

Adaptive Geometrical Action Line for
Musculoskeletal Simulation



Jorim Geers

Utrecht University, Virtual Human Technology Lab
Supervised by dr. N. G. Pronost

ICA-3472345

January 25, 2015

Abstract

This master thesis project is focused on the comparison of the different types of action lines for a musculoskeleton and our proposed adaptive action line. Also the modeling of the fibers inside the muscle models is an important part, because the fibers will provide us information about the moment arms of the muscle models. The fibers are modeled using fiber templates by morphing the fibers of the template to the inside of the muscle model. For the representation of the fibers we use a mass-spring system and the muscle models are also represented by mass-spring systems. The proposed adaptive action line will start with a shape that fits inside the muscle model and during the simulation the adaptive action line will adapt according to its deformed shape. In the experiments the various action line representations will be compared to each other and our proposed adaptive action line representation based on their moment arm values throughout the simulations. The results of the experiments show that our adaptive action line has the potential to become an action line representation that can be used for other researches.

Contents

1	Introduction	3
2	Related Works	4
2.1	FEM and FVM Deformation	4
2.2	Mass-spring Deformation	6
2.3	Other Deformations Methods	7
2.4	Projects Relation	9
3	Fiber Modeling	10
3.1	Fiber Templates	10
3.2	The Scalar Field	11
3.3	Contours	12
3.3.1	Contour Making	13
3.3.2	Contour Matching	14
3.4	Fiber Morphing	16
3.5	Mass-Spring System	18
4	Action Line Representation	21
4.1	Commonly Used Action Lines	21
4.2	Building Methods for Action Lines	22
4.3	Adaptive Action Line	22
5	Moment Arm Calculation	24
6	Results	25
6.1	The Setup	25
6.1.1	Bullet Physics Engine	25
6.1.2	Muscle and Bone Models	25
6.2	The Experiments	27
6.2.1	Comparison Common and Built Action Lines	27
6.2.2	Adaptive Action Line Results	31
6.2.3	Comparison Common and Adaptive Action Lines	34
7	Conclusion and Future Work	37
	References	39
A	Moment Arm Simulation Graphs	41
A.1	Biceps Femoris A	41
A.2	Biceps Femoris B	42
A.3	Gracilis	43
A.4	Iliopsoas D	44
A.5	Iliopsoas F	44

A.6	Inferior Gemellus	45
A.7	Pectineus	45
A.8	Piriformis	46
A.9	Quadratis Femoris	46
A.10	Rectus Femoris	47
A.11	Sartorius	47

1 Introduction

In the past, researchers have developed methods to simulate humans but also other creatures. The goal was, and still is, to make those simulations as realistic as possible and therefore they also did research in the field of the musculoskeleton. The musculoskeleton is a skeleton that also includes muscles which are attached to the bones of the skeleton. Because humans use their muscles to make a motion, the field of research of the musculoskeleton is very promising. Researchers are already able to simulate different types of muscles and use those simulations in various ways, e.g. make a character maintain balance or walk based on the muscle properties.

The simulations of the musculoskeleton use the muscle properties to calculate the moment arm or the muscle force that the muscle is capable of. To calculate the moment arm we need a line of action of the muscle. The action line is basically a path through the muscle that represents along which way the muscle will contract. The better the representation of the action line, the better the force calculation will become, but also more complicated. Researchers usually use action lines that are just represented by polylines. A disadvantage of these simple polyline methods is that the action line most likely will be on the outside of the muscle if the muscle will bend.

Because we like our simulations fast but also as precise as possible (or needed), we did research on the representation of the action line and propose a method that will adapt the action line to have a better fit than the polylines that others often use. This adaptive action line will be based on a degree threshold that will determine how quickly it is allowed to become more detailed.

We will consider the straight action line and a couple of other action line representations and compare those with our method to measure the difference in moment arms and therefore on the control of musculoskeletal models.

The goal of this master thesis project is to compare commonly used action lines with an adaptive action line in a simulation environment that also has more focus on speed than on accuracy of the muscle deformation. Based on the results of the simulation an action line representation method will be chosen as a best action line representation. Also the maximum degree threshold for which the representation of the deformation is still good enough will be determined for the adaptive action line.

In this master thesis report the modeling of the fibers of the muscle will be discussed in Section 3. The representation of the different types of action lines will be discussed in Section 4. The calculation of the moment arm will be discussed in Section 5. Then the setup and the results of the experiments will be discussed in Section 6. The end conclusion will be made in Section 7 and possible future work is also discussed there.

2 Related Works

A lot of research has been done in the field of muscle tissue deformation. An overview of researches [13] shows it is possible to use different methods to deform muscle tissue. Mass-spring models can be used for this kind of deformation. More popular methods are the Finite Element Method (FEM) and Finite Volume Method (FVM). The popularity of these two methods is probably due to the fact that it is harder to keep a constant volume with mass-springs systems and a constant volume is one of the essential properties of muscle tissue. Fiber representations in the muscle models are often different, but if the deformation method is the same, the fiber representation is sometimes used from another research. Also the researches cover different action line representations. In this Section the researches are sorted by their deformation method and if they use an action line, the representation of the action line will be discussed. We start with discussing literature that uses FEM or FVM deformation. Then the literature with mass-spring deformations is discussed and after that we discuss the literature that uses different types of deformation.

2.1 FEM and FVM Deformation

The popularity of FEM is clearly visible in the number of researches that use FEM. To start with, FEM is used to model and manipulate various alphabetic characters [20]. In those characters there are some muscle action lines (see Figure 1 (left)) that can be activated and which causes contraction along the muscle action lines. With the use of controllers it is possible to control the contractions and make the characters perform motions like jumping. The alphabetic characters do not have any muscle fiber information, unlike with the digitized fiber field method [18]. In addition to the FEM deformation the authors embed digitized fiber fields in the model by having a surface wrapped over the digitized fibers. Then the wrap surface is deformed to match the target mesh geometry and this will result in fibers fields that fit in the muscle model (see Figure 1 (right)). Those digitized fiber fields contain the data of the fibers in the specific muscle, e.g. the average fiber direction and the number of fibers in the muscle. By using those fibers and their adjusting length and volume it gives a better visual representation of the deformed muscle. One digitized fiber field is only useable for its corresponding muscle, but if someone would like a more generic approach the fiber template [24][5] would be a good choice. The muscles are also deformed using the FEM method, but for the fiber representation the authors use templates that contain information about the fiber structure for different muscle types. Using these templates the authors are able to track each individual fiber inside the muscle models and calculate the moment

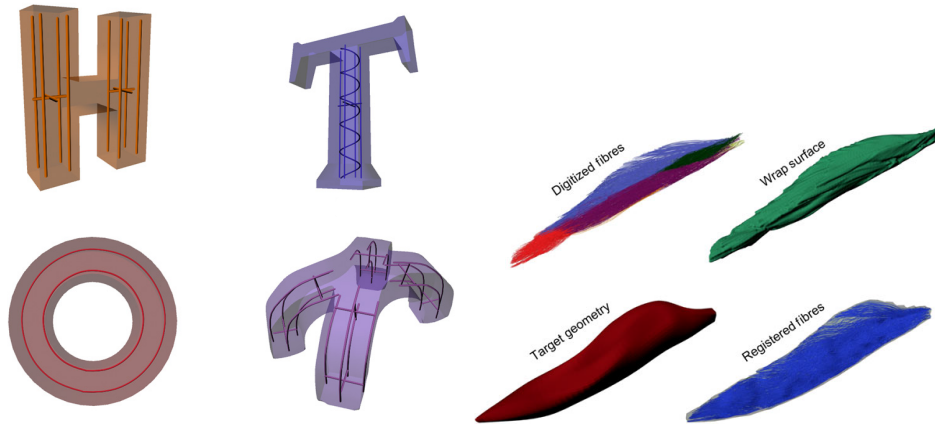


Figure 1: Left: the alphabetic characters with different types of action lines e.g. straight and circular (image from [20]). Right: the digitized fibers and their wrapping surface will result in the registered fibers once the wrapping surface is mapped to the target geometry (image from [18]).

arm of the fibers. The fiber templates are less specific than the fiber fields, because the digitized fiber fields are unique for every muscle model but one fiber template can be used for multiple muscle models. Another possibility for representing fibers of a muscle model is using NURBS [16]. Here a torus is used as an example and the fibers are obtained by shrinking the the original torus mesh (see Figure 2). Contractions are simulated along the fibers and their direction. Another method for obtaining the fibers was to have the fibers aligned parallel along one axis. For now we only assumed muscles consisting of a single model, but there is also research in muscles consisting of multiple models [4]. The multiple models of the muscle are linked via barycentric embeddings which should decrease the finite element resolution that is needed to predict deformation of a muscle. Examples of different types of models the authors use are 1D Hill-type representations for the fibers and 3D FEM for muscle and tendon materials.

The FVM method is less popular compared to FEM, but should not be underestimated. The researches that use FVM [21][22] claim that their deformation method is better than FEM. According to the authors, the deformation method needs fewer calculations and does not need to precompute and store any extra information compared to FEM. The authors also show how to calculate the stress of a muscle based on the activation level of that muscle. To represent the fibers of the muscles, the authors use B-spline solids.

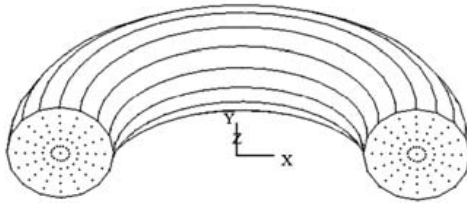


Figure 2: The fibers of the torus when they are obtained by shrinking the original torus mesh (image from [16]).

2.2 Mass-spring Deformation

The mass-spring systems are also used in a lot of research, but most of it is older research or used for fast visual representation. An older but important research shows the compression and stretching of a muscle model [17]. This muscle model is based on a mass-spring system and has an action line through its center. The authors also show that the angular spring (see Figure 3 (left)) is an important part of their mass-spring system, because that type of spring makes the deformation of the muscle more realistic. Besides complete muscle models made of a mass-spring system there are also mass-spring systems that are used for representing action lines [3]. The authors use force fields to prevent collisions of the mass nodes of the mass-spring system with the bones when the action line is adjusted due to movement of the bones. The muscle mesh is weighted against the action line by parameterization of the vertices of the mesh. With this parameterization every vertex knows where it lies along the action line and what its distance to the action line is. When the action line changes length or position, the position of the mesh vertices can be recalculated using their parameterization. So the mesh depends on the action line properties and this results in a muscle mesh that will stretch when the action line is also stretched. If we combine this research with the work of Nedel [17] we get the research project of Lee [14]. The difference is that the action line in this method is a quadratic Bezier curve instead of a polyline consisting of springs. The shape of the muscle is also based on the shape of the action line and its change in length, i.e. if the action line stretches, so does the muscle shape (see Figure 3 (right)). Another difference is that the used springs are working on the control points for the action line and not directly on the muscle. With their method the authors are able to simulate an entire human body with 200 muscles in real-time. This research is mainly focused on getting good visual muscles as quick as possible, but another research focuses more on accuracy of the simulation of breathing [25]. Springs are used for forces between individual ribs and mass-spring systems are also used for the deformable bodies (see Figure 4 (left)). Using the mass-springs system the authors are able to produce breathing motions through muscle activation that causes the ribs

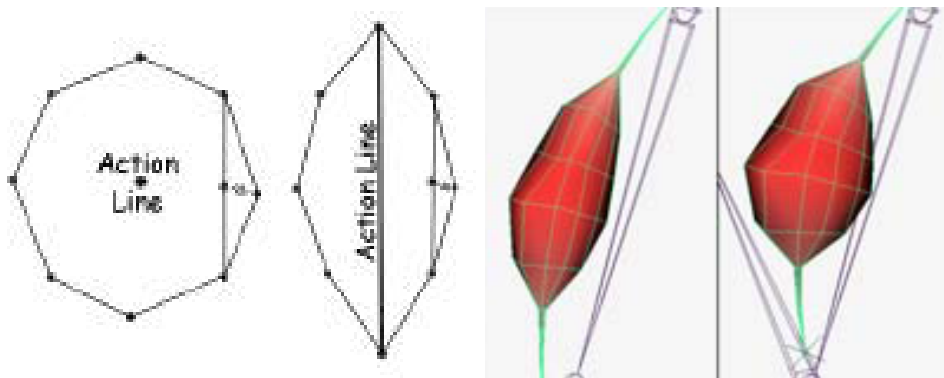


Figure 3: Left: the action line is visible in the muscle model and the horizontal and vertical angular springs are illustrated (image from [17]). Right: The deformation of the muscle model. The muscle shape is compressed, because the action line becomes shorter and the muscle shape still follows the curve of the quadratic Bezier (image from [14]).

to move and deform their gut-body. There is also research that animates the face using a mass-spring system [15]. The mass-spring system is used for modeling the different layers of muscle and skin of the face (see Figure 4 (right)). The layers are connected to each other and the muscle layer is attached to the bones with springs. When the skull moves, the layers of skin and muscle will deform accordingly. A texture is matched to the topmost layer of the skin and that results in an actual face on the mesh. The authors are able to produce several good facial animations with the addition of eyelids opening and closing, teeth being visible when the mouth opens, and using a rotatable neck. As we can see with the different researches is the mass-spring system usable for various goals and we will be using it for the simulation of the muscle models and the fibers.

2.3 Other Deformations Methods

In other literature it is not quite clear what deformation methods they use (FEM, FVM or mass-spring systems) or they use another approach for deformations. There is a research that deforms the shape of the muscle depending on their corresponding profile curve [6]. Those profile curves are based on the origin and insertion points of the muscle and when a bone moves, the origin and insertion area are possibly moved closer to each other which will cause the profile curve to change (see Figure 5). The action line of the muscles is either a straight line or a cubic curve when there would be multiple line segments for the action line representation. Another research gives a short overview of two different strategies to define the fibers in a muscle [10].

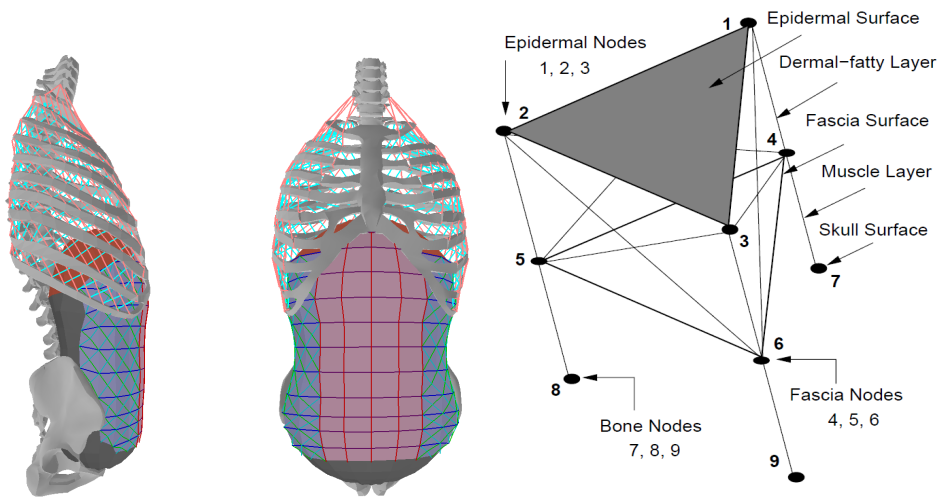


Figure 4: Left: the springs between the ribs and the modeling of the gut-body (image from [25]). Right: the different layers of the face and an overview of how those layers are connected together (image from [15]).

The surface-first strategy changes the shape of the mesh first when there is a movement in the bones and then it will decompose the shape of the muscle into fibers based on the fiber templates [5]. The fibers-first strategy first decomposes the shape of the muscle into fibers and when the fibers are transformed into a mass-spring system the fibers are changed according to the movement of the bones. A different research estimates the accuracy of muscle moment arms in MRI-based muscle model [2]. Here the action lines are represented by multiple line segments and via points and wrapping surfaces. The moment arms that are presented are from both the computer simulation and the experiment using real bones and muscles. In some cases the moment arms are quite similar (e.g. the semitendinosus moment arms are almost equal with the hip rotation experiment), but they can also be different (e.g. the psoas moment arm values end with decreasing values in the computer simulation, but the real life experiment has increasing values at the end).

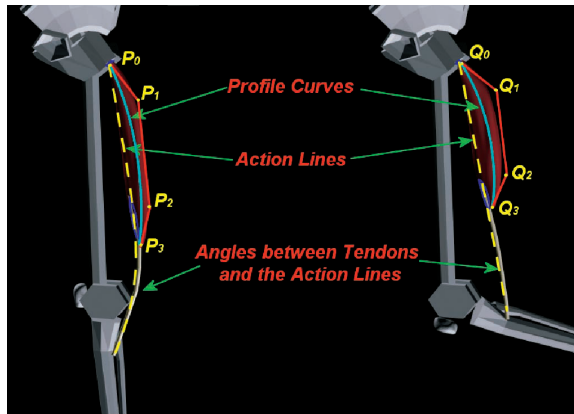


Figure 5: Visualization of the profile curves and the action line (image from [6]).

2.4 Projects Relation

In this project the fiber templates will be used for the modeling of the fibers, but it will be in a mass-spring deformation environment instead of a FEM deformation environment. We are going to compare the different action line representations that are used in the literature (e.g. the straight action lines and the quadratic Bezier) to see which action line representation will perform best. The comparisons are made based on moment arm values which will be calculated using the same formula as the related literature [24][5].

The fact that not all action lines in the literature take collisions with bones into account, the action lines could penetrate bones and this would not correctly represent the direction of the contraction of the muscles. We will propose a method for an action line representation that should solve this, but still has a simple representation and shape. The proposed action line representation will be compared to the action line representations from the literature to find out if it will perform well.

3 Fiber Modeling

In order to represent the muscles we do not only need to have a 3D model mesh of the muscle, but we would also like to model the inside of the muscle. The inside of the muscle will be represented by fibers which we will generate. The generation of the fibers is based on the method described in [11]. The fibers will be based on the fiber templates which will be explained in Section 3.1. The fibers are generated in a couple of steps and those steps contain generating the scalar field which will be discussed in Section 3.2. After the generation of the scalar field the contours will be made and that process is described in Section 3.3. After we found the contours, we can morph the fibers from the template into the contours and this is described in Section 3.4. And in Section 3.5 is described how to go to a mass-spring system from the morphed fibers.

3.1 Fiber Templates

Every muscle consists of numerous fibers and every fiber can cause a contraction for the muscle. Because fibers are the building stones of a muscle it is important to have a good representation of the fibers. There has been research done in this field and a leading technique comes from Blemker and Delp [5]. Blemker and Delp describe fiber templates that will represent fibers based on different geometry.

According to Blemker and Delp there are four different types of muscles and each type will have its own fiber template. There are parallel muscles, pennate muscles, curved muscles and fanned muscles. Each of these templates is a unit cube and uses a set of spline curves that form the basis of interpolation. The most simple case is the parallel muscle type and that only uses one set of spline curves, but the most complex case is the fanned muscle type and that template uses four sets of spline curves (see Figure 6). Every template has an origin area and an insertion area i.e. areas where the fibers will start and where they will end. Where the origin and insertion area are will depend on the type of the fiber template, but in the case of the simple parallel muscle type the origin and insertion area will respectively be the top plane and the bottom plane of the fiber template.

Our experiments are focused on the muscles that can be modeled using the simple parallel fiber template. What fiber template for what muscle model is given by anatomical data. Our method can easily use other fiber templates as long as the spline curves of the templates are represented as polylines (preferably with small line segments for better accuracy).

By only using the simple parallel muscle type template we can also simplify the fiber representation to a straight line instead of a spline curve. The number of fibers is variable and can be defined by the user. The fibers will be distributed evenly over the origin and insertion plane.

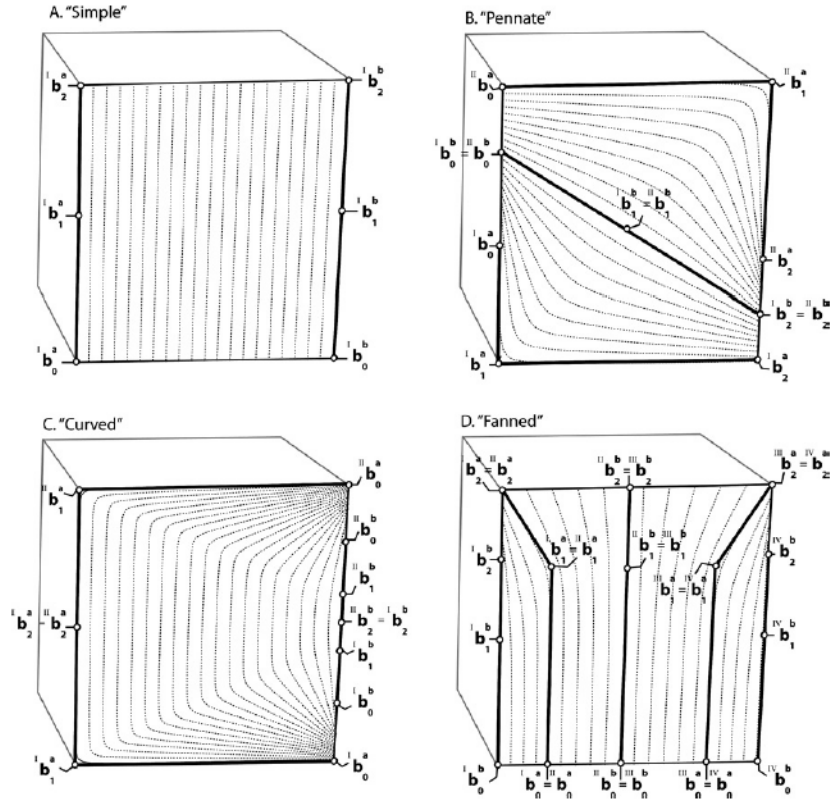


Figure 6: The different types of muscle fiber templates. The parallel muscle (A), the pennate muscle (B), the curved muscle (C) and the fanned muscle (D). Image from [5].

3.2 The Scalar Field

To know how a muscle model propagates from the origin to the insertion a scalar field method is used in [11] and that scalar field method is based on [7]. The scalar field will be computed based on the 3D model mesh of the muscles. We need to compute a scalar field u with scalar value u_i for every vertex v_i of the mesh such that we will not have local extrema except at the boundaries i.e. the origin and insertion area of the muscle. Without the local extrema we will be able to get to the origin or insertion area by going to a neighboring vertex with either a lower or higher scalar value than the current (randomly) chosen vertex. Such a method is described in [7] and will fulfill the following constraints. If vertex v_i is part of origin area, the scalar value u_i must be k_{min} and if v_i is part of the insertion area, u_i must be k_{max} . The constant values k_{min} and k_{max} have to be positive real numbers e.g. 1 and 5. If v_i does not belong to the origin area or the insertion area,

then the constraint is:

$$u_i - \sum_{j \in \Omega} w_{i,j} \cdot u_j = 0 \quad (1)$$

where Ω is the set of vertices connected to v_i by the use of an edge $e_{i,j}$. $w_{i,j}$ is the discrete harmonic weight which will be assigned to the edge $e_{i,j}$ and $w_{i,j}$ is calculated by using

$$r_{i,j} = \frac{1}{2} \cdot (\cot \alpha_{i,j} + \cot \beta_{i,j}) \quad (2)$$

$$w_{i,j} = \frac{r_{i,j}}{\sum_{j \in \Omega} r_{i,j}} \quad (3)$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are the triangle angles opposite the edge. This means that an edge belongs to two triangles, otherwise one of the angles would just be 0.

Because the constraints are linear we can rewrite them into a vector form and get the linear system $A \cdot u = b$, where u is the scalar field we seek. In the linear system b is a vector such that b_i is equal to k_{min} if vertex v_i belongs to the origin area, or b_i is equal to k_{max} if vertex v_i belongs to the insertion area and for the other vertices $b_i = 0$. Matrix A in the linear system contains 1 on its diagonal and $A_{i,j} = -w_{i,j}$ for non-boundary vertices, otherwise $A_{i,j} = 0$. Now that A and b are known, we can solve the linear system and find our scalar field u using a LU decomposition. Both [11] and [7] mention that any LU decomposition solver will work, but because the linear system is not only diagonally dominant, it is also sparse and it would be more efficient to use a solver that supports such matrices.

Calculating the scalar field could be heavy on the computation time, especially if the used 3D mesh is very detailed i.e. has a lot of vertices. Because the scalar field computation is a preprocessing step it is recommended to save the scalar values and just load them for simulations. In Figure 7 you see a scalar field computed for a 3D model mesh of the Rectus Femoris.

3.3 Contours

Now that we can compute the scalar field for a mesh, we can create the contours of the mesh based on those scalar values (like in [11]). A contour is an ordered list of points on the mesh with the same scalar value and when the points are connected we have a shortest path over the outside of the mesh. How to create the contours is explained in Section 3.3.1. After the contours of the mesh are found, the contours of the mesh should be matched to the fiber template (see Section 3.1) to be able to morph the fibers in the contour later. The process of matching the contours is described in Section 3.3.2.

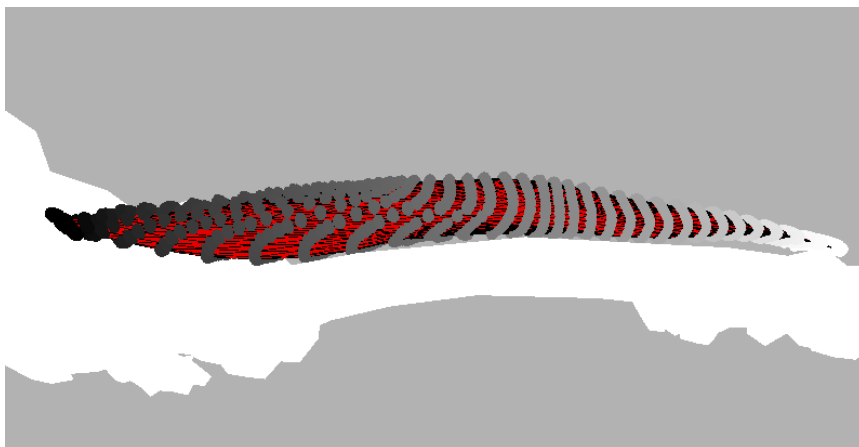


Figure 7: The scalar field of the Rectus Femoris visualized. The highest value is represented with a white color (insertion) and the lowest value is represented with a black value (origin).

3.3.1 Contour Making

A contour will be created based on a sample value s in the interval $\langle k_{min}, k_{max} \rangle$. With s known we can use a piece-wise linear function to find points on the 3D model mesh where $u = s$ and when those points are found we can make a piece-wise linear contour line.

In [11] the contour points are found and interpolated over the triangles of the 3D mesh, but we will find and interpolate the points along the edges of the mesh. The problem with the contour based on the triangle-interpolated points is that certain corners of the mesh will be skipped (see Figure 8 (a)) and you get a smoothed version of the contour instead of the complete outline. When using the points along the edges of the mesh, the contour will go around the whole mesh (see Figure 8 (b)) and this will not result in a smoothed version which we would like to avoid, because in our opinion a contour should follow the complete outside of the mesh shape and not take possible shortcuts between different triangles.

Finding points on the edges where the scalar value equals sample value s is a matter of linear interpolation between vertices v_i and v_j . Once those points are found they have to be placed in the correct order to make the contour. We make use of the 3D mesh topology to make it easy to obtain the contour. This ordering is illustrated in Figure 9. We pick the first point (p1) of the found contour points and see on what edge it lies. The next step is to find one of the two triangles the found edge belongs to (in the example the yellow triangle in (b)) and once we found that triangle we can determine the other edge that contains a point of the contour. After we

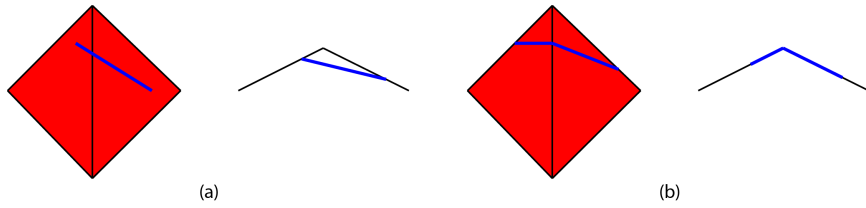


Figure 8: Example with front and top views of the difference between the triangle-interpolated points and the points along the edges. (a): in the top view it is clearly visible that the corner is skipped. (b): the corner is taken into account when the points are along the edges.

found that next edge and its contour point (p2) the search continues for the next triangle (yellow triangle in (d)). This process is repeated until all the contour points have been placed in the correct order and then we have our contour of the mesh, in the example this ends when p1 is connected to p2, and p2 is connected to p3.

The number of contours can be specified by the user. In the implementation the number of contours influences the interval $\langle k_{min}, k_{max} \rangle$ (and thus also the scalar field values), because k_{max} becomes the number of contours specified. That means the user must specify more than two contours to have more contours than just the ones at the origin and insertion area.

Figure 10 gives an example of contours of a 3D mesh.

3.3.2 Contour Matching

The contours of the mesh have been found, so now it is time to match those contours to the fiber template. This matching is important because without the matching we would not know how to morph the fibers from the inside of the template to the inside of the mesh contours. Part of the matching also involves slicing the fiber template. Because there are a certain number of contours, the fiber template will have the same number of slices. With the same number of fiber template slices and 3D mesh contours, the matching will be from a contour to its corresponding slice of the fiber template.

The slice of the fiber template and the contour could have different number of points for their representing polygons. The slice of the fiber template always has 4 points (the 4 corner points of the slice) so the contour must contain at least 4 points. If the contour does not have 4 or more points, the contour segments will be subdivided to ensure that it will have 4 or more points.

Now suppose that the contour consists of m segments and $m \geq 4$ and thus it could be represented by the sequence of vertices $P_0 \dots P_m$ where $P_m = P_0$. The contour of the fiber template slice is formed of $n \leq m$ segments and thus could be represented by the sequence of vertices $Q_0 \dots Q_n$ where $Q_n = Q_0$.

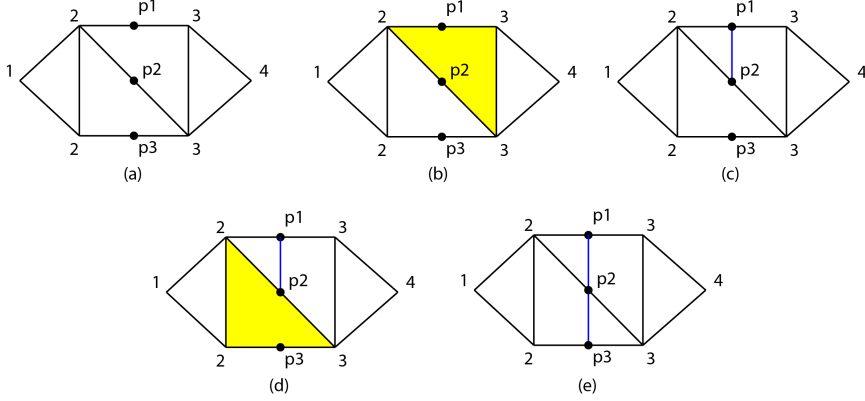


Figure 9: Example of the contour ordering. The numbers 1 up to 4 are the scalar values of the vertices. The contour will be based on the scalar value 2.5 and we find the points p1, p2, and p3 (a). We start with p1 and see that its edge belongs to the yellow triangle (b). While looking in the triangle edges for an edge with a contour point along it we find p2 (c). We find a new triangle (d) and connect the other point we found with that triangle (e).

It is the task to subdivide the contour of the fiber template slice in m segments and establish the one-to-one correspondance between the vertices of the mesh contour and the vertices of the fiber template contour.

We start with parametrization of the two polygons and assign the parameter tP_i to vertex P_i and assign parameter tQ_j to vertex Q_j . To find those parameters we use:

$$tP_i = \frac{\sum_{k=1}^i |P_k - P_{k-1}|}{\sum_{k=1}^m |P_k - P_{k-1}|}, tQ_j = \frac{\sum_{k=1}^j |Q_k - Q_{k-1}|}{\sum_{k=1}^n |Q_k - Q_{k-1}|} \quad (4)$$

From these equations we can see that $tP_0 = tQ_0 = 0$ and that $tP_m = tQ_n = 1$.

The next phase is to match the corners of the fiber template contour with vertices of the mesh contour. Starting with matching P_0 and Q_0 because their assigned parameters are similar (both are 0). After the first matching the algorithm starts to look for a vertex P_i whose assigned parameter is bigger than the assigned parameter of the first unmatched vertex Q_j of the fiber template contour i.e. $tP_i > tQ_j$. We will check if tP_i is closer to tQ_j than tP_{i-1} and the closest of those two will be the one that will match with Q_j . If a vertex P_i is not matched with a Q_j , then a point will be linearly interpolated between Q_{j-1} and Q_j which will have a direct match with P_i . The linear interpolation is based on the parameter values:

$$Q_{P_i} = Q_{j-1} + \left(\frac{tP_i - tQ_j}{tQ_j - tQ_{j-1}} \right) (Q_j - Q_{j-1}) \quad (5)$$

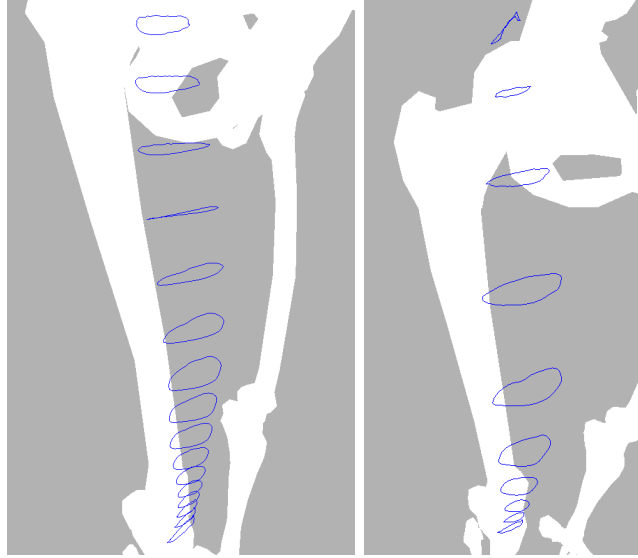


Figure 10: On the left is an image of the contours of the Rectus Femoris based on 20 slices. On the right is an image of the contours of the Rectus Femoris based on only 10 slices.

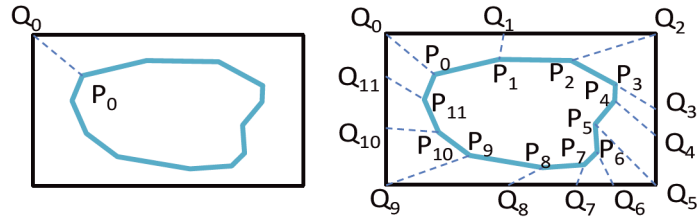


Figure 11: An example of the matching process between the template contour Q and the mesh contour P . Image from [11].

Once there is a match with a vertex of the fiber template contour, the algorithm will continue with the next vertex of the template contour until all of the mesh contour vertices have been matched to an interpolated vertex or a vertex of the template contour. Figure 11 shows an example of the matching process and Figure 12 shows a result in the matching process and the one-to-one correspondence.

3.4 Fiber Morphing

With the match between the fiber template and the mesh contour it is possible to morph the fibers from the template to the inside of the mesh contour. The morphing is based on the algorithm described in [9]. A morphing example can be seen in Figure 13.

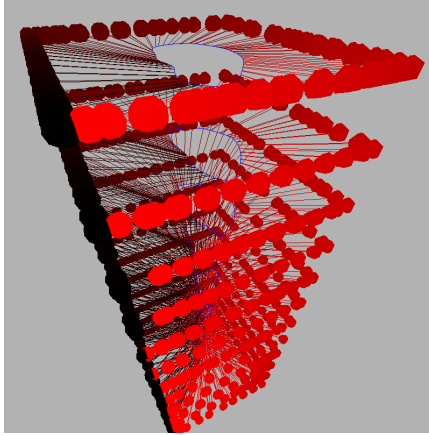


Figure 12: A result of the matching process on the contours of the Rectus Femoris.

For the morphing a couple of things are needed. Of course the contour of the mesh and the matching information are needed, but also the position of the fiber in the template slice is needed. The fiber position can be found by interpolating the spline curves (or in our case the straight line) that represent the path of the fiber on the inside of the fiber template. Once the fiber positions are known the algorithm will assign a weight λ_i to every matched template contour point Q_i for every individual fiber position. Because Q_i is matched to a mesh contour point P_i , the weights found for Q_i can be used with P_i . The morphed fiber position F_j is then:

$$F_j = \frac{\sum_{i=1}^m \lambda_i \cdot P_i}{\sum_{i=1}^m \lambda_i} \quad (6)$$

because the sum of weight λ_i can be bigger than 1.

The disadvantage of this method is that it does not handle concave parts of a contour very well. In such cases it will smooth out the concave part because with the weight calculation only direct neighboring vertices are considered (for Q_i this would be Q_{i-1} and Q_{i+1}) and this is not good enough to incorporate the concave parts. Due to this smoothing of concave parts some points will be placed outside of the contour instead of the inside of the contour. An example of this disadvantage can be seen in Figure 14.

Although the algorithm has this disadvantage, it does not influence our results. The used 3D muscle model meshes are mostly convex or in some cases slightly concave, but do not cause any fibers to be placed outside the mesh contours.

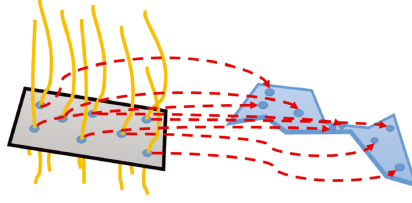


Figure 13: An example of how the morphing of the fibers works. Image from [11].

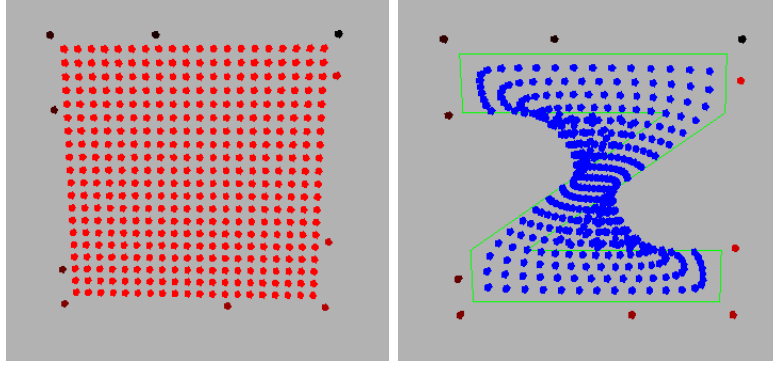


Figure 14: The left image shows the original shape, a square template with on its inside 20x20 points. The right image shows the result when trying to morph the left image to a (green) Z-shape.

3.5 Mass-Spring System

A mass-spring system consists of nodes with a mass and those nodes are connected to each other with springs. The nodes and springs are affected by internal and external forces. The internal force that affects the mass-spring system is the spring force:

$$F_s(i, j) = k_{i,j} \left(\vec{l}_{i,j} - l_{i,j}^{rest} \frac{\vec{l}_{i,j}}{|\vec{l}_{i,j}|} \right) \quad (7)$$

where $F_s(i, j)$ is the spring force of the spring between node i and node j , $k_{i,j}$ is the stiffness coefficient of the spring, $l_{i,j}^{rest}$ is the resting length of the spring and $\vec{l}_{i,j}$ is the direction vector to go from node i to node j .

One of the external forces that affect the mass-spring system is gravity:

$$F_g(i) = m_i \vec{g} \quad (8)$$

where $F_g(i)$ is the gravity force affecting node i , m_i is the mass of node i , and \vec{g} is the gravitational acceleration. Another type of external force is the

damping force, i.e. friction. There are two types of damping forces: node friction and spring friction. Node friction is calculated by:

$$F_{d_{node}}(i) = -c_d \vec{v}_i \quad (9)$$

where $F_{d_{node}}(i)$ is the node friction at node i , c_d is the damping coefficient of node i , and \vec{v}_i is the velocity of node i . Spring friction is calculated by:

$$F_{d_{spring}}(i, j) = -c_s(\vec{v}_i - \vec{v}_j) \quad (10)$$

where $F_{d_{spring}}(i, j)$ is the spring friction of the spring between node i and node j , c_s is the damping coefficient of the spring, and \vec{v}_i and \vec{v}_j are the velocities of node i and node j , respectively. The total force affecting a node can be written down as:

$$F_{total}(i) = F_g(i) + F_{d_{node}} + \sum_{j \in \Phi} (F_s(i, j) + F_{d_{spring}}(i, j)) \quad (11)$$

where Φ denotes the set of springs that are connected to node i . With the total force known, the new position of a node can be calculated for every iteration.

To get from the morphed fibers to a mass-spring system is easy. Although a conversion from a set of fibers to a mass-spring system is mentioned in literature ([10]), it is not mentioned how that conversion has been done. In our research every morphed fiber point in every slice will become a node of the mass-spring system. Between the slices there will be springs that connect the nodes belonging to the same fiber. The connections between nodes in the same slice are a more complicated, because the depth of a fiber inside the muscle affects the stiffness of the spring that connects to other nodes. The deeper a spring has to go, the more stiff it becomes, because a fiber should be kept in place by its surrounding fibers. The decision on how to connect the fibers with each other is already made in the fiber template. The stiffness value of a spring is based on linear interpolation between c_s^{min} and c_s^{max} , where c_s^{max} is the stiffness of the inner most springs and c_s^{min} is the stiffness of springs that connect the outer most nodes of the mass-spring system. The outer most nodes are also connected to the mesh, because when the mesh deforms, the inside of the mesh should also deform and that means that the mass-spring system will be affected. The user can specify how many springs should be connected from one node to vertices of the mesh. The number of connections should be more than 0, otherwise the mass-spring system remains unaffected by the mesh. There is not a maximum number of connections, but the number of connections is usually less than 4. The user can also specify the stiffness of the springs that connect the outermost nodes of the mass-spring system to the 3D model mesh. The spring stiffness of c_s^{min} and c_s^{max} are also user-defined.

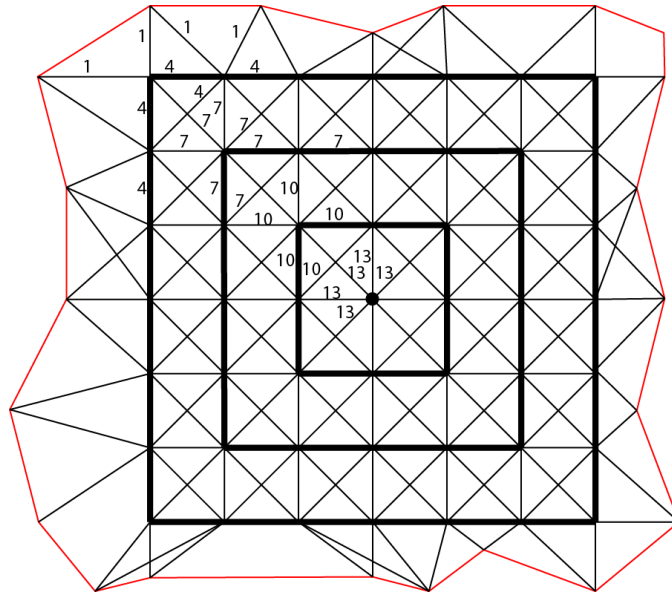


Figure 15: In this image is an example of a mass-spring system network with $c_s^{min} = 4$, $c_s^{max} = 13$, the spring stiffness for connections to the mesh is 1, and the number of connections to the mesh per node is 2. The red line is a mesh contour and every spring connecting to the contour has a stiffness of 1. The first layer of nodes will connect to each other with a stiffness of 4 and those nodes will connect to the next layer with springs that have a stiffness of 7. This continues until the center is reached and everything connected to the point in the middle will have springs with a stiffness of 13.

Figure 15 shows an example of the connections between the mesh and the nodes of the mass-spring system with different stiffness values. Figure 16 shows the result of a created mass-spring system.

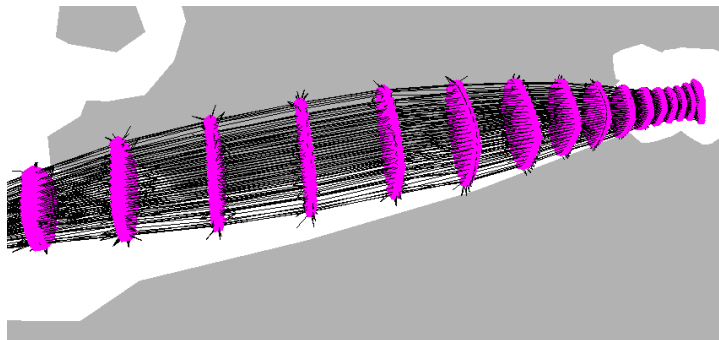


Figure 16: A result of the mass-spring system creation for the Rectus Femoris. The pink parts are the nodes and the black lines are the springs of the mass-spring system.

4 Action Line Representation

In this Section the representation of the action lines will be described. In Section 4.1 are the most commonly used action lines described. In Section 4.2 is described how to build a new action line based on a muscle model mesh itself. Our adaptive action line is described in Section 4.3.

4.1 Commonly Used Action Lines

In the literature there are a couple of commonly used representations for the action lines: straight action lines and action lines with via points ([3],[6],[13],[17]), a quadratic curve as an action line ([14]) and a cubic curve as an action line ([6]).

A straight action line is the simplest representation, because it is a straight line from the point of origin to the point of insertion. The action line with via points is a representation that consists of line segments. The first segment is from the point of origin to the first via point, then the other segments connect the other via points, and the last segment is from the last via point to the point of insertion. A via point will be attached to a bone, so when the bone moves, the via point will also move. The quadratic curve and cubic curve do also have some sort of via point, but this will function as a control point to influence the curve. Both curve representations are based on the Bezier curve:

$$Q(u) = \sum_{j=0}^d P_j B_j^d(u), \quad (12)$$

$$B_j^d(u) = \binom{d}{j} u^j (1-u)^{d-j}, \quad (13)$$

where $0 \leq u \leq 1$, Q_u is the collection of points that make the curve, d is the degree of the curve, P_j is a control point (which includes start and end position), and B_j^d is the Bernstein basis function. The differences between the quadratic curve and the cubic curve are that the quadratic curve only has one control point while the cubic curve has two control points, and the quadratic curve has $d = 2$, but the cubic curve has $d = 3$. The curves will go from the point of origin to the point of insertion and also the curve will go towards its control points, but not necessarily go through its control point.

There is also another representation for action lines and that is an action line for every fiber. In this case every fiber will have via points that will move according to the shape of the muscle. In our case the via points of a fiber will be the mass-spring nodes that make the corresponding fiber. By using the positions of the mass-spring nodes, the via points will move according to the deformation in the mass-spring system and the action line of the fiber will still go from origin to insertion while going through a number

of via points. This method is multi-action line, but requires more storage and computation.

4.2 Building Methods for Action Lines

A new type of action line would be nice if it is based on the muscle itself, because the direction of contraction of the muscle, i.e. the action line, is based on the shape of the muscle and the fiber architecture that is inside of the muscle. For building a new action line there are a couple of options we came up with ourselves. The first option would be to use the contour of the muscle model mesh. With the use of the found contours of the muscle slices (see Section 3.3.1) we can make an average position per contour using every point of the contour. Every slice would give a point which can be used as a via point. This way an action line can be built from origin to insertion with as many via points as muscle slices. However, with this building method we are depending on the density of the contour points along the contour, because if there are a lot of contour points around the left part of the contour, the averaged point will be more to the left while it should be more to the right based on the general shape of the contour. This dependency could cause some inaccuracy.

The second option would be to use the centroid of the muscle fibers. This means looking at the morphed fibers (see Section 3.4) at each slice and take the average of every fiber point in the slice. Every averaged point will become a via point. This results in an action line from origin to insertion with via points at every slices, just like the building method with the contour average but representing better its architecture because it is based on the position of the fibers in the muscle instead of being based on its external muscle model shape. However, this building method could be affected by the amount of fibers that are simulated, probably making it less precise with a small number of fibers than with a large number of fibers.

The third option is to build the action line based on the center of the fiber template. For this method the fiber template is used to describe the exact center of the template and use that corresponding fiber. Of course this fiber needs to be morphed into the mesh contours like the other fibers. After the morphing of the fiber there are the known points of the fiber at every slice of the muscle. Using the fiber points at each slice as via point, an action line is built from origin to insertion with a via point for every slice of the muscle. Because this building method is based on only one fiber, it is possible that is not accurate enough for an action line representation.

4.3 Adaptive Action Line

Our adaptive action line is an action line that has a flexible number of via points. This is for the purpose of keeping an action line as simple as

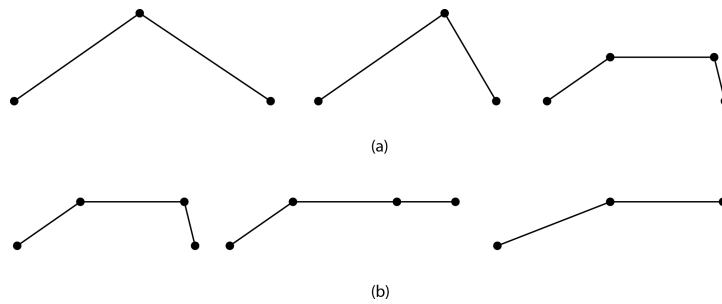


Figure 17: A simple case for adding an extra via point (a) and a simple case for removing a via point (b).

possible but still have a good representation of the shape of the muscle model mesh. The adaptive action line starts with an already built action line with via points, the building method could vary between the three methods mentioned in the previous section.

From the built action line as a starting point, the adaptive action line could decide to add or remove via points to make the representation simpler or more detailed, depending on its needs. Removing and adding via points will be based on the angles between the line segments at the via points. If an angle at a via point is almost 0 or 180 degrees, it could be simplified by removing that via point and his via point neighbor, and replace them with a via point between them. This replacement is to simplify the action lines structure, but also keep it from changing to much by directly removing that single via point, e.g. a muscle goes around a very sharp edge and causes the via point to have an angle of almost 0 degrees but when you would remove the via point you would completely lose that part of the muscle representation in the action line.

If an angle at a via point has changed it could indicate that there should be more detail at that position in the action line because there the action line has changed. The threshold for a changing angle at via points is variable and can be specified by the user. To find out if an angle has changed, the comparison is made with the angle with which the via point started. If the threshold is surpassed, then the current via point will be removed, but two new via point will be added. Both those via points will be placed at the middle of the line segments of the original via point. This adds more detail to the action line and could also smooth the action line a little.

Because the simplification replaces two via points with one via points and the adding replaces one via point with two via points it would be possible to regain the original action line. For the new via points it is unknown to what bone that should attach, so they will be attached to the same bone as the removed via point. In Figure 17 illustrates an easy example for the adding and removing actions.

5 Moment Arm Calculation

The moment arm of a muscle action line can be calculated using different methods as described in [19]. The easiest method to calculate the moment arm is based on virtual work and is formulated as:

$$ma = \frac{\delta l}{\delta \theta} \quad (14)$$

where ma is the moment arm, δl is the difference in length of the action line, and $\delta \theta$ is the difference in the angle of the joint of interest. The angle θ is in radians and not in degrees. The length l can be used in any type of length measurement, but the more conventional length measurement is cm so we used that too. Although to get to cm, a conversion is needed from the Bullet coordinate system measurements to the length measurement cm. For that we used the average length of the fibula ([1]) which is 40.5 cm and with the length of the fibula 3D model mesh a conversion value could be found and used.

The joint angle of interest is simple, because during the simulation only one joint angle at a time will change. That means that the joint that is changing, is the joint of interest and its angle can be used for the calculation of the moment arm of the action lines. If multiple joints would change their rotation at the same time a combined angle is needed to calculate the moment arm and then Equation 14 could not be used anymore and a more sophisticated calculation is required like the other moment arm calculation methods mentioned in [19]. For instance, there is the velocity-level kinematics method mentioned:

$$ma = \frac{\delta l}{\delta \theta} = \frac{\delta l/dt}{\delta \theta/dt} \equiv \frac{\dot{l}}{\dot{\theta}} \quad (15)$$

but also mentioned is that \dot{l} can be difficult to calculate. In the literature also the partial velocity method is mentioned which is based on $ma = PNC_\theta$ a partial velocity matrix P , a coupling matrix C_θ , and a block diagonal matrix N . But because those matrices are difficult to obtain (sometimes not possible at all) and this makes the method not very useful.

The last method mentioned in the literature is the generalized force method and uses forces instead of velocities. For this method we again need a coupling matrix C_θ , but also a generalized forces matrix f and a muscle tension scalar s , resulting in the formula:

$$ma = \frac{f^T C_\theta}{s} \quad (16)$$

but like the other methods it is difficult to obtain the matrices.

Because we do not have the necessary information available for those matrices we decided to use Equation 14 for the moment arm calculations.

6 Results

6.1 The Setup

In this section the experiment setup will be explained. Section 6.1.1 is about the used physics engine for the simulation. After that we will elaborate in Section 6.1.2 what muscles and bones are used and what their settings are in the simulation.

6.1.1 Bullet Physics Engine

Bullet Physics Engine (<http://bulletphysics.org>) is an open source physics engine. The choice to do the simulation in Bullet was based on the fact that soft deformable bodies should be used for the muscles and Bullet is able to simulate those kinds of objects. Also using an open source engine was preferred because the source code should be accessible for changes.

As mentioned before, Bullet will handle the simulation of the soft bodies, but also the rigid bodies. The collision between the bodies is also handled by Bullet and no changes have been made there. The collision checks will cost a lot of simulation time, especially with (multiple) high-detailed soft body models.

Attaching a soft body to a rigid body is possible with the anchor points in Bullet. These anchor points only attach one vertex of the soft body and does not affect the neighboring vertices. Because the neighbor vertices need to be affected by those anchor points, we adjusted it so that the neighbor vertices are moved towards the anchor point by replacing the vertices at halfway between them and the anchor point and adjusting the velocities to the correct values for the extra distance that the vertices traveled.

Maintaining the volume of a soft body is a difficult task, but this task is integrated in Bullet and can be used directly for the soft bodies. The maintaining of the volume is not a guarantee, but more of an approximation, so this has to be considered.

Because this experiment is not about the rendering of the muscles, the frame rendering of Bullet is used.

6.1.2 Muscle and Bone Models

The simulation needs 3D models meshes of muscles and bones and those meshes are coming from [12]. The bone meshes will be loaded into Bullet as rigid bodies, and the muscle meshes will be loaded as soft bodies. The soft bodies are based on a mass-spring system and the spring stiffness is set to its maximum value (which is 1).

For the simulation we focus on the lower extremity and the right leg in specific. The human skeleton is loaded into Bullet in separate bones or bone groups. The most important bones that will collide with the muscle

meshes are the foot, the fibula, the tibia, the femur, the pelvis, the tailbone, the spine and the thorax. The used muscles meshes are from the following muscles: Iliopsoas D (better known as Psoas Major), Iliopsoas F (better known as Iliacus), Gracilis, Inferior Gemellus, Biceps Femoris A, Biceps Femoris B, Pectineus, Piriformis, Rectus Femoris, Quadratis Femoris and Sartorius. Every muscle model will have 100 fibers to model the inside of the muscle. Also every model will consist of 20 slices, except for the Quadratis Femoris which consists of 5 slices, and the Inferior Gemellus which consists of 6 slices. Those two models have a lower number of slices because they are smaller in model size and do not need 20 slices to have a good modelling of their inside. The settings for the mass-spring system have been manually found by visual observations and are not selected through an optimization process. The result that the settings for the mass-spring system give is a stable system (i.e. the mass-spring system will come to a resting state when there is no external movement by the bones and/or model mesh) that remains inside the muscle model. The values are the same for every muscle model because they just worked fine in the different models. The node damping coefficient c_d is set to 0.00003, the spring damping coefficient c_s is set to 0.05, the minimum spring stiffness c_s^{min} is set to 1.0, the maximum spring stiffness c_s^{max} is set to 1.1. Those two values are pretty close, because based on visual observations a bigger difference in values would give a less stable mass-spring system. The fact that the two values are close makes the gradual increased stiffness values for the inner layers not worth it. The spring stiffness of the springs between the mesh and the outermost nodes is set to 0.75. The number of outgoing springs from the outermost nodes to the mesh is set to 3 for every muscle model except the Inferior Gemellus which is set to 12, because it has only a few vertices at the top in its 3D model mesh and with a low number of connecting springs the mass-spring system would fall down and remain outside of the muscle model.

The vertices of the 3D muscle models that should be attached to the bones are user specified and manually found by looking at the areas where the muscles should attach to in a real life situation. At those areas the vertices are examined individually and the vertices that should stay attached to the bones are noted. The muscle models have their origin and insertion area based on the attached vertices. After the origin and insertion areas are known we can compute the scalar field in the preprocessing part of the simulation. The scalar field values are saved in files which can be read when a new simulation with the same origin and insertion areas is started.

For the movement of the bones there are two joints: the knee joint and the hip joint. When those two joints are simulated, they will perform a flexion. The knee joint will have a flexion from 0° to 90° . The hip joint will have a flexion from -30° to 95° . The flexion angles of the two joints are based on the hip and knee movement range mentioned in [5], [8], and [23].

The adaptive action line is considered straight when the angle between two

line segments is smaller than 5° or bigger than 175° . When the adaptive action line is considered straight, the adaptive action line will be adjusted by removing a via point.

6.2 The Experiments

With the experiments there are results which will be discussed here. In Section 6.2.1 the comparison between the commonly used action lines and our constructed action lines is made. In Section 6.2.2 the adaptive action line results are discussed and the angle threshold for the adaptive action line is reviewed. In Section 6.2.3 the comparison will be made between the commonly used action lines and the adaptive action line with the best angle threshold. Important to mention is that the resulting moment arm values have been smoothed using a fourth degree polynomial (like in [5]) and therefore they start with the name "Poly" in the graphs.

6.2.1 Comparison Common and Built Action Lines

We need a ground truth before we can compare action lines and see which action lines perform well. We make the assumption that the action lines per fiber gives the ground truth moment arm values and action line trend that should be replicated during the simulation. Comparisons with the ground truth will be made with RMS (Root Mean Square) values of the error in moment arm values and the standard deviation in the error of moment arm values is also calculated. The graphs in Appendix A are mostly used for comparing the moment arm trend, because this shows whether or not moment arms should increase or decrease. The graphs will also be used for globally comparing (smoothed) moment arm values. The RMS values have been calculated by $x_{RMS} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$ where $x_1 \dots x_n$ are the errors in moment arms values. We will always have to consider both the RMS values and information of the graphs because both the moment arm trend and the RMS value have to be close to the ground truth.

With the first results we hope to see a difference between the commonly used action lines (i.e. the straight action line, the action line with via points, the quadratic Bezier action line, and the cubic curve action line) and the action lines that used the different building methods (i.e. the action line built from the center of the contour, the action line built from the centroid of the individual fibers, and the action line built from the center of the fiber template). Also based on the first results, an action line building method will be chosen as most promising to make it an adaptive action line. Only one building method will be picked, because when the adaptive action line will adapt it does not take into account what its starting shape was, but we would like it to start with a potentially good action line shape.

In Figure 18 there is an overview of the RMS values of the error of the

	Biceps Femoris A		Biceps Femoris B		Gracilis		Iliopsoas D	Iliopsoas F	Inferior Gemellus	Pectineus	Piriformis	Quadratis Femoris	Rectus Femoris B		Sartorius		Average
	knee	hip	knee	hip	knee	hip	hip	hip	hip	hip	hip	hip	knee	hip	knee	hip	
Straight	6,85 / 3,87	3,59 / 2,08	7,11 / 5,09	3,86 / 1,24	3,58 / 2,80	3,77 / 1,91	4,58 / 2,56	1,62 / 1,31	2,36 / 1,33	1,07 / 0,71	1,14 / 0,78	1,92 / 1,33	5,21 / 2,85	9,98 / 6,96	4,05 / 2,49		
Via points	8,68 / 5,04	3,37 / 2,09	7,04 / 5,04	2,04 / 1,32	3,60 / 2,79	2,07 / 0,79	4,39 / 2,22	1,86 / 1,36	2,53 / 1,36	1,82 / 0,82	1,85 / 0,76	2,37 / 1,57	3,98 / 2,41	14,87 / 8,70	4,32 / 2,59		
Quadratic	6,63 / 3,64	3,27 / 1,93	7,11 / 5,09	3,25 / 0,98	3,59 / 2,80	3,01 / 1,58	4,29 / 2,25	1,66 / 1,33	2,36 / 1,32	1,20 / 0,69	1,20 / 0,74	1,93 / 1,34	4,50 / 2,56	9,99 / 6,96	3,86 / 2,37		
Cubic	6,63 / 3,65	3,34 / 1,95	7,09 / 5,08	3,31 / 1,00	3,58 / 2,83	2,90 / 1,54	4,39 / 2,27	1,67 / 1,33	2,57 / 1,37	1,19 / 0,69	1,18 / 0,75	1,92 / 1,33	4,36 / 2,50	10,44 / 7,29	3,90 / 2,40		
Center of contour	7,07 / 3,92	4,05 / 2,26	7,09 / 5,07	3,24 / 1,05	4,34 / 3,41	3,71 / 1,17	6,06 / 2,51	3,22 / 1,79	8,04 / 2,08	3,65 / 1,29	3,60 / 1,09	4,49 / 1,80	4,26 / 2,52	44,84 / 12,40	7,69 / 3,03		
Centroid of fibers	7,06 / 3,92	4,06 / 2,27	7,07 / 5,06	3,23 / 1,05	4,35 / 3,42	3,66 / 1,16	6,13 / 2,52	3,23 / 1,79	8,10 / 2,09	3,65 / 1,30	3,59 / 1,09	4,48 / 1,79	4,25 / 2,51	44,81 / 12,43	7,69 / 3,03		
Center of template	7,14 / 3,96	4,03 / 2,25	7,06 / 5,06	3,13 / 1,13	4,28 / 3,36	3,53 / 1,05	6,54 / 2,54	3,19 / 1,78	7,61 / 2,03	3,46 / 1,65	3,55 / 1,08	4,39 / 1,64	4,11 / 2,40	44,54 / 12,62	7,61 / 3,04		

Figure 18: A table containing RMS values and standard deviations for the various muscle models and action lines. The first value of the action lines is the RMS value or the errors and the second value is the standard deviation of the error values.

moment arm values when compared to the ground truth.

The first muscle we will take a look at is the Inferior Gemellus. In Figure 19 (left) we can see our ground truth for this muscle which is the collection of the moment arms of the individual fibers. In Figure 19 (right) we see the moment arms of the various action lines, both common and ours. When comparing those two graphs with moment arms it is visible that the action lines of the center of the contour, the centroid of the fibers, and the center of the template have a worse fit in the range of the moment arm values of the individual fibers than the other action lines. For this muscle we can conclude that the commonly used action lines are better than our constructed action lines, because they fit better in the range of moment arm values of the individual fibers. Because the three building methods result in almost equal moment arm trends, we cannot pick a most promising building method based on themoment arm trends, but the center of the template results in the lowest RMS value of the errors.

With the Biceps Femoris A (Appendix A.1.1 for individual fibers and Appendix A.1.2 for the action lines) the action line with via points does not fit in the range of moment arm values of the fibers, even though it has the lowest RMS error. The other action lines have their moment arm values in the range of the moment arm values of the fibers. The centroid of the fibers building method results in a slightly lower RMS value than the other two building methods.

The Biceps Femoris B is affected by both the knee joint and the hip joint. For the knee joint (Appendix A.2.1 and A.2.2) it is clearly visible that none of the action lines have resulting moment arm values that stay within the moment arm value range of the individual fibers, because the moment arm values are either too high or too low. The three action lines from the building

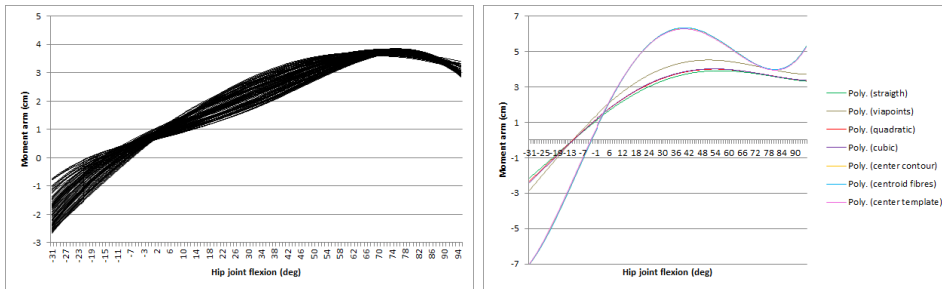


Figure 19: Left: the moment arms of the individual fibers in the Inferior Gemellus (also in Appendix A.6.1). Right: the moment arms of the commonly used action lines and the action lines with three different building methods (also in Appendix A.6.2).

methods result in the highest RMS values with the center of the template action line as best option (lowest RMS value). With the hip joint (Appendix A.2.5 and A.2.6) we can see that all action lines are almost the same and none of them follow the trend of the fibers, but they all stay in the range of the moment arm values of the individual fibers. The center of the template action line results in the lowest RMS error.

The Gracilis is also affected by the two joints and at the knee joint (Appendix A.3.1 and A.3.2) there is no good fit between the moment arms of the action lines and the moment arms of the individual fibers. According to the RMS values, the via points action lines performs best and the center of the template has the lowest RMS value of the three constructed action lines, but also has the highest standard deviation. For the hip joint (Appendix A.3.5 and A.3.6) we can see that only the moment arms of the three action lines resulting from the building methods do not perfectly fit in the range of moment arm values of the fibers, and they do have higher RMS values with the center of the template action line as the lowest RMS value.

With the Iliopsoas D (Appendix A.4.1 and A.4.2) and Iliopsoas F (Appendix A.5.1 and A.5.2) the individual fibers have smaller moment arm values than all the moment arms of the action lines. With the Iliopsoas F the action line from the center of the template has higher peaks than the center of the contour and the centroid of the fibers action lines, and also has a higher RMS value.

The Quadratus Femoris (Appendix A.9.1 and A.9.2) gives similar results as the Inferior Gemellus. This means the most promising building method is based on the RMS value and that would be the center of the template is this case.

With the Pectineus (Appendix A.7.1 and A.7.2) the moment arms of the individual fibers show a different trend than the moment arms of the action lines. The lowest RMS value is from the quadratic Bezier action line.

With the Piriformis (Appendix A.8.1 and A.8.2) the moment arms of the center of the fiber template action line does not follow the moment arm trend of the fibers and also does not stay in the moment arm value range of the fibers. The center of the template does also differ a lot from our other two constructed action lines and the individual fibers, because it has constantly positive moment arm values where the fibers start with negative and go to positive moment arm values. The difference is also visible in the RMS values, but the center of the template action line has a slightly lower RMS value than the other two constructed action lines, but a much (in comparison) higher standard deviation.

With the Rectus Femoris (Appendix A.10.1 and A.10.2) all the action lines result in moment arms that follow the same trend as the moment arms of the individual fibers. Our three constructed action lines result in much higher moment arms than the four commonly used action lines and the individual fibers. The RMS error values of the commonly used action lines are lower than our three constructed action lines and the center of the template action line has the lowest RMS error.

The Sartorius spans both the knee joint and the hip joint. For the knee joint (Appendix A.11.1 and A.11.2) none of the moment arms of the action lines follow the trend of the moment arms of the fibers and the center of the template action line has the lowest RMS value. For the hip joint (Appendix A.11.5 and A.11.6) we can see almost the same thing, because none of the moment arms of the action lines follow the trend of the moment arms of the fibers. All the moment arms of the three built action lines are now almost equal, but the center of the template has a slightly lower RMS value.

Summary

With the above stated observations about the action lines of the three building methods, the first conclusion is that the results vary from bad to good. The center of the contour and the centroid of the fibers methods are basically the same (averaged RMS value and averaged standard deviation are equal) and give better results than the center of the fiber template building method. One may have noticed that the center of the fiber template has the lowest RMS value with a lot of muscle models, but when that is the case the moment arm values and trend are very similar to the other two building methods (e.g. Biceps Femoris B). When the center of the fiber template does not have the lowest RMS error or a higher standard deviation, the moment arm values and trend differs more from the ground truth than the other two building methods (e.g. with the Piriformis). The center of the template building method is thus the least promising method to use as an adaptive action line.

In these experiments only the straight line fiber template is used, which affects the fibers but not the contours. If the template would be changed, e.g. to the pennate fibers, the fibers would adjust accordingly, but the contours

	Biceps Femoris A		Biceps Femoris B		Gracilis		Iliopsoas D	Iliopsoas F	Inferior Gemellus	Pectineus	Piriformis	Quadratis Femoris	Rectus Femoris B	Sartorius		Average
	knee	knee	hip	knee	hip	hip	hip	hip	hip	hip	hip	hip	knee	knee	hip	
1 degree	74,17 / 71,79	116,92 / 115,36	108,70 / 106,68	4,04 / 1,46	3,76 / 2,95	464,36 / 438,31	189,94 / 175,78	2,70 / 2,33	109,10 / 99,40	87,07 / 84,69	2,15 / 1,77	402,24 / 399,25	400,76 / 392,05	738,60 / 720,28	193,18 / 186,58	
2 degrees	70,70 / 66,91	376,58 / 373,40	473,51 / 468,87	20,70 / 19,14	91,47 / 88,19	866,10 / 640,14	84,75 / 80,68	2,69 / 2,33	98,22 / 92,35	87,34 / 84,42	2,34 / 1,98	402,24 / 399,25	584,59 / 553,50	761,55 / 743,81	280,20 / 258,21	
3 degrees	56,62 / 53,69	239,72 / 234,47	429,88 / 425,82	21,94 / 20,38	31,60 / 30,64	487,72 / 465,88	72,30 / 69,39	3,02 / 2,70	112,20 / 101,08	125,19 / 121,32	2,27 / 1,88	402,23 / 399,25	381,19 / 372,43	533,19 / 527,04	207,08 / 201,86	
4 degrees	74,54 / 71,59	298,09 / 294,11	305,62 / 303,12	210,91 / 209,07	1098,14 / 890,22	485,64 / 452,67	62,26 / 58,93	2,94 / 2,58	254,23 / 177,13	160,23 / 153,64	2,80 / 2,18	402,23 / 399,25	380,61 / 372,12	533,12 / 526,98	305,10 / 279,54	
5 degrees	59,41 / 56,47	313,95 / 309,73	274,66 / 272,27	210,94 / 209,08	180,88 / 177,04	771,26 / 564,13	117,06 / 108,75	3,04 / 2,69	245,67 / 185,25	85,25 / 81,48	2,38 / 1,97	473,81 / 471,06	383,46 / 376,07	451,52 / 445,59	255,23 / 232,97	
6 degrees	56,72 / 53,70	298,67 / 294,79	274,66 / 272,27	210,94 / 209,08	362,63 / 311,10	274,21 / 256,27	109,31 / 97,72	3,02 / 2,69	352,99 / 255,10	86,16 / 76,83	29,09 / 28,59	114,31 / 111,38	388,18 / 380,62	410,69 / 399,74	212,26 / 196,42	
7 degrees	60,40 / 56,47	278,25 / 273,77	274,66 / 272,27	210,93 / 209,05	952,26 / 742,07	254,47 / 237,45	214,18 / 188,80	3,21 / 2,81	110,49 / 82,10	110,63 / 105,62	40,76	123,51 / 121,24	1656,86 / 1311,91	385,50 / 376,30	334,10 / 287,19	
8 degrees	54,32 / 49,46	3363,21 / 1071,35	274,66 / 272,27	210,90 / 209,08	394,49 / 340,72	393,30 / 366,80	146,14 / 132,34	3,20 / 2,82	250,62 / 188,60	384,42 / 306,98	43,01 / 41,56	162,33 / 155,83	1014,84 / 926,25	382,49 / 373,19	362,71 / 316,95	
9 degrees	186,56 / 169,37	508,96 / 356,52	274,66 / 272,27	210,92 / 209,07	213,99 / 196,89	446,06 / 419,66	169,21 / 151,22	3,21 / 2,80	339,44 / 239,08	277,73 / 223,38	42,99 / 41,53	116,93 / 116,07	453,51 / 440,82	386,32 / 375,67	259,32 / 229,60	
10 degrees	55,57 / 52,54	446,78 / 376,33	274,66 / 272,27	210,92 / 209,07	224,82 / 210,64	270,16 / 244,20	108,27 / 100,86	3,24 / 2,77	223,17 / 153,16	222,72 / 205,02	42,62 / 41,16	125,61 / 123,17	321,20 / 297,01	386,18 / 375,65	208,28 / 190,28	
15 degrees	334,28 / 298,07	337,71 / 322,82	224,48 / 223,10	220,69 / 217,36	540,77 / 450,23	393,43 / 363,39	104,17 / 91,89	3,24 / 2,72	207,30 / 159,75	93,08 / 83,47	60,69 / 52,31	117,07 / 115,96	455,82 / 427,66	438,11 / 416,70	252,20 / 230,39	
20 degrees	82,76 / 73,65	243,40 / 227,35	224,48 / 223,10	244,25 / 239,48	233,49 / 229,54	268,48 / 259,64	102,77 / 94,76	7,56 / 7,22	185,80 / 140,59	86,27 / 76,26	47,31 / 43,39	154,24 / 148,62	356,70 / 332,82	371,87 / 361,60	186,39 / 175,57	

Figure 20: A table containing RMS values and standard deviations for the various muscle models and degree thresholds. The first value of the degree thresholds is the RMS value of the errors and the second value is the standard deviation of the error values.

stay the same. Because an action line would be more realistic if it is based on the inside of the muscle model (which can also change), it is better to say that the centroid of the fibers building method is the most promising method. The centroid of the fibers building method will be used as the adaptive action line from this point on.

6.2.2 Adaptive Action Line Results

The experiments with the adaptive action line were about finding a maximum degree threshold for which the representation of the deformation is still good enough. We simulated every degree to 10 degrees and then 15 degrees and 20 degrees.

The first thing to mention about the overall experiments is the possibility of high moment arm values. This is because of the change in the number of via points. When via points are removed or added, the length of the action line is changed significantly and can cause a high or low moment arm, because the moment arm is calculated as $ma = \frac{\delta l}{\delta \theta}$. By smoothing the values using a fourth degree polynomial we can smooth out the highest and lowest values and by comparing the RMS (Root Mean Square) values and standard deviations of the errors from Figure 20 we can still make good comparisons.

With the Biceps Femoris A (see Figure 21 or Appendix A.1.1, A.1.3 and A.1.4) the moment arm trends of the adaptive action lines look similar to each other from the 1 degree to the 8 degree threshold. This also holds

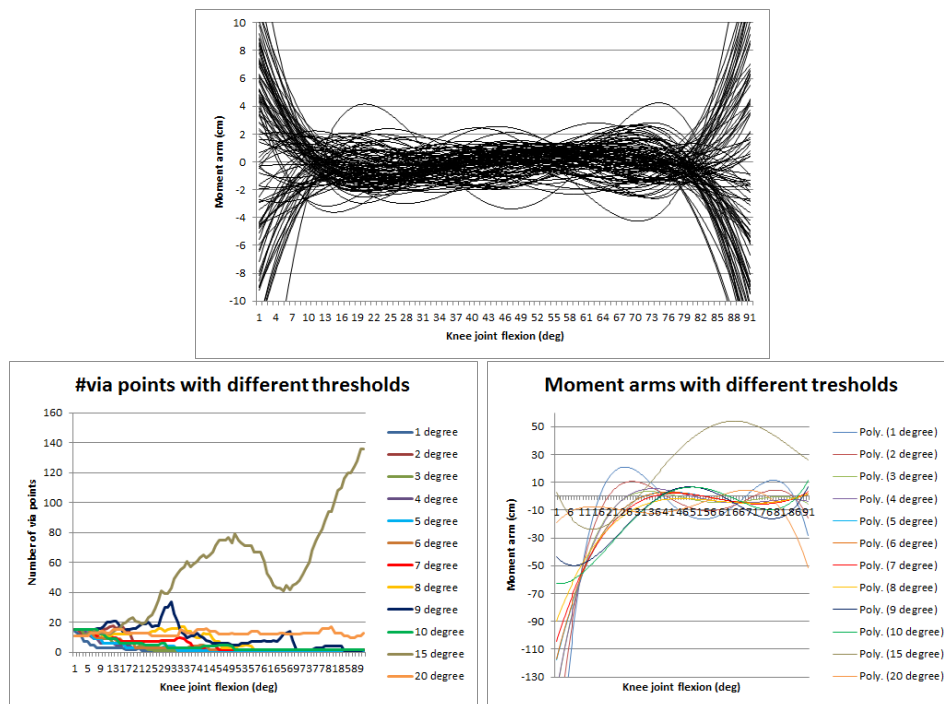


Figure 21: The results of the Biceps Femoris A. Top image shows the moment arms of the individual fibers. Bottom-left image shows the number of via points of the adaptive action line for every degree threshold and bottom-right image shows the moments arms of the adaptive action lines with different degree thresholds.

for the number of via points throughout the simulation and the RMS error value is also close. The big outsider here is the 15 degree threshold which has the highest RMS error value.

The Biceps Femoris B shows that for the knee joint (Appendix A.2.1, A.2.3 and A.2.4) the 1 degree to the 6 degree threshold are quite similar in their moment arm values and their trend. Again, this also holds for the number of via points. The 8 degree threshold shows different behavior compared to the others, because it generates over 160 via points and the moment arm value is over 300 cm, resulting in a high RMS value. None of degree thresholds generates moment arm that can have a perfect match with the moment arm of the fibers, but the 7 degree threshold is the best in following the trend. For the hip joint (Appendix A.2.5, A.2.7 and A.2.8) the moment arm values of the 1 degree and 3 degree thresholds are closest to the moment arm values of the individual fibers and the trends are quite similar. The lowest RMS values come from the 1 degree threshold for both joints.

With the knee joint of the Gracilis (Appendix A.3.1, A.3.3 and A.3.4) the moment arms of the adaptive action lines show that the 1 degree to 3 degree

thresholds are closer to the values of the moment arms of the individual fibers, but the best candidate is only the 1 degree threshold based on its overall moment arm value and RMS value. With the hip joint (Appendix A.3.5, A.3.7 and A.3.8) none of the degree thresholds result in a moment arm trend that corresponds perfectly to the moment arm trend of the fibers. We can say that the 1 degree to the 3 degree threshold are the best because they have the lowest RMS value. The 4 degree threshold suddenly generates a large number of via points and the moment arm values go up to 800 cm resulting in a high RMS error.

The Iliopsoas D (Appendix A.4.1, A.4.3 and A.4.4) the 2 degree and 5 degree threshold are the two outsiders which can be due to the generation of large numbers of via points. There is unfortunately no perfect match between the moment arms of the adaptive action lines and the individual fibers, but the 7 degree threshold has the lowest RMS value.

For the Iliopsoas F (Appendix A.5.1, A.5.3 and A.5.4) there is also no perfect match for the moment arms of the adaptive action lines compared to the moment arms of the individual fibers. From the 1 degree to the 6 degree threshold the moment arms of the adaptive action line are quite similar. The lowest RMS error comes from the 4 degree threshold.

When looking at the Inferior Gemellus (Appendix A.6.1, A.6.3 and A.6.4) we can see that the moment arms are very similar for every degree threshold, except for the 9 degree and 10 degree thresholds. Except for the 9 degree and 10 degree thresholds the adaptive action line has a perfect match to the moment arm values of the fibers. The number of via points is very small and the different degree thresholds do not make much difference in the number of via points. The RMS error values are very close, but the 2 degree threshold has the lowest RMS value.

With the Pectineus (Appendix A.7.1, A.7.3 and A.7.4) most of the thresholds generate a large number of via points (up to 275 via points), but the lowest RMS value is from the 2 degree threshold and that degree threshold does not generate a lot of via points. The 1 degree to the 3 degree threshold produce almost similar moment arm values.

The Piriformis (Appendix A.8.1, A.8.3 and A.8.4) does not have a perfect match between the moment arms of the fibers and the moment arms of the different thresholds. The 1 degree to 7 degree thresholds produces most similar moment arm values. Also the 20 degree threshold shows some real potential, because it is close to a match for moment arms with the individual fibers and has a low RMS value.

With the Quadratus Femoris (Appendix A.9.1, A.9.3 and A.9.4) the 1 degree to 5 degree thresholds produce quite similar moment arms with low RMS values. The 15 degree and 20 degree threshold are far off according to the RMS error values, probably due to the seemingly unnecessary large number of via points.

The Rectus Femoris (Appendix A.10.1, A.10.3 and A.10.4) provides results

where the number of via points is quite stable for every threshold. Unfortunately, the moment arms of the adaptive action lines do not have a perfect match with the moment arms of the individual fibers. The 1 degree to 5 degree thresholds are almost similar, but the 6 degree to 20 degree thresholds have lower RMS values and are closer to the moment arm values of the individual fibers.

With the Sartorius we have a knee and hip joint. For the knee joint (Appendix A.11.1, A.11.3 and A.11.4) we have the 9 degree to 20 degree thresholds moment arms that follow the trend of the moment arms of the individual fibers. The 1 degree to 6 degree thresholds are also quite similar to each other, but some have higher RMS values than the 9 degree to 20 degree thresholds. For the hip joint (Appendix A.11.5, A.11.7 and A.11.8) the 1 degree to 10 degree thresholds, except for the 6 degree threshold, are quite similar in moment arm trend and value. Also the moment arm trend of the mentioned degree thresholds is the same trend as the moment arm values of the individual fibers. The lowest RMS value is from the 20 degree threshold, which also follows the moment arm value trend of the individual fibers.

Summary

Based on overall result the lower degree thresholds perform the best, but in some cases a higher degree threshold has a better performance. A higher degree can perform better in cases where the muscle model is long and straight e.g. the Sartorius and the Rectus Femoris, because it does not have to change much throughout the simulation because the muscle model will not deform much. The smaller degree thresholds will perform better with more complicated or short muscles e.g. the Iliopsoas F and Pectineus, because such muscle can deform a lot with just a small rotation of a joint.

When looking at the RMS values and their errors and the moment arm values and trends the best performing degree thresholds are the 1 degree to 3 degree thresholds, because they have low RMS values and follow the moment arm trend of the ground truth more often than the other degree thresholds. According to those best performing degree thresholds we can state that the maximum degree threshold for which the representation of the deformation still is good enough is the 3 degree threshold.

6.2.3 Comparison Common and Adaptive Action Lines

When looking at the best performing adaptive action line threshold, which is the 3 degree threshold, and the commonly used action lines we can see that there is a big difference and not just in the moment arm values but in the trend of the moment arm values as well. We can also look at the RMS error values from Figure 18 and 20. With the Biceps Femoris A (Appendix A.1.1, A.1.2 and A.1.4) the 3 degree threshold does not follow the trend of the moment arm of the fibers, but the quadratic curve and cubic

curve action line stay perfectly in range of the moment arm values of the individual fibers. When comparing the RMS errors, the adaptive action line has a higher RMS value. The Biceps Femoris B shows that for the knee joint (Appendix A.2.1, A.2.2 and A.2.4) there is no real winner, because both the adaptive action line and the commonly used action lines do not match the moment arms of the fibers. For the hip joint (Appendix A.2.5, A.2.6 and A.2.8) they are both good, because they both produce moment arms that follow the same trend as the moment arm values of the individual fibers. For both joints the RMS value of the adaptive action line is much higher. With the knee joint of the Gracilis (Appendix A.3.1, A.3.2 and A.3.4) the adaptive action line is the one that produces moment arms that follow the trend of the moment arms of the fibers best, but it has a higher RMS value. With the hip joint (Appendix A.3.5, A.3.6 and A.3.8) we can say that the four commonly used action lines are better because they have a perfect fit when the moment arms are compared to the moment arms of the fibers, and the adaptive action line has a higher RMS value than the four commonly used action lines. For the Iliopsoas D (Appendix A.4.1, A.4.2 and A.4.4) and Iliopsoas F (Appendix A.5.1, A.5.2 and A.5.4) the best follower of the moment arm trend of the individual fibers is the adaptive action line, because the four commonly used action lines are decreasing the moment arm value for most of the simulation. The Inferior Gemellus (Appendix A.6.1, A.6.2 and A.6.4) has nice results for both the adaptive action line and the four commonly used action lines. But the best fit compared to the moment arm values of the individual fibers is the adaptive action line, because they both have the peak in the moment arm value around 75° hip flexion, where the straight action line has its peak around 50° hip flexion (just like the other commonly used action lines). The RMS value of the adaptive action line is higher than the RMS value of the four commonly used action lines. For the Pectineus (Appendix A.7.1, A.7.2 and A.7.4) the best fit compared to the moment arms of the fibers are the commonly used action lines, because the adaptive action line has a high peak in moment arms near 80° hip flexion but there is no peak at that point in the moment arms of the individual fibers. The RMS value of the adaptive action line is also the highest compared to the commonly used action lines. The Piriformis (Appendix A.8.1, A.8.2 and A.8.4) has a good match between the moment arms of the straight action line and the moment arms of the fibers. So in this case the commonly used action lines win. With the Quadratus Femoris (Appendix A.9.1, A.9.2 and A.9.4) the moment arms of the adaptive line have the better match with the moment arms of the fibers. Just like with the Inferior Gemellus the place of the peak in moment arms is better with the adaptive action line than with any of the four commonly used action lines. With the Rectus Femoris (Appendix A.10.1, A.10.2 and A.10.4) the adaptive action line and the four commonly used action lines do not match the moment arm values or moment arm trend of the individual fibers. The adaptive action line does

have a high RMS error. The knee joint of the Sartorius (Appendix A.11.1, A.11.2 and A.11.4) shows that both the adaptive action line and the four commonly used action lines do not match the moment arm values or moment arm trend of the individual fibers. The hip joint (Appendix A.11.5, A.11.6 and A.11.8) does have a winner and it is the adaptive action line, because unlike the four commonly used action lines its moment arms follow the same trend as the moment arms of the individual fibers. For both joints the RMS value of the adaptive action line is higher than the RMS values of the commonly used action lines.

Summary

The RMS error values of the adaptive action line are high for most of the muscle models, but when it is low the adaptive action line comes close to (most of) the commonly used action lines as we can see with the Inferior Gemellus and the Quadratus Femoris. In the other cases we should also look at the moment arm trends, because that shows when the fibers contract or extend and that kind of behavior should be replicated. Based on the moment arm trend the adaptive action line wins with the Gracilis (knee), the Iliopsoas D, the Iliopsoas F, and the Sartorius (hip). When the adaptive action line does not perform well, it could be due to a bad decision in the attachments of a via point of the adaptive action line. Because of a bad decision in attachment it could have been better that a via point was attached to another bone than the bone which is attached to the origin or insertion area of the muscle. Based on these discussed results we can say that the adaptive action line is a very promising method, but not ready to completely replace the commonly used action lines.

Possible improvements for the adaptive action line are making better decision in the attachments of via points, better smoothing of moment arm values of the adaptive action line instead of using a fourth degree polynomial. Other points of improvement could be doing optimizations for the various settings for the muscle models e.g. the number of slices, or simply using different fiber templates for the various muscle models that we used.

7 Conclusion and Future Work

With the scalar field we can create the contours of the muscle models and combined with the fiber templates, the contours can be matched. With the matching information the fiber from the fiber template can be morphed to the inside of the muscle model. Once the fibers are morphed to the inside of the model, they can be made into a mass-spring system and the inside of the muscle is represented by the fibers based on the mass-spring system.

We have also seen the action lines that are commonly used and those action lines are straight action lines, straight action lines with via points, quadratic Bezier curves and cubic curves. The building methods for a new adaptive action line are based on the center of the contour, the centroid of the fibers and the center of the fiber template.

Based on the results of the experiments, the most promising building method for the adaptive action line is the centroid of the fibers. With the centroid of the fibers building method, the adaptive action lines were made and the experiments that followed were based on finding a maximum threshold for the degree threshold for the adaptive action line. The experiments included every degree to 10 degrees and then 15 degrees and 20 degrees. With the results of the experiments it became clear that the best degree thresholds were in the range from 1° to 3° , so the maximum threshold for which the representation of the deformation still looks good enough is 3° . This can be explained by the fact that with a small degree threshold, an action line will become more detailed or simplified at a quicker rate than with a high degree threshold. The best threshold to use still strongly depends on the muscle model that is used, because some models have the 10 degree threshold as the best threshold while other models have the 3 degree threshold as the best threshold. The fact that a high degree threshold performs better in certain cases, could be due to bad choices in attachments for low degree thresholds when the adaptive action line adds or removes a via point.

The final conclusion is that there is a difference between the commonly used action lines and the adaptive action which is also visible with the RMS values and standard deviations of the errors. Based on the RMS values and the moment arm values and trends we can say that the adaptive action line is a promising method that could be used regularly once the method is improved.

For future research a couple of things can be done to improve the adaptive action line method. The first improvement possibility is doing the experiments again but in a Finite Element Method (FEM) or Finite Volume Method (FVM) environment. This would improve the quality of the deformation of the muscle model and the position estimation of the fiber location, because with the mass-spring systems the deformation and the fibers positions are more flexible than one would normally encounter in real muscles. A disadvantage of using a FEM or FVM environment is that it would slow

down the simulation a lot.

The second improvement possibility is using optimization on several points in the method. The optimization could be done on the mass-spring system, which will result in an optimized stable mass-spring system with the best stiffness values and damping coefficients for the different muscle models. This optimization will improve the simulation of the separate fibers and could result in more stable and eventually more accurate moment arm values for the fibers.

A different improvement could be achieved by preprocessing the muscle model meshes. When we start with knowledge about the shape of the mesh and the distribution of the vertices, it could be possible to estimate the optimal number of slices for every muscle model. The optimal number of slices would improve the representation of the fibers and the adaptive action line, because both depend on the number of slices. Also the optimal number of spring connections between the fibers and the mesh could be estimated if the mesh is preprocessed and this will improve the deformation of the fibers when the muscle mesh is deformed.

It could also be possible to do an optimization for the via point attachments for the building methods and the adaptive action line by eliminating the possibility of bad choices in attachments. It could be beneficial if some via points are attached to the bone that belongs to the muscle origin instead of the insertion or vice versa. Even better could be if the via point would attach to the bone closest to the via point. Adjusting the attachment of via points could result in more stable moment arm values and possibly a better action line fitting.

Another possible improvement could be to use different fiber templates for the various muscle models instead of only using the simple parallel fiber template. If the fibers of the muscle models would be modelled using the best corresponding fiber template for the muscle it could improve the quality of the simulation and also increase the accuracy of the results from the simulation.

References

- [1] <http://www.einfopedia.com/human-bones-longest-or-largest-human-body-bones.php>.
- [2] Allison S. Arnold, Silvia Salinas, Deanna J. Asakawa, and Scott L. Delp. Accuracy of muscle moment arms estimated from mri-based musculoskeletal models of the lower extremity. *Computer Aided Surgery*, 5:108–119, 2000.
- [3] Amaury Aubel and Daniel Thalmann. Interactive modeling of the human musculature, 2001.
- [4] Yacine Berranen, Mitsuhiro Hayashibe, David Guiraud, and Benjamin Gilles. Real-time muscle deformation via decoupled modeling of solid and muscle fiber mechanics. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2014*, volume 8674 of *Lecture Notes in Computer Science*, pages 65–72. 2014.
- [5] Silvia S. Blemker and Scott L. Delp. Three-dimensional representation of complex muscle architecture and geometries. *Annals of Biomedical Engineering*, 33(5):661–673, May 2005.
- [6] F. Dong, G. J. Clapworthy, M. A. Krokos, and J. Yao. An Anatomy-Based Approach to Human Muscle Modeling and Deformation. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):154–170, April 2002.
- [7] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds, 2004.
- [8] Jason P. Halloran, Marko Ackermann, Ahmet Erdemir, and Antonie J. van den Bogert. Concurrent musculoskeletal dynamics and finite element analysis predicts altered gait patterns to reduce foot tissue loading. *Journal of Biomechanics*, 2010.
- [9] K. Hormann and M. S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics*, 25(4):1424–1441, October 2006.
- [10] J. Kohout, G. J. Clapworthy, Y. Toa, F. Dong, P. Kellnhofer, D. Cholt, and Y. Zhao. Technologies for modelling fibrous muscle in motion, 2012.
- [11] J. Kohout and M. Kukačka. Real-time modelling of fibrous muscle. *Computer Graphics Forum*, 2014.
- [12] Francis P. Laclé. Real-time collision-aware musculoskeletal model for virtual human animation, 2014.

- [13] Dongwoon Lee, Michael Glueck, Azam Khan, Eugene Fiume, and Ken Jackson. A Survey of Modeling and Simulation of Skeletal Muscle. *ACM Transactions on Graphics*, 28(4), July 2010.
- [14] Keng Siang Lee and Golam Ashraf. Simplified muscle dynamics for appealing real-time skin deformation, 2007.
- [15] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation, 1995.
- [16] Y.T. Lu, H.X. Zhu, S. Richmond, and J. Middleton. Modelling skeletal muscle fibre orientation arrangement. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(12):1079–1088, December 2011.
- [17] Luciana Porcher Nedel and Daniel Thalmann. Real time muscle deformations using mass-spring systems, 1998.
- [18] C. Antonio Sánchez, John E. Lloyd, Sidney Fels, and Purang Abolmaesumi. Embedding digitized fibre fields in finite element models of muscles. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2014.
- [19] Michael Sherman, Ajay Seth, and Scott Delp. How to compute muscle moment arm using generalized coordinates, 2010.
- [20] Jie Tan, Greg Turk, and C. Karen Liu. Soft Body Locomotion. *ACM Trans. Graph.*, 31(4):26:1–26:11, July 2012.
- [21] J. Teran, S. Blemker, V. Ng Thow Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle, 2003.
- [22] J. Teran, E. Sifakis, S. Blemker, V. Ng Thow Hing, C. Lau, and R. Fedkiw. Creating and simulating skeletal muscle from the visible human data set, 2000.
- [23] Jack M. Wang, Samuel R. Hamner, Scott L. Delp, Vladlen Koltun, and More Specifically. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. Graph.*, 2012.
- [24] Joshua D. Webb, Silvia S. Blemker, and Scott L. Delp. 3D finite element models of shoulder muscles for computing lines of actions and moment arms. *Computer Methods in Biomechanics and Biomedical Engineering*, 2012.
- [25] Victor B. Zordan, Bhrigu Celly, Bill Chiu, and Paul C. DiLorenzo. Breathe easy: Model and control of simulated respiration for animation, 2004.

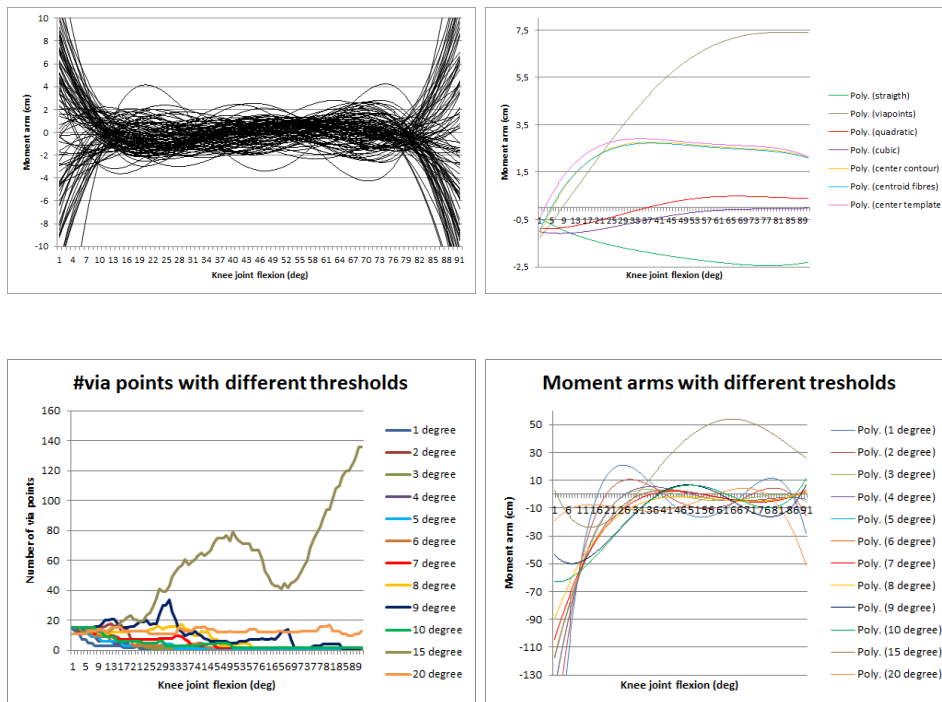
A Moment Arm Simulation Graphs

Here are all the graphs listed coming from the experiments. The graphs are listed in an alphabetic order based on the muscle names. When a muscle spans multiple joints, the muscle will have graphs for all those joints. In this case there are three muscles that are affected by the knee as well as the hip joint and those muscles are Biceps Femoris B, Gracilis, and Sartorius.

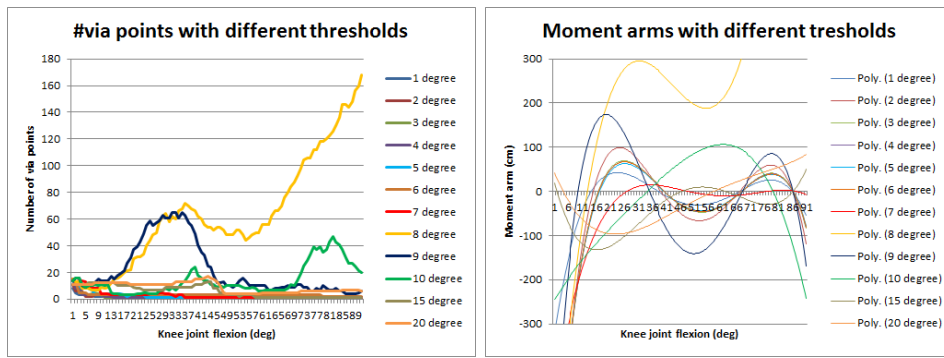
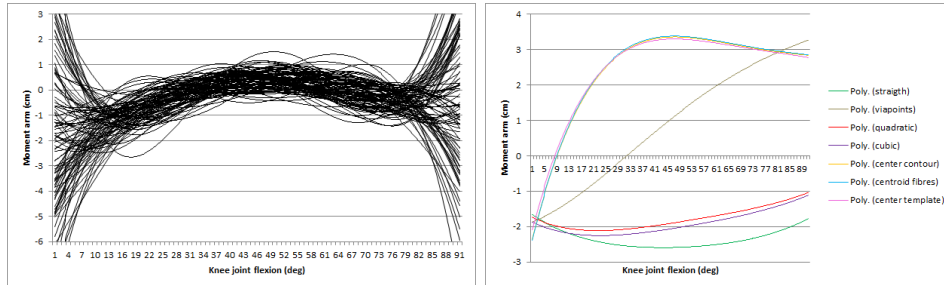
In every subsection the graphs are ordered the same. The first graph shows the smoothed action lines per fiber. The second graph shows the action lines that are commonly used and the building method action lines that can be used for the adaptive action line. The third graph shows the number of via points per degree threshold and the fourth is the corresponding graph that shows the adaptive action line moment arms per degree threshold.

The moment arm values (in the graphs of the fibers and the different degree thresholds) are smoothed using a fourth degree polynomial and thus start with the name "Poly" in the graphs.

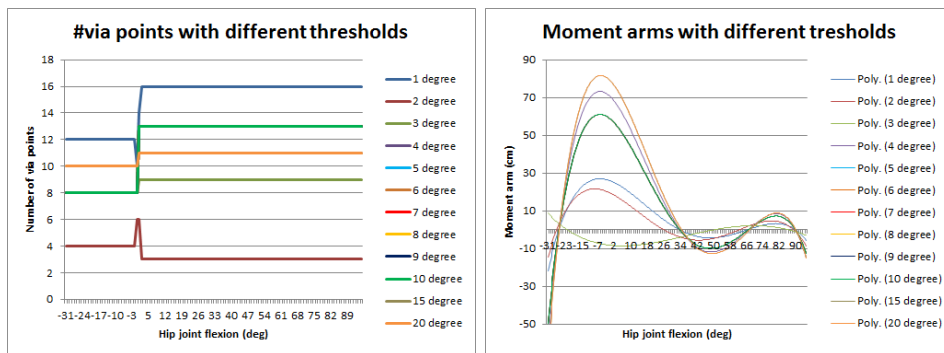
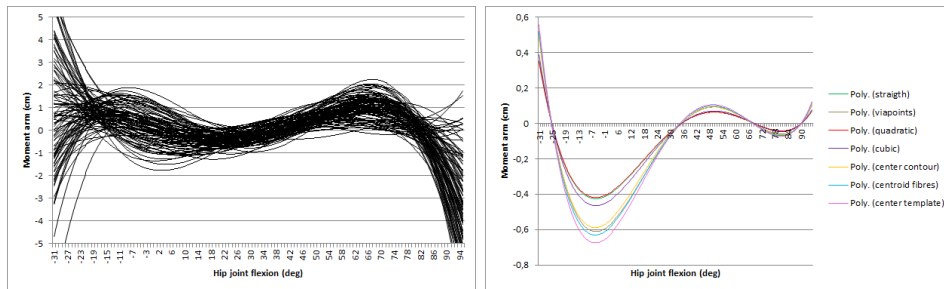
A.1 Biceps Femoris A



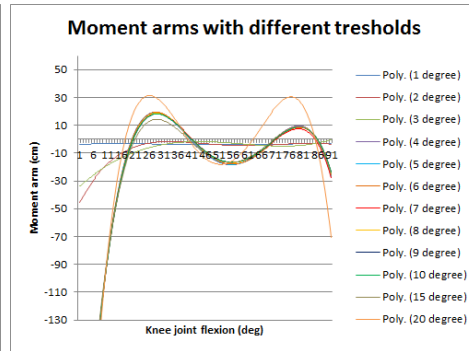
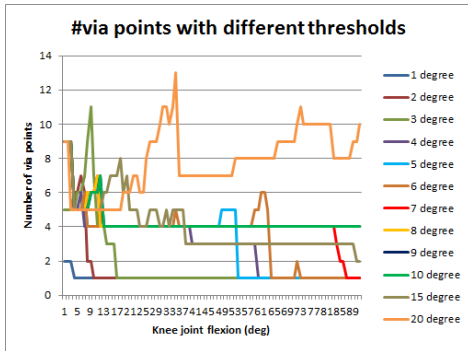
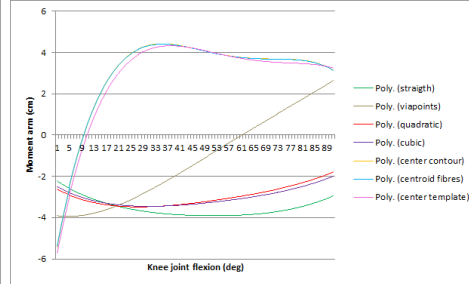
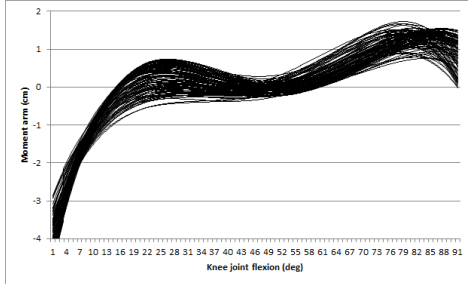
A.2 Biceps Femoris B



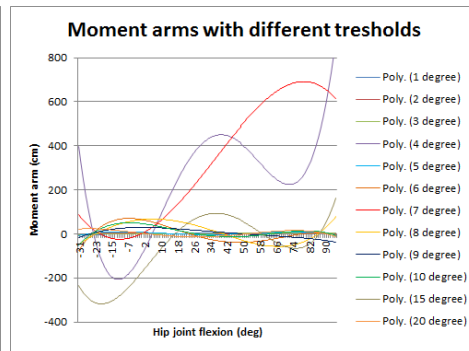
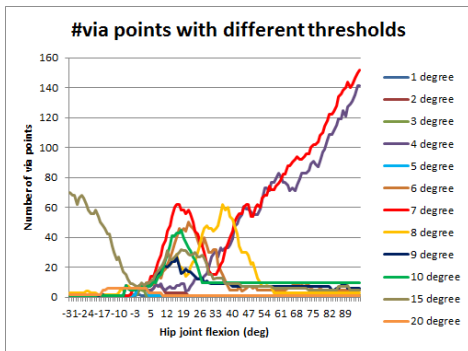
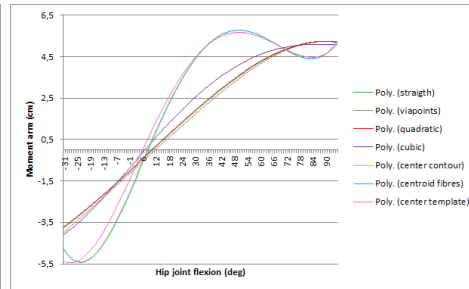
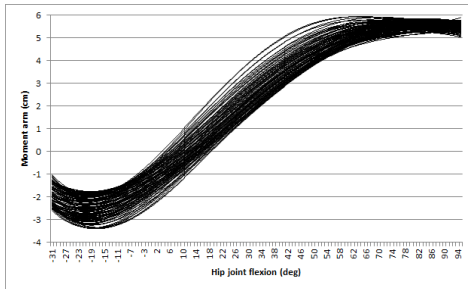
The following graphs are results from the hip flexion simulation.



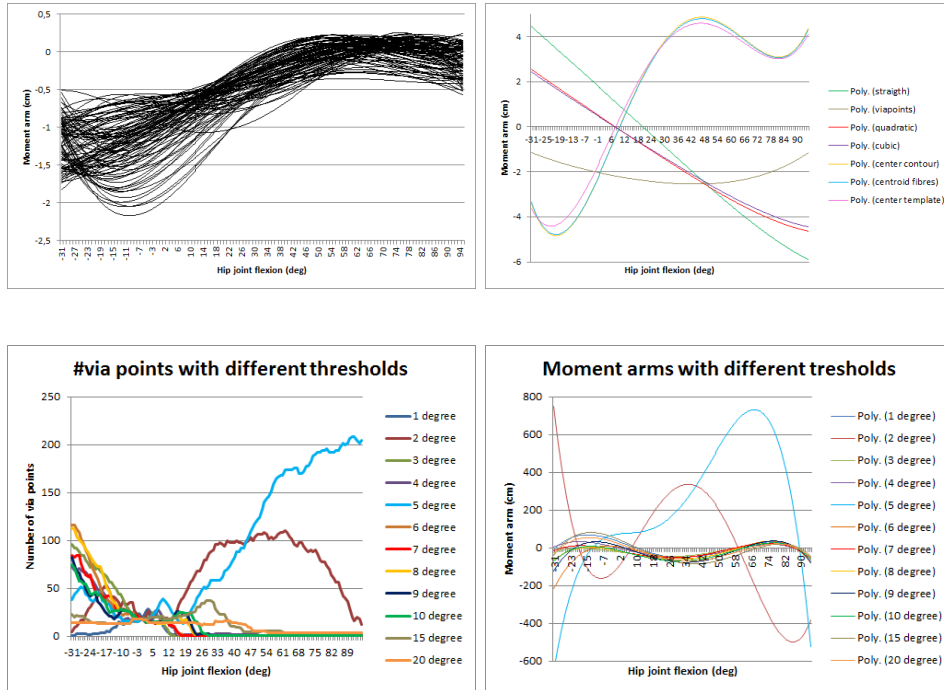
A.3 Gracilis



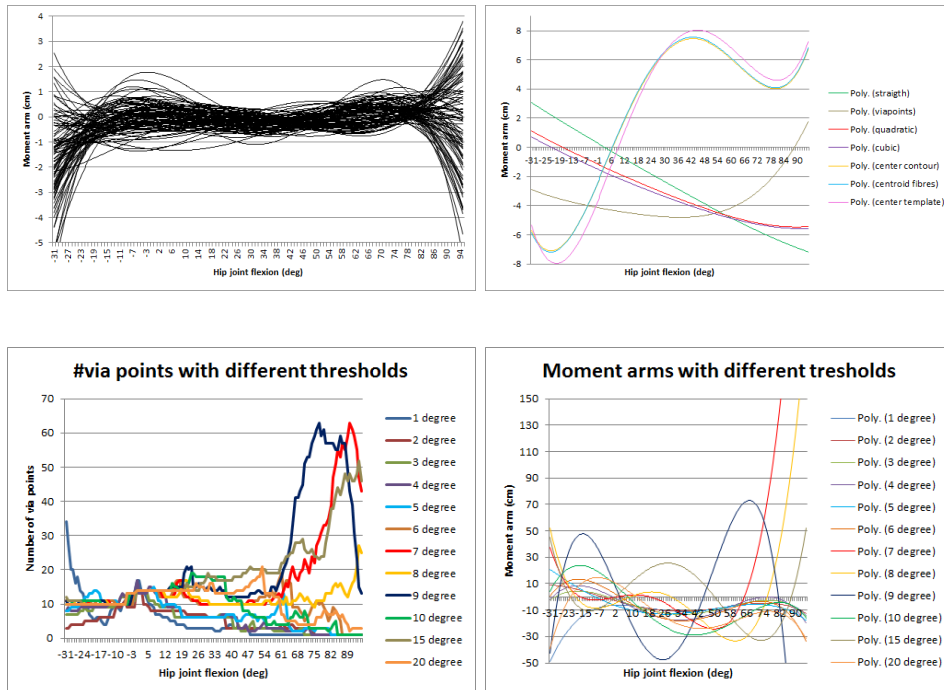
The following graphs are results from the hip flexion simulation.



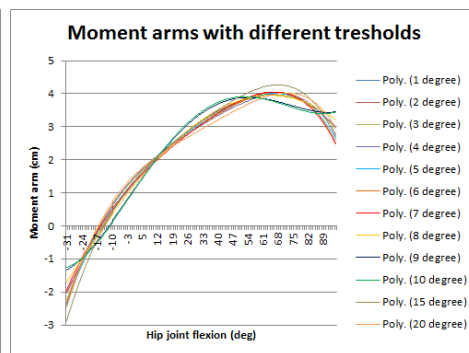
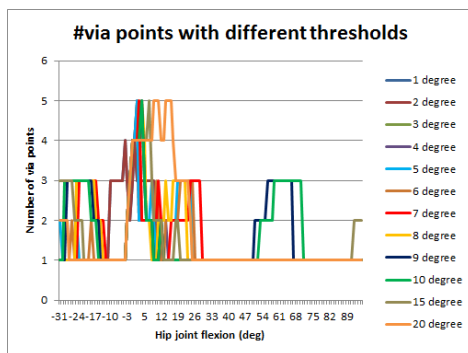
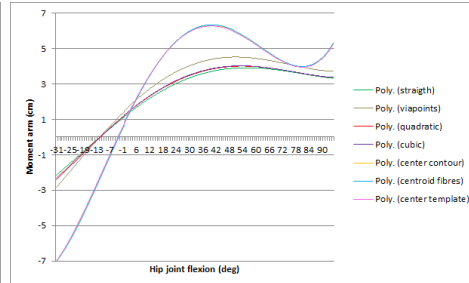
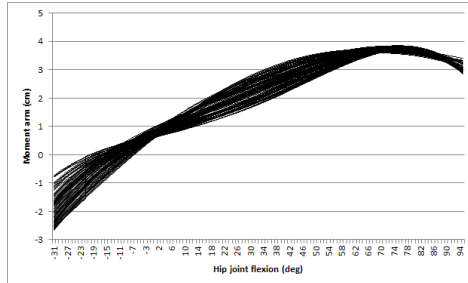
A.4 Iliopsoas D



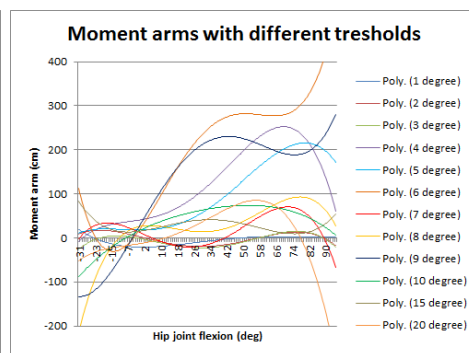
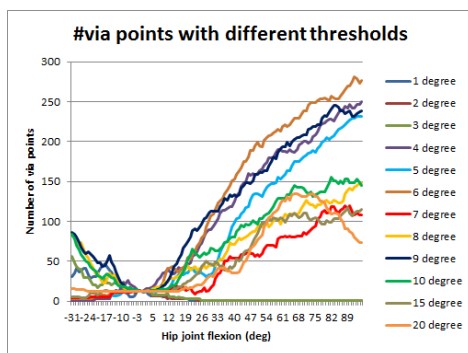
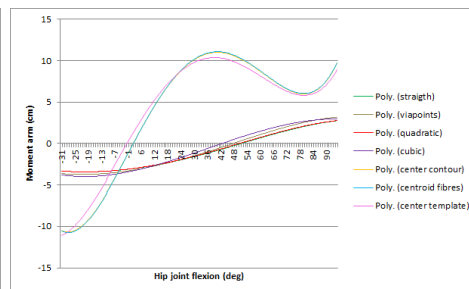
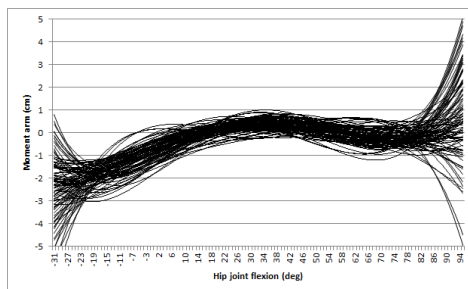
A.5 Iliopsoas F



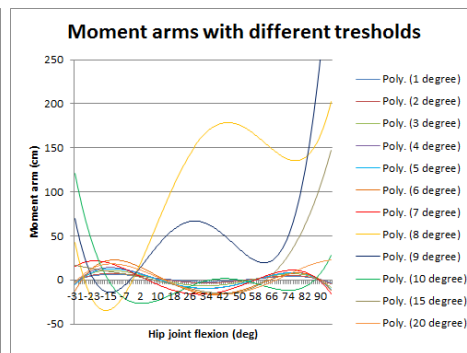
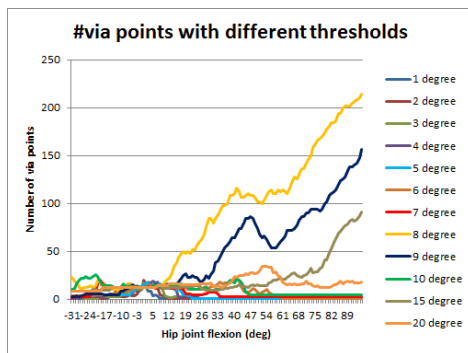
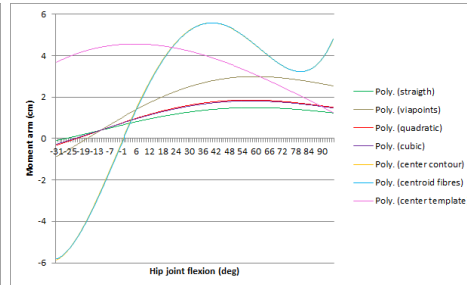
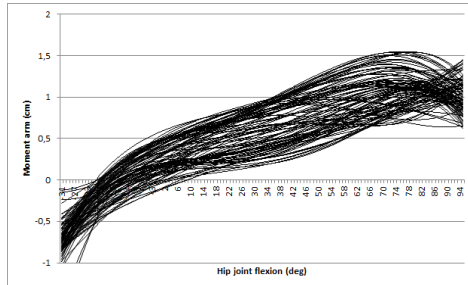
A.6 Inferior Gemellus



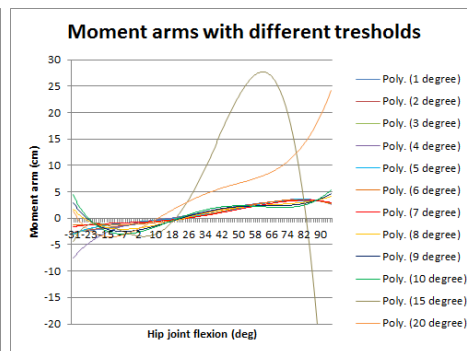
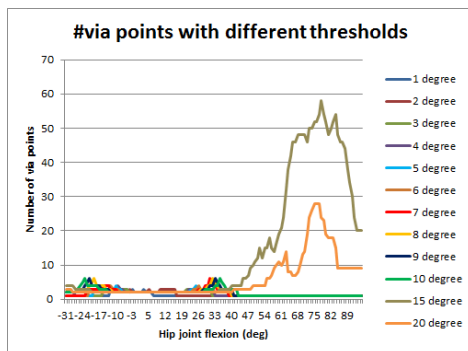
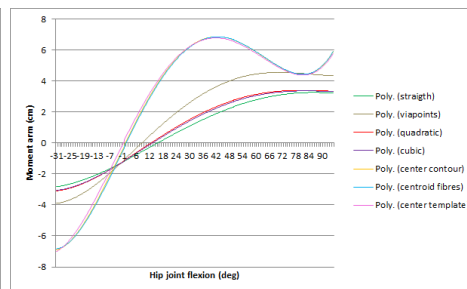
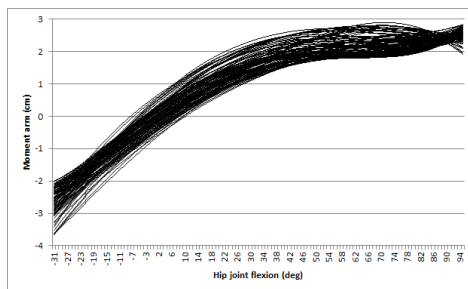
A.7 Pectineus



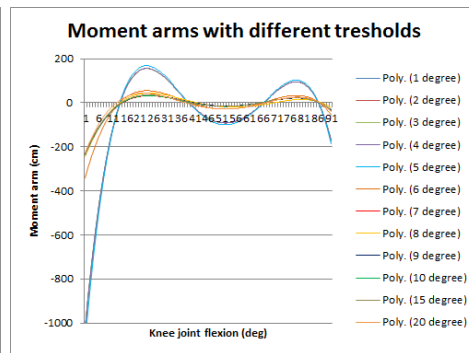
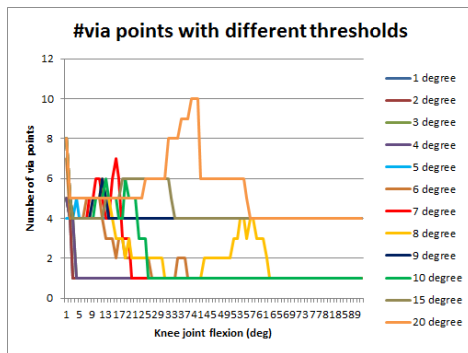
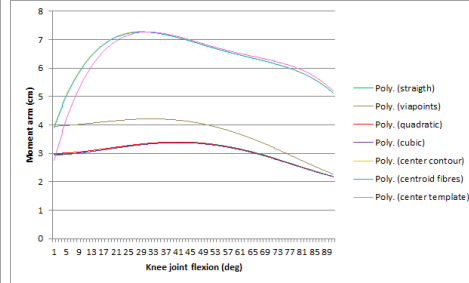
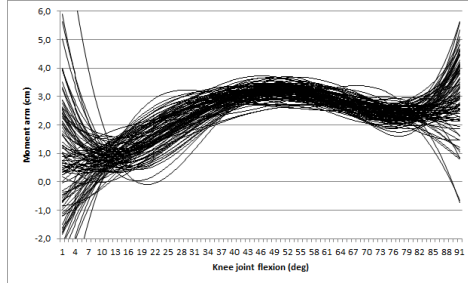
A.8 Piriformis



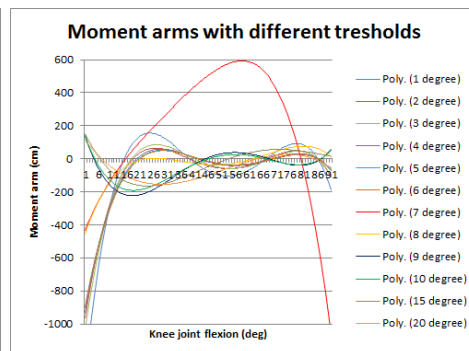
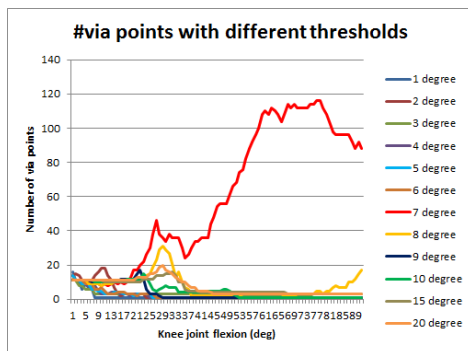
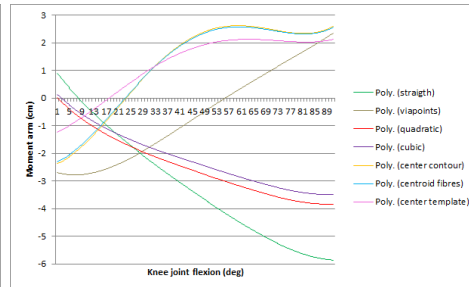
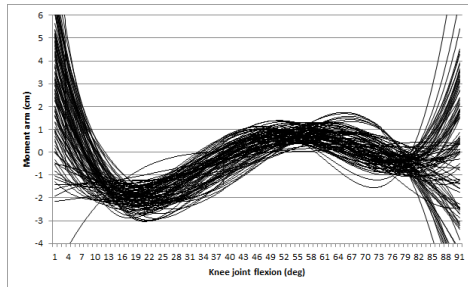
A.9 Quadratis Femoris



A.10 Rectus Femoris



A.11 Sartorius



The following graphs are results from the hip flexion simulation.

