

# Musculoskeletal Modeling of the Hand and Contact Object in Modelica

Shashank Swaminathan<sup>1</sup> Johan Andreasson<sup>2</sup>

<sup>1</sup>Novi, Michigan, USA, sh.swami235@gmail.com

<sup>2</sup>Modelon KK, Japan, johan.andreasson@modelon.com

## Abstract

The paper's primary goal is to develop a mathematical model that could be used towards the development and improvement of orthotic assist gloves. The model is constructed using component based modeling in the object-oriented declarative language Modelica, specifically the MultiBody Modelica library. Multiple hand models currently do exist; however, they are mainly causal, and require separate development and validation of mathematical solvers before use. By using Modelica, the model is constructed from the system's physical equations, thereby relieving issues regarding validity of the model's computational equations; the acausality inherent in Modelica allows for model development that more closely mirrors relations in the physical world. The model is scoped to be able to model the kinematics and dynamics of the hand when grasping a spherical object – both bone structure and muscle geometry and actuation are simplifications based off anatomy literature. The contact model is developed as a separate component from the hand system. The main design goal of the contact model is to represent the characteristics of a relatively rigid object that still maintains a degree of friction and pliability on the surface layer.

The main two grasps tested in the paper are the prehensile and precision grasps (powerful and dexterous grasps). The muscle actuation profiles per each finger are adjusted until the desired dynamic profile is achieved for each type of grasp. The main data points of interests are the joint angles and contact forces for each finger. Further verification of the model is done using the animation automatically generated by the tool. Simulation testing results indicate that the model can successfully simulate contractions at all levels of abstraction of the hand's components (basic bone-joint components, finger components, and the overall hand system). The results also indicate that both prehensile and precision grasps are possible, given appropriate muscle actuation and finger orientation parameter values.

**Keywords**—*musculoskeletal model of hand; Modelica; grasp model; orthotic gloves*

## 1 Introduction

### 1.1 Relevant Background and Definitions

Patients recovering from a stroke, or those that have Parkinson's disease, amongst many others, typically experience muscle weakness in the upper extremities. The use of orthotic devices in such situations is an effective method of returning a modicum of motor control to patients. Multiple such orthotic devices have been developed, including, but not limited to, gloves (Radder et al, 2015), (Adler, 2016), braces (Linn et al, 2012), and soft-muscle pneumatic tubes (Yanchev, 2015; Polygerinos, 2015). However, many of these devices, must be specially constructed per each patient, and requires multiple rounds of testing and data acquisition before completion. Constructing a mathematical model would enable a better understanding of the orthotic device, as well as optimize its construction. The prerequisite to developing a model of an orthotic device, is the development of a model for the underlying system, the hand.

Hand modeling has been typically done as a system of rigid bodies connected through revolute joints e.g. (Griffin et al, 2000). The papers derive the full set of equations of motion of the hand from this physical concept e.g. (Tarmizi, 2009).

From (Marieb, 2000), neural impulses trigger protein-based reactions that leads to overall muscle contraction, proportional to the neural impulse strength. Since the purpose of this paper is not to model the neural aspects, we will abstract this as an actuation request for a percentage of total muscle force.

In this paper also, the hand is modeled as composed of rigid bodies connected by revolute joints. The joints have restrictions on the total angle of rotation, and muscle actuation is added to the fingers appropriately.

(Hicks et al, 2015) observes that mathematical modelers have a dual responsibility of verifying and validating both the physical equations in the model, and the mathematical solving components of the model. We aim to significantly reduce this challenge by keeping the physics of the system well-removed from the mathematics required to solve the models. This is achieved by using Modelica (Modelica®, 2013) as the modeling language to describe the physical equations of

the hand, and Modelica-supporting tools – specifically Dymola (Dymola, 2017), OpenModelica (OpenModelica, 2016), JModelica (JModelica, 2016), and Wolfram SystemModeler (SystemModeler, 2015) – to mathematically solve the model. The choice of language was made due to Modelica's object-oriented unique nature; the systems can be broken into components, and each component's behavior can be represented solely through its physical equations. The separation of the physical and mathematical aspects of the system, with the user only interacting with the physical equations, and the tool handling the mathematical portion, enables focus only on the validity of the physical equations of the model.

The model utilizes the MultiBody library (Otter et al., 2003), which contains many components dealing with three-dimensional rigid bodies, further reducing the user burden.

To check the performance, the hand model performs prehensile (powerful) and precision (gentle) grasps around spherical objects. The grasps are derived from the taxonomy of grasps defined in (Cutkosky, 1990). To investigate such motion, a model of a contact object is also required; given the various levels of potential abstraction available while developing the contact object, this is addressed separately in the paper.

## 1.2 Objectives

The goal for the work described in this paper was to build a prototype mathematical model of the hand, in Modelica, that can describe the kinematic and dynamic interaction between the bones, joints, and natural or artificial muscles and tendons, such that it can be used to:

1. Simulate the curling and extension motion of the finger based on activation of the posterior muscles and anterior muscles.
2. Simulate different types of grasping motions; specifically simulate prehensile and precision grasping motions around a spherical object.
3. Visualize the simulation of the finger motions through three-dimensional animation.
4. Capture the contact forces on the fingers resulting from muscle actuation around the spherical object.

In Section 2, the physiological considerations in modeling the hand are discussed, including the necessary assumptions made. In Section 3, a closer look is taken at the hand model itself, involving both a component-by-component inspection, as well as a broader view at the package hierarchy. Section 4 follows with detail on the structure of the contact model developed in this paper. Section 5 handles the simulation of the models, as well as the corresponding analysis. Section 6 provides the final remarks and closes the paper.

## 2 Approach to the Physiology

The bones in the hand are treated as rigid bodies, and joints are modeled as a set of revolute joints, the number depending on the degrees of freedom in the joint's motion.

The muscles in the hand are composed of numerous sarcomeres (muscle fibers); these muscle fibers actuate in unison to produce the overall muscle force. The model of the muscle abstracts this actuation process into one total force - the input to the muscle component is the percent of the total muscle being actuated, and the output is the product of the percent value and the parameter value for the total muscle force (Marieb, 2000). This is done as sarcomeres actuate in an all-or-nothing manner; hence, for the muscle to vary the force of contraction, it must vary the total amount of sarcomeres firing – in essence, activating a portion, or percentage, of the total possible muscle force.

The muscle's complex structure is broken down into multiple line segments moving between attachment points, as an approximation to the curve, demonstrated in Figure 1.

The attachment points function as the skin, limiting the muscle to conform to the physiology of the hand itself. The nature of skin as a dividing middle layer between a bone and an object is included in the contact model. It acts as a buffer layer between the direct contact between the bone and object, serving to add a degree of compliancy. The tendons are assumed to act in conjunction with the muscles as massless bodies that connect contracting muscles with appropriate bone structures.

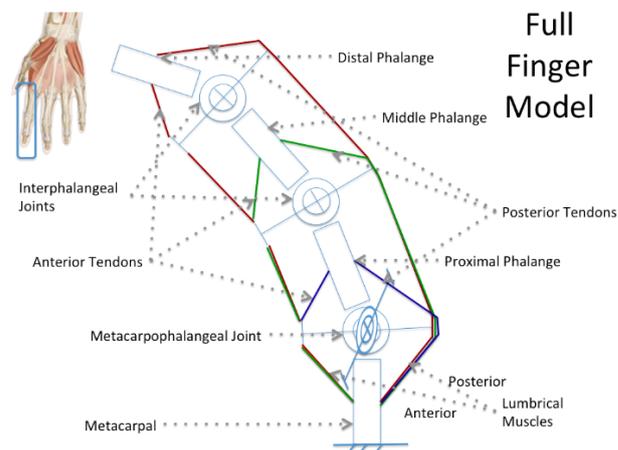


Figure 1: Finger Model Approximation schematic

## 3 Hand Model

### 3.1 Modeling Approach

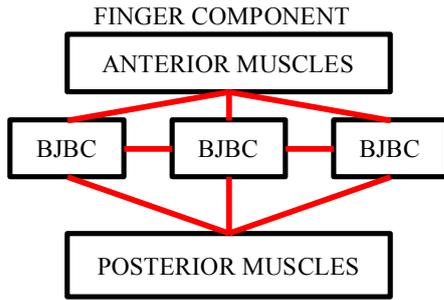
The musculoskeletal aspect of the hand can be broken down into component-based construction using bones and muscles. The basic component considered to have similar functional properties to the hand is called the

Bone-Joint-Bone component; it is constructed using two bones, a connecting revolute joint, and actuating muscles on both the anterior and posterior side.



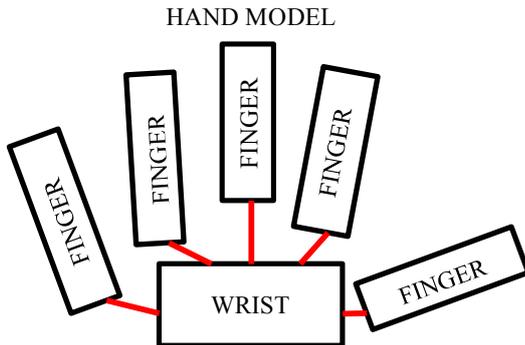
**Figure 2: Bone-Joint-Bone Component Structure**

A finger can be considered an extension of this idea; rather than having two bones and one joint, there are multiple bones, and multiple joints, between joints for normal flexing motion as well as sideways motion.



**Figure 3: Finger component structure**

The hand itself can be thought of as the joint workings of multiple fingers in unison, connected through a bone structure representing the wrist.

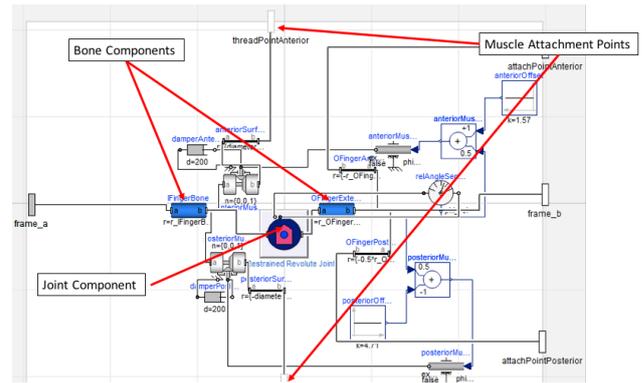


**Figure 4: Hand Model Structure**

The modeling approach relies on the component breakdown detailed above. By relying on the basic Bone-Joint-Bone component structure, the finger bones and the overall hand are constructed. Muscle components are added as appropriate to actuate the joints present.

### 3.2 Bone-Joint-Bone Component

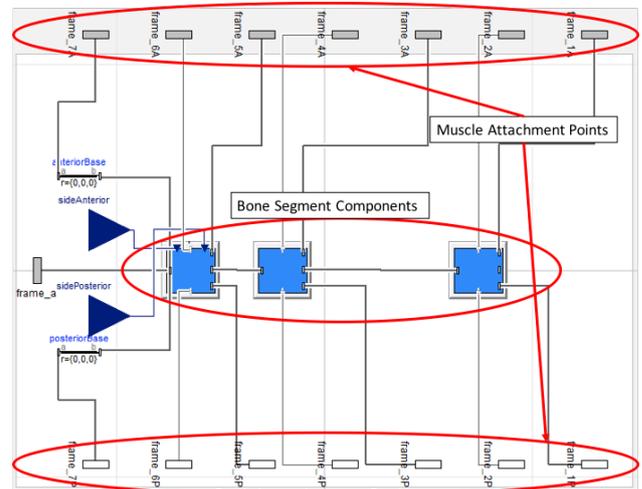
This component (BJBC) represents the basic structure of the bones and joints in the hand. The component is constructed using two rigid bodies representing bones, connected by a revolute joint representing a finger joint; there are attachment points designated on the bones as areas the muscle will actuate upon. The Double-Joint-Bone (DBJBC) component is an extension of this idea, with an additional degree of motion added to the joint, to allow sideways motion.



**Figure 5: Schematic of a Basic Muscle-Joint component**

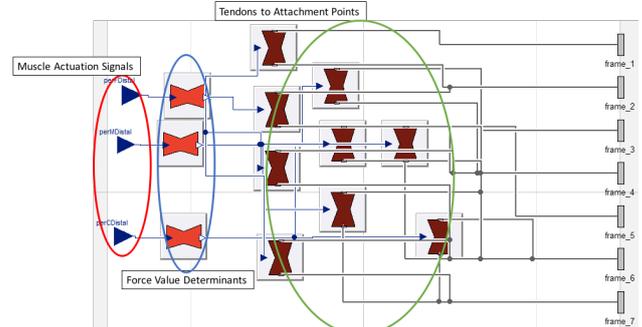
### 3.3 Finger Component

The finger is constructed by fusing two BJBC's and one DBJBC, to make four bones (metacarpal and phalanges) connected by three joints (metacarpophalangeal and interphalangeal joints) – as seen in Figure 6A.



**Figure 6A: Schematic of the Finger Bone Model**

There are muscle components for both the anterior and posterior side, connected to the bone at the attachment points (as seen in Figure 6B).



**Figure 6B: Schematic of the Finger Muscle Model**

The finger model additionally contains elements that model interface to a contact object, and is discussed in the next section; this is shown in Figure 6C.

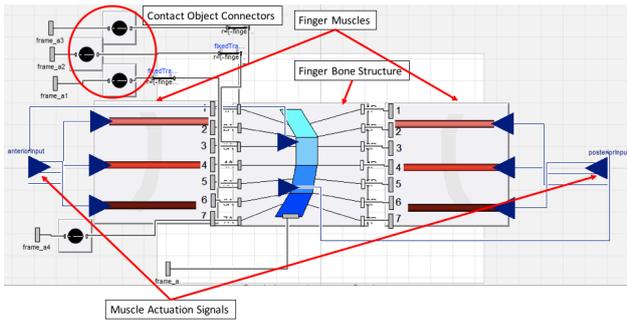


Figure 6C: Schematic of the Full Finger Model

### 3.4 Hand Component

The model is created by instantiating multiple finger components, each at a different position and orientation relative to the inertial frame. The non-opposable fingers each have axes of rotation rotated slightly (about 15 degrees) relative to each other, while the opposable finger’s axis of rotation is almost opposite to the axes of the other fingers.

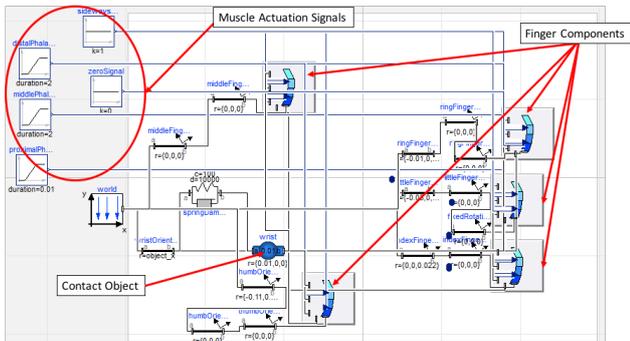


Figure 7: The Hand Component

### 3.5 Package Structure

The overall PowerGrab library consists of one main package, PowerGrabStructure (as seen in Figure 8), and a separate package for test models, named PowerGrabTestingRig (not shown in the figure). The division was made so that the main models can be assuredly independent of the testing models and other older versions.

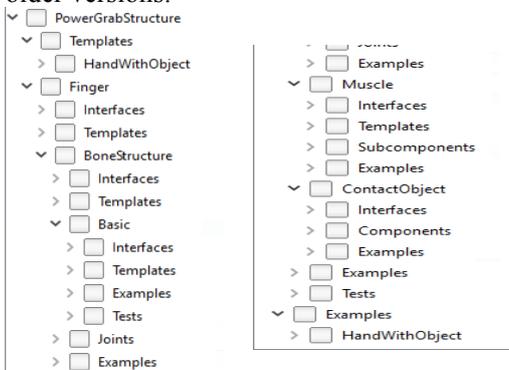


Figure 8: Package Structure

The PowerGrabStructure package contains the main components of the library, including the bone structures,

the muscle components, and contact object models. There also exist examples for each type of system, namely the basic bone-joint system, the finger-and-contact system, and the hand-and-contact system.

## 4 Contact Object

The contact object is modeled essentially as a semi-rigid sphere – a combination of nonlinear spring and damper systems that only exerts a force on the bone when a contact event occurs (below is a diagram of the contact object).

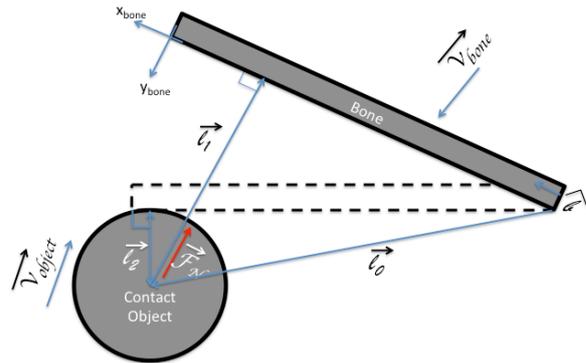


Figure 9: Spherical Object Contact Model

The object is represented as the combination of a point defining the center of the contact object, and a connector component between the object center and the potential point of contact on the bone. Each connector component details the contact dynamics between the contact object and the specific bone segment the component is connected to. Having separate connectors per each bone segment allows there to be multiple contact points per finger, one per each bone segment (for a maximum of 4 points per finger). However, as the connector follows a straight line, the contact is restricted to occur at a single point per each bone segment. Furthermore, as the contact model is designed for a spherical object, later models for other object shapes must be independently developed.

#### 4.1.1 Determining the point of contact

We define vector  $\vec{l}_0$  as the direct path from the base of the bone to the center of the contact object. Next, we define a vector perpendicular to the bone,  $\vec{l}_1$ , by subtracting the projection of the vector  $\vec{l}_0$  along the length of the bone from  $\vec{l}_0$ . Should the magnitude of  $\vec{l}_1$  fall below the radius of the object, we can then determine that contact has occurred.

$$\vec{l}_1 = \vec{l}_0 - (\vec{l}_0 \cdot \hat{i}_{bone})\hat{i}_{bone}$$

The actual implementation of this strategy in Modelica is simplified using a relative position sensing

component and a prismatic joint. A prismatic is attached to the base of the bone, and is free to slide along the x-axis, as resolved in the frame of the bone. Using a relative position sensing component between the center of the contact object and the nonmoving base end of the prismatic joint, resolved in the frame of the bone, the vector for the relative position of the center of the object is found. The component of that vector along the bone is determined (as the x-component of the vector, as it is resolved in the frame of the bound), and is subtracted from the overall relative position vector. This leaves us with the component of the vector that is perpendicular to the bone, which is the vector desired.

If and as the finger slips along the object, both the number and location of contact points will be updated accordingly.

#### 4.1.2 Determining the force of contact

As the contact object is spherical in nature; this allows for the following abstraction: the object is a spring that has a relaxed length of 0, with a nonlinear stiffness that becomes nonzero only when the stretched length is below a certain threshold (thus creating a zone of nonzero stiffness described by a threshold radius  $tR$ ). The force equation is thus as follows:

$$F_N(\vec{l}_1) = \begin{cases} k(tR - |\vec{l}_1|) * \hat{l}_1, & |\vec{l}_1| \leq tR \\ 0, & |\vec{l}_1| > tR \end{cases}$$

Apart from the normal contact force between the object and the bone, an additional force representing the effect of skin on contact is also applied. This is considered as a "buffer layer", as the skin will meet the contact object before the bone, thus acting as a buffering between the two. The skin is considered to have some pliability, and is therefore modeled as a spring connection between the contact object and the point of contact on the bone. Due to modeling purposes, the skin is assumed to be layered around the contact object rather than the bone itself, as it allows for the approximation that the skin-caused buffering force  $F_{Buffer}(|\vec{l}_1|)$  acts in the same manner as the normal contact force, albeit at a larger threshold range. This extension in force and range is reflected in the parameters  $bC$  and  $bR$ , respectively.

$$F_{Buffer}(|\vec{l}_1|) \begin{cases} bC, & |\vec{l}_1| \leq tR + bR \\ 0, & |\vec{l}_1| > tR + bR \end{cases}$$

#### 4.1.3 Determining the friction due to contact

The friction due to the contact between the bone and contact object is represented as a damping on the sliding motion across the surface of the contact object. The magnitude of damping is  $|\vec{F}_N| * |\vec{v}_{surface}|$ . The normal force magnitude is equal to the magnitude of the contact force on the bone, and the surface velocity is determined as the magnitude of the result of subtracting the vector

component of the relative velocity between the bone and contact object that is parallel to the radius from the overall relative velocity vector.

$$\vec{v}_{rel} = \vec{v}_{bone} - \vec{v}_{object}$$

$$\vec{v}_{surface} = \vec{v}_{rel} - \left( \vec{v}_{rel} \cdot \frac{\vec{l}_1}{|\vec{l}_1|} \right) \frac{\vec{l}_1}{|\vec{l}_1|}$$

## 5 Simulation

### 5.1 Component Testing

The purpose of the component tests is to determine if the component's performance conforms to the expected result. To test the bone structure components, muscle components are instantiated in the test models, to actuate the bone structures.

#### 5.1.1 Bone-Joint-Bone Component Test

The muscle actuation profile alternates between actuating the anterior muscle and actuating the posterior muscle, with small intervals of overlap. As seen in Figure 10, the limits on rotation are -0.5 and 1.6 radians, and the system can successfully reach those limits following sustained muscle activation. As the desired functionality is for the component to be able to undergo such motion, we conclude that the Bone-Joint-Bone Component can adequately support our needs.

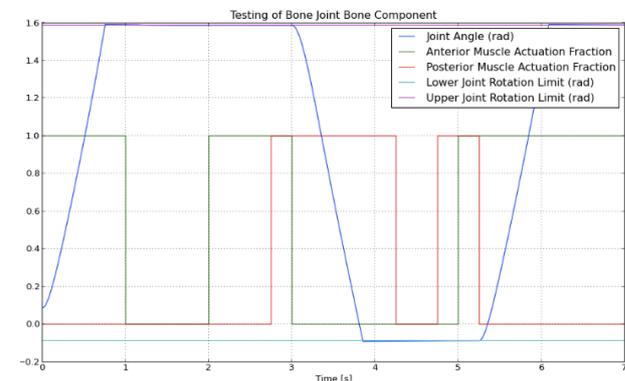
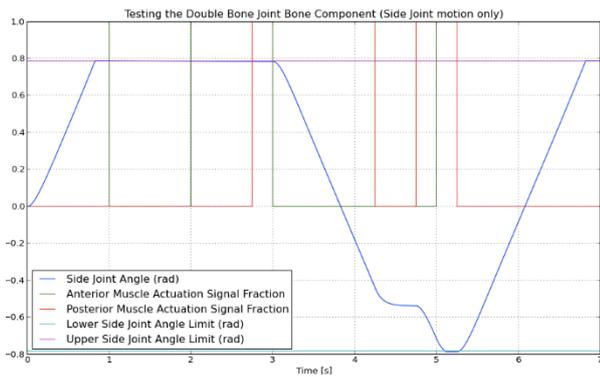


Figure 10: (Unit Level) test of BJB component

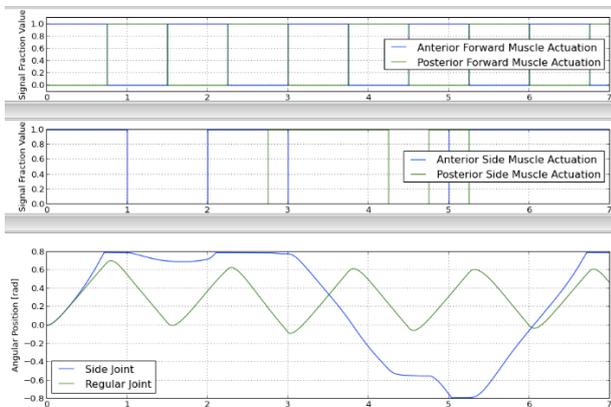
#### 5.1.2 Double-Bone-Joint-Bone Testing

Like the Bone-Joint-Bone component's test, the DBJBC component also utilizes muscle components to actuate the bones in the system. As seen in Figure 11, we actuate the side muscles using the same muscle activation profile used for testing BJB component, while keeping the other muscles inactive. The following angular displacement occurs in the Side Joint (note that the limits on the angle of rotation is different between the joints).



**Figure 11: (Unit Level) test of the DBJB component: Side joint actuation only**

Of special interest is the relationship between sideways motion and forward motion. It is reasonable to expect that, due to coupled dynamics, forward motion will cause motion sideways, and vice versa. In Figure 12, both the Forward Joint and the Side Joint are actuated. For the Forward Joint, it is sequential activation of the anterior and posterior muscles. The Side Joint follows the same activation profile used in the previous tests, and correspondingly experiences motion as seen in Figure 12 (bottom graph). There are also slight additional movements in the side joint, in conjunction with change in direction of movement in the forward joint, which can be attributed to the coupled dynamics.



**Figure 12: (Unit Level) test of the DBJB component, with simultaneous side and forward actuation**

Similar to the requirements for the Bone-Joint-Bone component, the requirement for this component is to be able to undergo such movement given appropriate actuation, without too much deviation from smooth motion. As such, we determine that this component is suitable for use.

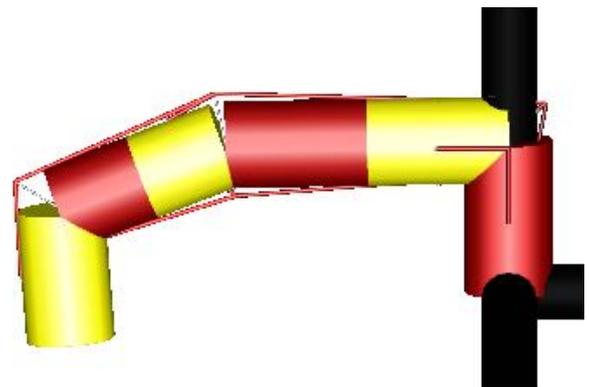
**5.1.3 Finger Component Test**

**5.1.3.1 Finger Testing without Contact Object - Results:**

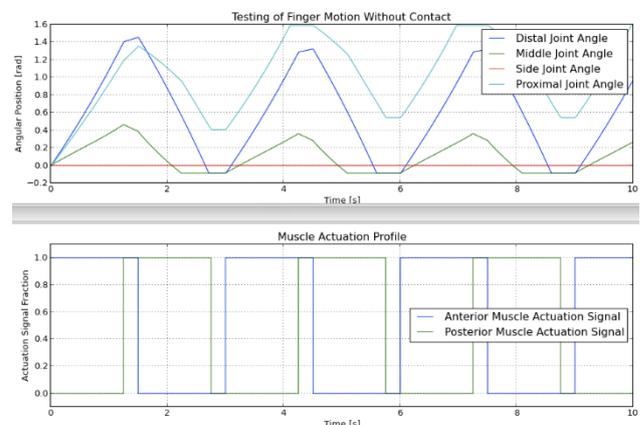
The finger muscles were sequentially actuated to enable flexion and extension. The activation is done in

a square wave pattern, as seen in Figure 14 (bottom), and alternates the anterior actuation with the posterior actuation. The resulting joint angles (Figure 14 top) indicate that the model successfully captures contracting motion, with the finger curling when the anterior muscles are actuated, and extending when the posterior muscles are actuated. A screen capture of the animation of the testing is seen in Figure 13.

These test results, in conjunction with the test results shown in the previous section, indicate that the first goal of the paper has been satisfied (to simulate the curling and extending motion of the fingers through actuation of the muscles.



**Figure 13: Animation of No Contact Finger Component Test**

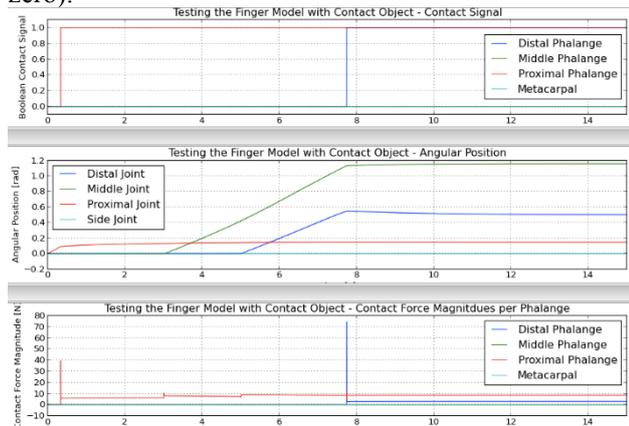


**Figure 14: Curling and extension tests for Finger model**

**5.1.3.2 Finger Testing with the Contact Object - Results:**

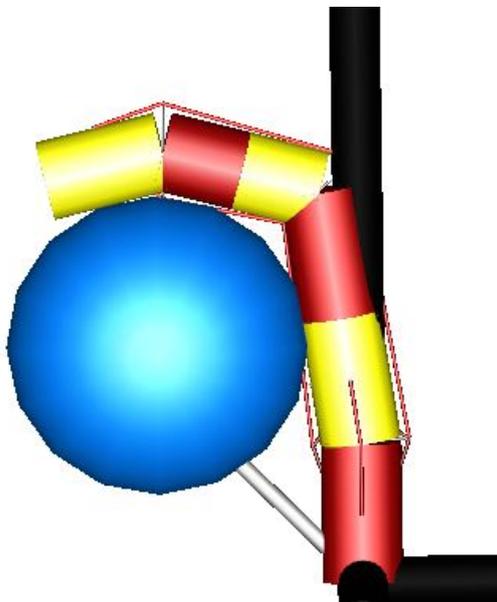
The previous test was repeated, but with the addition of a spherical contact object placed in front of the finger. The actuation profile is a staggered sequential activation from the proximal phalange to the distal phalange. The test's goal is to have the finger curl around the contact object when the object is positioned both directly in front of the finger, and positioned in front with a small offset to the side. As seen in Figure 15 below, when the object is directly in front, the finger curls around the object without slipping to the side. (Note that the middle

phalange and distal phalange contact the object at the same time, and that the side joint's angle is constant at zero).



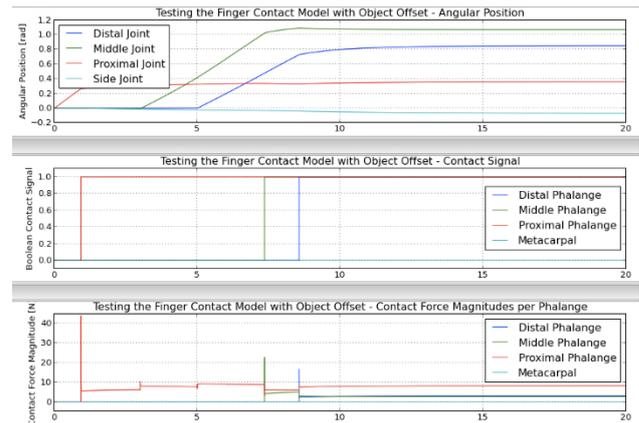
**Figure 15: Finger contacting spherical object “head-on”**

Figure 16 shows a screen capture of the animation of the head-on contact with the spherical object.

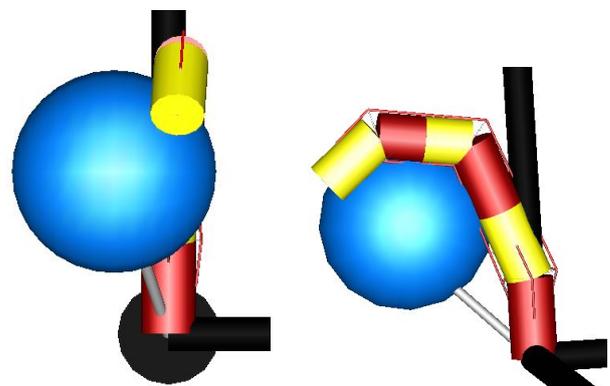


**Figure 16: Animation of the Finger contacting spherical object “head-on”**

The experiment is repeated with the object placed with a small offset. As seen in Figures 17 and 18 below, the finger still contacts the object, but proceeds to slide across the surface for a short period.



**Figure 17: Finger contacting spherical object with offset**



**Figure 18: Animation of Finger contacting spherical object (front and side views)**

The finger, during each trial, undergoes motion that conforms to expectation on how it should behave, and so is considered successfully tested.

A limitation observed is that the frictional force model, along with the high stiffness associated with the object's normal force, causes computational strain on the numerical solver during model simulation.

## 5.2 Hand Grasping Tests

The two grasps tested for in simulation were the prehensile and precision grasps. The precision grasp is a grasping motion that relies on relatively minimal muscle actuation to lightly hold the contact object; a prehensile grasp is when the muscles in the hand actuate to fully grab, and squeeze, the contact object in question (Cutkosky et. al., 1990).

The testing of the hand model, consisting of five finger component instantiations, is similar to the contact object test for the individual finger component. The muscles of the hand actuate, and cause the hand to contract. The purpose of the test is to determine if the hand can both perform a prehensile circular grasp around the ball, or a precision circular grasp around the ball. Separate actuation profiles were used for the prehensile grasp and the precision grasp.

Correspondingly, the object was placed at two different locations, depending on the type of grasp.

The results of the prehensile grasp (Figure 19) indicate that the fingers make and maintain contact with little sliding through the testing – all except the little finger – as indicated by the minimal movement in the side-joint’s angle. The contact object’s small radius of only 2.75 cm, in comparison to the fingers of average length 15 cm with diameter 2 cm, was chosen to demonstrate a typical hand grasp. The opposable fingers can maintain a firm grasp on the object, like how actual hands maintain holds on small objects. Some slipping occurs because there is not enough friction between the surfaces. The sliding motion could not be further reduced by increasing the friction, as doing so caused numerical issues; however, the slipping did not occur indefinitely, due to the side joint’s resistance to motion (as mentioned in Section 3.2.1).

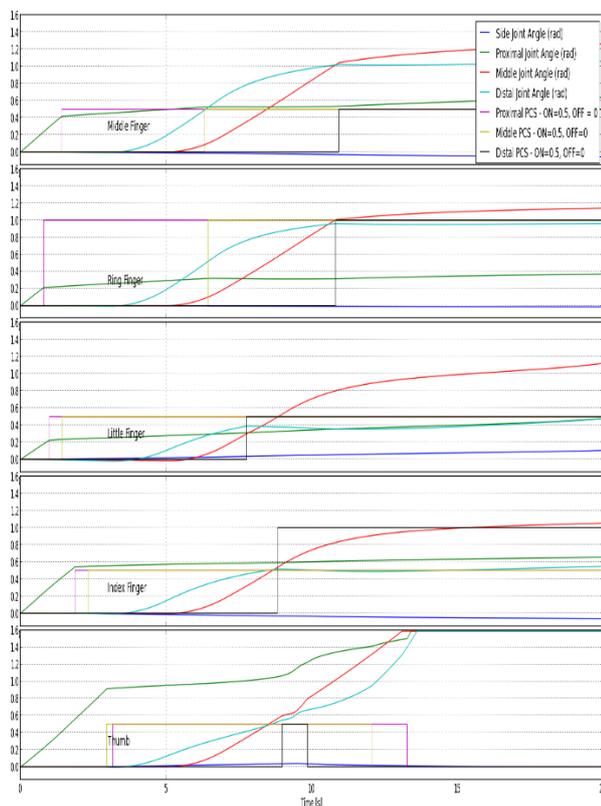


Figure 19: Prehensile Grasp Simulation

For the precision grasp, only the proximal phalanges were actuated, and the object was located closer to the distal phalanges. As seen from the results in Figure 20, contact occurs only as the distal phalanges – the little finger does not contact the object entirely, as it does not reach the object. Looking at the joint angles, we see that the proximal joints’ motion stops soon after the distal phalange contacts the object. The distal phalanges’ start to bend backwards upon with the object, as reasonably expected – once the distal phalanges stop bending backwards, the proximal phalanges’ motion stops as

well. During the entire grasp, there is minimal slipping exhibited at all the contact points.

The hand testing, for both the prehensile and the precision grasps, displayed both the contracting motion and the grasping characteristics desired.

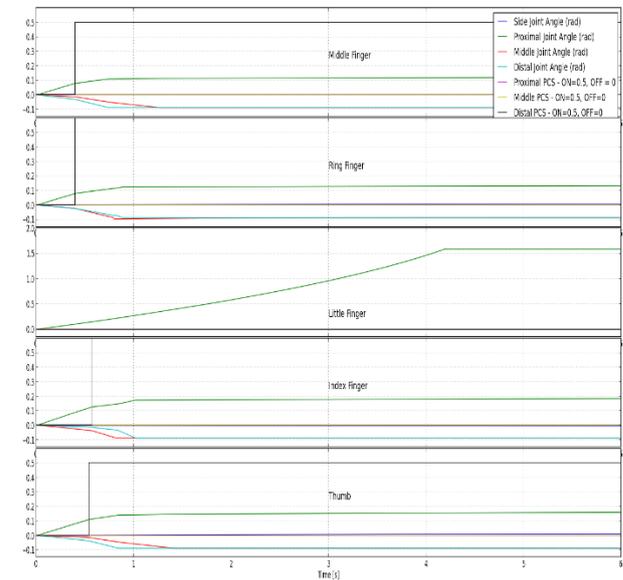


Figure 20: Precision Grasp Simulation

## 6 Conclusions

### 6.1 Results Summary

#### Implication with respect to Paper Goals:

The primary goals of this study were to be able to:

1. Simulate the curling and extension motion of the finger based on activation of the posterior muscles and anterior muscles.
2. Simulate different types of grasping motions; specifically - specifically simulate prehensile and precision grasping motions around a spherical object.
3. Visualize the simulation of the finger motions through three-dimensional animation.
4. Capture the contact forces on the fingers resulting from muscle actuation around the spherical object.

As seen from the results, goals 1, 2, 3 and 4 were successfully achieved.

### 6.2 Development Review

The initial construction of the model was done in Wolfram SystemModeler; the later models were developed in Dymola, leveraging its user-friendly refactoring and model creation capabilities. For compilation, debugging, and simulation capability, JModelica was used. OpenModelica was then separately utilized for animation (with simulation). All three services, apart from OpenModelica’s lack of a

CVODES solver, did not require major tool-specific code alterations to run. This allowed for smooth transition between tools during model development and testing. The non-tool-specific functionality of Modelica, and the support for Modelica from the tools SystemModeler, Dymola, JModelica, and OpenModelica, were extremely useful for this package's creation.

### 6.3 Further Work

Much of this model was based off simplifications to get the models and simulations running – future work should focus on refining these model assumptions.

#### 6.3.1 Musculoskeletal Model Improvement

Models of both the bone structure and the muscle geometry can be improved from its current state. The bone's physical parameters, including bone lengths and finger orientations, should reflect the actual structure of the bones. Currently, most parameters were chosen from an average of measurements taken of a group of volunteers; parameters not gathered from measurements were chosen arbitrarily. Using data selected from studies on hand dimensions and appropriately large ranges of volunteers would improve the validity of the resulting forces and motion involved in grasping. The muscles are currently modeled through linear line force segments between a limited number of attachment points to approximate the muscle's curve – this could be expanded by better representing the multiple interacting muscles and tendons, and their corresponding muscle geometries. Further improvements with the muscle actuation dynamics are also possible.

#### 6.3.2 Contact Model Improvements

The current contact model is based around a spherical object; this should become more generalized, for multiple geometrical shapes and surfaces. Furthermore, the model currently assumes that each interacting bone will have a maximum of one tangential contact point, and that skin acts as a mere buffer – the models should account for the hand's relative flexibility and pliability. Lastly, the friction model is constructed as a type of surface damping, but it would be more appropriate to include Coulombic friction as well. As the contact object was developed in an ad-hoc manner, improvements can be made by integrating standard existing contact model approaches.

#### 6.3.3 Testing Improvements

The testing of the hand grasping motion should be improved such that the muscle actuation profiles are not arbitrary pulses, but an imitation of natural activation profiles. Moreover, future work should also integrate experimental datasets from muscular grasps to make testing result analysis more accurate.

## References

- Cutkosky, M. R., & Howe, R. D. (1990). Human Grasp Choice and Robotic Grasp Analysis. In S. T. Venkataraman & T. Iberall (Eds.), *Dextrous Robot Hands* (pp. 5-31). Springer-Verlag.
- Griffin, W. B., Findley, R. P., Turner, M. L., & Cutkosky, M. R. (2000). Calibration and Mapping of a Human Hand for Dexterous Telemanipulation. *ASME IMECE 2000 Conference Haptic Interfaces for Virtual Environments and Teleoperator Systems Symposium*. Retrieved from [http://www-cdr.stanford.edu/DML/publications/griffin\\_asme00.pdf](http://www-cdr.stanford.edu/DML/publications/griffin_asme00.pdf)
- Adler A, "GM-NASA Space Robot Partnership brings "Power" Glove to Life", GM Corporate News announcement, 2017 July 6<sup>th</sup>, <http://media.gm.com/media/us/en/gm/home.detail.html/content/Pages/news/us/en/2016/jul/0706-gm-nasa.html>
- Linn D. M., Ihrke A. C., Diftler M. A., "Human grasp assist device and method of use", US Patent No. 8255079 B2, 2012.
- Polygerinos P, , Galloway K. C., Savage E., Herman M., O'Donnell K, and Walsh J. C., "Soft Robotic Glove for Hand Rehabilitation and Task Specific Training", 2015 IEEE International Conference on Robotics and Automation, May 2015, doi: 10.1109/ICRA.2015.7139597
- Yanchev T, "Power Assist Gloves", <https://www.youtube.com/watch?v=gzfZCTYREww> 2015
- van Nierop, O. A., van der Helm, A., Overbeeke, K. J., & Djajadiningrat, T. J.P. (2007). A natural human hand model. *The Visual Computer*, 24(1). <http://dx.doi.org/10.1007/s00371-007-0176-x>
- Gustus, A., Stillfried, G., Visser, J., Jörntell, H., & van der Smagt, P. (2012). Human hand modelling: kinematics, dynamics, applications. *Biological Cybernetics*, 106(11). <http://dx.doi.org/10.1007/s00422-012-0532-4>
- Wan Tarmizi, W. F. B., Elamvazuthi, I., & Begam, M. (2009). Kinematic and Dynamic Modeling of a MultiFingered robot Hand. *International Journal of Basic & Applied Sciences*, 9(10). Retrieved from <http://ijens.org/index.htm>
- Marieb, E. N. (2000). *Essentials of human anatomy and physiology* (6th ed.). San Francisco: Benjamin Cummings.
- Otter, M., Elmquist H, Mattson S. E., "The New Modelica Multibody Library", *Proceedings of the 3<sup>rd</sup> International Modelica Conference*, Linköping, 2003
- Tiller, M. (2014). *Modelica by Example*. Retrieved from <http://book.xogeny.com/>
- Modelica® (2013) - A Unified Object-Oriented Language for Physical Systems Modeling, Language Specification <https://modelica.org/documents/ModelicaSpec32Revision2.pdf>
- Hicks J. L., Uchida T.K., Seth A., Rajagopal A., Delp S.L., "Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulation environment", *Journal of Bioengineering*, Vol 137, Feb 2015.
- Radder, B., Kottink AIR, van der Vaart N, et. al, "User-centred input for a wearable soft-robotic glove supporting hand function in daily life", 2015 IEEE International

Conference on Rehabilitation Robotics (ICORR), Singapore, 2015, doi: 10.1109/ICORR.2015.7281249

Dymola (2017) Copyright © Dassault Systèmes, 1992-2016  
<http://www.3ds.com/products-services/catia/products/dymola/>

SystemModeler (2015) Copyright © 2015 Wolfram Research, Inc. <http://wolfram.com/system-modeler/>

OpenModelica (2016) Copyright Open Source Modelica Consortium (OSMC) <https://www.openmodelica.org/>

JModelica (2016) from <http://jmodelica.org/>

## Author's Notes



I was first introduced to Modelica during an internship at Xogeny in the summer of 2014, as a freshman in high school. This led to me modeling a flat linear motor I had previously built by hand. The linear motor consisted of multiple solenoids sequentially activated by an Arduino, to propel a cart down metal tracks. It was developed in an ad hoc manner, and adjusting the design was difficult; I felt that modeling was a nicer way of doing these kinds of tasks.

When noticing elderly persons having trouble opening doors and holding jars, I began thinking of ways to devise an orthotic glove to assist them. To gain a better understanding of the system, I decided to first develop a mathematical model of the hand. I began working on the PowerGrab project as a summer intern at Modelon KK, under the supervision of Dr. Andreasson. Since then, I have continued to work on developing the code for the PowerGrab library, as well as writing the corresponding paper, with the continued guidance of Dr. Andreasson's comments and feedback.

### **My Background:**

Shashank was born in Michigan, USA, and is currently a senior in high school. He was a summer intern at Modelon KK and Modelon, Inc., and is currently in the Wolfram Mentorship Program.