an event driven model of two concurrent components where the sensor is constantly measuring forces that could be pulled by the 'ExperimentManager', or even pushed onto it (bi-directional communication). With the sensor always measuring forces, it could easily detect collisions or overloading of objects within the workspace (noticed as spikes in force and moment) and alert the ExperimentManager to halt the experiment. Maturing the concurrency of the architecture is a challenging effort, but there are additions, such as the one mentioned above, that are identifiable, achievable and worthwhile.

There are also limitations on the study of the mechanical response of the knee to large loads. Given the possibly large deflection of the bones, the principle of superposition, invoked by cutting studies in pursuit of in situ forces in the various tissues, may not hold throughout the complete range of loading. Techniques may need to be developed to provide measurement on the size of bone deflection and estimate the potential error induced in any ensuing calculations of in situ force. Given the generality of the software in the biomechanical testing platform, additional equipment could be integrated to track bone motion and interlace this redundant data on bone deflection with that from obtained from the robot and force-moment sensor during the experiment.

It is precisely this ability of the software to accommodate additional software and hardware equipment is what represents the most significant innovation achieved with this work. This is the result of a concerted effort to evolve a software architecture that can represent, as accurately as possible, the domain of biomechanics of the knee. More broadly, this architecture is guided by the principles of modularity, testability and versatility:

Modularity of the testing platform was achieved by partitioning the software into logically distinct but interfacing components, each encapsulating the subset of the overall expertise: motion tracking, hardware communications, control algorithms, joint motion descriptions, 3D graphics and user interface. With the software subdivided, maintenance became more accessible and less costly because corrections and improvements are narrowed to smaller areas of code that are logically independent.

Testability, as pursued for the platform, is evidenced by the ability to perform automatic and targeted verifications of the functional integrity of critical portions of its algorithms. Leveraging the modularity of the software, the interface to each of the components has been

made interoperable with both the biomechanical testing platform (where the component performs) and a verification harness (where it practices). The verification harness, a piece of software itself, can exercise each software component individually, validating its operation in relation to a set of known specifications. Several significant advantages must be cited for automatic verification of individual components. The most important is that core kinematics or control code, for example, can be verified in the absence of any hardware (such as the robotic arm). During the development of control algorithms for the original robotic/UFS testing system, this was not readily possible. With the current platform, verifications could be performed remotely, in the absence of the robot, force moment sensor or even the specimen. This significantly minimized exposure of the testing equipment to the inherent danger of development testing. It also reduced overall development costs because procuring hardware for an experiment takes time.

The versatility of the testing platform is the result of creating components with a sufficiently distinct interface to enforce an encapsulation of logical dependencies and expertise. For example, the MPS, which contains logic exclusively about coordinate system operations, is independent of any software constrained by robot manufacturer dependencies. Otherwise, the MPS could not function or be tested in the absence of a robotic arm. Given its mathematical generality, that would be a significant disadvantage, and one that was problematic with the legacy robotic/UFS testing system. Another example is the abstraction and encapsulation of the control algorithms themselves. Given that they are represented on the basis of 6-DOF position and orientations and 3D forces and moments (also, purely mathematical constructs), the algorithms should be reusable for the purpose of controlling computational analyses, i.e., not just experiments. Beside providing an avenue for automatic verification of the software (simple

canonical computational models could be used to certify algorithm integrity), encapsulation of the components would allow the reuse of the software in contexts were robotic/UFS testing experiments operate concurrently with computational analyses.

Enabling experiments and computational analyses to occur concurrently would constitute a significant departure from the existing paradigm for combining these two approaches towards the study of the knee joint. Previously, the experimentalist would conclude his/her work when the measurement is made, with the computational analyst starting thereafter to run his/her models. This can be said of research targeted to the knee joint where the computational analysis has historically occurred on a set of experimental data measured on a different set of subjects/specimens, by a researcher at a different laboratory. It is a reflection of the nature of the effort and expertise once required for each kind of approach. However, it now acts as a barrier to their consolidation, which is necessary for computer models to fulfill their promise of formulating fully validated and robust predictions of experimental behavior.

The bioengineer must be able to formulate a model based on the response of a subject to one set of loading conditions and use it to predict the response to another. He or she must be able to verify the accuracy of model predictions by reproducing such conditions experimentally: mismatches could be used to direct model refinements, while agreements could help chart confidence intervals. Concurrency is required for the two approaches. In other words, the paradigm of experimenting first and computing later is not rich enough for the bioengineer to navigate his/her inquiry into the level of validity and refinement of a subject-specific computational model intended to estimate the forces that occur in the knee joint *in vivo*. It is this eventual strategic integration of experiments and computations that is intended to become accessible with this work.

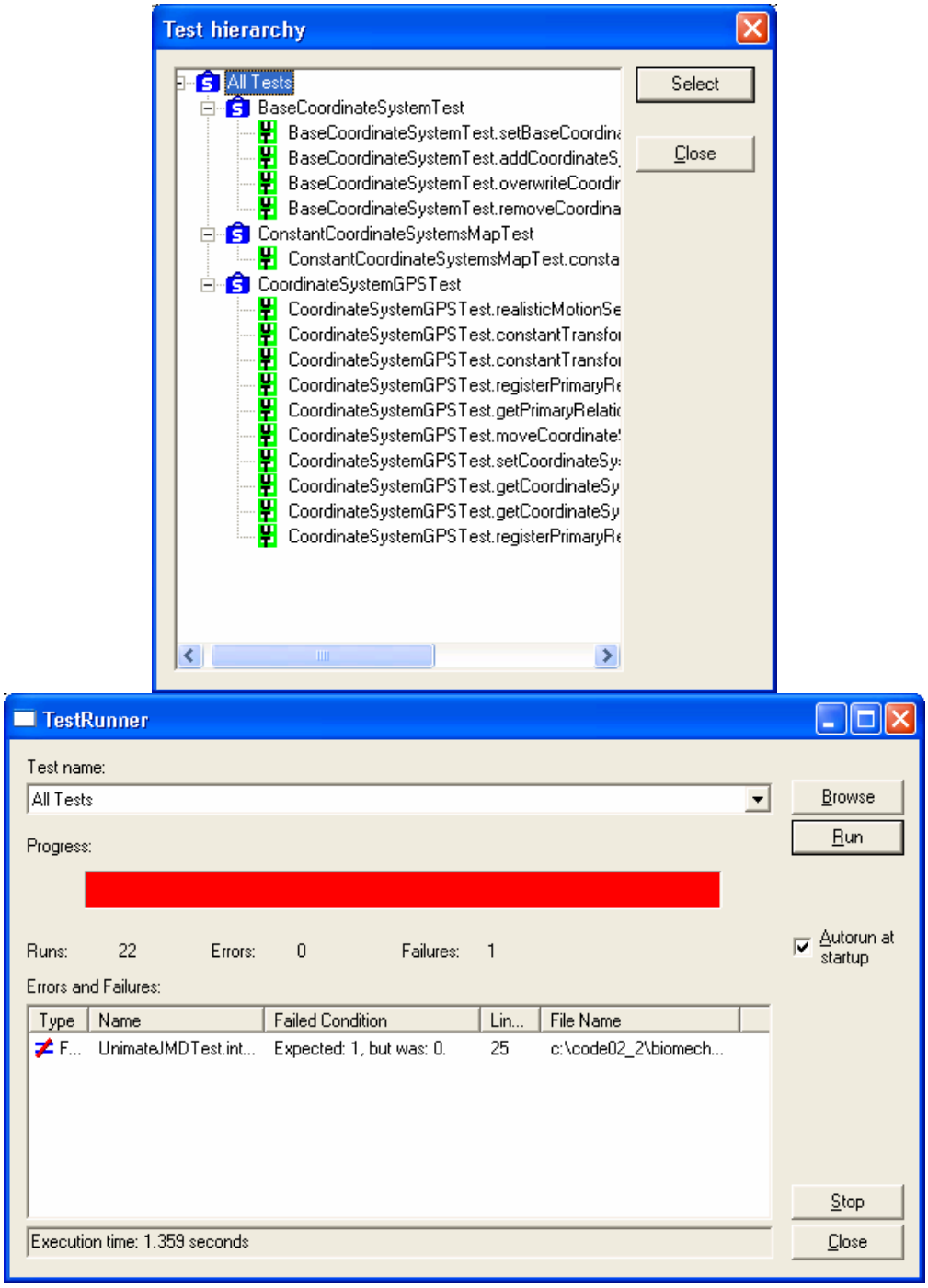


Figure 26 Test harness verifying MPS functionality. Above the tests are selected and below an error is detected

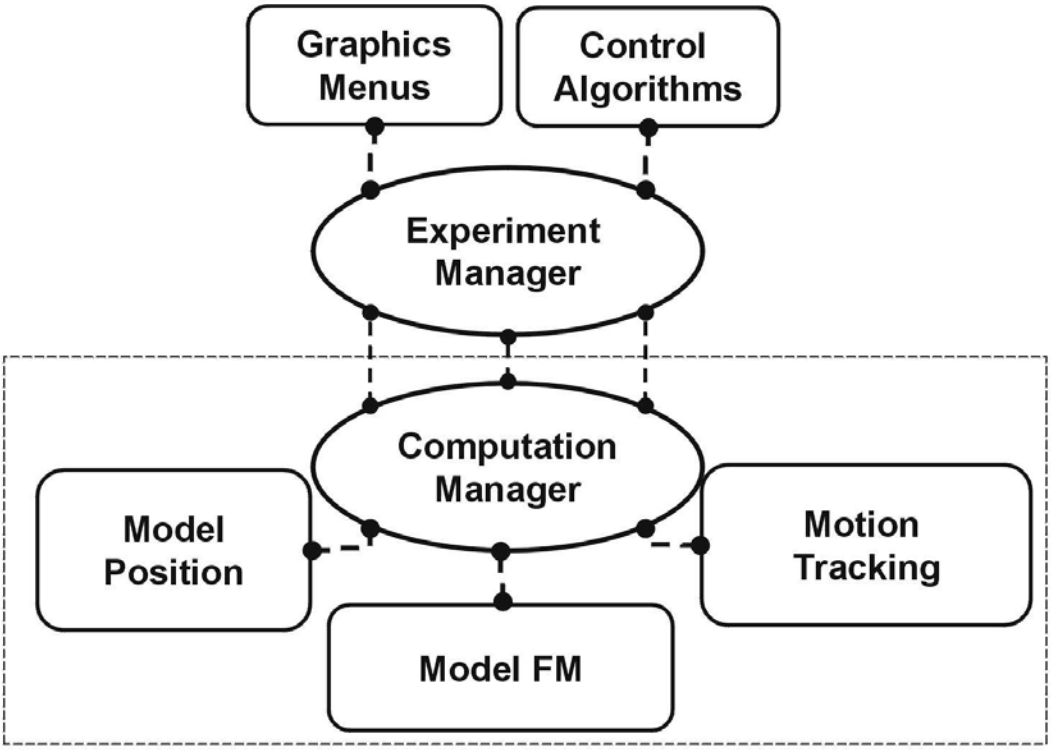


Figure 27 Architecture of the platform as it integrates computational analyses. The control algorithms, graphics and user interface remain intact. The experiment manager interacts indirectly with the MPS through the computation manager, which has substituted the robot and force-moment sensor by “model position” and “model force moment”

12.0 FUTURE DIRECTIONS

The RBSM or Rigid Body-Spring Method used for computational analysis of the knee joint will be implemented within the biomechanical testing platform[46]. This computational method simplifies the deformation of the articular cartilage and ligaments as that of the antagonistic action of a bed of compressive springs and strands of tensile springs, respectively. Given the architectural design and generality of the formulations for the kinematics and kinetics of the knee, this should be straightforward. The RBSM could be used to formulate automatic integration tests of the complete platform, as a complete experiment could be executed with the computational model, instead of an actual specimen. This test could be used to verify the integrity of the algorithms and overall functionality of the biomechanical testing platform.

12.1 OPTIMIZATION OF LIGAMENT PARAMETERS

One interesting application of the platform with concurrent operation of experiments and computation is the estimation of the ligament reference strain in a computational model of the knee joint.

A wealth of experimental data and representations are available for the mathematical modeling of the tensile properties of the ligaments[42, 91]. To successfully apply such models, a

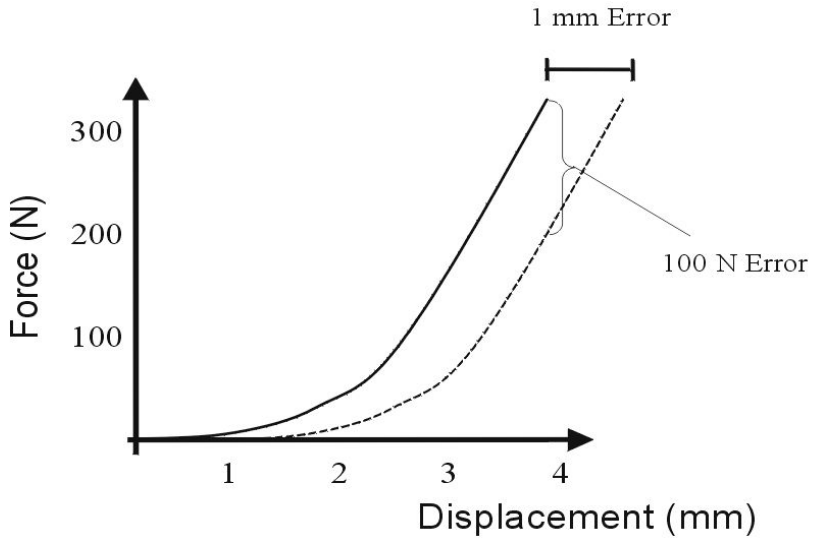


Figure 28 Effect of a 1 mm error in the estimation of the reference strain for a ligament with a typical force-displacement curve

reference strain is necessary to indicate the loading of the ligaments at some reference configuration of the knee. Acquiring such data would require invasive and even destructive methods that limit the breadth of experimentation possible with the specimen, so this data is often estimated. Estimates come with error: for reference strains, depending on the ligament and loading condition; an error of 1 mm can translate into force prediction errors in the order of ten to hundreds of Newtons.

For a full model of the knee the reference strain of all ligaments of the knee need to be known. In other words, to model a knee joint with an ACL, PCL, MCL and LCL a multivariate optimization for the 4 respective reference strains is necessary. The paradigm for optimization is currently to measure the response of the knee to a few select loading conditions and to validate the optimization with a select few others, all arbitrarily chosen. The problem is that for any given loading condition, one (or more) ligament may not be recruited to the extent that its function is visible, and the optimization of its values can be coarse or inaccurate. Ideally, a

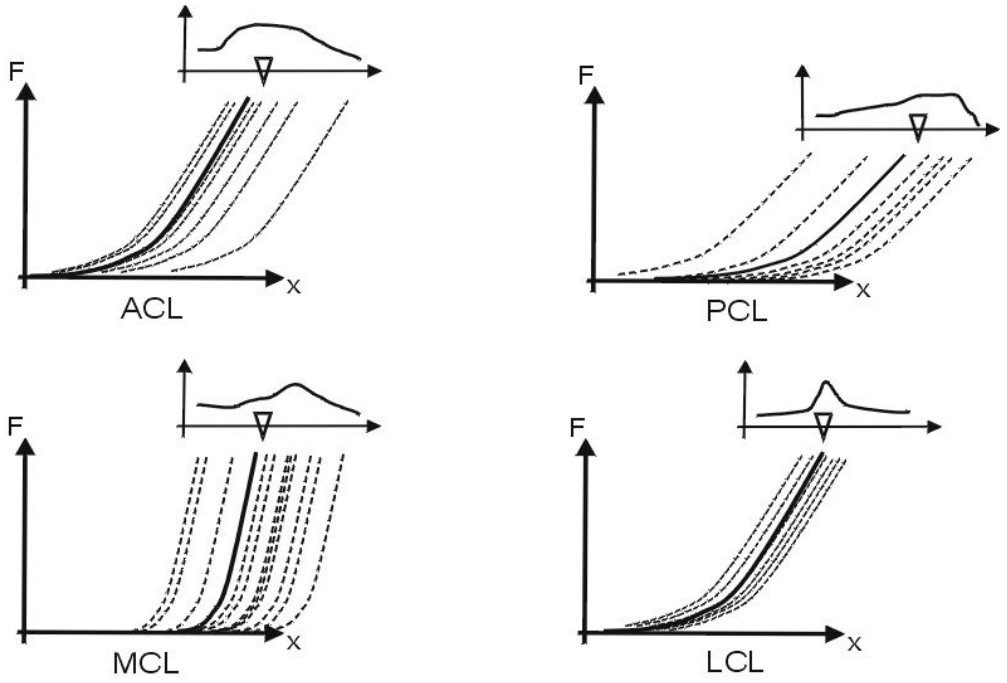


Figure 29 Optimal reference strains gathered for a number of loading conditions for the various ligaments. The small axes display distributions (incidence vs reference strain value) suggesting the likelihood that a given reference strain is correct

representative sample of loading conditions would be selected instead, where every ligament is exercised, even if not all at once. In fact, this is the premise of robust design. From the sample of loading conditions, the data could be assembled into distributions that reflect the likelihood of each value of reference strain for the ligaments

Collecting all such data for a rich enough sample of loading conditions could yield confidence intervals for a robust design that matches an ample set of mechanical responses of the knee.

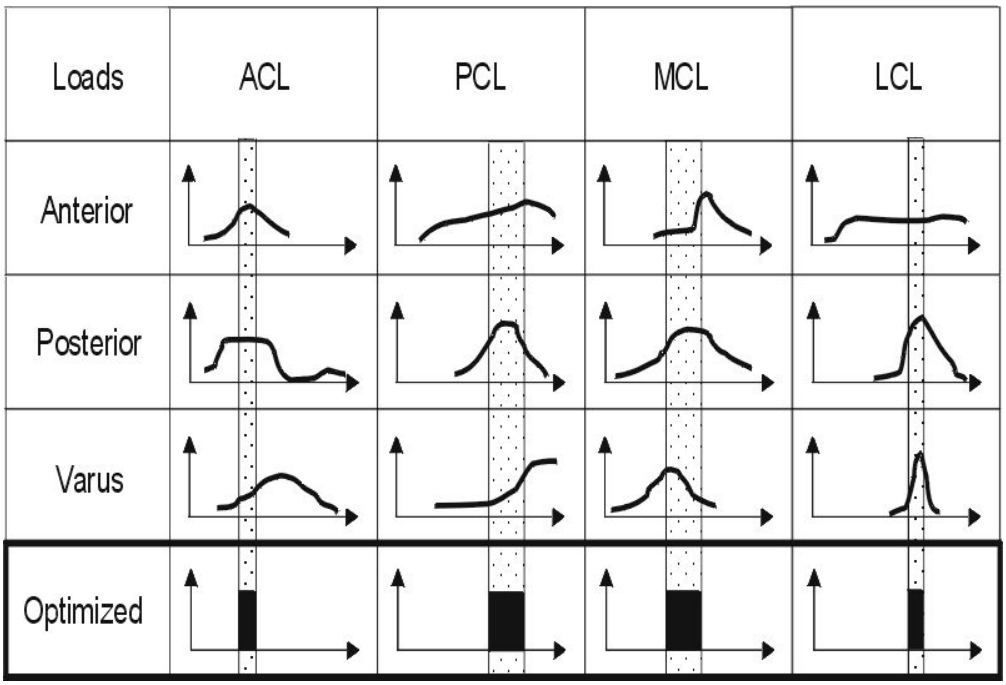


Figure 30 Building confidence intervals for the robust combination of reference strains for all ligaments

Gathering a sufficient sample of loading conditions may not be easy. More problematic is the fact that each specimen can only be tested experimentally for a limited period of time before its mechanical properties deteriorate. Therefore, how can the experimentalist know that enough data has been gathered for the computational analyst to complete his/her analysis? How can the computational analyst know, after the experiment, whether the optimal set of reference strains gathered from the data does in fact produce a model that has a verifiable experimental response?

One answer is to build the dataset of optimizations suggested in the previous figures from the wealth of force/moment and position/orientation data obtained throughout the iteration process for applying a variety of loading conditions. Typically experimentalists only report these values for the converged positions in the process of iteration. However, this data could be collected for unconverged data points, as well. As the experimental knee specimen is tested, the

computational model could operate concurrently, producing combinations of optimized reference strains for each iteration point that are compiled into the distributions suggested in the figures. This method would take advantage of the somewhat random perturbations that occur throughout the iteration process of the quasi-static control algorithms. It would be executed until the robust design reaches a sufficiently narrow confidence interval. At this point, the experimental test would be modified to perform model validation: the experimentalist would now have a quantitative basis with which to select a loading condition that truly challenges the model.

Furthermore, the stiffness properties reflected locally to the site of perturbations could also be correlated to that of the overall knee (ratio of range of motion to range of force of the specimen) and serve as a criterion to classify specimens according to their directional and rotatory stiffness. Given that overall directional and rotatory stiffness measures are part of standard clinical exams (KT2000, for example), this already collected in vivo data could guide the selection of cadaver specimens and/or computer models that match the behavior for a living subject, with an assessment of the range of validity of any measurements and estimations. This could further specialize the recruitment of experimental and computational data into representative categories to a given patient case in the process of elaborating conclusions about his/her injury mechanism, treatment, and rehabilitation.

By replacing the paradigm of experimenting first and computing later with one in which the two approaches operate concurrently, the bioengineer may be able to formulate a basis to select the best optimization method, together with a challenging validation scheme. He or she would have significantly greater freedom in navigating his/her inquiry into the level of validity and refinement of a subject-specific computational model and be able to use it for the aim of accurately measuring the forces that occur in the knee joint *in vivo*.

APPENDIX A

COORDINATE SYSTEM CONVENTIONS

Position control devices such as a robotic manipulator are attributed two coordinate systems, by default, and these fulfill the minimal task of tracking the position and orientation of the end-effector with respect to the fixed base. These are the end-plate (*E*) and world (*W*) coordinate systems and are fixed by the manufacturer to landmarks on the equipment (and cannot be redefined). However, two copies of these coordinate systems are also often available and can be reconfigured to suit the application are also provided. They are the tool (also called tool-center-point or *TCP*) and user (*U*) coordinate systems. Each of these can be offset to a new position or orientation with respect to their native references (end-plate and world, respectively).

The force-moment sensor is attributed one coordinate system by default. This is designated as the sensor (*S*) system. It is convenient to make an axis in the S system to be aligned with the normal to the force-moment measuring surface of the sensor. It is also convenient to align the TCP system with the sensor system, so that their x, y and z axes exactly superimposed, i.e. S = TCP.

For the case of the FANUC S-900 W robot originally used in the development of the software, the z-axis of the S system pointed normally to the ATI sensor measuring plate. When

clamped to the robot, S = TCP and the orientation of TCP matched the E system, and was offset along its z-axis by 88.5 mm.

$$
\begin{aligned}
&{}^{W}_{U}R \equiv I \\
&{}^{E}_{S}R \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 88.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{12-1}
$$

It was of little use to edit the U system relative to W. In fact, this is considered somewhat dangerous, as applications for the robot written by other parties may yield surprising results to such an inadvertent modification. Editing the TCP relative to E was considered less dangerous, as it didn't actually change the relative orientation of the two systems, but just their translation.

Given the acronyms displayed in the table, a coordinate system such as 'US_R' in the code comes to mean ${}^{U}_{S}R$, as per the derivations of this work. Additionally, a transformation such as ${}^{S}R_{Motion}$, signfying motion relative to the sensor, is represented as 'S_R_Mot' in the code.

Table 6 Acronyms commonly used for coordinate systems in the biomechanical testing platform

Symbol	Name
U	User (Global)
S	Sensor (Force-Moment)
F	Femur
T	Tibia
E	End-effector
W	World (Global)
TCP	Tool Center Point

APPENDIX B

INITIALIZING THE MPS COORDINATE SYSTEMS

The figure demonstrates typical locations, 11 in all, for the landmarks on the tibia used to spatially digitize its corresponding coordinate system. Analogous locations are digitized for the femur, as well. It is important to note that the digitization of points 1 and 2 (medial and lateral) is more critical for the femur because they directly determine the direction of the flexion-extension axis in the JMD (the axis runs exactly through these points). For the femur, they are typically selected to be on the insertions of the MCL and LCL. For the tibia, digitization of the diaphyseal axis is more critical because it directly determines the proximal distal axis of the JMD. Such points are selected on the surface of the cylindrically shaped potting material that surrounded the diaphysis. Point 3 is generally selected someplace on the anterior aspect of the bone, so that the epiphyseal plane and the diaphyseal axis intersect at approximately right angles.

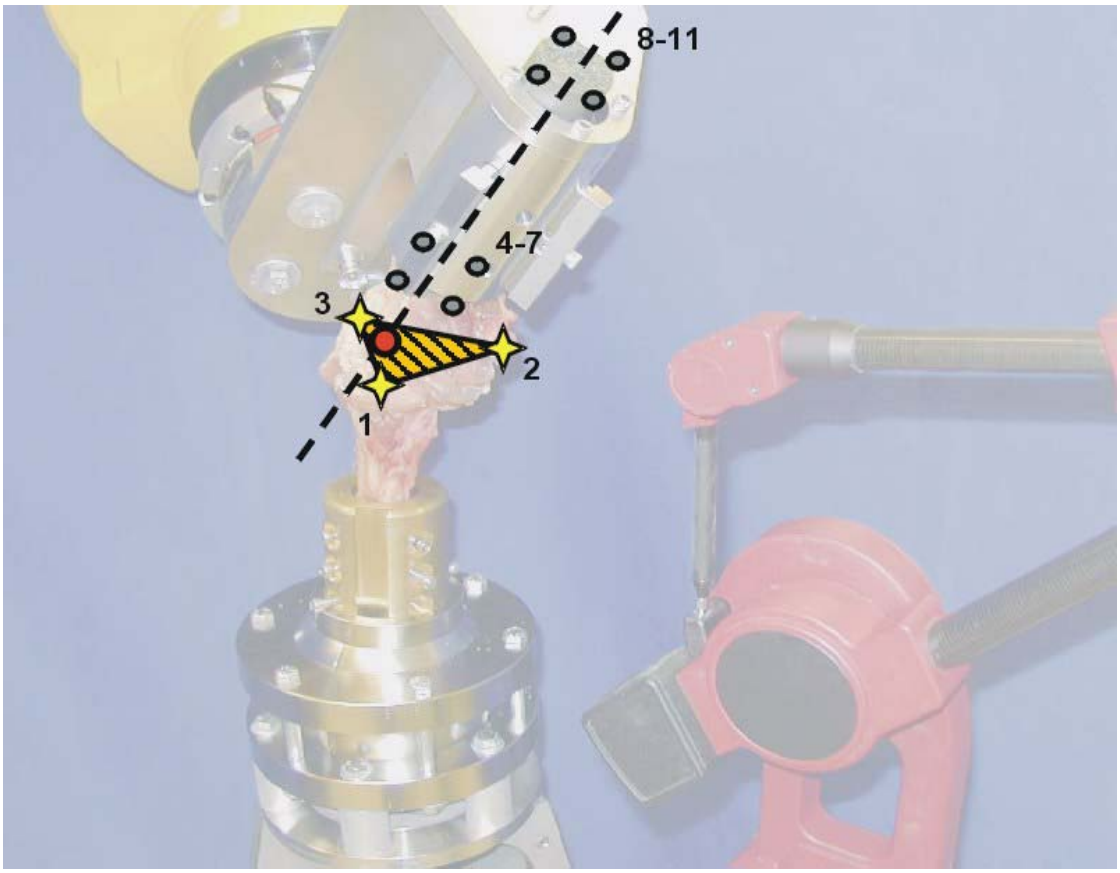


Figure 31 Digitization the spatial location on the tibia to initialize the MPS coordinate systems. Together, points 1, 2, and 3 define an epiphyseal plane. Points 4-7 and 8-11 define a diaphyseal axis. The origin of the tibial coordinate system is located at the intersection of the diaphyseal axis with the epiphyseal plane

APPENDIX C

TARING THE FORCES AND MOMENTS AT THE SENSOR

Counts are the units provided by the manufacturer to translate the analog elongation of the underlying strain gages into a digital value of force and moment at the measuring plate. There is a linear relationship between the value of the measurement of force and moment in counts and its corresponding value in N and N-mm when the sensor is within it normal operating range. Therefore, a slope and intercept must be known to get useful measurements.

Whereas the slope is provided by the manufacturer and can be queried from the sensor through its API, the intercept must be measured experimentally. This is because the intercept not only depends on intrinsic factors to the UFS, but also on extrinsic artifacts such as bolting loads generated during the rigid attachment of the tibial clamp, which vary from one experiment to the next. As a result, a method was adopted to identify this intercept more generally.

Every measurement of force and moment made with the sensor while the clamp is attached has confounded both the weight of the clamp and the bolting load. Bolting loads are the result of over-constraint at the imperfectly mating surfaces of the sensor and the tibial clamp caused by the attachment bolts. For a rigid attachment, they are fixed in magnitude and orientation relative to the sensor; the clamp weight, on the other hand is fixed relative to space. Thus, measurements can be taken in various sensor orientations such that the effect of the clamp

weight cancels, isolating the effect of bolt loads. For example, measuring force in the z-direction while the sensor is oriented with and against gravity will cancel the effect of the clamp weight, leaving an idea of the sensor z-direction bolt force. At the same time, x and y moments cancel, suggesting the x and y bolt moments. This process can yield tare forces and moments in all sensor channels (x, y, z).

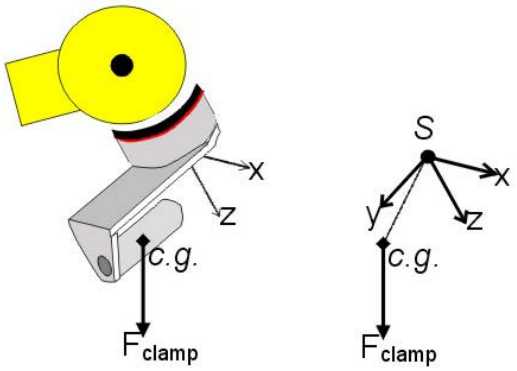


Figure 32 Clamp weight relative to the sensor

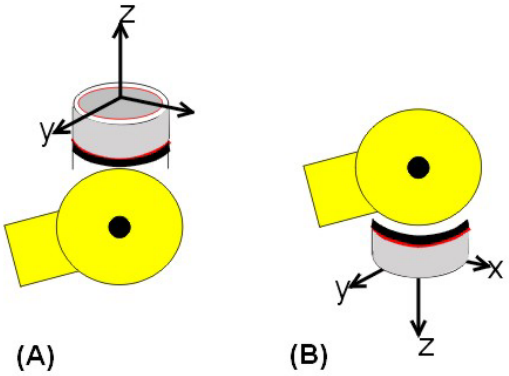


Figure 33 Calibration of the z-axis using two measurements “Z-Up” (A) and “Z-Dn” (B)

APPENDIX D

RESULTANT FORCES AND THE JMD

The calculation of a resultant force F_{res} in 3D space is straightforward, given the Cartesian components $\left(F_x, F_{\mathbf{y}}, F_z\right)$ in an arbitrary coordinate system:

$$F_{res} = \sqrt{F_x^2 + F_{\mathbf{y}}^2 + F_z^2} \tag{12-2}$$

The same statement in JMD is **only approximately true and should be used judiciously**. That is:

$$F_{res} \approx \sqrt{F_{ML}^2 + F_{AP}^2 + F_{PD}^2} \tag{12-3}$$

The reason is that the JMD axes do not generally form an orthogonal system: The ML and AP axes are orthogonal; so are the AP and PD axes. However, the ML and PD axes are **not** orthogonal. In fact, the ML and PD axes deviate from orthogonality by an angle that is, by construction, equal to the VV rotation.

Given these facts, approximation 11-3 is good as long as the VV rotation is small, because in such case, the deviation of the JMD from an orthogonal description is small. If some VV rotation is present, the approximation is good unless both ML and PD components are

Table 7 Approximate magnitude calculated from JMD components on a 100 N force, as a function of VV angle.

VV Angle (º)	Approximated Magnitude (N)
-10	108.2
-9	107.4
-8	106.7
-7	105.9
-6	105.1
-5	104.3
-4	103.4
-3	102.6
-2	101.7
-1	100.9
0	100
1	99.1
2	98.2
3	97.4
4	96.5
5	95.6
6	94.7
7	93.8
8	92.9
9	92
10	91

simultaneously large. The following table shows the error expected for a force of magnitude 100 N distributed along the ML and PD axes (70.7 N along each axis), as a function of VV angle: The error is in the order of 1% per degree of VV rotation.

The loading conditions considered in this study (AP and VV) involve load targets that both minimize ML and PD forces. AP loads, in addition, feature low VV rotations. Consequently the approximation is deemed appropriate for this study.

APPENDIX E

EXPERIMENTAL DATASET

On the following pages are the results to the experiments, fully tabulated and graphed. Some of the tables and figures have been sampled earlier on throughout the body of this document. Appendix A contains the kinematics of the knee in response to the two loading conditions, 100 N A-P and 3 N-m V-V tibial loads at the two flexion angles tested, 35° and 90°.

Table 8 Kinematics of the knee in response to a 100 N A-P tibial load at 35° of flexion

(A)

Applied Load (N)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-100	0.0	1.1	-3.3	0.5	-0.7	0.5	-0.4	0.4	1.6	2.0
-86.6	-0.1	1.0	-2.8	0.4	-0.6	0.4	-0.3	0.4	1.1	1.8
-75	-0.3	0.9	-2.4	0.4	-0.5	0.3	-0.3	0.4	0.8	1.5
-66.6	-0.3	0.7	-2.1	0.3	-0.5	0.2	-0.2	0.3	0.5	1.3
-50	-0.4	0.6	-1.6	0.3	-0.4	0.1	-0.1	0.3	0.1	1.0
-33.3	-0.4	0.6	-1.1	0.2	-0.2	0.1	-0.1	0.1	-0.1	0.8
-16.6	-0.3	0.4	-0.5	0.2	-0.2	0.1	0.0	0.1	-0.1	0.7
-6.6	-0.1	0.1	0.0	0.2	0.0	0.0	-0.1	0.1	-0.1	0.3
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6.6	-0.1	0.1	0.3	0.4	0.1	0.2	0.1	0.3	-0.5	1.0
16.6	-0.2	0.6	2.8	2.0	0.7	0.6	0.5	0.8	-2.3	3.4
33.3	-0.4	0.6	3.5	2.0	0.9	0.6	0.6	0.8	-2.7	3.5
50	-0.7	0.6	4.1	1.9	1.0	0.6	0.8	0.9	-3.2	3.7
66.6	-1.1	0.8	4.9	1.9	1.3	0.7	1.1	1.1	-4.1	3.9
75	-1.4	0.8	5.3	1.7	1.4	0.6	1.2	1.2	-4.4	3.9
86.6	-1.7	0.7	5.7	1.5	1.6	0.7	1.4	1.1	-5.0	3.8
100	-2.1	0.8	6.3	1.4	1.8	0.6	1.8	1.4	-5.5	4.1

(B)

Applied Load (N)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-100	0.2	1.4	-4.0	1.0	-0.7	0.3	-0.6	0.7	2.3	3.8
-86.6	-0.2	1.0	-3.3	0.7	-0.5	0.3	-0.4	0.6	1.2	2.5
-75	-0.5	0.9	-2.9	0.5	-0.5	0.1	-0.3	0.5	0.5	1.7
-66.6	-0.5	0.8	-2.6	0.5	-0.4	0.1	-0.2	0.5	0.4	1.5
-50	-0.5	0.6	-2.1	0.4	-0.4	0.1	-0.1	0.4	0.0	1.1
-33.3	-0.6	0.6	-1.6	0.2	-0.3	0.1	0.1	0.2	-0.3	0.7
-16.6	-0.5	0.5	-0.9	0.3	-0.3	0.2	0.1	0.2	-0.3	0.6
-6.6	-0.3	0.3	-0.3	0.3	-0.1	0.1	0.1	0.2	-0.2	0.3
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6.6	-0.1	0.0	0.1	0.1	0.1	0.1	0.0	0.0	-0.1	0.1
16.6	0.8	0.9	4.3	1.5	1.0	0.4	0.3	0.2	-1.5	0.7
33.3	0.9	1.1	8.6	3.4	1.6	0.8	3.4	2.0	-1.7	1.2
50	0.9	1.9	12.3	2.5	2.6	1.1	5.8	1.3	1.1	3.5
66.6	2.9	2.4	16.8	0.3	4.2	1.1	6.5	2.5	9.3	4.0
75	4.0	2.9	17.9	0.2	5.0	1.0	6.0	2.9	11.7	4.0
86.6	4.2	3.0	18.3	0.2	5.3	1.2	5.8	3.0	12.1	3.9
100	4.5	2.8	19.2	0.4	5.7	1.1	5.4	2.9	13.0	4.2

Table 9 Kinematics of the knee in response to a 100 N A-P tibial load at 90° of flexion

Applied Load (N)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-100	-0.6	1.6	-2.8	0.7	-0.9	0.5	0.2	0.7	0.0	3.7
-86.6	-0.7	1.5	-2.5	0.7	-0.8	0.4	0.3	0.7	-0.1	3.4
-75	-0.7	1.4	-2.1	0.7	-0.7	0.4	0.4	0.6	-0.2	3.1
-66.6	-0.7	1.1	-1.8	0.6	-0.6	0.3	0.4	0.5	-0.3	2.7
-50	-0.7	1.0	-1.3	0.4	-0.5	0.3	0.5	0.6	-0.4	2.2
-33.3	-0.7	0.9	-0.9	0.4	-0.4	0.3	0.6	0.6	-0.5	1.9
-16.6	-0.6	0.8	-0.4	0.3	-0.2	0.3	0.5	0.6	-0.5	1.6
-6.6	-0.4	0.5	0.0	0.2	0.0	0.1	0.5	0.5	-0.6	1.1
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6.6	0.0	0.2	0.1	0.1	0.0	0.1	0.0	0.1	0.0	0.2
16.6	0.0	0.4	0.4	0.3	0.1	0.2	0.2	0.4	-0.2	0.6
33.3	0.0	0.6	1.6	1.0	0.5	0.4	0.8	1.3	-0.6	0.9
50	0.1	0.7	3.0	1.6	0.8	0.6	1.7	2.0	-0.6	0.9
66.6	0.3	0.6	3.9	1.6	1.0	0.6	2.1	2.2	-0.5	0.9
75	0.3	0.6	4.3	1.5	1.1	0.6	2.3	2.3	-0.4	1.0
86.6	0.3	0.6	4.6	1.5	1.2	0.6	2.5	2.5	-0.3	1.3
100	0.2	0.6	5.0	1.4	1.3	0.6	2.6	2.7	-0.5	1.7

Table 10 Kinematics of the knee in response to a 3 N-m V-V tibial load at 35° of flexion

(A)

Applied Load (N-m)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-3.0	1.0	1.2	0.2	0.4	-0.8	0.6	-3.7	1.9	-0.2	2.9
-2.6	0.7	1.1	0.3	0.5	-0.6	0.6	-3.3	1.9	-0.7	2.5
-2.3	0.7	1.2	0.2	0.4	-0.7	0.6	-3.1	1.7	-0.7	2.5
-2.0	0.5	1.0	0.2	0.4	-0.6	0.5	-2.8	1.8	-0.8	2.3
-1.5	0.1	0.9	0.1	0.4	-0.5	0.4	-2.0	1.2	-0.7	1.9
-1.0	0.1	0.9	0.1	0.5	-0.1	0.3	-0.8	0.9	-0.2	1.7
-0.5	-0.1	0.5	-0.1	0.3	0.0	0.2	0.1	0.3	0.0	1.0
-0.2	-0.1	0.3	0.0	0.2	0.1	0.2	0.1	0.3	0.1	0.8
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	0.0	0.1	-0.1	0.1	0.0	0.1	0.1	0.1	0.0	0.4
0.5	-0.1	0.2	-0.1	0.2	0.0	0.1	0.5	0.3	-0.2	0.6
1.0	-0.5	0.4	0.3	0.5	0.1	0.2	1.3	1.0	-0.6	1.2
1.5	-1.3	0.9	1.2	1.5	0.3	0.5	2.8	1.8	-1.6	1.8
2.0	-1.7	1.2	1.6	1.8	0.3	0.6	3.6	2.2	-2.2	2.1
2.3	-2.2	1.6	2.2	1.9	0.4	0.6	4.3	2.5	-2.6	2.4
2.6	-2.4	1.7	2.4	2.0	0.4	0.8	4.7	2.7	-2.8	2.6
3.0	-2.7	1.9	2.5	2.0	0.3	0.7	5.2	2.9	-3.0	2.7

(B)

Applied Load (N-m)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-3.0	1.1	1.6	0.6	0.5	-1.0	0.8	-4.7	1.6	-0.8	1.5
-2.6	0.8	1.5	0.6	0.5	-0.9	0.8	-4.1	1.6	-1.3	1.0
-2.3	0.8	1.5	0.6	0.5	-0.9	0.7	-4.0	1.6	-1.2	1.0
-2.0	0.6	1.4	0.6	0.4	-0.9	0.8	-3.7	1.7	-1.3	1.0
-1.5	0.2	1.2	0.5	0.3	-0.7	0.6	-2.8	1.4	-1.2	1.0
-1.0	-0.1	0.9	0.2	0.5	-0.4	0.2	-1.5	0.5	-0.8	1.2
-0.5	-0.2	0.3	0.1	0.2	0.0	0.2	0.0	0.4	-0.3	0.5
-0.2	-0.2	0.3	0.1	0.1	0.0	0.1	0.1	0.3	-0.3	0.5
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	-0.2	0.2	0.0	0.1	0.0	0.1	0.2	0.2	-0.3	0.4
0.5	-0.3	0.2	0.0	0.2	0.0	0.0	0.5	0.3	-0.5	0.6
1.0	-0.6	0.2	0.3	0.3	0.0	0.1	1.5	0.8	-0.9	1.3
1.5	-1.5	0.5	1.7	0.9	0.0	0.5	3.7	0.3	-1.7	1.9
2.0	-2.2	0.9	3.8	2.0	0.3	0.9	5.9	1.0	-1.9	1.9
2.3	-2.7	1.2	5.6	2.8	0.6	1.3	7.4	1.5	-2.3	2.3
2.6	-3.2	1.4	7.4	3.9	1.1	1.7	8.8	2.4	-2.8	2.3
3.0	-3.5	1.6	8.6	4.0	1.1	1.9	10.3	2.6	-2.8	2.1

Table 11 Kinematics of the knee in response to a 3 N-m V-V tibial load at 90° of flexion

Applied Load (N-m)	Translations (mm)						Rotations (deg)			
	Medial		Anterior		Proximal		Varus		Internal	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
-3.0	2.2	0.5	-0.2	0.7	0.0	0.8	-3.8	1.5	0.0	3.7
-2.6	1.4	0.5	-0.3	0.7	0.0	0.6	-2.6	0.9	-0.1	3.4
-2.3	0.9	0.6	-0.3	0.5	0.0	0.4	-2.0	0.9	-0.2	3.1
-2.0	0.7	0.6	-0.3	0.5	0.0	0.4	-1.6	0.4	-0.3	2.7
-1.5	0.0	0.8	-0.3	0.4	-0.1	0.3	-0.4	0.5	-0.4	2.2
-1.0	-0.3	0.8	-0.3	0.3	0.0	0.2	0.1	0.5	-0.5	1.9
-0.5	-0.4	0.8	-0.2	0.2	-0.2	0.1	0.5	0.7	-0.5	1.6
-0.2	-0.5	0.6	-0.2	0.2	-0.1	0.1	0.6	0.5	-0.6	1.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	-0.1	0.1	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.2
0.5	-0.4	0.3	0.0	0.2	-0.1	0.1	0.5	0.3	-0.2	0.6
1.0	-0.9	0.6	0.2	0.6	-0.1	0.2	1.3	0.8	-0.6	0.9
1.5	-1.1	0.6	0.5	1.2	-0.1	0.4	2.0	1.4	-0.6	0.9
2.0	-1.5	0.8	0.9	1.6	-0.2	0.6	2.9	1.9	-0.5	0.9
2.3	-1.7	1.0	1.1	1.8	-0.2	0.6	3.4	2.3	-0.4	1.0
2.6	-1.8	1.0	1.3	1.9	-0.3	0.7	3.8	2.5	-0.3	1.3
3.0	-2.0	1.0	1.5	2.0	-0.3	0.8	4.4	2.6	-0.5	1.7

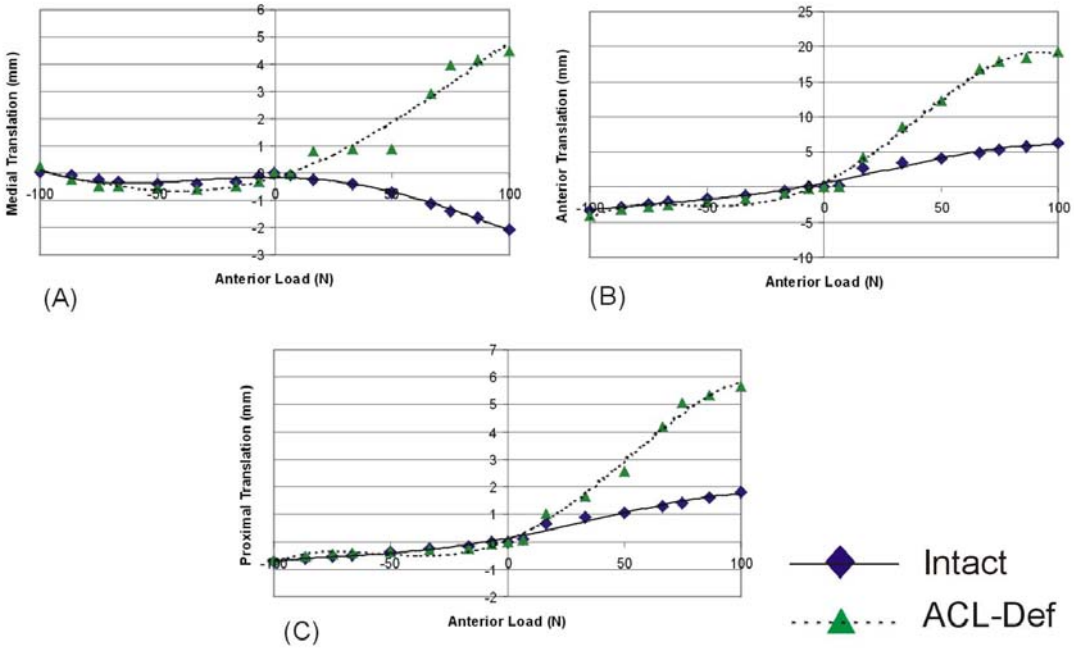


Figure 34 Translation of the knee in response to a 100 N A-P tibial load at 35° of flexion

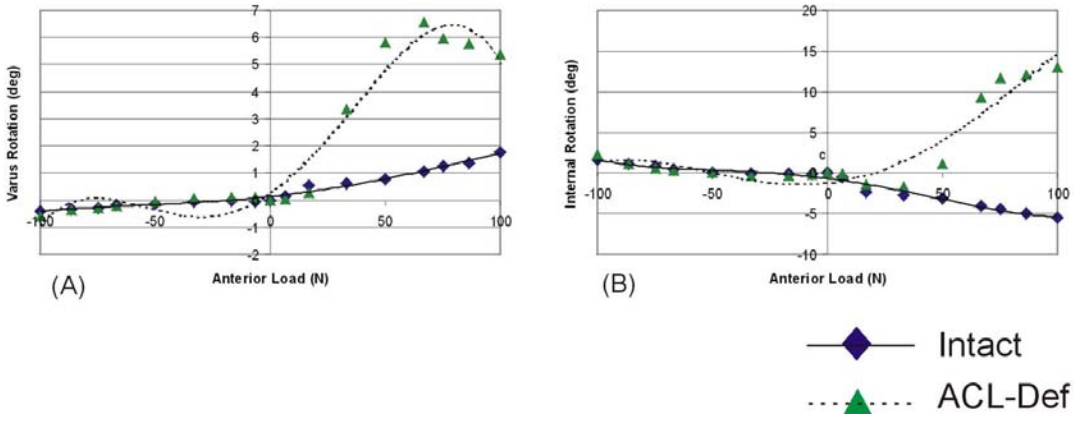


Figure 35 Rotation of the knee in response to a 100 N A-P tibial load at 35° of flexion

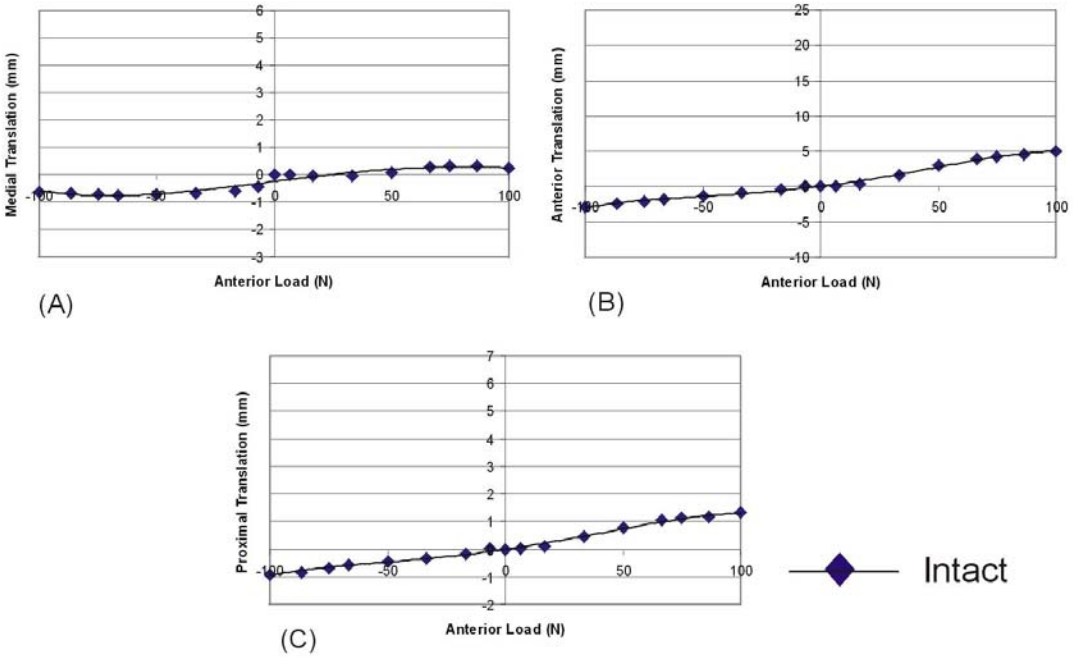


Figure 36 Translation of the knee in response to a 100 N A-P tibial load at 90° of flexion

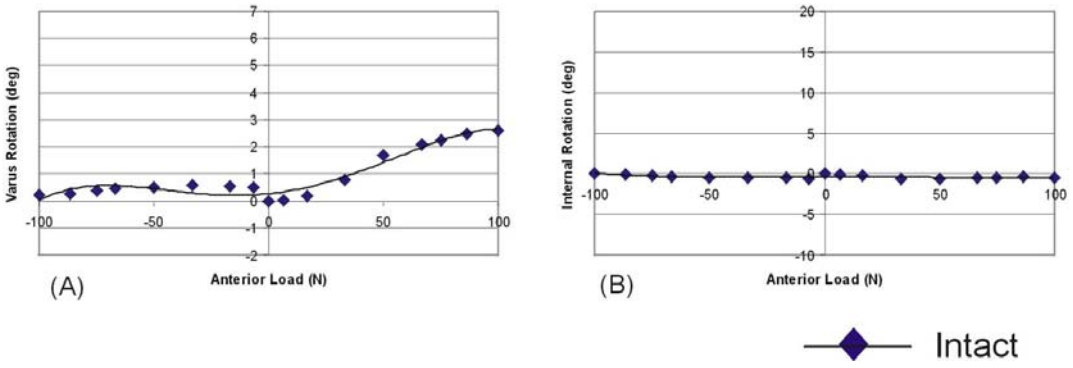


Figure 37 Rotation of the knee in response to a 100 N A-P tibial load at 90° of flexion

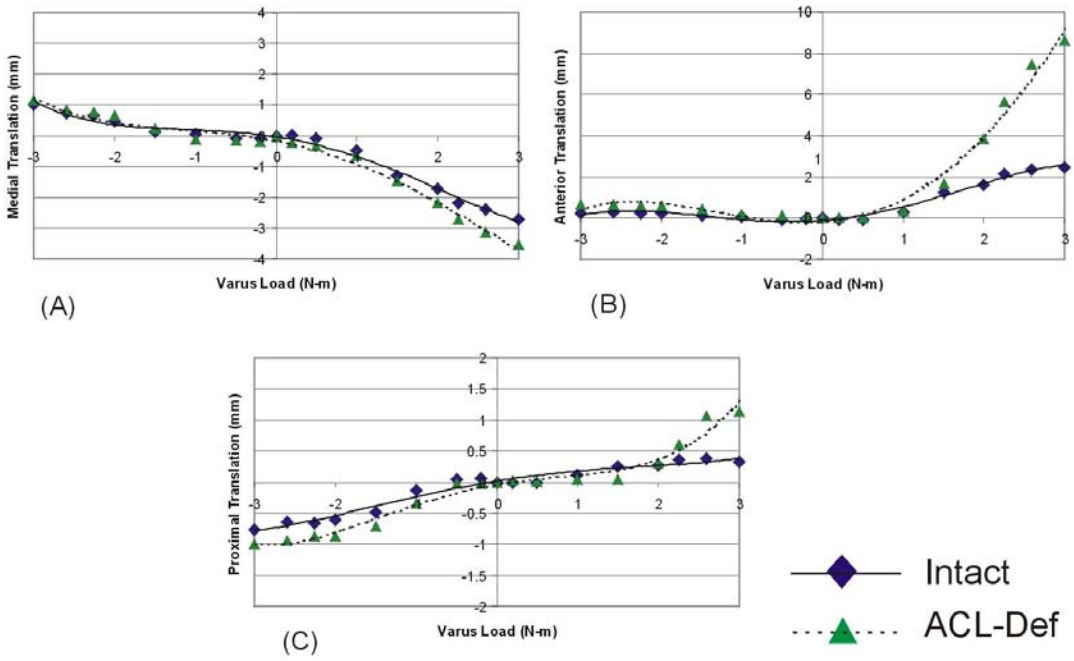


Figure 38 Translation of the knee in response to a 3 N-m V-V tibial load at 35° of flexion

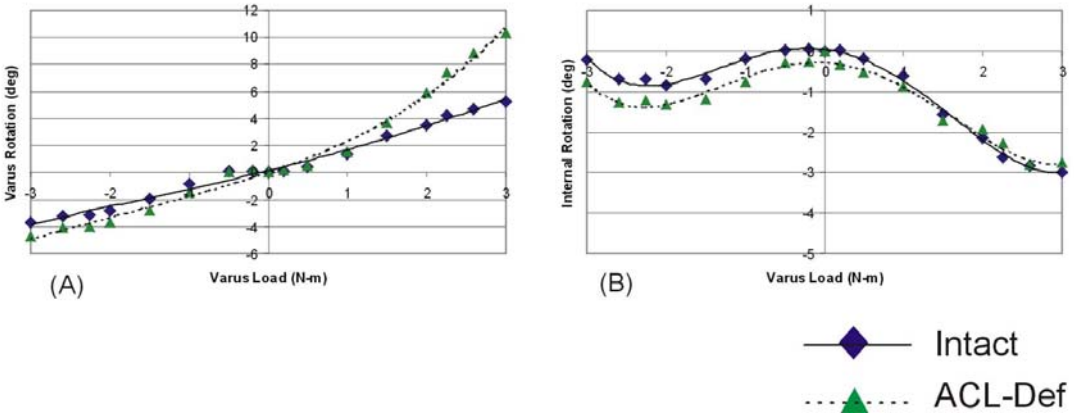


Figure 39 Rotation of the knee in response to a 3 N-m V-V tibial load at 35° of flexion

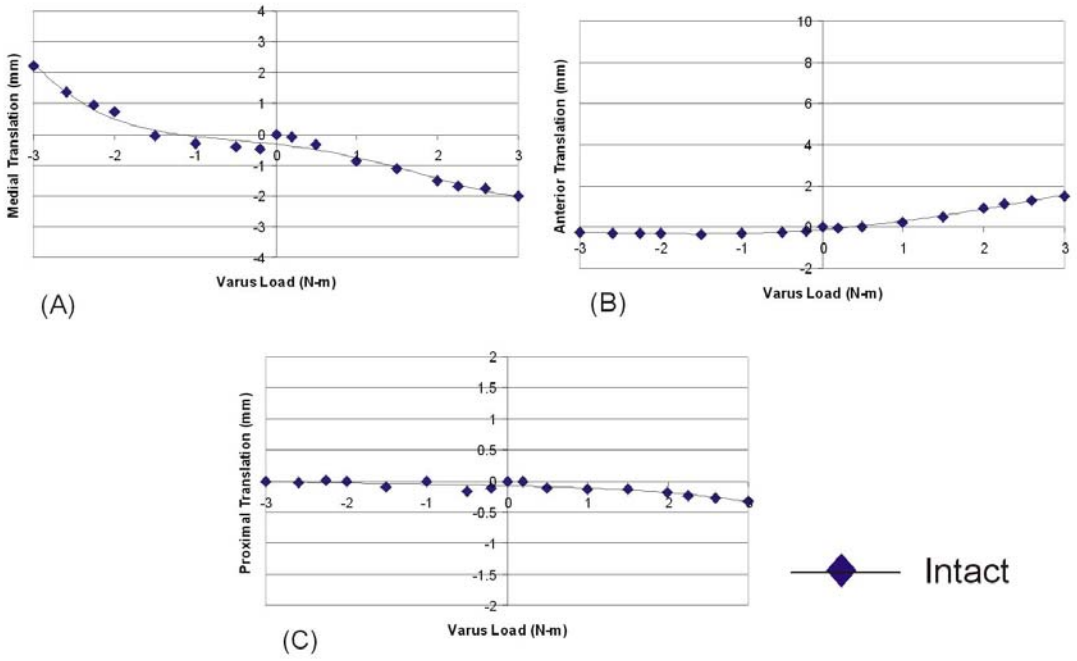


Figure 40 Translation of the knee in response to a 3 N-m V-V tibial load at 90° of flexion

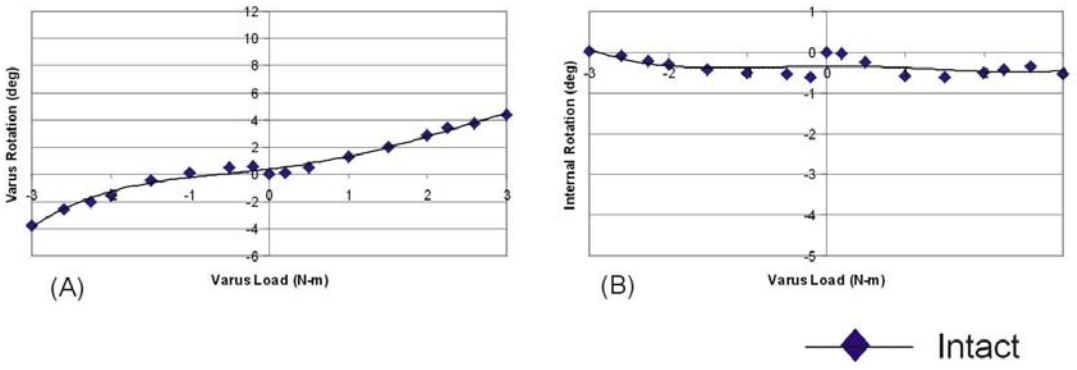


Figure 41 Rotation of the knee in response to a 3 N-m V-V tibial load at 90° of flexion

BIBLIOGRAPHY

1. Chao, E.Y., et al., *Normative data of knee joint motion and ground reaction forces in adult level walking.* Journal of Biomechanics, 1983. **16**(3): p. 219-33.

2. Inman, V.T., *Human locomotion.* Canadian Medical Association Journal, 1966. **94**(20): p. 1047-54.

3. Seedhom, B. and N. Wallbridge, *Walking activities and wear of prostheses.* Annals of Rheumatic Diseases, 1985. **44**(12): p. 838-843.

4. Schmalzried, T., *Quantitative assessment of walking activity after total hip or knee replacement.* Journal of Bone & Joint Surgery - American Volume, 1998. **80**(1): p. 54-59.

5. Woo, S.L.-Y., et al., *Biomechanical considerations of joint function*, in *Osteoarthritis*, Moskowitz, Editor. 2001, W.B. Saunders. p. 145-169.

6. Livesay, G.A., et al., *Evaluation of the effect of joint constraints on the in situ force distribution in the anterior cruciate ligament.* Journal of Orthopaedic Research, 1997. **15**(2): p. 278-84.

7. Wilson, D.R., et al., *The components of passive knee movement are coupled to flexion angle.* Journal of Biomechanics, 2000. **33**(4): p. 465-73.

8. Lafortune, M.A., et al., *Three-dimensional kinematics of the human knee during walking.* Journal of Biomechanics, 1992. **25**(4): p. 347-57.

9. Allen, C.R., et al., *The importance of the medial meniscus in the ACL-deficient knee.* Journal of Biomechanics, 2000. **32**: p. 1147-1152.

10. Papageorgiou, C.D., et al., *The biomechanical interdependence between the ACL replacement graft and the medial meniscus.* American Journal of Sports Medicine, 2001. **29**(2): p. 226-231.

11. Ma, C.B., et al., *Interaction between the ACL graft and MCL in a combined ACL+MCL knee injury using a goat model.* Acta Orthopaedica Scandinavica, 2000. **71**(4): p. 387-393.

12. Kanamori, A., et al., *The in situ forces in the medial and lateral structures of the intact and ACL deficient knee.* Journal of Orthopaedic Science, 2000. **5**(6): p. 567-571.

13. Wilson, D.R., J.D. Feikes, and J.J. O'Connor, *Ligaments and articular contact guide passive knee flexion.* Journal of Biomechanics, 1998. **31**(12): p. 1127-36.

14. Davis, M., W. Ettinger, and J. Neuhaus, *The association of knee injury and obesity with unilateral and bilateral osteoarthritis.* American Journal of Epidemiology, 1989. **130**: p. 278-288.

15. Ettinger, W., et al., *Long-term physical functioning in persons with knee osteoarthritis from NHANES I: effects of co-morbid medical conditions.* Journal of Clinical Epidemiology, 1994. **47**: p. 809-815.

16. Davis, M., et al., *Knee osteoarthritis and physical functioning: Evidence from the NHANES I Epidemiologic Followup Study.* J Rheumatology, 1991. **18**: p. 591-598.

17. Rejeski, W., T. Craven, and W. Ettinger, *Self-efficacy and pain in disability with osteoarthritis of the knee.* Journal of Gerontology B: Psychological Science, 1996. **51**: p. 24-29.

18. Miyasaka, K., *The incidence of knee ligament injuries in the general population.* American Journal of Knee Surgery, 1991. **4**(1): p. 3-8.

19. Hewson, G.J., R. Mendini, and J. Wang, *Prophylactic knee bracing in college football.* American Journal of Sports Medicine, 1986. **14**: p. 262-266.

20. Feagin, J.J., K. Lambert, and R. Cunningham, *Considerations of the anterior cruciate ligament injury in skiing.* Clinical Orthopaedics, 1987. **216**: p. 13-18.

21. Deibert, M., *Skiing injuries in children, adolescents and adults.* Journal of Bone & Joint Surgery - American Volume, 1998. **80**(1): p. 25-32.

22. Kaplan, N., T.L. Wickiewicz, and R.F. Warren, *Primary surgical treatment of anterior cruciate ligament ruptures. A long-term follow-up study.* American Journal of Sports Medicine, 1990. **18**(4): p. 354-8.

23. Aglietti, P., *Long-term study of anterior cruciate ligament reconstruction for chronic instability using the central one-third patellar tendon and a lateral extraarticular tenodesis.* American Journal of Sports Medicine, 1992. **20**(1): p. 38-45.

24. Bach, B.R., *Arthroscopically assisted anterior cruciate ligament reconstruction using patellar tendon autograft: Five to nine-year follow-up evaluation.* American Journal of Sports Medicine, 1998. **26**(1): p. 20-29.

25. Fleming, B.C., et al., *The strain behavior of the anterior cruciate ligament during bicycling. An in vivo study.* American Journal of Sports Medicine, 1998. **26**(1): p. 109-18.

26. Fleming, B.C., et al., *The strain behavior of the anterior cruciate ligament during stair climbing: an in vivo study.* Arthroscopy, 1999. **15**(2): p. 185-91.

27. Fleming, B.C., et al., *An in vivo comparison of anterior tibial translation and strain in the anteromedial band of the anterior cruciate ligament.* Journal of Biomechanics, 1993. **26**(1): p. 51-8.

28. Beynnon, B.D., et al. *Measurement of anterior cruciate ligament strain during non-weight and weight bearing conditions*. in *XVII ISB Congress*. 1999. Calgary.

29. Andriacchi, T.P., et al., *A point cluster method for in vivo motion analysis: applied to a study of knee kinematics.* Journal of Biomechanical Engineering, 1998. **120**(6): p. 743-9.

30. Ishibashi, Y., et al., *The effect of anterior cruciate ligament graft fixation site at the tibia on knee stability: evaluation using a robotic testing system.* Arthroscopy, 1997. **13**(2): p. 177-82.

31. Stone, J., et al., *Assessment of PCL graft performance using robotics technology.* Journal of Sports Medicine, 1996. **24**(6): p. 824-828.

32. Kanamori, A., et al., *The forces in the anterior cruciate ligament and knee kinematics during a simulated pivot shift test: A human cadaveric study using robotic technology.* Arthroscopy, 2000. **16**(6): p. 633-9.

33. Hoher, J., et al., *Mechanical behavior of two hamstring graft constructs for reconstruction of the anterior cruciate ligament.* Journal of Orthopaedic Research, 2000. **18**(3): p. 456-61.

34. Debski, R.E., et al., *Effect of capsular injury on acromioclavicular joint mechanics.* Journal of Bone & Joint Surgery - American Volume, 2001. **83**(9): p. 1344-1351.

35. Gilbertson, L., T. Doehring, and J. Kang, *New methods to study lumbar spine biomechanics: delineation of in-vitro load-displacement characteristics using a robotics/UFS testing system with hybrid control.* Operative Techniques in Orthopaedics, 2000. **10**(4): p. 246-53.

36. Carlin, G.J., et al., *In-situ forces in the human posterior cruciate ligament in response to posterior tibial loading.* Journal of Biomedical Engineering, 1996. **24**: p. 193-197.

37. Takai, S., et al., *Determination of the in situ loads on the human anterior cruciate ligament.* Journal of Orthopaedic Research, 1993. **11**: p. 686-695.

38. Woo, S.L.-Y., et al., *Biomechanics of the ACL: Measurements of the in situ force in the ACL and the knee kinematics.* Knee, 1998. **5**: p. 267-288.

39. Rupert, M., et al., *Influence on sensor size on the accuracy of in-vivo ligament and tendon force measurements.* Journal of Biomechanical Engineering, 1998. **120**(12): p. 764-769.

40. Testud, L. and A. Latarjet, *Anatomia Humana*. 5a ed. ed. Vol. Vol 1: Osteologia y Artrologia. 1971, Milan, Italy.

41. Fung, Y.C., *Elasticity of soft tissues in simple elongation.* American Journal of Physiology, 1967. **213**(6): p. 1532-44.

42. Woo, S.L.-Y., et al., *Tensile properties of the human femur-anterior cruciate ligament-tibia complex. The effects of specimen age and orientation.* American Journal of Sports Medicine, 1991. **19**(3): p. 217-25.

43. Torzilli, P.A., R.L. Greenberg, and J. Insall, *An in vivo biomechanical evaluation of the anterior-posterior motion of the knee. Roentgenographic measurement technique, stress machine, and stable population.* Journal of Bone & Joint Surgery - American Volume, 1981. **63**(6): p. 960-968.

44. Andriacchi, T.P. and E.J. Alexander, *Studies of human locomotion: past, present and future.* J Biomech, 2000. **33**(10): p. 1217-24.

45. Yu, B., et al., *Reproducibility of the kinematics and kinetics of the lower extremity during normal stair-climbing.* Journal of Orthopaedic Research, 1997. **15**: p. 348-352.

46. Blankevoort, L., et al., *Articular contact in a three-dimensional model of the knee.* Journal of Biomechanics, 1991. **Vol. 24**(No. 11): p. 1019-1031.

47. Crowninshield, R., M.H. Pope, and R.J. Johnson, *An analytical model of the knee.* Journal of Biomechanics, 1976. **9**: p. 397-405.

48. Grood, E.S. and M.S. Hefzy, *An analytical technique for modeling knee joint stiffness - Part I: Ligamentous forces.* Journal of Biomechanical Engineering, 1982. **104**: p. 330-337.

49. Hirokawa, S. and R. Tsuruno, *Three-dimensional deformation and stress distribution in an analytical/computational model of the anterior cruciate ligament.* Journal of Biomechanics, 2000. **33**(9): p. 1069-77.

50. Li, G., et al., *A validated three-dimensional computational model of a human knee joint.* Journal of Biomechanical Engineering, 1999. **121**(6): p. 657-62.

51. Markolf, K.L., et al., *Combined knee loading states that generate high anterior cruciate ligament forces.* Journal of Orthopaedic Research, 1995. **13**(6): p. 930-5.

52. Markolf, K.L., et al., *Direct measurement of resultant forces in the anterior cruciate ligament. An in vitro study performed with a new experimental technique.* Journal of Bone & Joint Surgery - American Volume, 1990. **72**(4): p. 557-67.

53. Markolf, K.L., D.C. Wascher, and G.A. Finerman, *Direct in vitro measurement of forces in the cruciate ligaments. Part II: The effect of section of the posterolateral structures.* Journal of Bone & Joint Surgery - American Volume, 1993. **75**(3): p. 387-94.

54. Fujie, H., et al., *The use of robotics technology to study human joint kinematics: a new methodology.* Journal of Biomechanical Engineering, 1993. **115**(3): p. 211-7.

55. Sakane, M., et al., *In situ forces in the anterior cruciate ligament and its bundles in response to anterior tibial loads.* Journal of Orthopaedic Research, 1997. **15**(2): p. 285-93.

56. Chao, E.Y., *Justification of triaxial goniometer for the measurement of joint rotation.* Journal of Biomechanics, 1980. **13**(12): p. 989-1006.

57. Grood, E.S. and W.J. Suntay, *A joint coordinate system for the clinical description of three- dimensional motions: application to the knee.* Journal of Biomechanical Engineering, 1983. **105**(2): p. 136-44.

58. Fujie, H., et al., *Forces and moments in six-DOF at the human knee joint: mathematical description for control.* Journal of Biomechanics, 1996. **29**(12): p. 1577-85.

59. Beck, K., *Test-driven development*. 2002: Addison-Wesley.

60. Link, J. and P. Frolich, *Unit testing in Java: How to test drive the code*. 2003: Morgan Kaufmann.

61. Blankevoort, L., R. Huiskes, and A.D. Lange, *The envelope of passive knee joint motion.* Journal of Biomechanics, 1988. **21**(9): p. 705-20.

62. Li, G., et al., *The importance of quadriceps and hamstring muscle loading on knee kinematics and in situ forces in the ACL.* Journal of Biomechanics, 1999. **32**(4): p. 395-400.

63. Blacharski, P.A. and J.H. Somerset, *A three-dimensional study of the kinematics of the human knee.* Journal of Biomechanics, 1975. **8**: p. 375-384.

64. Lucchetti, L., et al., *Skin movement artefact assessment and compensation in the estimation of knee-joint kinematics.* Journal of Biomechanics, 1998. **31**(11): p. 977-84.

65. Sheehan, F.T., F.E. Zajac, and J.E. Drace, *Using cine phase contrast magnetic resonance imaging to non-invasively study in vivo knee dynamics.* Journal of Biomechanics, 1998. **31**(1): p. 21-6.

66. Ateshian, G., et al., *Finite deformation biphasic material properties of bovine articular cartilage from confined compression experiments.* Journal of Biomechanics, 1997. **30**(11-12): p. 1157-1164.

67. Suh, J., Z. Li, and S. Woo, *Dynamic behavior of a biphasic cartilage model under cyclic compressive loading.* Journal of Biomechanics, 1995. **28**(4): p. 357-364.

68. Lewis, J., W. Lew, and J. Schmidt, *A note on the application and evaluation of the buckle transducer for knee ligament force measurement.* Journal of Biomechanical Engineering, 1982. **104**: p. 125-128.

69. Holden, J., et al., *In vivo forces in the anterior cruciate ligament: Direct measurements during walking and trotting in a quadruped.* Journal of Biomechanics, 1994. **27**: p. 517-526.

70. Kennedy, J., R. Hawkins, and R. Willis, *Strain gauge analysis of knee ligaments.* Clinical Orthopaedics and Related Research, 1977. **129**: p. 225-229.

71. Bach, J. and M. Hull, *Strain inhomogeneity in the anterior cruciate ligament under application of external and muscular loads.* Journal of Biomechanical Engineering, 1998. **120**: p. 497-503.

72. Stroustrup, B., *The C++ programming language*. 3rd ed. 1997, Reading, MA: Addison Wesley.

73. Rector, B.E. and J.M. Newcomer, *Win32 Programming*. Addison-Wesley Advanced WIndows Series, ed. A.R. Feuer. 1997, Reading, MA: Addison Wesley Longman.

74. Josuttis, N.M., *The C++ standard library: a tutorial and handbook*. 1999, Reading, MA: Addison-Wesley.

75. Craig, J.J., *Spatial descriptions and transformations*, in *Introduction to robotics : mechanics and control*. 1986, Addison-Wesley: Reading, MA. p. 15-59.

76. Paul, R.P., *Solving kinematic equations*, in *Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators*. 1981, MIT Press: Cambridge, MA. p. 65-84.

77. Paul, R.P., *Differential Relationships*, in *Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators*. 1981, MIT Press: Cambridge, MA. p. 85-118.

78. Greenwood, D.T., *Principles of Dynamics*. Second Edition ed. 1988, Englewood Cliffs, NJ: Prentice Hall.

79. Rudy, T.W., et al., *The effect of the point of application of anterior tibial loads on human knee kinematics.* Journal of Biomechanics, 2000. **32**: p. 1147-1152.

80. Gamma, E., et al., *Design Patterns: Elements of reusable object-oriented software*. Addison-Wesley professional computing series. 1995, Reading, MA: Addison-Wesley.

81. Blaszczak, M., *The Document/View architecture*, in *Professional MFC with Visual C++ 6*. 1999, Wrox Press Ltd.: Birmingham, U.K. p. 155-237.

82. Rector, B. and C. Sells, *ATL Internals*. Object Technology Series, ed. B.J. Rumbaugh. 1999, Reading, MA: Addison-Wesley.

83. Grimes, R., *Professional ATL COM programming*. 1998: Wrox.

84. Blaszczak, M., *Writing multithreaded appllications with MFC*, in *Professional MFC with Visual C++ 6*. 1999, Wrox Press Ltd.: Birmingham, U.K. p. 595-655.

85. Wright, R.S. and M. Sweet, *OpenGL SuperBible*. 2nd Ed ed. 1999, Indianapolis, IN: Waite Group.

86. Garlan, D. and M. Shaw, *An introduction to sotfware architecture*, in *Advances in Software Engineering and Knowledge Engineering*, V.A.a.G. Tortora, Editor. 1993: Singapore. p. 1-39.

87. Shaw, M., *Patterns for Software Architectures*, in *Pattern Languages of Program Design: First Annual Conference on the Pattern Languages of Programming*, J.C.a.D. Schmidt, Editor. 1994, Addison-Wesley. p. 453-462.

88. Lafore, R., *Objects and classes*, in *Object-Oriented Programming in C++*. 1995, Waite Group Press: Corte Madera, CA.

89. Meyers, S., *Effective C++: 50 Specific ways to improve your programs and designs*. 2nd Edition ed. 1997, Reading, MA: Addison-Wesley.

90. Lakos, J., *Large-scale C++ software design*. 1996, Reading, MA: Addison-Wesley.

91. Woo, S.L.-Y., G.A. Johnson, and B.A. Smith, *Mathematical modeling of ligaments and tendons.* Journal of Biomechanical Engineering, 1993. **115**: p. 462-473.