

Robotica

<http://journals.cambridge.org/ROB>

Additional services for **Robotica**:

Email alerts: [Click here](#)

Subscriptions: [Click here](#)

Commercial reprints: [Click here](#)

Terms of use : [Click here](#)



Articular human joint modelling

Ibrahim I. Esat and Neviman Ozada

Robotica / Volume 28 / Special Issue 02 / March 2010, pp 321 - 339
DOI: 10.1017/S0263574709990592, Published online: 07 December 2009

Link to this article: http://journals.cambridge.org/abstract_S0263574709990592

How to cite this article:

Ibrahim I. Esat and Neviman Ozada (2010). Articular human joint modelling. Robotica, 28, pp 321-339 doi:10.1017/S0263574709990592

Request Permissions : [Click here](#)

Articular human joint modelling

Ibrahim I. Esat and Neviman Ozada*

Department of Mechanical Engineering, Brunel University West London, Uxbridge, London, UK

(Received in Final Form: October 27, 2009. First published online: December 7, 2009)

SUMMARY

The work reported in this paper encapsulates the theories and algorithms developed to drive the core analysis modules of the software which has been developed to model a musculoskeletal structure of anatomic joints. Due to local bone surface and contact geometry based joint kinematics, newly developed algorithms make the proposed modeller different from currently available modellers. There are many modellers that are capable of modelling gross human body motion. Nevertheless, none of the available modellers offer complete elements of joint modelling. It appears that joint modelling is an extension of their core analysis capability, which, in every case, appears to be musculoskeletal motion dynamics. It is felt that an analysis framework that is focused on human joints would have significant benefit and potential to be used in many orthopaedic applications. The local mobility of joints has a significant influence in human motion analysis, in understanding of joint loading, tissue behaviour and contact forces. However, in order to develop a bone surface based joint modeller, there are a number of major problems, from tissue idealizations to surface geometry discretization and non-linear motion analysis. This paper presents the following: (a) The physical deformation of biological tissues as linear or non-linear viscoelastic deformation, based on spring-dashpot elements. (b) The linear dynamic multibody modelling, where the linear formulation is established for small motions and is particularly useful for calculating the equilibrium position of the joint. This model can also be used for finding small motion behaviour or loading under static conditions. It also has the potential of quantifying the joint laxity. (c) The non-linear dynamic multibody modelling, where a non-matrix and algorithmic formulation is presented. The approach allows handling complex material and geometrical nonlinearity easily. (d) Shortest path algorithms for calculating soft tissue line of action geometries. The developed algorithms are based on calculating minimum 'surface mass' and 'surface covariance'. An improved version of the 'surface covariance' algorithm is described as 'residual covariance'. The resulting path is used to establish the direction of forces and moments acting on joints. This information is needed for linear or non-linear treatment of the joint motion. (e) The final contribution of the paper is the treatment of the collision. In the virtual world, the difficulty in analysing bodies in motion arises due to body interpenetrations. The collision algorithm proposed in the paper involves finding the shortest projected ray from one body to the other. The projection

of the body is determined by the resultant forces acting on it due to soft tissue connections under tension. This enables the calculation of collision condition of non-convex objects accurately. After the initial collision detection, the analysis involves attaching special springs (stiffness only normal to the surfaces) at the 'potentially colliding points' and motion of bodies is recalculated. The collision algorithm incorporates the rotation as well as translation. The algorithm continues until the joint equilibrium is achieved. Finally, the results obtained based on the software are compared with experimental results obtained using cadaveric joints.

KEYWORDS: Software; Joint Modelling; 6DOF; Tissue wrapping; forced contact based articulation; unilateral; bilateral; constraints

1. Introduction

Multibody modelling is essential in various research areas such as robotics, biomechanics, ergonomics and general engineering. In terms of biomechanics it is a challenge to understand the kinematics and dynamics of multibody musculoskeletal systems. Thus, in kinematics- and dynamics-based modelling of musculoskeletal structures, kinematic joint modelling, deformable tissue (muscle, ligament and cartilage) modelling, and finding tissue line of actions, contact dynamics and collision detection and response techniques are required. From an anatomic point of view, real human joints are kinematically unconstrained joints that allow 6 degrees of freedom (DOF) between articulating bone surfaces. However, many industrially available analysis software treat the joints as standard mechanism joints, such as spherical and revolute joints. In real life, constraining factors that restrict joint movement are strong ligaments, tendons and muscles that surround joints tightly. The geometric cavity of bone surfaces can only contribute to joint stability along with surrounded tissues. Constructing unconstrained joint models with the interaction of surrounded tissues would possibly answer many unanswered questions relevant to joints such as the biomechanics of joint injuries and abnormal functionality. For example, when loads at diarthrodial joints exceed the strength level, tendons and ligaments are stressed beyond their capacity, which results in tissue failure. Moreover, osteoarthritis which is reported to be the most common disease in elderly adults is the main reason of pain and abnormal joint functionality due to ligament and/or cartilage defect.¹ However, understanding of such biomechanical behaviour is impossible by treating joints as constrained. Unconstrained joint modelling is the key

* Corresponding author. E-mail: neriman.ozada@brunel.ac.uk

for the necessary understanding for medical treatment, rehabilitation⁸⁹ and repairing and restoring of damaged tissue and joints. Computationally developed human body models that are used to study such biomechanical concepts have many advantages in clinical applications. Particularly, quantitative evaluation of surgical procedures along with accurate virtual models has become a valuable tool in medical applications.^{2,3} Through the development of three-dimensional (3D) computer graphics and computer hardware capacity, many commercial and non-commercial software packages have been developed for multibody analysis and simulations. For more than three decades a variety of virtual human models have been developed^{4–7} to analyse human joint configurations and movements and to mimic realistic human motions in the virtual world. In order to model anatomic joints, some fundamental modelling concepts such as kinematics and dynamics including of rigid and deformable biological tissues (bone, muscle, ligament and cartilage) as well as contact and collision have to be handled. In order to create model-based musculoskeletal structures, 3D solid data are needed to construct body meshes. Most commonly, computed tomography (CT), magnetic resonance imaging (MRI) and digitizer devices are used to extract precise anatomical data.^{8,9} Through the development of 3D increased computer graphics and computer hardware capacity, researchers are more capable in building graphics-based models for complex human, animal, robotic and mechanic multibody systems. Computationally developed human body models are broadly used to study the concept of joint diseases such as arthritis, joint injury mechanisms, induced accident musculoskeletal disorders, surgical procedures in the virtual world. In the field of orthopaedics, it is essential to understand normal and abnormal joint mechanics as well as altered joint mechanics after reconstruction surgery. Thus, many experimental diagnostic devices have been marketed such as KT-1000, KT-2000 arthrometers (MEDmetric Corporation, CA, USA), Rolimeter and several joint laxity measuring devices.^{10,11} However, the reliability and repeatability of these techniques are always questioned due to the skin movement around joints.^{12,13} Recent availability of realistic 3D virtual human models offers a great deal of advantage, as researchers and clinicians can use virtual models for diagnosing clinical joint problems and for designing and testing implants and instrumentations. Along with virtual models, a number of surgical devices like haptic and force feedback devices and surgical navigation systems have been developed to facilitate real-time dynamic surgery simulation.¹⁴

2. Literature Review

2.1. Articulated multibody modelling

Many pioneering researchers have studied dynamic joint modelling for upper limb^{15–17} and lower limb extremities.^{18–22} Furthermore, graphics-based kinematic and dynamic musculoskeletal modelling techniques have been applied through commercial multibody dynamics software such as Anybody,²³ SIMM^{2,24,25} (software for interactive musculoskeletal modelling; Musculographics Inc., IL,

USA), Lifemod/BodySIM²⁶ and VIMS (virtual interactive musculoskeletal system).²⁷ In the early 1970s the ADAMS²⁸ (automatic dynamic analysis of mechanical systems) and DADS (dynamic analysis and design software) software had been developed for general mechanical multibody design and simulations. Also, the Lifemod software has been commercialized as a plug-in to ADAMS for advanced dynamic modelling of musculoskeletal motion. SD/Fast (Symbolic Dynamics, Mountainview, CA, USA) is also a well-known dynamic engine used by the SIMM software. The SIMM is an interactive computer graphics package that has been developed to model musculoskeletal bodies to analyse neurological, orthopaedic and surgical disorders. Chao *et al.*²⁷ developed the VIMS software to understand complex musculoskeletal disorders. The VIMS software is widely used in the planning of total joint replacement, functional joint simulation and for rehabilitation purposes. The extended capability of the software has been utilized by clinicians and researchers to specifically study the shoulder rhythms,²⁹ joint contact stresses and external fixator effects. More recently, a graphics-based software called musculoskeletal modelling in simulink (MMS) has been developed by Davoodi and Loeb.³⁰ The software is capable of modelling and controlling musculoskeletal movements and providing comprehensive framework for virtual prototyping of prosthesis. Similarly, for the purpose of the development of rehabilitation devices in orthopaedics and ergonomics and in analysing musculoskeletal disorders, the Anybody software^{23,31} can be an appropriate choice. Through the study of currently available musculoskeletal modelling software packages, it has been found that the Lifemod software is the most capable and the closest to our proposed modelling software, which used validating the proposed modeller.

2.2. Modelling of deformation

The modelling of deformation is commonly performed based on physical and geometrical methods. In surgical simulations, the deformation modelling of biological tissues³² is essential. In the film and game industries, hair, cloth, skin, tissue, face and body deformations with the environmental interactions are extensively performed. In terms of physical deformation, many engineering materials and most of the biological tissues (passive skeletal muscle, tendon, ligament and cartilage) are represented by the combination of Newtonian viscous fluids and Hookean elastic solid constitutive equations. Due to both elastic and viscous properties of biological tissues, they are generally modelled as viscoelastic materials. The most commonly used linear viscoelastic material models are the Maxwell model, the Kelvin–Voigt model and the Burger model. In addition to the viscoelastic modelling of biological tissues, biphasic, poroelastic and triphasic material models are also widely used.^{33,34} Moreover, under dynamic (periodic) and static (time-invariant) loads, elastic spring and viscous damper non-linearities exist in many mechanical and biological system behaviours. The material non-linearity implies that the stress–strain or force–deflection relationship is not linear. Probably the non-linearity originates from intrinsic material properties at cellular (for biologic materials) and

molecular levels and geometric effects. Existing models and theories to predict non-linear viscoelastic behaviour have been comprehensively surveyed by Drapaca *et al.*³⁵ One of the qualitative studies was provided by Viidik and Ekholm³⁶ for understanding the mechanical behaviour of collagen-fibred tissues along with non-linear spring-based tissue modelling. Fung³⁷ suggested that for small deformations in soft tissues the linear viscoelastic models are sufficient; however, for finite deformations the non-linear viscoelastic behaviour has to be taken into account. In addition to this, the first continuum mechanics-based quasi-linear viscoelastic model theory was developed by Fung.³⁷ Nevertheless, Mooney-Rivlin and Ogden models³⁸ are widely used as strain energy-based material models that predict stress-strain behaviour of incompressible non-linear viscoelastic, elastic and hyperelastic materials (rubber, soft tissue, arterial walls and brain tissue). Further, non-linear and quasi-linear models have been provided by Funk *et al.*³⁹ to predict biomechanical behaviours of injured ankle ligaments. Comprehensive information about the inhomogeneity, incompressibility, non-linearity and quasi-linear and linear viscoelasticity and their applicability in soft tissue modelling are provided by Cowin and Doty.⁴⁰ A detailed review of finite element methods, constitutive modelling methods and experimental methods to characterize and model the ligaments has been given by Weiss and Gardiner.⁴¹ Skeletal muscles have also been represented as linear and non-linear viscoelastic materials in tensile and compression loading experiments.⁴² Nevertheless, in addition to passive material modelling for musculoskeletal tissues, force generation has to be taken into account for the modelling of complete dynamic muscle activities. The most popular Hill's muscle model⁴³ was introduced by A. Hill in 1938. The model is composed of three elements: elastic element (SE) in series with active contractile element (CE), both of which are in parallel with passive element (PE). Thus, the most commonly used muscle model technique is Hill's muscle model type, which has also been used by many authors such as Blum *et al.*⁴⁴ and Winters and Wang.⁴⁵ Furthermore, muscle contraction has been investigated and modelled by Zajac *et al.*,⁴⁶ which is more sophisticated than Hill's approach concerning muscle and tendon architecture. Another sophisticated model has been presented by Huxley,⁴⁷ which is built on the cross-bridge theory, involving thermodynamic energy transfer at the molecular level of muscle contractions. An important aspect of this type of model is accounting mechanical properties such as permeability of elements in cross-bridge activation at the molecular level. The intention of presenting the deformable modelling literature is to highlight the non-linear material behaviours and the corresponding modelling applications. In many comprehensive material modelling-based studies, the deformation modelling of biological tissues has been taken as a focused application that provided validated results. It is important to notice that more sophisticated deformation models may provide more accurate results and have better concordance with experimental outputs. In spite of these advantages, more sophisticated models are more complex as more data and longer computational processes are required, which decrease the efficiency of the analysis. As well as permeability, non-

linearity and rate-dependent properties of cartilage tissue need to be incorporated for an accurate analysis of joint contact with realistic ligament, tendon and muscle models. However, the combination of all required realistic and fundamental properties in tissue modelling has not been presented in multibody modelling software packages, which lead to missing realistic tissue interactions in anatomic fidelity. Thus, soft tissue deformation modelling concerning fundamental material properties of soft tissue and muscles has been proposed with the newly developed multibody modeller to provide realistic tissue interactions during joint movements through the guidance of the newly developed tissue line of action and moment arm model.

2.3. Line of action modelling

In addition to modelling of biological tissue deformation, description of moment arms and line of actions are important factors in the modelling of musculoskeletal structures. Transferring moment arm knowledge into medical treatments, surgical applications and pre- and post-processing of surgical treatment reveals significant importance. For example, ligament and muscle-tendon moment arm knowledge yields to understand accurate loading and the range of joint articulations and helps surgeons to plan for tissue reconstructive surgery or tissue transfer surgery. Without an accurate line of action modelling, post-operative limbs can exhibit abnormal joint articulations, due to altered tissue length which can result in wrong moment arms, and inconvenient interaction with surrounded tissues and bone. Thus, it is essential to understand the concept of tissue deflection and change in moment arm relative to the performed task.⁴⁸ Therefore, experimental and geometrical tissue length measurements, moment arm calculations as well as computational modelling techniques have been widely developed.^{49,50} The obtained tissue origin and insertion points and moment arms provide kinetic insight during joint articulation. In considering the moment arm estimation, kinematic modelling reveals significant importance where the centre of joint rotation is required in order to describe moments of joints. Therefore, the definition of the joint centre of movement is based on the joint kinematic model. In bilateral joint constraint-based kinematic joint models, the centre of movement is fixed and remains the same during joint articulations, which are also referred to as idealized kinematic joints. In spite of bilateral joint constraints, unilateral contact constraints are utilized where the joint centre of rotation is not fixed but the mobility is constrained when contact is active and reveals more complex analysis. Most researchers applied bilateral joint constraint-based modelling along with the modelling of tissue line of actions and moment arms. To our knowledge there is only one sophisticated method^{51,52} that presents geometric contact-based kinematic joint modelling which is utilized to estimate ligament length and moment arms according to performed task. In this particular application, the interest was the modelling of ligament line of actions for distal radioulnar joint (DRUJ) without concerning tissue deformation. Rather, it is essential to describe how moment arms and tissue lengths are changing along with muscle contractions and joint articulations. The most commonly

performed moment arm measurement technique was first presented by An *et al.*⁵⁰ in which the moment arm estimation is performed with respect to change in tissue displacement (excursion) at joint articulation relative to change in angle. This procedure is widely used to predict moment arms for finger, knee,^{53,54} shoulder and elbow joints.⁵⁵ In such applications, it is not required to define the kinematic joint centre of rotation explicitly. However, if the moment arm estimation is proposed to link to the dynamic joint motion analysis, the joint kinematic model is required. In considering geometric modelling of tissue line of actions and moment arms, the straight line model of tissue paths is very commonly utilized, which is composed of only a fixed origin and insertion points. The straight line method with simplified deformation and wrapping facilities is used in the Lifemod software, where tissue deflection and wrapping is allowed in the range of the straight path rather than following realistic tissue paths. Thus, musculoskeletal tissues such as deltoid muscles do not appropriately wrap within a curvature that highly effects force vector representation. Other most commonly used geometric models is the centroid line model^{56,57} in which the tissue path passes from the centre of the tissue as a curved line via cross-sectional centroid points between the origin and the insertion attachments. In the centroid line model the deflection of the tissue path is simulated on the basis of centroid points rather than on a straight line. In considering some pioneering musculoskeletal models, Murray *et al.*⁵⁸ presented a 'specimen-specific' kinematic model of the elbow joint along with surrounded musculoskeletal tissues and the data have been extracted from MR images for analysing peak moment arms of elbow muscles. The scope was to determine the muscle moment arms of the elbow with respect to flexion and extension movements and clarify the functional capacity of muscles and peak moment arms. The elbow joint was modelled as a hinge joint where the centre of rotation axis passes through the centres of the capitulum and trochlear grooves. Muscle lines of actions were represented as a set of line segments with a series of points passing through the muscle centroid points. Moment arm estimation⁵⁶ was based on the derivation of muscle-tendon excursion with respect to the change in joint angle. More comprehensive computational models of musculoskeletal bodies have appeared with more sophisticated tissue wrapping schemes. In the modelling of tissue wrapping, some problems arise with representing tissue-constraining geometries. Apart from bone-tissue attachments, tissues interact with each other and wrap around multiple anatomic structures. In order to represent geometric surfaces of these structures, simple geometric shapes, such as the sphere and cylinder, are used. These geometric shapes represent constraining anatomical surfaces (i.e. bone and tissue), which are also referred to as obstacles. More recently, attention has been paid to tissue shortest path wrapping modelling on simple geometric obstacles and anatomic surfaces.⁵⁹ The preliminary study provided by Garner and Pandey⁶⁰ introduced the obstacle-set method in which the tissue shortest paths wrap on separated single obstacles (cylinder, sphere) with a series of straight and curved line segments on the human upper limb. These segments are joined together via fixed and movable dynamic points.⁶¹

More comprehensively, Charlton and Johnson⁶² discussed the analytical and numerical solutions of path wrapping on cylinder, sphere and complex surfaces. Furthermore, Gao *et al.*⁶³ proposed to model tissues as stretched strings wrapped around anatomic bone geometries, tissue surfaces and simple geometric shapes rather than wrapping only simple shape obstacles. In a recently performed study, Audenaert *et al.*⁶⁴ proposed an optimal wrapping method on sphere and cylinder considering all possible shortest paths. The global optimal method was applied to construct the shoulder deltoid muscle wrapping on sphere and cylinder and was compared with the popular obstacle-set method.⁶⁰ In the comparison of wrapping on sphere, conjunction has been achieved in describing muscle length, moment arms and wrapping radii between two methods. Additionally, a novel multi-object wrapping method has been developed by Marsden and Swailes⁶⁵ to model tissue wrapping for multiple obstacles by employing the theory of geodesics. In the analytic geodesic method, the wrapping path is described as part of an arc and a helix (referred to as geodesic) wrapped around a sphere and cylinder. The path is described as spring-like anodes interconnected via dynamic centroid points. In this method, friction is neglected and standard energy minimization approach is applied. In this field, the modelling and experimental approach has been presented by Gatti *et al.*,⁶⁶ who proposed to compare predicted and measured moment arms for shoulder rotator cuff muscles during arm elevations. As a conclusion, none of the multibody dynamics packages and physics engines are designed to deal with local joint dynamics along with precise geometric description of tissue line of actions, bone surface and contact surface. What is provided in this paper is a platform (although not fully validated) that will analyse predominantly joint kinematics and dynamics of neighbouring bones with the interaction of soft tissues wrapped around anatomic tissue and bone surfaces. Another problem we have found with the Lifemod software is the way in which it deals with muscle wrapping. What is provided is not a general-purpose muscle-wrapping algorithm, but a user-guided muscle path description that can work only if the path remains more or less unchanged during the motion.

2.4. Contact and collision evaluation

Before applying contact dynamics the potentially contacting points or vertices are checked with collision detection methods in virtual environments. The collision detection depends on the simulation, object geometry as well as the approximation of physics to represent body state in collision tests. In general, objects are geometrically represented as a collection of regular primitive shapes (cube, sphere, cylinder, etc.) consisting of primitive elements (patches, triangles). In collision detection, these primitive elements are object features that are checked between each object pair. Respectively, relative object configurations, detection of intersections, distance calculation between objects and boundaries, computation of separation distance between colliding objects, penetration depth calculation and many similar concepts are referred to as proximity queries.⁶⁷ For the sake of simplicity, in most of the multibody dynamic

simulations, polygonal surface approximations are used to simplify the object geometry for more efficient distance calculations. Precise calculations and accurate detections in collision detection need sophisticated algorithms, which are neither easy to implement nor efficient. Most of the developed collision detection algorithms are originally designed based on proximity queries for convex objects such as searching close features of convex objects, separating axis testing between many polytopes⁶⁸ and computing translational distances and penetration depths between convex object pairs.⁶⁹ Additionally, hierarchical boundaries, spatial partitioning and GPU (graphical processing unit)-assisted collision detection algorithms are driven for faster proximity query applications. However, collision detection for non-convex objects has rarely been taken into account because it requires the decomposition⁷⁰ of non-convex objects into convex objects, which is quite time-consuming. Various literature surveys on collision detection have been provided by several authors.^{71,72} The most commonly used collision detection libraries are RAPID,⁷³ I-Collide, V-Collide, Q-Collide, Voronoi-clip and SOLID. These libraries are widely reported in the UNC GAMMA Research Group.⁷⁴ RAPID is a practical library and has been widely implemented in many virtual applications and software packages such as ADAMS and Lifemod. Once the potential collision is detected, the contact dynamic analysis has to be applied by employing collision response models for contacting bodies.

Contact is a generic problem that can be assessed as a major problem of realistic physical simulations. The major focus in developing a method for contact problems is on the robustness and capability of describing multiple simultaneous contacts and impulsive contacts and the capability of employing resistive forces to avoid body penetrations. In contact and collision problems, the solution approach depends on the surface geometry, smoothness and roughness of contacting bodies, frictionless and frictional contact models, material types of bodies and the time of contact period such as resting or instantaneous contact. In multibody dynamics, the motion of interacting bodies can be subjected to bilateral joint constraints or unilateral contact constraints⁷⁵ with or without friction. Unilateral contact constraint basically means a constraint to prevent penetration between two bodies and differs from a bilateral joint constraint. On employing unilateral contact constraints, the equation of motion becomes more complex than on introducing bilateral kinematic joint constraints. Unilateral contact constraints are included into equations of motion as constraint equations only when they are active or remain in the proposed contact trajectory. In considering the unilateral contact constraints, there are two types of modelling approaches: regularized modelling and non-smooth modelling. These are employed to model contact dynamics with collision response. Regularized modelling expresses the contact phenomena in terms of spring attachment(s) between contacting bodies, which is evaluated only at the condition of contact and neglected if the distance between contacting bodies is not in the range of the contact trajectory. Specifically for rigid bodies with unilateral contact constraints, the regularized modelling solution is either based

on penalty methods or complementarity formalism-based methods.^{75–77} Thus, the procedure of penalty-based methods starts with the calculation of the penetration depths of rigid bodies at every time step and generates spring forces to prevent unrealistic penetration. Where these forces are created with respect to penetration depths. Even if there is a spring or spring-dashpot attachment between rigid bodies to prevent the penetration and calculate the contact forces, the contact condition is assumed as rigid contact. Unless the unilateral contact is described for deformable objects the contact only possesses local deformations in terms of spring deformations at contact points. However, due to the spring attachments between bodies, undesired oscillation and stiff system problems can occur, but these can be handled numerically. In addition to the RAPID collision detection library, the Lifemod software performs penalty-based contact methods, along with the Parasolid geometric modelling kernel, which is the component of UGS' Unigraphics Solution Inc. However, there is not much information about the contact dynamics available on Lifemod and ADAMS manuals. It is known that applying the unilateral contact constraints like penalty-based methods through employing spring attachments adds constraint equations into the system equations when unilateral contacts are active. There are many applications that have gone into contact modelling, such as Trinkle⁷⁸ developed a complementary-based method to tackle with the frictional contact between multi-rigid bodies which is capable of describing the equation of motion instantaneously with the geometric changes. Thus, the complementary problem yields to two unknowns such as acceleration and forces at contact. The method is then followed by discrete time integration (temporal discretization). Analytic, multi-rigid body contact formulation has also been introduced by Pang and Trinkle⁷⁹ to solve rigid contact problem concerning Coulomb friction through complementarity formulation. The aim was to determine instantaneous accelerations of the object, contact forces and contact interactions (rolling, sliding) due to applied forces. For the sake of simplicity, in program implementations, Signorini's boundary condition works very well in frictional contact problems and has been performed in haptic simulations.⁸⁰ In considering diarthrodial (synovial) joint motion, contact stresses and friction occur between bones. During joint movement, joint contact points move and contact stress is reduced along with naturally increased cartilage contact area. Articular (hyaline) cartilage is prominently found in diarthrodial (synovial) joints which covers bone contact surfaces and provides friction resistance between joint contact surfaces. Thus, an accurate contact model has to be driven with respect to anatomical surface geometry and constitutive material models and applying general and simply contact formulations such as Signorini's approach.⁸¹ Cartilage contact modelling has been proposed by many authors concerning its elastic and viscoelastic mechanical behaviour⁸² and surface geometry.⁸³ In the proposed contact and collision modelling, the collision detection algorithm has been designed to tackle with non-convex objects along with predefined potential colliding areas and simple culling. In addition, a new collision response method is proposed which is similar to the penalty-based

collision response method; however, through this method, constraint equations are not added into the dynamic system equations even if the contact is active. Thus the joint still possesses 6 DOF. The dynamic contact and multibody analysis are strongly performed based on the anatomic soft tissue constraints among their interactions without influencing the total mobility of the joint. Moreover, in addition to describing unconstrained joint contact dynamics, the change in geometric contact points on the volumetric deformable cartilage model without employing finite element analysis can be depicted.

3. Mathematical Formulation

The force-based, dynamic and surface geometry guided musculoskeletal joint modelling is the main contribution of the proposed multibody joint modelling. The generic joint model has been elaborated based on the linear and non-linear dynamic multibody modelling in Euler coordinates, constitutive Kelvin–Voigt viscoelastic tissue deformation model, simplified Hill’s muscle model type, developed tissue line of action and shortest path algorithms, deformable cartilage contact and developed collision detection and collision response algorithms. According to the proposed methodology, none of the existing kinematic and dynamic musculoskeletal joint modelling procedures follow the perspective of not simplifying anatomical joints into the idealized joints (planar, revolute and spherical).⁸⁴ Simply some theories from existing models (Kelvin–Voigt model and Hill’s model) have been adapted in order to establish the multibody joint model. Consequently, in the proposed novel joint model, some adapted models are combined with uniquely developed joint modelling theories and algorithms. Biological soft tissues (ligament and tendon) are regarded as spring or spring-dashpot elements where the deformation of the tissues is taken through their shortest paths. Developed effective algorithms are driven to define the shortest paths of tissues and moment arm calculations, which are then incorporated in dynamic equations of the joint. The other consideration is the definition of contact between two (or more) joint surfaces before assembling the equations of motion of a joint. As explained before, current mechanism modellers assume that the joints are of a particular type and position constraints imposed by joints are predetermined. This makes the formulation of the motion equations relatively simple. However, if the contact conditions between two surfaces are defined entirely by the surface geometry, then the problem exhibits a number of challenges. If the contact is rigid then one needs to have exact analytical formulae describing the contact surfaces and establishment of contact mobility. This is not a trivial task and in case of multiple contacts, the problem becomes even more complicated. In order to have an analysis tool that would be practical and useful for medical practitioners and orthopaedic device designers or in surgical navigation, one needs to base the bone surface definition data to what is currently available in the literature. Common surface data formats, such as VRML, are based on triangulations of the surface. Therefore, what is investigated in the analysis of joint motion and joint loads would involve accurate description of contact

without volumetric intersection of bodies. Given the fact that triangulation of surfaces will never be smooth or regular, the developed algorithm to accomplish this task should not violate volumetric integrity of bodies and at the same time should ensure that physical properties of contact (cartilage) is accurately defined.

3.1. Linear multibody modelling

In the proposed linear system, force–deflection relationship is linear and the system geometry remains unchanged due to motion. The force–deflection relationship depicts the material linearity or non-linearity. The change of forcing terms in multibody (internal or external) relates to geometrical linearity or non-linearity. In other words, if the system is geometrically linear it means that the geometry of bodies remains unaffected by motion. Gyroscopic effects are assumed to be small and negligible. These assumptions are acceptable for most of small motion problems. As for articular and skeletal motion, such formulation seems to be irrelevant; however, small motion analysis is needed for two reasons, firstly for calculating the initial equilibrium position of joints accurately and secondly (which is more important) to introduce the concept of joint stiffness (as a quantitative description of joint laxity). In Newton–Euler formulations, generalization of the equations of internal reactions and external forces are needed. The internal reactions due to damping and stiffness elements have to be expressed in a unified and structured fashion for formulation of the stiffness matrix (damping matrix structure is identical to stiffness matrix structure except that stiffness coefficients need to be replaced by damping coefficients). After establishing the stiffness/damping matrix, generalization of the equations of linear momentum (mass–acceleration equations) and angular momentum (inertia–turning moment equations) is required. It is assumed that deflections in the three principal axes of a spring remain orthogonal and orthogonality is preserved throughout the deflection cycle, as is already implied by geometrical linearity. In order to assemble the equations of motion, the internal forces acting on individual bodies due to their motion relative to each other are required. The assembly of equations is carried out taking into account body i and body j connected by spring k_p . Bodies are rigid and all the inertial properties are assumed to be known. The bodies i and j are shown in Fig. 1.

The motion of a rigid body is described by the translational and rotational motion of its centre of mass o_i given by \mathbf{x}_i and its rotation θ_i . In this description θ_i is an infinitesimally small rotation and it can therefore be treated as a vector. There is no such restriction on \mathbf{x}_i . In order to describe forces acting on this body, the location of force (p) is given by \mathbf{r}_i , which is measured relative to a local axes system, $o_i x_i y_i z_i$. Although the axes can be located at any position, for simplification of analysis it will be assumed that the point coincides with the centre of mass. It is also assumed that the axes system is fixed to the body and the body is rigid. The absolute displacement of p_i is measured relative to the local axis and is given by \mathbf{d}_i . Now assume that body j exists with similar descriptions of motion. All the parameters described for body i exist for body j . Therefore, in order to describe the body j , all that is needed is to replace suffix i by j . Now the small displacement

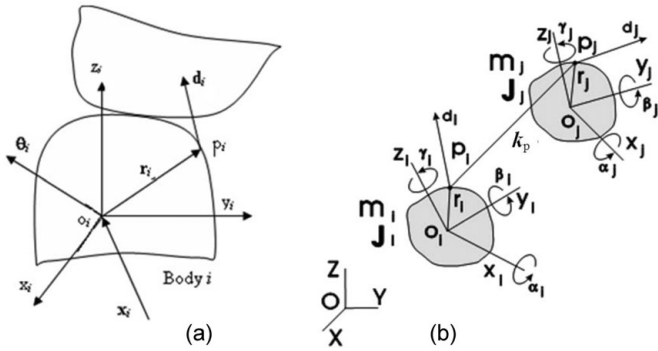


Fig. 1. (a) Bodies *i* and *j* in contact; (b) bodies *i* and *j* connected by spring k_p .

of \mathbf{d}_i and \mathbf{d}_j can be described as follows:

$$\begin{aligned} \mathbf{d}_i &= \mathbf{x}_i + \boldsymbol{\theta}_i \times \mathbf{r}_i, \\ \mathbf{d}_j &= \mathbf{x}_j + \boldsymbol{\theta}_j \times \mathbf{r}_j. \end{aligned} \tag{1}$$

\mathbf{d}_i can be expressed in matrix form as

$$\mathbf{d}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{ip} & -y_{ip} \\ 0 & 1 & 0 & -z_{ip} & 0 & x_{ip} \\ 0 & 0 & 1 & y_{ip} & -x_{ip} & 0 \end{bmatrix} \begin{Bmatrix} x_{io} \\ y_{io} \\ z_{io} \\ \theta_{ix} \\ \theta_{iy} \\ \theta_{iz} \end{Bmatrix}. \tag{2}$$

Replacing the suffix *i* with *j*, \mathbf{d}_j can also be represented in matrix form. Introducing the notations

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & z_{ip} & -y_{ip} \\ 0 & 1 & 0 & -z_{ip} & 0 & x_{ip} \\ 0 & 0 & 1 & y_{ip} & -x_{ip} & 0 \end{bmatrix} \tag{3}$$

and

$$\mathbf{u}_i = \begin{Bmatrix} x_{io} \\ y_{io} \\ z_{io} \\ \theta_{ix} \\ \theta_{iy} \\ \theta_{iz} \end{Bmatrix}, \tag{4}$$

Eq. (2) can be written as

$$\mathbf{d}_i = \mathbf{A}_i \mathbf{u}_i. \tag{5}$$

Forces and moments acting on body *i* at position *p* may be expressed as

$$\begin{Bmatrix} F_{ipx} \\ F_{ipy} \\ F_{ipz} \\ T_{ipx} \\ T_{ipy} \\ T_{ipz} \end{Bmatrix}, \text{ where } \begin{Bmatrix} T_{ipx} \\ T_{ipy} \\ T_{ipz} \end{Bmatrix} = \mathbf{r}_{ip} \times \mathbf{F}_{ip}. \tag{6}$$

Then Eq. (6) may be written in matrix form as

$$\begin{Bmatrix} T_{ipx} \\ T_{ipy} \\ T_{ipz} \end{Bmatrix} = \mathbf{r}_{ip} \times \mathbf{F}_{ip} = \begin{bmatrix} 0 & -z_{ip} & y_{ip} \\ z_{ip} & 0 & -x_{ip} \\ -y_{ip} & x_{ip} & 0 \end{bmatrix} \begin{Bmatrix} F_{ipx} \\ F_{ipy} \\ F_{ipz} \end{Bmatrix}. \tag{7}$$

Therefore,

$$\begin{Bmatrix} F_{ipx} \\ F_{ipy} \\ F_{ipz} \\ T_{ipx} \\ T_{ipy} \\ T_{ipz} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -z_{ip} & y_{ip} \\ z_{ip} & 0 & -x_{ip} \\ -y_{ip} & x_{ip} & 0 \end{bmatrix} \begin{Bmatrix} F_{ipx} \\ F_{ipy} \\ F_{ipz} \end{Bmatrix}. \tag{8}$$

Noting that the matrix in Eq. (8) is the transpose of the matrix introduced in Eq. (4), the force equation is written as

$$\mathbf{f}_{ip} = \mathbf{A}_i^T \mathbf{F}_{ip}. \tag{9}$$

Note that \mathbf{F}_{ip} is a force vector (acting at point *p*) described in the local axes frame. Normally, force is described in the global axes. This equation can be rewritten as

$$\mathbf{f}_{ip} = \mathbf{A}_i^T \mathbf{T}_i^T \mathbf{F}_{ip(3)}. \tag{10}$$

$\mathbf{F}_{ip(3)}$ is in the global axes frame of dimension 3. Now, pre-multiplying both sides of the equation with \mathbf{T}'_i , we have

$$\mathbf{F}_{ip} = \mathbf{T}'_i \mathbf{A}_i^T \mathbf{T}_i^T \mathbf{F}_{ip(3)}. \tag{11}$$

The left-hand side of the equation has dimension 6, where

$$\mathbf{T}'_i = \begin{bmatrix} \mathbf{T}_i & 0 \\ 0 & \mathbf{T}_i \end{bmatrix}. \tag{12}$$

Now, forces acting on body *i* are written as

$$\mathbf{F}_{ip(3)} = \mathbf{k}_p (\mathbf{T}_j \mathbf{d}_{jp} - \mathbf{T}_i \mathbf{d}_{ip}) \tag{13}$$

and

$$\mathbf{F}_{ip(3)} = \mathbf{k}_p (\mathbf{T}_j \mathbf{A}_{jp} \mathbf{u}_j - \mathbf{T}_i \mathbf{A}_{ip} \mathbf{u}_i). \tag{14}$$

\mathbf{u} vectors are still in the local frame and these need to be replaced by \mathbf{U} :

$$\mathbf{F}_{ip(3)} = \mathbf{k}_p (\mathbf{T}_j \mathbf{A}_{jp} \mathbf{T}_j'^T \mathbf{U}_j - \mathbf{T}_i \mathbf{A}_{ip} \mathbf{T}_i'^T \mathbf{U}_i). \tag{15}$$

In this case,

$$\mathbf{T}'_i = \begin{bmatrix} \mathbf{T}_i & 0 \\ 0 & \mathbf{T}_i \end{bmatrix}, \tag{16}$$

\mathbf{k}_p is described in the global axes frame and \mathbf{f}_{ip} is a force vector (all six forces) in the global axes frame.

$$\mathbf{M}_i \ddot{\mathbf{u}}_i = \sum_p \mathbf{f}_{ip}, \tag{17}$$

$$\mathbf{M}_i \mathbf{T}'_i{}^T \ddot{\mathbf{U}} = \sum_p \mathbf{f}_{ip}, \tag{18}$$

$$\mathbf{T}'_i \mathbf{M}_i \mathbf{T}'_i{}^T \ddot{\mathbf{U}} = \mathbf{T}'_i \sum_p \mathbf{f}_{ip}, \tag{19}$$

$$\mathbf{T}'_i \mathbf{M}_i \mathbf{T}'_i{}^T \ddot{\mathbf{U}} = \sum_p \mathbf{F}_{ip}. \tag{20}$$

Substituting forces from Eqs. (11) and (15), the equations of motion are written as

$$\begin{aligned} \mathbf{T}'_i \mathbf{M}_i \mathbf{T}'_i{}^T \ddot{\mathbf{U}}_i + \sum_p (\mathbf{T}'_i \mathbf{A}_i{}^T \mathbf{T}'_i{}^T \mathbf{k}_p \mathbf{T}_i \mathbf{A}_{ip} \mathbf{T}'_i{}^T) \mathbf{U}_i \\ - \sum_p (\mathbf{T}'_i \mathbf{A}_i{}^T \mathbf{T}'_i{}^T \mathbf{k}_p \mathbf{T}_j \mathbf{A}_{jp} \mathbf{T}'_j{}^T) \mathbf{U}_j = 0 \end{aligned} \tag{21}$$

and

$$\begin{aligned} \mathbf{T}'_j \mathbf{M}_j \mathbf{T}'_j{}^T \ddot{\mathbf{U}}_j - \sum_p (\mathbf{T}'_j \mathbf{A}_j{}^T \mathbf{T}'_j{}^T \mathbf{k}_p \mathbf{T}_i \mathbf{A}_{ip} \mathbf{T}'_i{}^T) \mathbf{U}_i \\ + \sum_p (\mathbf{T}'_j \mathbf{A}_j{}^T \mathbf{T}'_j{}^T \mathbf{k}_p \mathbf{T}_j \mathbf{A}_{jp} \mathbf{T}'_j{}^T) \mathbf{U}_j = 0. \end{aligned} \tag{22}$$

The final equations of motion described in matrix form are ready to be assembled in the global matrix for the overall system.

3.2. Non-linear modelling

3.2.1. Formulation for non-linear stiffness elements. In this section we propose to present the formulation of a general multibody system that exhibits non-linearity. If the stiffness characteristics of the spring are non-linear this can conveniently be expressed in terms of a force deflection curve. According to the assumption that is given above, the deflection in three orthogonal planes remains uncoupled. In other words, three force equations are described in terms of a series of spring parameters coupled with higher order terms of deflection in principal axes directions. No equation contains any variable from any other two directions. This representation scheme is generally used in industry and has been found to be satisfactory for many practical applications. Matrix representation of this is possible, since each order of non-linearity contains terms that are separated from each other:

$$\mathbf{f} = \mathbf{k}_1 \mathbf{x} + \mathbf{k}_2 \mathbf{x}^2 + \mathbf{k}_3 \mathbf{x}^3 + \dots + \mathbf{k}_n \mathbf{x}^n. \tag{23}$$

Although the first term \mathbf{x} is a vector, \mathbf{x}^2 and higher order terms are vectors containing terms such as $\mathbf{f} = (f_x \ f_y \ f_z) \mathbf{x}^n =$

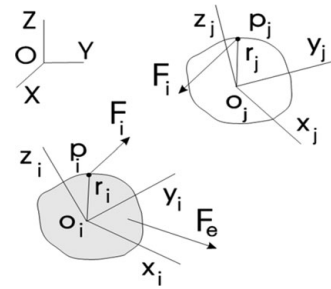


Fig. 2. Free body diagram of bodies *i* and *j*.

$(x^n \ y^n \ z^n)$. In order to facilitate the assembly of stiffness elements to the global matrix a typical conversion of the following type is needed:

$$\mathbf{F} = \mathbf{K}_1 \mathbf{X} + \mathbf{K}_2 \mathbf{X}^2 + \mathbf{K}_3 \mathbf{X}^3 + \dots + \mathbf{K}_n \mathbf{X}^n. \tag{24}$$

Unfortunately, such transformation is neither possible nor meaningful. This is one of the important reasons why non-linearity is generally formulated based on piecewise linearization. Although piecewise linearization ensures that tensorial operation is applicable and the formulations given for linear motion can be used, the system is now described by one of *S* number of stiffness combinations, where *S* is given by

$$S = \prod_{s=1}^n r_s,$$

where r_s is the number of linear segments on the *s*th spring. A single mass with four mounts and each with five linear segments, the formulation based on piecewise linearization would require $625 \ 6 \times 6$ stiffness matrices. Although in terms of memory allocation this may not be too unmanageable, during the simulation process at every time step the stiffness matrix will need to be reassembled by selecting the correct segments and of course this will slow the simulation.

3.2.2. A non-matrix-based motion analysis. This is based essentially on the free body diagrams, and the internal reactions explicitly appear in the equations of motion. Each body possesses 6 DOF and spring reactions (and other forces acting on it) are treated as external forces. A typical free body diagram of bodies *i* and *j* is shown in Fig. 2. Axes systems $ox_i y_i z_i$ and $ox_j y_j z_j$ are fixed to the bodies *i* and *j* respectively. The orientation of the body is measured in terms of the Euler angles. In this analysis, it is assumed that the local axes and the principal axes coincide. Spring force acting at a point on the body is due to relative displacements of the ends of the spring. In a system with bodies *i* and *j* this is determined due to motion of the attachment positions of the spring on body *i* relative to body *j*.

The equation of motion which is describing the motion of body *i* is given by

$$\mathbf{M}_i \ddot{\mathbf{x}}_i = \mathbf{F}, \tag{25}$$

$$\text{where } \mathbf{F} = F_e + F_i \tag{26}$$

In Eq. (26), suffixes ‘e’ and ‘i’ mean external and internal forces (not to be confused with the body index). \mathbf{F}_i can then

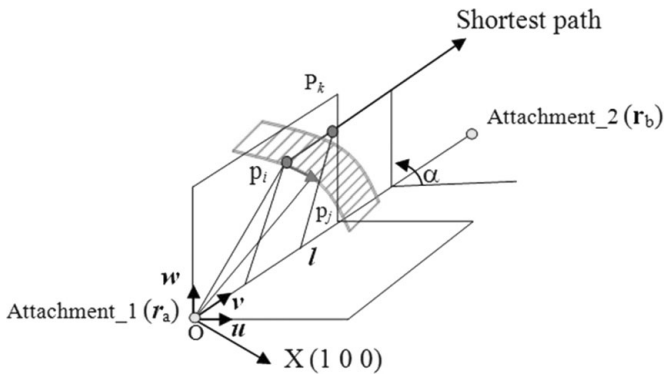


Fig. 3. Illustration of proposed shortest path between attachment points.

be expressed as

$$F_i = kx + k_2x^2 + k_3x^3 + \dots + k_nx^n. \quad (27)$$

The x is described as the displacement of the spring along its principal axes. This is important because it is assumed that experimental data exist for displacement of soft tissue along the main direction of stretch. When it was implemented and coded into the program, it was assumed that the lateral stiffnesses of tissues are small or negligible.

3.2.3. *Displacements of a point on a body.* Force acting on a rigid body at a point p due to springs and dampers is a function of the displacement and velocity of that point. This may be described by

$$r_p = R_i + r_i, \quad (28)$$

where R_i is the position vector of the centre of origin of the axes system fixed on the rigid body and r_i is the position vector of a point on the body relative to the moving axes system. In order to analyse the dynamics of a multibody system, interaction of at least two bodies should be considered. Let us assume that these bodies are named as i and j . p_i and p_j are two points on these bodies as shown in Fig. 3. In order to analyse the motion of bodies i and j , the internal forces acting on the individual bodies due to their motion relative to each other need to be expressed. Motion of the origin of axes system i is given by (x_i, y_i, z_i) and the angular rotation of axes is given by $(\alpha_i, \beta_i, \gamma_i)$; similarly the motion of body j is described by (x_j, y_j, z_j) and $(\alpha_j, \beta_j, \gamma_j)$:

$$d_i = (x_i \ y_i \ z_i) + (\alpha_i \ \beta_i \ \gamma_i) \times (a_i \ b_i \ c_i), \quad (29)$$

$$d_j = (x_j \ y_j \ z_j) + (\alpha_j \ \beta_j \ \gamma_j) \times (a_j \ b_j \ c_j). \quad (30)$$

In order to calculate the reactions on each body, relative displacements between connecting points need to be calculated. The relative displacement is given as

$$d = T_j^T d_j - T_i^T d_i, \quad (31)$$

where $T_i^T d_i$ transforms local deflection to the global deflection, and thus d is in the global axes frame. It is more

convenient to deal with the non-linearity in the principal axes system of springs. Therefore, deflection in the reference (local) frame of the spring is given by

$$d_l = T_s d, \quad (32)$$

where T_s is the transformation matrix from the global to the local axes frame of the spring. Then the polynomial expression for forces may be defined as follows.

If $d_l = (x_l \ y_l \ z_l)$, then forces acting in the spring (or soft tissue) may be calculated as

$$F_{x_l} = kx_l + k_2x_l^2 + k_3x_l^3 + \dots + k_nx_l^n. \quad (33)$$

Therefore, Eq. (33) gives the forces in the local frame of the spring and similar expressions may be written for y_l and z_l deflections, obtaining $F_l = (F_{x_l} \ F_{y_l} \ F_{z_l})$. In order to apply the force F_i and $-F_i$ on bodies i and j , they need to be expressed in body frames i and j respectively:

$$F_i = T_i T_s^T F_l, \quad (34)$$

$$F_j = T_j T_s^T F_l. \quad (35)$$

In Eqs. (34) and (35), transformation of forces from the local axes frame of springs to the global and from the global to the local body axes frame is carried out. Having calculated the forces on bodies, now moments can also be calculated. Each moment equation is carried out in its own axes frame.

$$M_i = r_i \times F_i \quad (36)$$

and

$$M_j = r_j \times F_j. \quad (37)$$

Till now no equation of motion compiled in matrix form; forces are available only in numerical form, provided displacements and velocities of end points of springs are known. Since numerical integration techniques such as the Runge–Kutta method requires velocities and displacements for integration, these should be available (this should be equally true even if the solution was analytical). This approach enables us to analyse motion of a multibody system in time domain however complex the non-linearity may be. The main steps of this approach are listed below. Compared to piecewise linearization, this method needs no checking as to which linear segments should be added to the stiffness matrix and in this method the stiffness or inertia matrix is never assembled. Inertia matrix may be needed if one does not choose to use the principal axes systems of individual bodies. However, operating in the principal axes systems, each equation of each body is to be divided with appropriate mass or inertia elements, e.g. $\ddot{x}_i = F_{ix}/m_i$.

Steps of Analysis

1. Find the displacements of all tissue attachment points and cartilage contact points (cartilage contact is discretized).
2. Find the relative displacements between these points. When this is implemented for bone surfaces, this step becomes more complicated as the displacement is not the

- Euclidian distance between the displacements of these points. This is explained later in Section 5.
3. Transform these displacements to the local axes of the springs (muscles/tendons).
 4. Obtain forces in local axes.
 5. Transform forces to the global axes; from the global axes transform them to the individual body axes system, divide them with appropriate mass/inertia and add them to the state space formulation.
 6. Perform numerical integration obtaining new displacement and velocity vectors. These are to be converted to the global axes frame (although not needed for dynamics analysis) for surface calculations.

In the integration of the equations of motion these steps are performed at each time interval.

Having established the equations of motion, either in matrix or non-matrix form, there are additional information which the analysis treatment requires. These relate to the geometry of the objects in interaction. The very first problem relates to the muscle force action points (in our equations, the positions of p_i and p_j). The tendons articulating a joint because of muscle force do so by pulling bones at the attachment points; the line of action should go through these points provided there is no other surface intersecting the line. However, if there is any soft or hard tissue between those points then the connecting tissue 'wraps around' the obstacle(s). The next section deals with this problem of calculating the effective line of force action in the presence of obstructive surfaces or objects.

3.3. A muscle wrapping algorithm

In order to generate a tissue path between origin (Attachment_1) and insertion (Attachment_2) points are chosen on the surface and a straight line is drawn between them (Fig. 3). After describing the origin and insertion points a local cylindrical coordinate system needs to be established to investigate possible paths. The overall system already has a global coordinate system where all local coordinate systems are established and run with respect to the global coordinate system. The origin of a local cylindrical coordinate system is always assumed to be at origin point O . Also, the axis system at the origin point must be established by introducing three orthogonal vectors such as u , v and w . Let us introduce the vector v as the central axis of the local cylindrical coordinate system. The straight line between the attachment points is referred to as vector l shown in Fig. 3.

The vector l is written as

$$l = r_b - r_a. \quad (38)$$

All unit vectors are extracted to establish the local cylindrical coordinate system.

$$v = \frac{(r_b - r_a)}{|r_b - r_a|} \quad (39)$$

In order to have the vector w perpendicular to vector v , an arbitrary vector X is introduced to the system as $X =$

$(1 \ 0 \ 0)$.

$$w = \frac{X \times v}{|X \times v|}, \quad (40)$$

$$u = v \times w. \quad (41)$$

Before running the algorithms, it is required to obtain projections of points that surround bone surfaces within the particular cylindrical volume. Projections are required to be obtained on the created Ouw plane. Projection of a typical point (i) can be found as

$$l_i = r_i - r_a. \quad (42)$$

Projection on u is given by

$$p_u = (r_i - r_a) \cdot u \quad (43)$$

and projection on w is given by

$$p_w = (r_i - r_a) \cdot w. \quad (44)$$

From here the angle measured from positive u is given by

$$\alpha_i = \tan^{-1} \frac{p_w}{p_u}. \quad (45)$$

The first part of the algorithm divides the 360° into equal intervals (36 segments taken to be 10° each) and then calculates the angular position of each point and determines to which segment the point belongs. The algorithm calculates the total number of points in each segment and it also calculates the segment covariance, which is given as

$$\text{cov} = \sum (p_w^2 + p_u^2). \quad (46)$$

The first algorithm tested considered the total number of points in deciding the shortest path. Here the assumption is that the points on the surface spread evenly and the segment with the least number of points identifies the shortest path. The covariance is a better measure if the points are not evenly distributed, but there is no significant advantage over simple point summation if the spread is uniform.

After identifying the segment which will be taken to create the shortest path, the second part of the algorithm deals with the convex hull. This is a necessary step since tendons or ligaments under tension will wrap around bone surface and follow straight lines between different sections of bones when tension dictates. However, what is required in our case is not a standard implementation of the convex hull as a 3D surface but finding a specific line on the convex hull surface. Therefore, the algorithm will assume that the shortest path is a convex hull line on the mid-plane of the segment with the shortest path (minimum covariance or minimum number of points). Therefore, this section of algorithm will find the points on this curve (curve due to the mid-plane intersecting with the surface). In order to implement this we need to find the point of intersection of a line segment and the mid-plane. Once the intersection formulation is established, the intersection can be performed for all the line segments of the

surface triangles, as the surface is determined by triangular patches (VRML2 format is used). Consider a line segment connecting points i and j (part of a triangle made of points i , j , k) as shown in Fig. 3.

The plane/line segment intersection point is given by

$$\mathbf{l}_p = \mathbf{l}_i + \lambda(\mathbf{l}_j - \mathbf{l}_i). \quad (47)$$

In this case the intersection is to be on the mid-plane, and the projection of the intersection point on \mathbf{u} and \mathbf{w} will have an angle equal to the mid-plane angle (say this angle is α).

$$\tan \alpha_i = \frac{\mathbf{p}_w}{\mathbf{p}_u}, \quad (48)$$

$$t_p = \tan \alpha_i, \quad (49)$$

$$t_p = \frac{(\mathbf{l}_i + \lambda(\mathbf{l}_j - \mathbf{l}_i)) \cdot \mathbf{w}}{(\mathbf{l}_i + \lambda(\mathbf{l}_j - \mathbf{l}_i)) \cdot \mathbf{u}}, \quad (50)$$

$$t_p((\mathbf{l}_i + \lambda(\mathbf{l}_j - \mathbf{l}_i)) \cdot \mathbf{u}) = (\mathbf{l}_i + \lambda(\mathbf{l}_j - \mathbf{l}_i)) \cdot \mathbf{w}. \quad (51)$$

Solve this for λ

$$\lambda = \frac{\mathbf{l}_i \cdot (\mathbf{w} - t_p \mathbf{u})}{(\mathbf{l}_i - \mathbf{l}_j) \cdot (\mathbf{w} - t_p \mathbf{u})}, \quad (52)$$

$$\mathbf{l}_p = \mathbf{l}_i + \frac{\mathbf{l}_i \cdot (\mathbf{w} - t_p \mathbf{u})}{(\mathbf{l}_i - \mathbf{l}_j) \cdot (\mathbf{w} - t_p \mathbf{u})}(\mathbf{l}_j - \mathbf{l}_i). \quad (53)$$

Now the point \mathbf{r}_p can be calculated by

$$\mathbf{r}_p = \mathbf{r}_a + \mathbf{l}_i + \frac{\mathbf{l}_i \cdot (\mathbf{w} - t_p \mathbf{u})}{(\mathbf{l}_i - \mathbf{l}_j) \cdot (\mathbf{w} - t_p \mathbf{u})}(\mathbf{l}_j - \mathbf{l}_i). \quad (54)$$

After calculating the intersection point \mathbf{r}_p and the points being collected along the line between the attachment points, the shortest path is calculated. It is possible that having started along a 'shortest' path, testing the potential paths for the remaining distance may offer further improvement. For this, the neighbouring paths need to be checked to find out whether there is a shorter one for the remaining distance. Every path will have a total covariance measured as

$$\text{cov}(i) = \sum_i (\mathbf{p}_w^2 + \mathbf{p}_u^2), \quad (55)$$

where i is the stage index. This is the summation for all points within a slice. The covariance along the line on the mid-plane of each slice will be different than 'slice covariance'. However, this is a relative measure and it is expected that the lowest mid-plane line covariance would belong to the lowest slice covariance. The reason why the testing of the remaining distances is needed is that a particular slice may have a high covariance because only a section has a very high peak that distorts the measure of that section; if the high peak section is excluded, then it could have had the lowest covariance. It means that the path may start along a slice to avoid the high peak section but later along the path may change back to the so-called path with the lower covariance. In order to implement this, the concept of 'residual covariances' has been introduced. This algorithm divides the length into

segments and travel along the path starts with the preferred path (of lowest covariance); but along the path, neighbouring paths are checked for lower covariances for the remaining journey. An example of the implementation of the lowest path algorithm is shown in Fig. 4. Having calculated the shortest path the moment arms can easily be obtained by checking the body membership of the surface points along the path. The last point belonging to body i and the first point belonging to body j give the coordinates of the moment arms (or p_i and p_j in the presented multibody formulation). The shortest path example is shown in Fig. 4.

3.4. Collision, contact and equilibrium

It is not always possible to have human skeletal surface models being presented in an assembled form. Even if this is the case, the actual equilibrium position of bones will depend on the loading and dynamics of the assembly. Therefore, it is important that the motion starts from an equilibrium position. In our analysis this will be taken to be the static equilibrium position. It will be assumed that the bones are placed as close as possible to their natural positions, but starting from this approximate position an algorithm is required to move them to their static equilibrium position. To achieve this, the collision algorithm is developed. Since our objects (bones) are of rather complex shapes, collision between two bone objects involves finding the shortest distance along the direction of motion. Although it is possible to move bodies towards each other (the direction of motion to be calculated based on tendon–muscle forces) and incrementally calculate the nearest distance, this is a very inefficient method. The proposed method tests every point on the moving surface, which has a surface normal with an angle less than 90° to the direction of motion against the other object(s) surface. Having calculated the forces, in the absence of any reacting force from the contact surface there will be a pulling force between the bodies. The algorithm assumes that at least one of the multibodies is stationary and others are moving relative to the stationary body. The current software allows the user to enter ligament connections interactively, and on entering the unloaded lengths the tension in each connection can be calculated, enabling the calculation of the relative direction of motion \mathbf{V} . Now, on the moving body i each point on the surface is projected towards body j and one would expect to collide with a point on the surface of the body j . However, not all points on the bodies need to be investigated. Only those points with positive surface normals, making angles less than $\pi/2$ with the vector \mathbf{V} , need to be considered. Considering two points p_i and p_j on bodies i and j respectively, the following algorithm will determine if these two points are in the range of the collision path and if they need to be considered for collision. Not all points satisfying the collision criterion will collide as only the shortest distance among all eligible pairs will collide.

The following statement determines whether points are on the positive side of the surface with respect to the motion vector \mathbf{e}_v , which is the unit vector of \mathbf{V} .

Consider points p_i and p_j only and only if $\mathbf{n}_{p_i} \cdot \mathbf{e}_v > 0$ and $\mathbf{n}_{p_j} \cdot -\mathbf{e}_v > 0$.

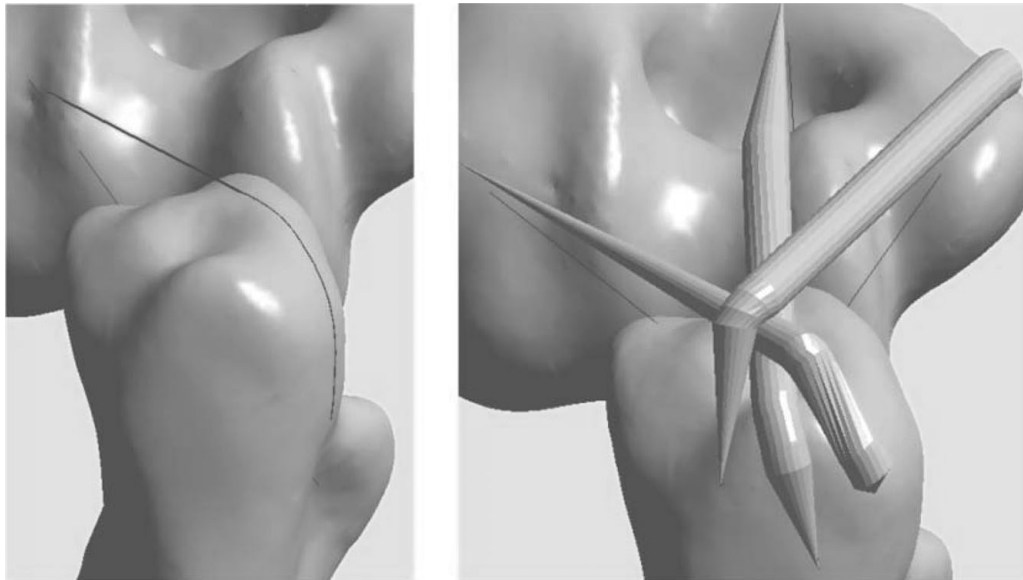


Fig. 4. Shortest path of a ligament wrapping between ulna and humeral bones of the elbow joint.

If the above condition is satisfied, then the points are checked to see if they are in their collision path. The condition for this is given by

If $|l_{ij} \times e_v| < \text{tol}$, then points i and j will collide. 'tol' is the tolerance and can be determined by point density

$$\text{tol} = \rho \sqrt{\frac{\text{Surface area}}{\text{No. of points}}}, \quad (56)$$

where ρ is a factor taken to represent irregularity in surface point distribution density; this is taken to be sufficiently high enough to cater any density fluctuations which ensures selection with a reasonable safety factor. Taking it to be 5 ensures a selection with a reasonable safety factor. In the release version of the software, an option will be given to the user to choose the value for ρ . Note that the cross product term gives the distance h_{ij} , as shown in Fig. 5. The current algorithm is based on the point clouds but future algorithms may use intersection of the motion vector with the surface simplexes to ensure more precise calculation of collision surfaces. Now the final stage of the algorithm needs to compare all eligible pairs to find the shortest among them, which will give the collision point. The collision distance between points i and j is given by

$$d_{ij} = l_{ij} \cdot e_v. \quad (57)$$

Collecting all the elements of the algorithm, if $n_{pi} \cdot e_v > 0$ and $n_{pj} \cdot -e_v > 0$ and $|l_{ij} \times e_v| < \text{tol}$ and $\text{abs}(l_{ij} \cdot e_v) < \text{Mind}_{ij}$, then $\text{Mind}_{ij} = l_{ij} \cdot e_v$. When all the points of bodies i and j are tested, Mind_{ij} will give the points i and j which will satisfy the collision conditions.

After the bodies are brought together at the collision point, the contact spring is attached (representing cartilage stiffness

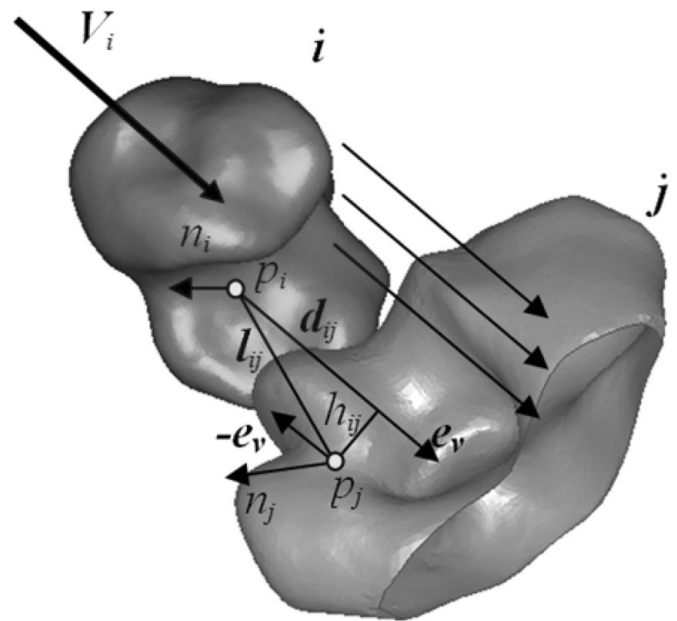


Fig. 5. Illustration of collision algorithm along with applied directional velocity V_i .

only in the l_{ij} and lateral stiffness taken to be 0.001 of the normal stiffness). Loading the system again in this new position makes the moving body translate (sliding on the surface relative to the contact point) and rotate. If there are more than one contact points then the resultant sliding will ensure sliding translation on all contact surfaces as the mathematical formulation given the above ensures this. Since the motion now includes rotation as well as translation, each body on the moving surface will have different velocity vectors. And instead of calculating e_v in the algorithm given above for all the moving points, it now needs to be calculated

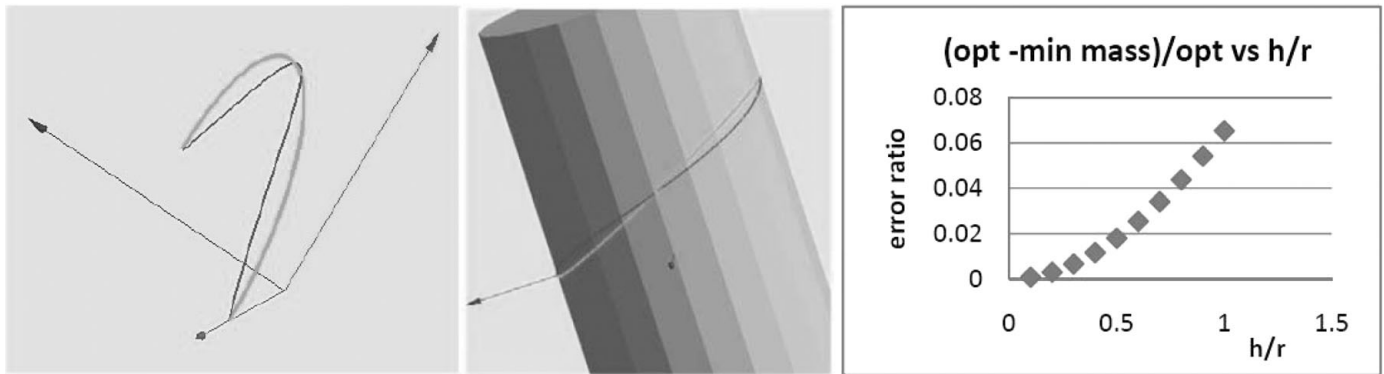


Fig. 6. A comparative study between helix and ellipsoidal arcs: ellipsoidal, minimum mass (solid line); helix, optimum shortest path (dashed line).

for each point on the body i as

$$e_{vi} = \text{Unit vector} (V_i + w_i \times p_i). \quad (58)$$

4. A Study of Effectiveness of the Muscle Wrapping and Collision Algorithms

4.1. The muscle wrapping algorithm

The muscle wrapping algorithm is a heuristic algorithm. And being a heuristic algorithm, it is very difficult to compare against other published material. The true comparison can only be possible if identical surface data (mesh geometry and topology) are used and if the published algorithms by others are available for testing. This makes the comparative study very difficult, if not impossible. Instead, the proposed algorithm is tested against a possible analytical shortest path. The simplest surface for this is a cylindrical surface. Because a cylindrical surface can be 'cut opened' to a plane, the shortest distance between two points is a straight line. Or, rolled back to a cylinder, the line becomes a helix. In our case the path between two points on a cylindrical surface will lie on a plane and on the cylindrical surface this will be an ellipsoidal arc. In other words, the study will therefore involve the error between an ellipsoidal arc and a helix, both curves starting at the same point and finishing at the same point. Due to space restrictions in this report a full derivation will not be given, but it can be shown that the maximum error is observed between these curves when the finishing point is exactly at the opposite face of the cylinder relative to the starting point. h is a parameter measuring the finish point relative to the start point along the central axis of the cylinder. A full derivation is available from the authors of the paper. Even for the maximum error curve, the error between the analytical solution and the proposed method is 7%. This tends to increase continuously; however, such a study goes beyond what is likely to be observed in realistic situations. Figure 6 shows extracts from the study; paths from the minimum mass algorithm (solid line) and the analytical solution (dashed line) are shown. The percentage of error (optimum curve length –

minimum mass curve length)/optimum curve length is plotted versus h/r ratio (a measure of planar inclination of the minimum mass plane).

4.2. Collision algorithm

For collision several methods were tested: (A) The collision algorithm, which is provided by DInsight (the company which provided the graphics platform), a part of KernelCAD.OCX, is effectively a nearest distance algorithm and requires iterative application of the method for finding the collision point; (B) the exhaustive search method which involved finding the nearest distance by searching every vertex on one body against every vertex on the other and similar to DInsight and iterating towards the collision point; and (C) the method proposed in this project. There are two versions of the method, one assumes no rolling and projection of vertices of one body on the other along the lines of velocity (or relative velocity) ensuring a single-step solution and the developed algorithm returns 'time to collision' and the collision point. The method is of $(O) n^2$ complexity; however, it only applies after substantial data reduction. The second version has conditional iteration added to the algorithm to deal with rolling as well as translation. If angular rotation is small, the combined motion of translation and rotation of individual points may be assumed to be linear. The assumption is true only if the angle of rotation is less than (10°) ; this is taken to be the angle which $\sin(\theta)$ can be approximated to (θ) . However, if the predicted time implies that an angular rotation is more than this, then the motion is halved and collision prediction algorithm is re-run. This is different than moving bodies and testing for possible collision.

The following results are found for 70,000 vertices (50,000 on one and 20,000 on the other):

- Exhaustive search took 21 min.
- DInsight (KernelCAD) software took 70 s for a single-step nearest distance calculation. It is estimated (by the company) that it would take approximately 10 iterations for the software to be converted to work as a collision tool with $e-4$ accuracy (although one would not need

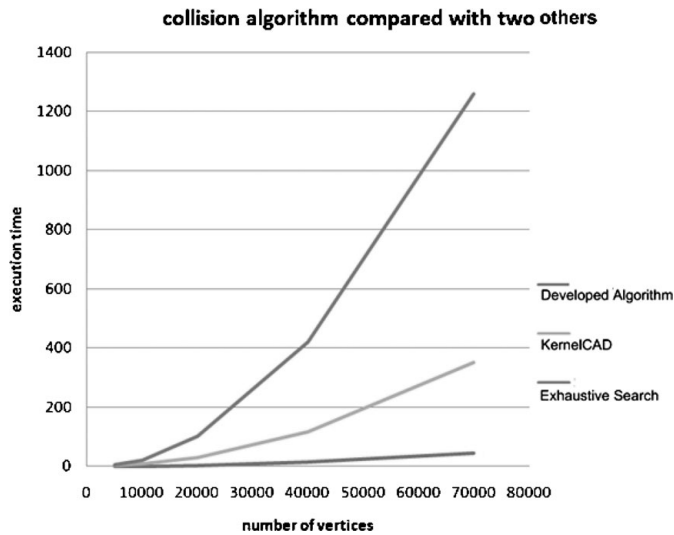


Fig. 7. Comparing collision execution time.

this). This translates to approximately 11 min, half of the exhaustive search. Their results were plotted with an assumption of 5 iterations and total time of which is calculated based on single iteration time. The software company promised to implement full collision during the next few months. It further promised to implement our algorithm in their library as an option. If this is done then one could perform a reliable comparison.

C. Predictive search proposed in this thesis was 45 s.

Examples chosen had not required any iteration during the translation combined rolling motion. The efficiency and algorithm performance as a time to execution is plotted against the number of vertices in the search (Fig. 7). The results show that there is some overhead which influences the lower number of vertex sets; however, as the vertex number increases, the overhead influence becomes a small part of overall processing time.

5. Software Implementation

The Musculoskeletal Joint Modeller (MJM) software has been developed based on the formulation and the theory presented above. The software performs the following steps:

1. Initialize, read bone surfaces, identify number of bones, number them, find mass/inertia properties, find principal inertias and principal axes system.
2. Enter into time integration loop, and in each step perform the following steps:
 - a. Update geometry, use displacement and velocity of each body to relocate them in space
 - b. Perform muscle wrapping and muscle tension.
 - c. Obtain muscle segments, centroid lines, force action points on bodies and force vectors.
 - d. Obtain equation of motion and perform time integration.

A pseudo code is given as follows:

PSEUDO CODE

Read system parameters, obtain mass and inertia of bodies, obtain principal inertias and axes, convert spring attachment points to local axes frames. With this there is no need for mass matrix conversion.

If time \geq total integration time, then stop, or else continue.

For $i = 1$ to N tissue attachments (including muscle and ligament)

For each tissue attachment perform muscle wrapping algorithm:

Obtain cylindrical coordinates joining two end points of the attachment and perform minimum mass or perform minimum covariance to obtain the 'shortest path'.

Along this path perform 'convex hull'.

Along the path and r distance away from the surface (using surface normals), construct the 'centroid' path.

Record the mass number from which the centroid points are created.

Starting from the point of attachment, follow the centroid points until the mass number changes.

The last point before the mass number changes is the point which is to be used in calculating the moment.

Arm: This is the 'spring attachment point'. At the end of the muscle wrapping algorithm, the following are obtained:

(1) spring attachment points, (2) on which body the point is located, (3) direction of the tissue at that point, (4) extension of the tissue, (5) tension in the tissue (based on polynomial or any other formulation) and (6) direction of force.

For each tissue attachment find how many sub-segments exist and create a new tissue segment list (this way now we can handle each tissue path going over multibodies)

Next tissue

Perform collision algorithm to find a collection of contact points (surface contact points within a range).

Two algorithms are currently being used; the user can choose any one:

A. *Collision based on the algorithm came with the graphics platform component.*

This is based on stepwise shortest distance calculations, a method used by all collision analysis algorithms currently available in commercial code.

B. *Algorithm developed and presented in this paper which performs 'time to collision' predictive analysis.*

If the angle between the surface normal and the direction of relative motion between bodies is more than 90° , then exclude the vertex from the search.

If the radial distance between the approaching vertices (fly-past) is greater than a tolerance, then exclude the pair from the search.

For the remaining vertices compare minimum distances (Mind) to find the smallest

Mind/approach velocity = time to collision

The following statement finds the Mind: If $\mathbf{n}_{pi} \cdot \mathbf{e}_v > 0$ and $\mathbf{n}_{pj} \cdot -\mathbf{e}_v > 0$ and $|\mathbf{l}_{ij} \times \mathbf{e}_v| < tol$ and $abs(\mathbf{l}_{ij} \cdot \mathbf{e}_v) < Mind_{ij}$, then $Mind_{ij} = \mathbf{l}_{ij} \cdot \mathbf{e}_v$.

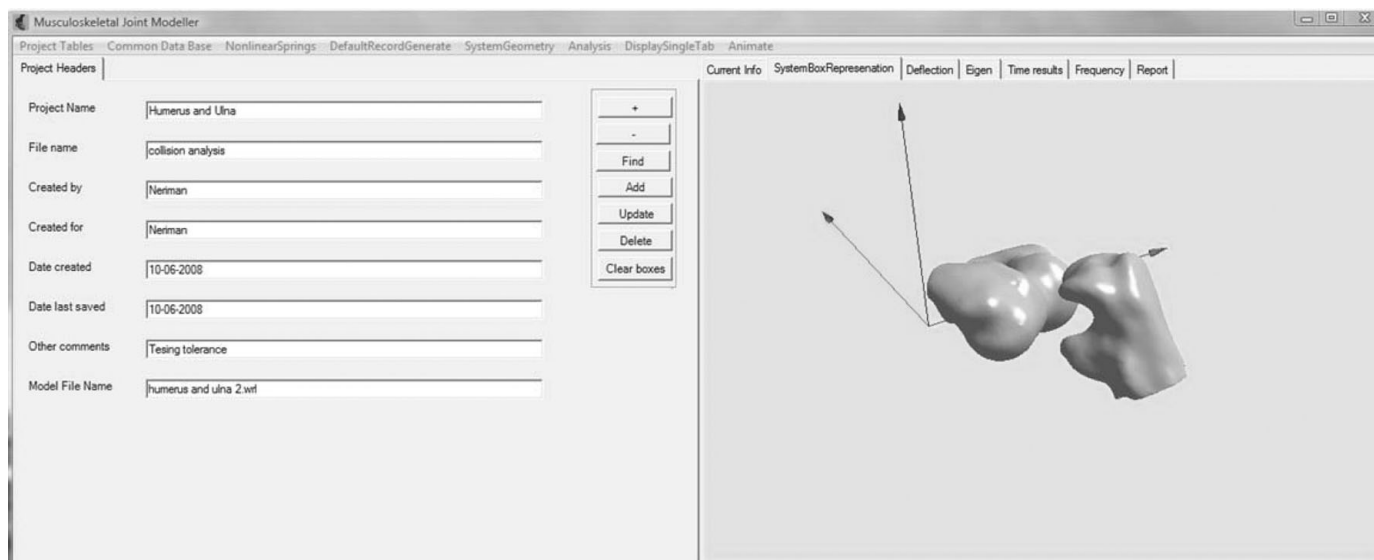


Fig. 8. The front end of the Musculoskeletal Joint Modeller (MJM) software.

At this stage all connections between bodies are established

For $i = 1$ to total ligament segments

Find body numbers attached at each end of the ligament segment (ligament segment defined above).

Attachment coordinates and force vectors are known.

Assemble these into the equations of motion (depending on the mass numbers) in state space form.

Next i

For $i = 1$ to total number of contact points

Find body numbers attached at each end of the ligament segment (ligament segment defined above).

Attachment coordinates and force vectors are known.

Assemble these into the equations of motion (depending on the mass numbers) in state space form.

Next i

Use the Runge–Kutta method to integrate.

Increment time step.

The software is general and is capable of dealing with multibody interactions. The front end of the software is shown in Fig. 8.

The software input window has five sections. The first section includes project descriptions such as title of the project etc. The second section has the spring definitions, including polynomial coefficients. The next section shows the mass definition, which includes how many mass is in the analysis system. This section also details mass-related information such as mass and inertial values, centre of mass relative to the global axes and initial Euler angles. The fourth section determines spring/damper attachments. The entry in this section involves x , y , z coordinates of spring ends relative to the local axes frames and initial lengths of springs. The fifth and final section involves the definition of forces. In this case the force directions are defined in the global axes, but the actual point on bodies are described in the local axes frames. This definition allows external loads such as forces due to acceleration of gravity (weights) to be defined precisely. Currently, alternative loading schemes are also considered.

The prototype software is database-driven and the tables of data are saved in MS Access.

The current testing involved all the algorithms described above. The efficiency of algorithms are not fully scrutinized and they can all be improved further. Particularly, the shortest path algorithm needs to be improved in order to make the MJM software to be a viable research or commercial tool. The actual contact algorithm seems to be efficient enough and bodies in contact seem to settle to their equilibrium position in no more than four steps. A typical solution of two bones under loading can take up to 2 min using a laptop computer with a Pentium 4 with a processor speed of 2.33 GHz. About 80% of the processing time is taken by the shortest path algorithm. Full testing of the program is to continue for some time and it is expected to be the subject of another publication.

6. Preliminary Results, Discussion and Conclusion

A preliminary study is carried out for an elbow joint. Bone surface parameters are obtained using a mechanical digitizer and converted into post-geometric processing software Geomagic Studio 9 in order to create 3D geometric mesh data. The experimental study involved measuring the elbow joint laxity based on various devices developed in our own labs. For the subjected specimen there was no tissue damage recorded and no artificial joint reconstruction. Thus, the performed experimental test was carried out to understand the normal joint laxity of the elbow (Fig. 9). The experiment was performed at 30° elbow flexion, 30° shoulder joint supination and 120° forearm supination with respect to the neutral position. Measurements showed that the varus-valgus elbow joint laxity is between 1.3° to 5° under 100 N varus-valgus loading conditions.

The initial software study involved simulating these results using the software developed (MJM). In order to perform the simulation, passive ligament,^{37,88} cartilage, muscle–tendon parameters and active muscle parameters^{85–87} were taken from the Lifemod software²⁶ and literature provided above. Initial elbow joint laxity measurement results through the



Fig. 9. Experimental-cadaveric study of elbow joint laxity.

joint modeller (MJM) showed much softer joint behaviour with up to 15° varus-valgus laxity ranges. This is probably due to the accuracy of the parameters used in Lifemod rather than the accuracy of either software since the comparison presented here was carried out against the real human tissue. Unfortunately, it was not possible to import our surface data to Lifemod to compare the software results against each other. Because of the type of investigation involved, comparison tends to be qualitative rather than exact and quantitative. Nevertheless, gradual stiffening of tissue parameters eventually gave laxity measurements that were comparable with the experimental results; when the applied load (at 30 cm away from the elbow joint) gradually increased to 100 N, a laxity measurement of 5° was observed from the software. At this stage of investigation only linear parameters were used.

The theories and algorithms to model a musculoskeletal structure is presented in this paper to form a framework for musculoskeletal joint analysis. It can be said that the focus of the paper is the joint behaviour rather than musculoskeletal motion, as is the case in all commercially available software packages. Currently, the analysis of the motion of rigid bodies in space interconnected by various standard types of joints is well established. A typical human joint, which can be approximated to a standard mechanical joint for the gross motion analysis, exhibits full 6 degrees of freedom. The local mobility of joints has a significant influence in human motion analysis as far as the local joint loads are concerned. Thus, an accurate analysis of joint motion is very important in medical applications such as implant evaluation or surgery assistance and design of medical instrumentation. Since the proposed modelling will involve interacting surfaces and relative motion of these surfaces, the way surfaces are defined becomes very important. The first geometrical problem arises because it is impossible to have a digitized surface to be completely smooth and it will be necessary to develop algorithms to ensure that the contact and collision are handled without violation of space occupied by bodies (no volumetric intersection among contacting bodies). The second geometrical problem, which is also related to surface modelling, involves finding the shortest path between two points on surfaces belonging two bodies. It is assumed that connecting tissue under tension takes the shortest path. This is true only if there is no surface friction. Calculating exact

geometry of muscles and tendons is a prerequisite for forming equations of equilibrium as the correct position of the force action points needs to be calculated accurately.

In this paper, (a) the physical deformation of biological tissues is modelled using linear or non-linear viscoelastic spring and spring-dashpot elements. These kinds of elements are used in modelling biological tissues, and data are available in literature. (b) The formulations presented deals with small motion dynamics as well as (c) the non-linear modelling, where in this case a non-matrix and algorithmic formulation is presented. In dealing with non-linearity, the equation of motion is solved in the global axis frame but evaluation of forces is carried out at the local axis frame of the non-linear 'springs'. This approach allows both geometrical as well as material non-linearity to be handled with ease. (d) The paper also presents the shortest path algorithms for calculating biological tissue line of action geometries. The developed algorithms are based on calculating minimum 'surface mass' and 'surface covariance'. An improved version of the 'surface covariance' algorithm is described as 'residual covariance'. The 'shortest surface' is further processed to ensure that the path lay on the convex hull. The resulting path is used to establish the direction of forces and moments acting on the bodies of the joint. The actual action points of the force line are determined between the last point of the convex hull path on the 'origin' body and the first point on the 'insertion' body. This information is needed for linear or non-linear treatment of motion. (e) The final contribution of the paper is the treatment of the collision. In the virtual world, difficulty in analysing bodies in motion arises because bodies can penetrate each other. The collision algorithm proposed in the paper involves finding the shortest ray projected from one body to the other. The projection of the body direction is determined by the resultant forces acting on it due to soft tissue connections under tension. This enables the calculation of collision condition of non-convex objects accurately. After the initial collision the subsequent analysis involves attaching special springs (crudely representing cartilage exhibiting stiffness normal to the surfaces only) at the 'collision points' and recalculating motion of bodies. Currently, the improved collision algorithm incorporates the rotation as well as translation. The cartilage stiffness calculated based on the number of contact points

with high enough stiffness values does not allow a penetration of more than 20%. Thus, stiffness is recalculated as the contact points are regenerated at each step. This ensures that the spread of contact area is accounted for and the explicit cartilage stiffness is now not needed. In other words, contact treatment is similar to the penalty method used by Lifemod, but with an important difference: in our case the contact stiffness is not designed to prevent penetration only (as it would be the purpose of the penalty method) but to describe cartilage behaviour as accurately as possible. The software has not been fully tested but preliminary results are encouraging and comparable with experimental results previously carried out in our research group.

References

1. K. Brandt, E. Radin, P. Dieppe and L. Van De Putte, "Yet more evidence that osteoarthritis is not a cartilage disease," *Ann. Rheum. Dis.* **65**, 1261–1264 (2006).
2. S. Delp, J. Loan, M. Hoy, F. Zajac, E. Topp and J. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Trans. Biomed. Eng.* **37**(8), 757–767 (1990).
3. E. Chao, R. Armiger, H. Yoshida, J. Lim and N. Haraguchi, "Virtual Interactive Musculoskeletal System (VIMS) in orthopaedic research, education and clinical patient care," *J. Orthop. Surg. Res.* **2**(2), (2007).
4. F. M. Blakeley, "Cyberman," Chrysler Corp., Detroit, MI (1980).
5. P. Bapu, S. Evans, P. Kitka, M. Korna and J. McDaniel, *User's Guide for Combiman Programs, Version 4*, University of Dayton Research Institute, Dayton, OH (1980).
6. Z. Mi, J. Yang and K. Abdel-Malek, "Optimization based posture prediction for human upper body," *Robotica* **27**, 607–620 (2009).
7. D. Thalman, R. Boulic, Z. Huang and H. Noser, "Virtual and real humans interacting in the virtual world," *Proceedings of the International Conference on Virtual Systems and Multimedia '95* (1995) pp. 48–57.
8. J. Fernandez, P. Mithraratne, S. Thrupp, M. Tawhai and P. Hunter, "Anatomically based geometric modelling of the musculo-skeletal system and other organs," *Biomech. Model. Mechanobiol.* **2**(3), 139–155 (2004).
9. N. Thalman and F. Cordier, "Construction of a human topological model from medical data," *IEEE Trans. Inform. Tech. Biomed.* **4**(2), 137–149 (2000).
10. J. Kupper, B. Loitz-Ramage, D. Corr, D. Hart and J. Ronsky, "Measuring knee joint laxity: A review of applicable models and the need for new approaches to minimize variability," *Clin. Biomech.* **22**(1), 1–13 (2007).
11. M. Safran, M. McGarry, S. Shin, S. Han and T. Lee, "Effects of elbow flexion and forearm rotation on valgus laxity of the elbow," *J. Bone Joint Surg.* **87**, 2065–2074 (2005).
12. H. Jonsson, J. Karrholm and L. Elmquist, "Laxity after cruciate ligament injury in 94 knees. The KT-1000 Arthrometer versus Roentgen Stereophotogrammetry," *Acta Orthop. Scand.* **64**(5), 567–570 (1993).
13. A. Ganko, L. Engebretsen and H. Ozer, "The Rolimeter: A new arthrometer compared with the KT-1000," *Knee Surg. Sports Traumatol. Arthrosc.* **8**(1), 36–39 (2000).
14. R. Habets, Computer assistance in orthopaedic surgery, *Ph.D. Thesis* (Technische Universiteit Eindhoven, 2002).
15. F. Van Der Helm, "Analysis of the kinematic and dynamic behaviour of the shoulder mechanism," *J. Biomech.* **27**(5), 527–550 (1994).
16. W. Maurel, 3D modelling of the human upper limb including the biomechanics of joints, muscles and soft tissues, *Ph.D. Thesis* (Ecole Polytechnique Federale De Lausanne, 1999).
17. R. Raikova, "A general approach for modelling and mathematical investigation of the human upper limb," *J. Biomech.* **25**(8), 857–867 (1992).
18. A. Engin and S. Tumer, "Improved dynamic model of the human knee joint and its response to impact loading on the lower leg," *J. Biomech. Eng.* **115**, 137–143 (1993).
19. M. Moeinzadeh, A. Engin and N. Akkas, "Two dimensional dynamic modelling of human knee joint," *J. Biomech.* **16**(4), 253–264 (1983).
20. Z. Ling, H. Guo and S. Boersma, "Analytical study on the kinematic and dynamic behaviours of a knee joint," *Med. Eng. Phys.* **19**(1), 29–36 (1997).
21. S. McLean, A. Su and J. Van Den Bogert, "Development and validation of a 3d model to predict knee joint loading during dynamic movement," *J. Biomech. Eng.* **125**(6), 864–874 (2003).
22. E. Abdel-Rahman and M. Hefzy, "Three dimensional dynamic behaviour of the human knee joint under impact loading," *Med. Eng. Phys.* **20**(4), 276–290 (1998).
23. M. Damsgaard, J. Rasmussen, S. Christensen, E. Surma and M. Zee, "Analysis of musculoskeletal systems in the anybody modeling system," *Simulat. Model. Pract. Theor.* **14**(8), 1100–1111 (2006).
24. S. Delp and J. Loan, "A graphics based software system to develop and analyze models of musculoskeletal structures," *Comput. Biol. Med.* **25**(1), 21–34 (1995).
25. S. Delp and J. Loan, "A computational framework for simulating and analyzing human and animal movement," *Comput. Sci. Eng.* **2**, 46–55 (2000).
26. Lifemodeller (2008). *LifeMOD™*. Retrieved August 20th, 2008, from <http://www.lifemodeler.com>
27. E. Chao, "Graphic based musculoskeletal model for biomechanical analyses and animation," *Med. Eng. Phys.* **25**(3), 201–212 (2003).
28. MSCSoftware (2008). *ADAMS*. Retrieved August 20th, 2008, from <http://www.mscsoftware.com>
29. H. Lin, Y. Nakamura, F. Su, J. Hashimoto, K. Nobuhara and E. Chao, "Use of Virtual, Interactive, Musculoskeletal System (VIMS) in modeling and analysis of shoulder throwing activity," *J. Biomech. Eng.* **127**, 525–530 (2005).
30. R. Davoodi and G. Loeb, "A software toll for faster development of complex models of musculoskeletal systems and sensorimotor controllers in simulink," *J. Appl. Biomech.* **18**, 357–365 (2002).
31. Anybody Technology (2006). *The AnyBody Modeling System™*, Trial Version. Retrieved December 10th, 2006, from <http://www.anybodytech.com>
32. I. Brouwer, V. Mora and D. Laroche, "A viscoelastic soft tissue model for haptic surgical simulation," *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Tsukuba, Japan (2007).
33. T. Atkinson, R. Haut and N. Altiero, "A poroelastic model that predicts some phenomenological responses of ligaments and tendons," *J. Biomech. Eng.* **119**(4), 400–405 (1997).
34. V. Mow, S. Kuei, W. Lai and C. Armstrong, "Biphasic creep and stress relaxation of articular cartilage in compression, theory and experiments," *J. Biomech. Eng.* **102**(1) 73–84 (1980).
35. C. Drapaca, S. Sivaloganathan and G. Tenti, "Nonlinear constitutive laws in viscoelasticity," *Math. Mech. Solid* **12**, 475–501 (2007).
36. A. Viidik and R. Ekholm, "Light and electron microscopic studies of collagen fibers under strain," *Anat. Embryol.* **127**(2), 154–164 (1968).
37. Y. Fung, *Biomechanics: Mechanical Properties of Living Tissues*, 2nd ed. (Springer-Verlag, New York, 1993).
38. R. Ogden, *Non-Linear Elastic Deformations* (Ellis Horwood, New York, 1984).
39. J. Funk, G. Hall, J. Crandall and W. Pilkey, "Linear and quasi-linear viscoelastic characterization of ankle ligaments," *J. Biomech. Eng.* **122**, 15–22 (2000).
40. S. Cowin and S. Doty, *Tissue Mechanics*, 2nd ed. (Springer, New York, 2007).

41. J. Weiss and J. Gardiner, "Computational modeling of ligament mechanics," *Clin. Rev. Biomed. Eng.* **29**(4), 1–70 (2001).
42. M. Loocke, C. Lyons and C. Simms, "Viscoelastic properties of passive skeletal muscle in compression: Stress-relaxation behaviour and constitutive modelling," *J. Biomech.* **41**, 1555–1565 (2008).
43. A. Hill, "The heat of shortening and the dynamic constants of muscle," *Proc. Roy. Soc. Lond. Ser. B, Biol. Sci.* **126**(843), 136–195 (1938).
44. E. Blum, C. Haun and J. Ryan, "A musculo-skeletal model of rat ankle motion and its experimental test on rat," *J. Biomech.* **40**(4), 891–899 (2007).
45. J. Winters and Y. Wang, "Integrating hill-based and neuro-fuzzy adaptive models to estimate history-dependent muscle mechanical behaviour," *5th World Congress of Biomechanics* **39**(1) (2006) p. S41.
46. F. Zajac, E. Topp and P. Stevenson, "A dimensionless musculotendon model," *8th Annual Conference IEEE Engineering in Medicine and Biology Society* (1986) pp. 601–604.
47. H. Huxley, "The mechanism of muscular contraction," *Science* **164**(3886), 1356–1366 (1969).
48. M. Krogt, C. Doorenbosch and J. Harlaar, "Muscle length and lengthening velocity in voluntary crouch gait," *Gait Posture* **26**(4), 532–538 (2007).
49. M. Pandy, "Moment arm of a muscle force," *Exerc. Sport Sci. Rev.* **27**, 79–118 (1999).
50. K. An, K. Takahashi, T. Harrigan and E. Chao, "Determination of muscle orientations and moment arms," *J. Biomech. Eng.* **106**(3), 280–282 (1984).
51. G. Marai, D. Laidlaw, D. Demiralp, C. Andrews, C. Grimm and J. Crisco, "Estimating joint contact areas and ligament lengths from bone kinematics and surfaces," *IEEE Trans. Biomed. Eng.* **51**(5), 790–799 (2003).
52. G. Marai, Data driven predictive modeling of diarthrodial joints, *Ph.D. Thesis* (Brown University, 2007).
53. W. Buford, Jr., C. Andersen, K. Elder and R. Patterson, "Verification of spline-path muscle models for a 3D simulation of the extremities," *Proceedings ISB, Zurich* (2001) p. 206.
54. C. Spoor, J. van Leeuwen, C. Meskers, A. Titulaer and A. Huson, "Estimation of instantaneous moment arms of lower-leg muscles," *J. Biomech.* **23**(12), 1247–1259 (1990).
55. W. Murray, A. Arnold, S. Salinas, M. Durbhakula, T. Buchanan and S. Delp, "Building biomechanical models based on medical image data: An assessment of model accuracy," *Book Chapter-Medical Image Computing and Computer-Assisted Intervention – MICCAI 1496* (1998) pp. 539–549.
56. K. An, F. Hui, B. Morrey, R. Linscheid and E. Chao, "Muscles across the elbow joint: A biomechanical analysis," *J. Biomech.* **14**(10), 659–669 (1981).
57. R. Jensen and D. Davy, "An investigation of muscle lines of action about the hip: A centroid line approach vs the straight line approach," *J. Biomech.* **8**(2) 103–110 (1975).
58. W. Murray, T. Buchanan and S. Delp, "Scaling of peak moment arms of elbow muscles with upper extremity bone dimensions," *J. Biomech.* **35**(1) 19–26 (2002).
59. W. Buford and C. Anderson, "Predicting moment arms in diarthrodial joints-3D computer simulation capability and muscle-tendon model validation," *Proceedings of the 28th IEEE EMBS Annual International Conference*, New York, USA (2006).
60. B. Garner and M. Pandy, "The obstacle set method for representing muscle path in musculoskeletal models," *Comput. Meth. Biomech. Biomed. Eng.* **3**(1), 1–30 (2000).
61. A. Carman and P. Milburn, "Dynamic coordinate data for describing muscle-tendon paths: A mathematical approach," *J. Biomech.* **38**(4), 943–951 (2005).
62. I. Charlton and G. Johnson, "Application of spherical and cylindrical wrapping algorithm in a musculoskeletal model of the upper limb," *J. Biomech.* **34**(9) 1209–1216 (2001).
63. F. Gao, M. Damsgaard, J. Rasmussen and S. Christensen, "Computational method for muscle-path representation in musculoskeletal models," *Biol. Cybern.* **87**(3), 199–210 (2002).
64. A. Audenaert and E. Audenaert, "Global optimization method for combined spherical-cylindrical wrapping in musculoskeletal upper limb modelling," *Comput. Meth. Programs Biomed.* **92**(1) 8–19 (2008).
65. S. Marsden and D. Swailes, "A novel approach to the prediction of musculotendon paths," *J. Eng. Med.* **222**(1), 51–61 (2008).
66. C. Gatti, C. Dickerson, E. Chadwick, A. Mell and R. Hughes, "Comparison of model predicted and measured moment arms for the rotator cuff muscles," *Clin. Biomech.* **22**(6) 639–644 (2007).
67. M. Lin and D. Manocha, "Collision and proximity queries," In: J. Goodman and J. O'Rourke (Eds.), *Handbook of discrete and computational geometry* (2nd ed., pp. 787–807). New York: CRC Press (2003).
68. K. Chung and W. Wang, "Quick collision detection of polytopes in virtual environments," *Proceedings of the ACM Symposium, Virtual Reality Software and Technology*, Hong Kong (1996).
69. S. Cameron, "Enhancing GJK: Computing penetration distance between convex polyhedra," *Proceedings of the IEEE International Conference on Robotics and Automation* **4** (1997) pp. 3112–3117.
70. C. Bajaj and T. Dey, "Convex decomposition of polyhedra and robustness," *SIAM J. Comput.* **21**(2), 339–364 (1992).
71. M. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," *Proceedings of IMA Conference on Mathematics of Surfaces* (1998).
72. P. Jimenez, F. Thomas and C. Torras, "3D collision detection: A survey," *Comput. Graph.* **25**(2), 269–285 (2001).
73. S. Gottschalk, M. Lin and D. Manocha, "OBB-tree: A hierarchical structure for rapid interference detection," *30th Annual Conference Series Computer Graphics* (30) (1996) pp. 171–180.
74. GAMMA (2008). Geometric Algorithms for Modeling, Motion, and Animation/Software and Models. Retrieved March 1st, 2008, from <http://gamma.cs.unc.edu>
75. F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts* (Wiley Series in Nonlinear Science) (Wiley, New York, 1996).
76. E. Drumwright, "Fast and stable penalty method for rigid simulation," *IEEE Trans. Vis. Comput. Graph.* **14**(1), 231–240 (2008).
77. M. Ferris, O. Mangasarian and J. Pang, "Complementarity: Applications, algorithms and extensions," Kluwer Academic Publishers, Netherlands (2001).
78. J. Trinkle, "Formulation of multibody dynamics as complementarity problems," *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois (2003).
79. J. Pang and J. Trinkle, "Complementarity formalisms and existence of solutions of dynamic multi-rigid contact problems with coulomb friction," *Math. Programming* **73**, 199–226 (1996).
80. C. Duriez, C. Andriot and A. Kheddar, "Signorini's contact model for deformable objects in haptic simulations," *IEEE/RSJ International Conferences on Intelligent Robotics and Systems*, Sendai, Japan (2004).
81. S. Berrenberg and R. Krause, "Efficient parallel simulation of biphasic materials in biomechanics," *6th International Congress on Industrial Applied Mathematics (ICIAM07) and GAMM Annual Meeting, Zurich* **7**(1), 1121101–1121102 (2007).
82. Y. Bei and B. Fregly, "Multibody dynamic simulation of knee contact mechanics," *Med. Eng. Phys.* **26**(9), 777–789 (2004).
83. S. Han, S. Federico, M. Epstein and W. Herzog, "An articular cartilage contact model based on real surface geometry," *J. Biomech.* **38**, 179–184 (2005).

84. J. Lenarcic and N. Klopčar, "Positional kinematics of humanoid arms," *Robotica* **24**, 105–112 (2006).
85. B. Garner and M. Pandy, "Estimation of musculotendon properties in the human upper limb," *Annu. Biomed. Eng.* **31**, 207–220 (2003).
86. K. Holzbauer, W. Murray and S. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analysing neuromuscular control," *Annu. Biomed. Eng.* **33**(6), 829–840 (2005).
87. J. Langenderfer, S. Jerabek, V. Thangamani, J. Kuhn and R. Hughes, "Musculoskeletal parameters of muscles crossing the shoulder and elbow and the effect of sarcomere length sample size on estimation of optimal muscle length," *Clin. Biomech.* **19**, 664–670 (2004).
88. D. Butler, E. Grood and F. Noyes, "Biomechanics of ligaments and tendons," *Exerc. Sport Sci. Rev.* **6**, 125–181 (1978).
89. M. Mihelj, "Human arm kinematics for robot based rehabilitation," *Robotica* **24**, 377–383 (2006).