Optimization of Muscle Activations for Cycling

Pratiksha Ganesan, Risheekesh Kesavan, Mikayla Schneider, Ravesh Sukhnandan

Abstract

The human body is ”redundant” in that it has more muscles than degrees of freedom, resulting in numerous ways to execute a given task. Identifying the objective function that our neural system minimizes to perform a given task is useful for understanding our musculoskeletal system in order to treat physical impairments, improve sports performance, and design bio-inspired technologies. Cycling is well-suited for this optimization project because it can be accurately modeled in the 2-D plane with six muscles and four segments. We can also choose the level of complexity for solving the equations of motion (static vs. dynamic), modeling the production of muscle forces (linear vs. non-linear), and application of muscle forces (as line forces vs. only torques). Our motivation is to understand the muscle forces that are activated during cycling and see if we can closely match our optimized activations to experimental data from literature. After running the optimizer we got the minimum activations for different crank angles of the cycle. The activations for the scaled problem obtained did not exactly match the EMG data obtained from literature,however it shows similar trends. The disparity might be due to a variety of factors like the gross simplification of the model, the assumption that only six muscles are in action .

Index Terms

Cycling, biomechanics, optimization, muscles, activations

I. Introduction

THE human body has more muscles than degrees of freedom causing the human muscle system to be “redundant” in nature [1]. Due to this, any task that involves muscle movement can be done in numerous ways- by the activation of different combinations of muscles that can generate the same joint torque in order to achieve the movement. This raises the question: how does the human body decide which muscle combination to use? The human body is incredibly efficient in its ability to rapidly configure our complex musculoskeletal system in order to achieve some desired task, such as grasping a cup of coffee, but we do not fully understand how each configuration and execution is planned by our neural system. The ”objective” of our neural system may depend on a number of factors including the context of the task, an individual’s current physical state, external conditions and more. Some objectives could be to minimize time, relative muscle force, distance traveled by each joint, energy cost and more. Once the objective function is known, it is easier to solve for individual muscle forces and the torques applied at each joint during activities such as cycling, walking, running, and reaching. The process of analyzing muscle forces and joint angles (kinetics and kinematics) has been used extensively in the field of modeling and simulation of human biomechanics, physical therapy and orthopedic surgery, optimization of sports performances and robotics [2].

Our project is focused on understanding the muscle activations required for a cycling task through an optimization approach. Through experimental research using EMGs, it has been determined that the muscle activation patterns are fairly common for different people performing the same action. Using experimental data of applied pedal forces and torques, we aim to optimize the lower limb muscle activation during cycling using muscle and joint parameters from literature, and compare it to previously gathered EMG data. In order to optimize muscle activation, the sum of squares of activation energy is considered as the objective function as it is a proxy for the metabolic cost. The comparison to the previously gathered EMG data will inform us about the performance and accuracy of the constructed model. If our optimized activations are similar to the experimental EMG data, this could indicate that our objective function is a good proxy for our neural system’s objective function while cycling. Given that we are making a lot of simplifications, activations that don’t match the experimental data might also indicate that we need to adjust the complexity of our model. The muscle activation problem is redundant in nature, with more muscles than degrees of freedom, making the solution non-trivial and ideal for optimization. The key trade-off for the problem is between complexity of the model and performance of the model. The key constraints arise from the maximum ability of different muscles to do different tasks.

II. Problem Statement

Our project seeks to identify the muscle activations required to cycle a stationary bicycle with prescribed torque and forces at the pedal with the goal of minimizing the energy cost. We will use a 2-D model of a cyclist driven by six muscles, all of which have different limits on the force they can produce and joints over which they act. The following equations describe our optimization problem in negative null form.

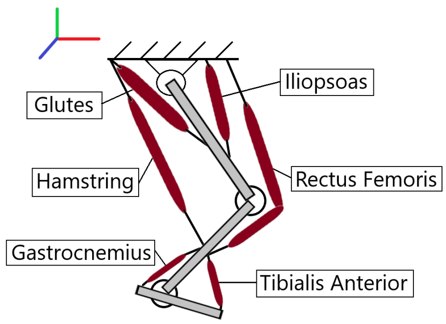


Fig. 1: Problem depiction with all 6 muscles.

Fig. 2: Coordinate definitions. h: hip joint, k: knee joint, A: ankle joint, T: toe, P: pedal center, C: crank center. The stationary global coordinate frame $i_0 j_0 k_0$ coincides with the hip joint such that gravity points in the $-j_0$ direction. Segment $\overline{h,k}$ represents the thigh, $\overline{k,A}$ represents the shank, $\overline{A,T}$ represents the foot. The foot is connected to the pedal by segment $\overline{T,P}$. The pedal is connected to the bicycle by segment $\overline{P,C}$.

$$\underset{\mathbf{a}\in\mathbb{R}^6}{\text{minimize}} \quad f(\mathbf{a}) = \sum_{i=1}^{6} a_i^2, \text{(sum of squared activations = metabolic cost) where } \mathbf{a} = [a_H, a_G, a_{IP}, a_{RF}, a_{GA}, a_{TA}] \tag{1a}$$

subject to

$$-F^{H,max} \times a^H \times r^h - F^{G,max} \times a^G \times r^h + F^{I,max} \times a^I \times r^h + F^{R,max} \times a^R \times r^h - M_\theta^{ext,h} = 0 \tag{1b}$$

$$-F^{H,max} \times a^H \times r^k + F^{R,max} \times a^R \times r^k - M_\theta^{ext,k} = 0 \tag{1c}$$

$$F^{C,max} \times a^C \times r^a - F^{T,max} \times a^T \times r^a - M_\theta^{ext,a} = 0 \tag{1d}$$

$$-\mathbf{a} \leq 0.1 \tag{1e}$$

$$\mathbf{a} \leq 1 \tag{1f}$$

$M_\theta^{ext,h}$, $M_\theta^{ext,k}$, and $M_\theta^{ext,a}$ are solved for at each angle by the equations in the Appendix. Essentially, these are known scalar parameters for each snapshot in time.

TABLE I: Model Parameters

Symbol	Description	Value used in Optimization Problem	Value from Literature	Units
r^h	Hip Radius	0.081	0.081	m
r^k	Knee Radius	0.035	0.035	m
r^a	Ankle Radius	0.052	0.052	m
$F^{G,max}$	Max Gluteals Force	15000	3000	N
$F^{IP,max}$	Max Iliopsoas Force	7500	1500	N
$F^{H,max}$	Max Hamstrings Force	15000	3000	N
$F^{RF,max}$	Max Rectus Femoris Force	6000	1200	N
$F^{GA,max}$	Max Gastrocnemius Force	3000	3000	N
$F^{TA,max}$	Max Tibialis Anterior Force	2500	2500	N

III. Analysis of Problem Statement

A. Simplifications and Assumptions

In our problem, we have three equality constraints (Eq. 1b-1d) which represents the sum of torques at the hip, knee and ankle respectively. However, we have 6 forces associated with the muscles of Fig. 1, and hence we have a redundantly actuated system.

While the muscle exerts a force on the bone it is attached to, we make the simplifying assumption that each muscle only exerts a pure moment about its associated joint equal to product of the muscle's activation, joint radius and the max force of the muscle, and disregard the force contribution of that muscle to the force equations. So, for a given muscle w that acts on joint u, the torque produced by this muscle on a segment of the leg $\overline{u,v}$ is given by $M^{w,u} = F^{w,max} \times \ r^u \times \ a^w$. Here, r^u, $F^{w,max}$ and a^w refer to the radius of the joint (Table I), maximum force of the muscle (Table I) and activation of the muscle, respectively. This serves to simplify the problem significantly, because if the actual muscle forces are included in the force equilibrium equation, then we would need to keep track of where the muscle inserts into the bone and how the line of force of the muscles changes with motion of the joint.

The next simplifying assumption made is to treat the problem as a quasistatic problem, where all time derivative terms (such as velocity and acceleration) are set to 0. When cycling, this is clearly not the case because the leg segments are in motion. However, prior optimization work on muscle activations have shown that dynamic and static optimizations of muscle forces can converge upon similar solutions [3].

The third simplifying assumption made is to treat the motion of the legs and the cycle as happening in the 2D-plane (xy), and ignore forces in the z direction (Fig. 2). While there is some out of plane motion of the legs, to a first approximation this is a valid simplifying approach because the pedal moves purely in a plane.

With the simplifying assumptions of planar motion, pure muscle torque and quasi-static motion, we can produce the hip, knee and ankle torque from existing pedal force data [4] using the following methodology. From the data set, we get the reaction force of the pedal on the foot in the global x and y directions at a given crank angle. We assume that the angle of the ankle is the same as the angle of the pedal, so we can solve the angle of the knee and hip using inverse trigonometry. With the force at the foot known and the angles of all the joints known, we then compute the net torques at the knee, hip and ankle to produce the pedal forces at the foot (Appendix A). We then use these torques in our equality constraints, and run our optimization to find the optimum activation for each muscle at this given pedal angle. Because we are assuming quasistatic motion, we can repeat this optimization procedure at each of the 360 points for a single pedal cycle to get the optimum activations at each of these angles.

TABLE II: Problem Classification

Attribute	**Classification**
Problem Class	Non-linear Programming (NLP)
Continuity	Objective and Constraints are continuous
Smoothness	Objective and constraints have continuous first and second derivatives
Convexity	This is a convex problem. The objective function $f(\mathbf{a})$ is convex and all constraints **g(x)** and **h(x)** are linear.
Undefined regions	No regions exist where the gradient is undefined.
Size	Number of variables: 6 ($a_1, a_2, ...a_6$) Number of equality constraints: 3 (sum of moment equations for the hip, knee and ankle) Number of inequality constraints: 6 (one upper and one lower bound for each of the six muscle activations)

As the activation is related to the force capacity of a given muscle for some neural input to that muscle, the activation cannot be smaller than 0 (i.e. 0%) or larger than 1 (i.e. 100%). However, it was observed in cycling that for the muscles measured, there was no muscle that showed identically 0 activations [1]. Hence, we set the lower bound to 0.1, which is in the range of some of the minimum activations from literature. However, this can be viewed as a practical constraint because there is not a physiological justification for setting all of the activations at precisely 0.1. If these constraints are active, the effect of these lower bounds on activation can be analyzed further via the Lagrange multipliers for that equality constraint and sensitivity analysis on the lower bound (see Sections IV and V).

We also chose to scale the max force of the iliopsoas, rectus femoris, hamstring and gluteals by a factor of 5. The reason for this is that there are more muscles than these four which act over the hip and knee, but weren't modelled to reduce the dimensionality and complexity of the problem. To compensate, we scaled these forces by an arbitrary value of 5. Further analysis of how this scale factor affects the activation profile will be discussed in the Sensitivity Analysis section.

Note that since we are minimizing muscle activations, it may be possible that the trivial solution of $\alpha = 0$ may be provided from the optimization. This will most likely not be the case as at each time step, we are including the pedal force information (see Eq. 4 and Eq. 7 of Appendix A). Since the foot needs to exert some non-zero force on the pedals, then this will most likely require some non-zero muscle force, and hence non-zero muscle activations.

Another simplification pertains to the muscle force production model. In reality, muscle tissue exhibits complex behavior that can be modelled as a neural-activation scaled non-linear spring and damper in parallel with a passive elasticity, which is in series with another passive elastic element [5]. In our beginning implementation, we will use a simpler model where the muscle force produced at any given moment is simply a linear function of the neural activation and the maximum muscle force (Eq. 1b-1d). This allows us to ignore the presence of viscous (time-varying) behavior of muscle tissue, which will allow us to not violate the static condition assumption.

B. Problem type and convexities/non-convexities

The functional relationship between the angle of the crankshaft of the bicycle and the pedal force is taken from Kautz et al. [4]. The problem can be classified as a non-linear programming optimization problem (NLP) as we have a continuous and non-linear objective function (Eq.1). As our objective function is convex and the equality and inequality constraints are all linear functions of our decision variables, *a*, then this is a convex optimization problem (Table II). All functions are continuous and smooth and defined at all points in the decision space. The gradients are defined at all points in the decision space. As a convex optimization problem, any local minima we find, and its associated minimizer, is guaranteed to be a global minimum. We will solve our non-linear optimization problem with MATLAB's *fmincon* solver using both sqp (an active set strategy) and an interior-point (barrier-function based) algorithm.

IV. Optimization Study

Our problem is a convex optimization problem as our objective is a convex function and our equality constraints are all linear (Table II. As the problem is convex, MATLAB's fmincon was used to evaluate the solution. Both SQP and interior-point algorithms were evaluated to compare their convergence and speed of execution. It was found that SQP converged more quickly than interior-point (8 *ms* vs 21 *ms*). This may be attributed to the fact that our the objective function is subject to twelve inequality constraints (lower and upper bounds of each of the six activations), and SQP works with an active set strategy thereby only considering constraints that are active and hence converges to a solution faster. Upon evaluation both algorithms produced the same global minima at all 360 points with exit flags of 1 for each point. Because SQP executed 2.6 times faster than that of the interior-point, we chose SQP as our algorithm for all further analysis.

Considering the base case at a crank angle of 0, the external moments at hip,knee and ankle were 39.48 Nm, -15.33 Nm, 4.39 Nm respectively. The lower bounds were 0.1 and upper bounds were 1 for all activations. The minimum of the total activations obtained was: 0.5676.

A. BASE CASE:

Parameters:
Crank angle = 0°
External moments: $M_0^{ext,h} = 39.48$ $M_0^{ext,k} = -15.33$ $M_0^{ext,a} = 4.39$
Minimum = 0.5676
Minimizer = [0.177, 0.323, 0.100, 0.100, 0.636, 0.100]
Lower bound = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
Upper bound = [0, 0, 0, 0, 0, 0, 0]

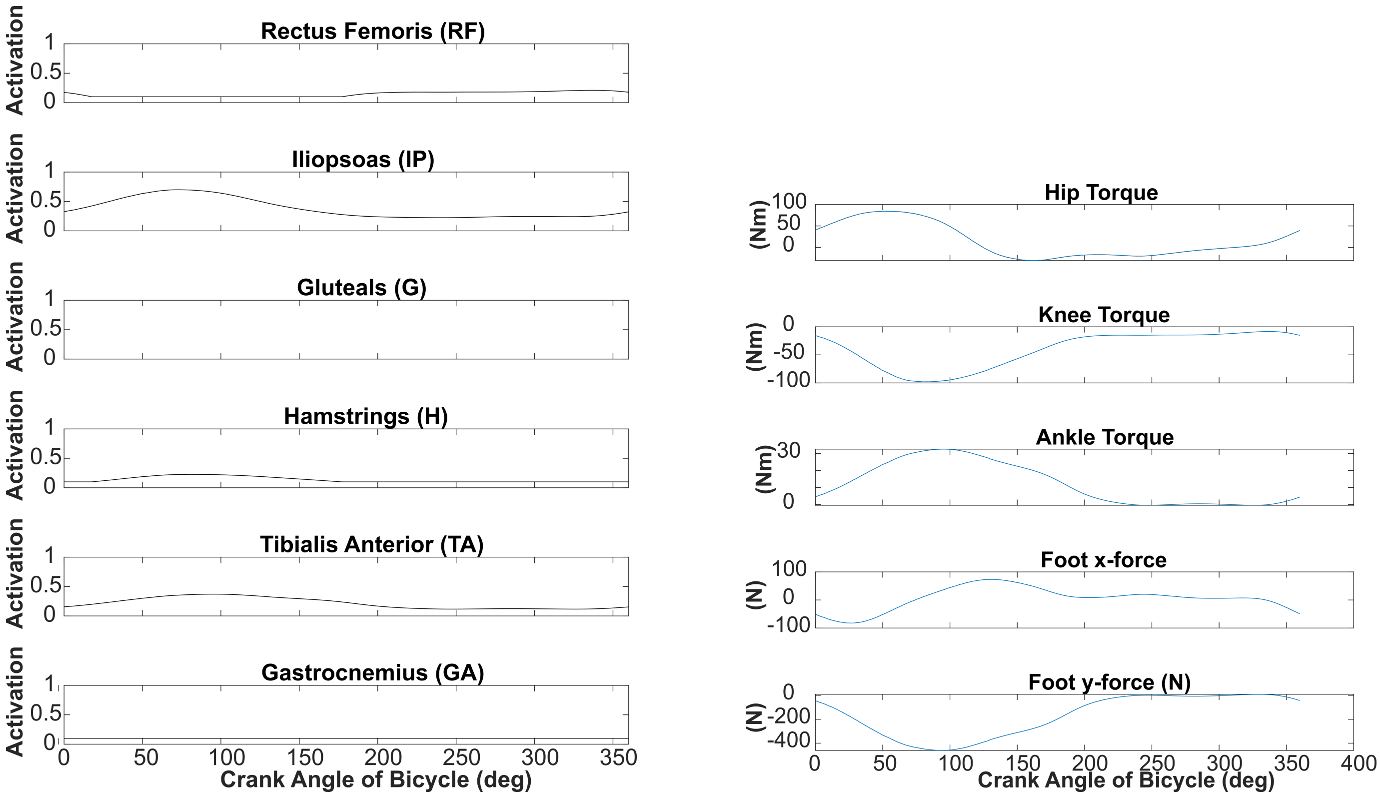
Termination message:
Exit flag = 1. Feasible point with lower objective function value found. Local minimum found that satisfies the constraints.Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied within the value of the constraint tolerance.

Local optimality:
Lagrange multipliers for equality and inequality constraints were determined, the complementary and non-negativity conditions were satisfied, and the hessian obtained is positive definite. Thus, KKT conditions were satisfied.

Global optimality:
As the objective function is a parabola (sum of squares), it is convex. As exit flags of 1 were obtained and the KKT conditions were satisfied, global minimum was obtained.

Justification of modelling assumptions:
When the lower bounds for activations were set to 0, the activations obtained from the algorithm tend to be very small values to the order of 10^{-31}. To make the solutions align more with experimental EMG, lower bound was set to 0.1, the solutions obtained with this were more in-line with the experimental data from literature. Realistically, when any physical activity is performed, all muscles would have some activation regardless of if they are being used or not.

Optimization plots:
It is evident from Table III that all the Lagrange multipliers of the equality constraints are non-zero, which means that they are active. The Lagrange multipliers are in order of 10-3 because of the scaling factor between the moments, maximum forces and the activations. As these moment equations represent physical laws that must hold, they cannot be relaxed or tightened.



(a) (b)

Fig. 3: a) Muscle activations as a function of crank angle. b) External forces and torques as a function of crank angle

TABLE III: Solutions and Lagrange multipliers

Crank angle	Activation (RF, IP, G, H, TA, GA)	External moment	Lagrange multiplier	Lower bound	Upper bound
0°	0.177	$M_0^{ext,h} = 39.48$	-1.06×10^{-3}	0.00	0
	0.323	$M_0^{ext,k} = -15.33$	7.78×10^{-4}	0.00	0
	0.100	$M_0^{ext,a} = 4.39$	-9.75×10^{-3}	1.49	0
	0.100			1.08	0
	0.634			0.00	0
	0.100			7.81	0
90°	0.100	$M_{90}^{ext,h} = 63.83$	-22.23×10^{-3}	0.38	0
	0.676	$M_{90}^{ext,k} = -97.34$	6.01×10^{-3}	0.00	0
	0.100	$M_{90}^{ext,a} = 32.45$	-1.31×10^{-2}	2.90	0
	0.225			0.00	0
	0.850			0.00	0
	0.100			10.39	0
180°	0.112	$M_{180}^{ext,h} = -24.63$	-8.89×10^{-4}	0.00	0
	0.270	$M_{180}^{ext,k} = -29.07$	9.96×10^{-4}	0.00	0
	0.100	$M_{180}^{ext,a} = 14.12$	-1.09×10^{-2}	1.28	0
	0.100			0.76	0
	0.709			0.00	0
	0.100			8.71	0
270°	0.180	$M_{270}^{ext,h} = -11.78$	-7.78×10^{-4}	0.00	0
	0.236	$M_{270}^{ext,k} = -14.66$	8.55×10^{-5}	0.00	0
	0.100	$M_{270}^{ext,a} = 0.2$	-9.25×10^{-3}	1.15	0
	0.100			1.10	0
	0.602			0.00	0
	0.100			7.42	0

At the four crank angles in Table III, the Lagrange multipliers of the lower bounds are zero for some bounds and non zero for others. The lower bounds of activations of gluteals, hamstring and gastrocnemius are most frequently active, with the rectus femoris active only at the 180°crank angle. These correspond with the general trends seen in the full activation profiles across an entire 360°cycle (Fig. 3a), where the gluteals and the gastrocnemius remain at the lower bound for the entire cycle. With regards to the choice of the lower bound of the activation as 0.1, we observe that the lower bound of the gastrocnemius is the most sensitive constraint as it has the largest Lagrange multiplier. As such, if we were to tighten the lower bound of the gastrocnemius from 0.1 to 0.11 for instance, we would expect to see the objective function at the 0°crank angle increase from 0.5676 to 0.6457. Hence increasing the lower bound has a tendency to lead to optimum activation profiles that have a higher sum of squared activations. We note that the non-zero lower bound in activation is physically realistic based on measured EMG data from cycling [1]. If indeed a lower bound that goes towards 0 results in lower metabolic cost, then it does open the question of why there is this base level of activation present in the EMG. This is beyond the scope of this paper, but potentially has to do with the nervous system keeping the muscle ”primed” to allow it to activate more quickly when needed [5]. The Lagrange multipliers of the upper bounds are all zero, indicating that the upper bounds of all muscle activations are inactive.

V. Sensitivity Analysis

The Lagrange multipliers for our lower bounds, upper bounds, and equality constraints can be seen in Table III. Since the Lagrange multipliers for the upper bounds are all zeros, we elected not to perform a sensitivity analysis on them but did so for the lower bounds. To do this, we tested a sample of 7 values ranging from -30% to +30% of the original lower bound value. For each optimization iteration, we held 5/6 muscles' lower bound value constant, and tested a new lower bound for the other muscle from the sample of 7 values. This resulted in a total of 42 (7 values * 6 muscles) unique optimization iteration, each of which were done over the full range of the crank angles (360 degrees). The outcome we examined was the % change in the objective function (the sum of the squared activations) compared to the result from our base values. Table IV shows a snapshot of the results for angles of 0, 90, 180, and 270, which are the angles for which we displayed the Lagrange multipliers in Table III. This table only shows the results for a -30% change. The value of the Lagrange multiplier (either zero or positive nonzero) is shown next to the resulting % change in the function evaluation, demonstrating that the changes in the lower bounds we induced did generally match the change expected from the Lagrange multipliers, except for the two cases shown in bold.

TABLE IV: Lower Bounds Sensitivity Analysis

Crank angle	Muscle Changed	Lagrange multiplier	% Change in Sum of Squared Activations
0°	Rectus Femoris	0	0
	Iliopsoas	0	0
	Gluteals	+	13.7
	Hamstrings	+	21.3
	Tibialis Anterior	0	0
	Gastrocnemius	+	7.8
90°	Rectus Femoris	+	1.5
	Iliopsoas	0	0
	Gluteals	**+**	**0**
	Hamstrings	**0**	**12.2**
	Tibialis Anterior	0	0
	Gastrocnemius	+	4.5
180°	Rectus Femoris	0	0
	Iliopsoas	0	0
	Gluteals	+	1.3
	Hamstrings	+	20.1
	Tibialis Anterior	0	0
	Gastrocnemius	+	12.1
270°	Rectus Femoris	0	0
	Iliopsoas	0	0
	Gluteals	+	20.0
	Hamstrings	+	22.4
	Tibialis Anterior	0	0
	Gastrocnemius	+	6.0

The other nonzero Lagrange multipliers were for the equality constraints, which were determined by the required external moments at the hip, knee, and ankle joints at each crank angle. Since the external moments are solved for by the equations of motion and implicitly determined by the parameters in Table 1, we chose to perform the sensitivity analysis on those parameters. These parameters were found in literature but had to be adjusted to fit our simplified problem. Specifically, the maximum force of each muscle is accurate but the leg is actuated by more than 6 muscles in real life, so just these 6 maximum forces aren't large enough to produce the forces needed for the cycling task. When initially solving the problem, we had to scale the first four max forces (for the gluteals, ilipsoas, hamstrings, and rectus femoris) by a factor of 5 to obtain a feasible solution. The comparison of the scaled values used in the optimization and the values from literature are contained in Table I. From there, we performed a sensitivity analysis on the maximum force parameters to examine how they changed the outcome. To do this, we created a sample of 11 values for each muscles that ranged from -30% of each F^{max} to +30% of F^{max}, increasing by 100N each value. For each optimization iteration, we held 5/6 muscles' F^{max} value constant, and tested a new F^{max} for the other muscle from the sample of 11 values. This resulted in a total of 66 (11 values * 6 muscles) unique optimization iteration, each of which were done over the full range of the crank angles (360 degrees). The table below shows a snapshot of the results for a crank angle of 90 degrees, and for F^{max} changes of -30% and +30%. The outcome reported is the % change in the activation (compared to the original F^{max} value) for each of the muscles. The rows show which muscle F^{max} was changed and the columns are the % change in activation for each of the muscles. It is interesting to note that changing the F^{max} value for one muscle does not necessarily change the activation for that same muscle. If that were the case, then the nonzero values would be along the diagonal (from top left to bottom right). Additionally, changing one muscle's F^{max} only affects one muscle's activation (either that same muscle or another muscle that acts on the same joint).

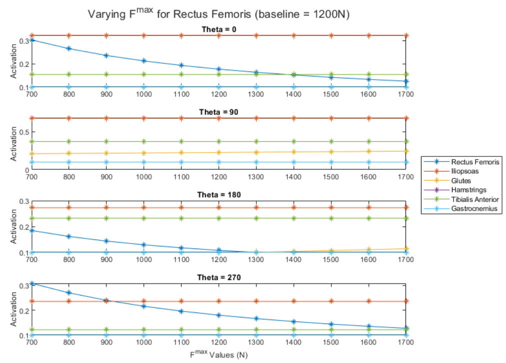


Fig. 4: Muscle Activations as Rectus Femoris maximum force is changed.

TABLE V: Parameters Sensitivity Analysis

% Change in F^{max}	Muscle Max Force Changed	RF	IL	GL	HA	TA	GA
-30% Change	Rectus Femoris	0	0	5.3	0	0	0
	Iliopsoas	0	-42.9	0	0	0	0
	Gluteals	0	0	-42.9	0	0	0
	Hamstrings	0	8.8	0	0	0	0
	Tibialis Anterior	0	0	0	0	-42.9	0
	Gastrocnemius	0	0	0	0	9.8	0
+30% Change	Rectus Femoris	0	0	-5.3	0	0	0
	Iliopsoas	0	23.1	0	0	0	0
	Gluteals	0	0	23.1	0	0	0
	Hamstrings	0	8.8	0	0	0	0
	Tibialis Anterior	0	0	0	0	23.1	0
	Gastrocnemius	0	0	0	0	-9.8	0

VI. Interpretation

We were able to solve our optimization problem for the local and global minima of each point; however, we wanted to evaluate how applicable these results would be for real-world applications. The complexity of the human body necessitated that we made many simplifications to our system which might not reflect reality. The first major limitation was actuating the lower leg with only 6 muscles. Obviously, this is not the reality and this simplification was reflected in the fact that the maximum muscles forces from literature were not able to produce the torques necessary for our experimental biking data. We also only included two biarticular muscles, which introduce more complexity into the model. Another important simplification was making our model two-dimensional instead of three-dimensional. While the cycling problem is well-suited to a 2D simplification, there are some details, such as stability in the 3rd axis, that might be lost in our model.

One simplification that we did address was the relationship between activation and the force output of the muscles. Initially, we modeled force as the product of activation and the maximum force constant (a linear relationship). In reality, there is a complex non-linear relationship between force, length, velocity, and activation of each muscle. The typical force-length relationship is shown in Fig. 5. We can approximate this relationship as an inverted quadratic function centered within the range of angles a muscle can cover. A corresponding force-length constant plot can be seen in Fig. 6. This is for the tibialis anterior muscle. We first found the typical range of angles for that muscle and selected the midpoint as the optimum angle. This was done for each of the angles, resulting in the final force-length constant profile shown in Fig. 7. Adding in this complexity to our model made the final results match experimental EMG data more closely, as can be seen in Fig. 8. This version of the optimization still resulted in exit flags of 1.

VII. Conclusions

In our project, we aimed to apply optimization techniques to find the muscle activations required to produce the forces on the pedal during cycling. While the dynamics of the musculoskeletal system include many complex phenomena, such as

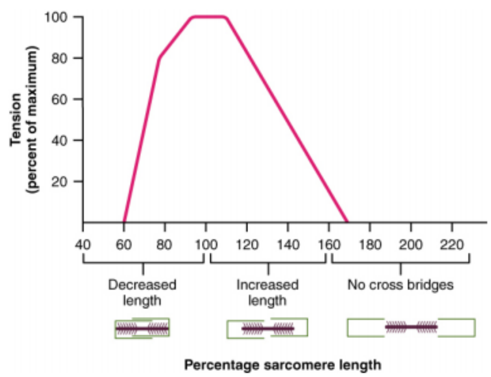


Fig. 5: Typical Force (called Tension here) vs. sarcomere (units within a muscle) length relationship curve.

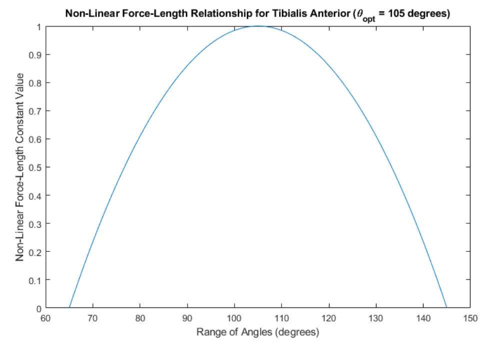


Fig. 6: Non-Linear Force-Length relationship for the Tibialis Anterior muscle.

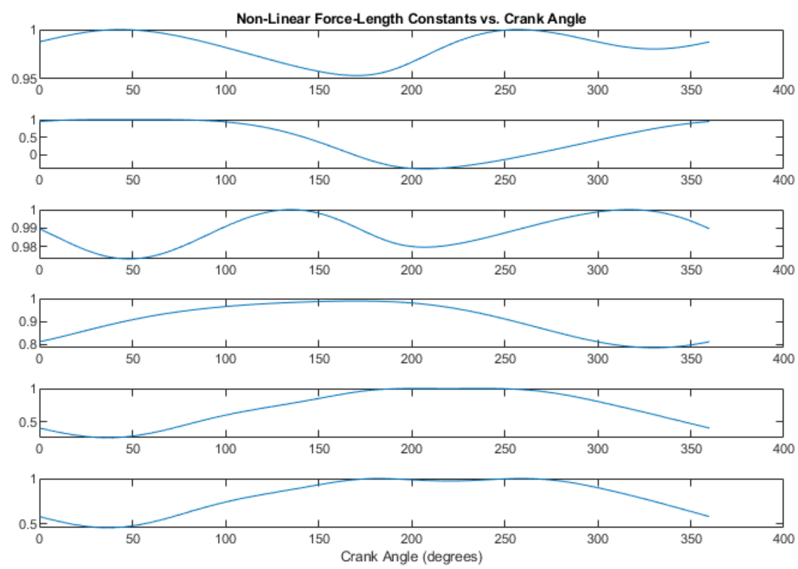


Fig. 7: Non-Linear Force-Length Constants for all muscles.

history-dependent force production, changing moment arms with joint excursion and forces associated with angular velocity and acceleration, we simplified our problem using the following three major assumptions: *i*) the motion of the leg and bicyle all happened in a 2D plane, *ii*) there were 6 muscles that actuated the knee, hip and ankle, and they acted as pure torques around their respective actuated joint and *iii*) the motion of the cyclist was quasistatic, where all terms which were a derivative of time were set to 0. We used the sum of square activations (Eq. 1), which in prior muscle activation optimization studies has been used as a proxy for metabolic cost. These simplifications allowed us to pose our optimization problem as a convex optimization problem, where our objective function was convex and non-linear, and the three equality constraints related to the sum of moments at the hip, knee and ankle were all linear in the decision variable (muscle activation, a). We developed a pipeline to calculate the joint angles and joint torques required in the equality constraints from measured pedal force data reported in [4]. We then used MATLAB's fmincon to solve our optimization problem. We found that if we did not scale the maximum forces of the muscles and used a lower bound of 0 on the activations, then the reported activations had a tendency to make the rectus femoris and the gluteals completely inactive for any crank angle. However, by scaling the muscles that actuated the hip and knee joints (the rectus femoris, gluteals, hamstrings and illiopsoas) by a factor of 5, increased the activation lower bound to 0.1 for all 6 muscles, and use a non-constant force-length relationship for the muscle (Fig 7) we were able to obtain activation profiles as functions of crank angle that qualitatively resembled the EMG profiles in [1] (Fig. 8). Hence, we were able to show that even with major simplifications made to the dynamics of the musculoskeletal system and its interactions with the bicycle, gradient-based optimization techniques can produce solutions to the muscle redundancy problem that resemble real physically measured muscle activations. In the future, incorporation of these additional complexities in our model, such as removing the quasistatic assumption to allow for non-zero time-derivative terms in the equations of motion and modelling

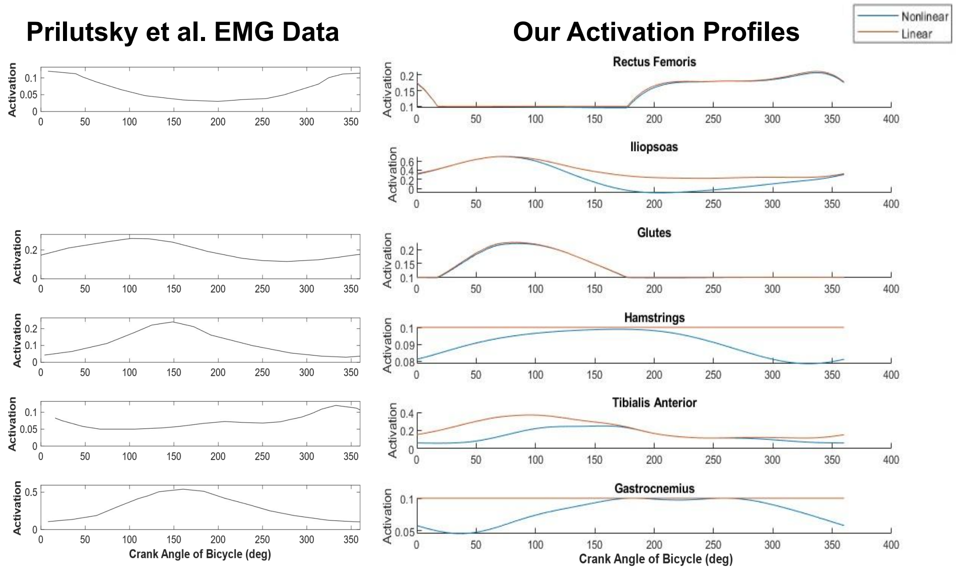


Fig. 8: Final Optimization results with and without the nonlinearity of the force-length relationship applied.

the non-linear force-activation and force-velocity terms, we may potentially be able to get even more realistic matches to the activation profiles reported in literature.

APPENDIX A
"PRE-PROCESSING:" SOLVING FOR JOINT ANGLES USING EXPERIMENTAL DATA AND EQUATIONS OF MOTION

For the equations below, each force or moment is labeled with the joint (hip, knee, ankle with h/k/a) and the axes over which it acts (x/y/z is 1,2,3). Constants such as mass and length are also labeled with the segment (thigh, shank, and foot) It is also identified as an external force if so (ext) or an applied force from the pedal. We are solving for the external forces, specifically the external moments, and will use those for the optimization of the muscle activations. The applied forces from the pedal are scalar values that are defined at each angle of the crank.

$$F^{\mathrm{ext,h1}} + F^{\mathrm{h1}} - F^{\mathrm{k1}} = 0 \quad \text{(sum of forces on the thigh in X)} \tag{2}$$

$$F^{\mathrm{a1}} - \left(F^{\mathrm{ext,k1}} + F^{\mathrm{k1}}\right) = 0 \quad \text{(sum of forces on the shank in X)} \tag{3}$$

$$F^{\mathrm{a1}} + F^{\mathrm{ext,a1}} - \left(F^{\mathrm{pedal1}}\right) = 0 \quad \text{(sum of forces on the foot in X)} \tag{4}$$

$$F^{\mathrm{ext,h2}} + F^{\mathrm{h2}} - \left(F^{\mathrm{k2}} + g\,m^{t}\right) = 0 \quad \text{(sum of forces on the thigh in Y)} \tag{5}$$

$$F^{\mathrm{a2}} + g\,m^{s} - \left(F^{\mathrm{ext,k2}} + F^{\mathrm{k2}}\right) = 0 \quad \text{(sum of forces on the shank in Y)} \tag{6}$$

$$F^{\mathrm{a2}} + F^{\mathrm{ext,a2}} + F^{\mathrm{pedal2}} - \left(g\,m^{f}\right) = 0 \quad \text{(sum of forces on the foot in Y)} \tag{7}$$

$$\begin{aligned} &2\,M^{\mathrm{ext,h3}} + 2\,M^{\mathrm{h3}} + F^{\mathrm{h1}}\,L^{t}\,\sin\left(\theta^{h}\right) + F^{\mathrm{k1}}\,L^{t}\,\sin\left(\theta^{h}\right) \\ &- \left(2\,M^{\mathrm{k3}} + F^{\mathrm{h2}}\,L^{t}\,\cos\left(\theta^{h}\right) + F^{\mathrm{k2}}\,L^{t}\,\cos\left(\theta^{h}\right)\right) = 0 \quad \text{(sum of moments on the thigh)} \end{aligned} \tag{8}$$

$$\begin{aligned} &2\,M^{\mathrm{a3}} + F^{\mathrm{a2}}\,L^{s}\,\cos\left(\theta^{h}+\theta^{k}\right) + F^{\mathrm{k2}}\,L^{s}\,\cos\left(\theta^{h}+\theta^{k}\right) \\ &- \left(2\,M^{\mathrm{ext,k3}} + 2\,M^{\mathrm{k3}} + F^{\mathrm{a1}}\,L^{s}\,\sin\left(\theta^{h}+\theta^{k}\right) + F^{\mathrm{k1}}\,L^{s}\,\sin\left(\theta^{h}+\theta^{k}\right)\right) = 0 \quad \text{(sum of moments on the shank)} \end{aligned} \tag{9}$$

$$\begin{aligned} &2\,M^{\mathrm{a3}} + 2\,M^{\mathrm{ext,a3}} + F^{\mathrm{a1}}\,L^{f}\,\sin\left(\theta^{a}+\theta^{h}+\theta^{k}\right) + F^{\mathrm{pedal1}}\,L^{f}\,\sin\left(\theta^{a}+\theta^{h}+\theta^{k}\right) \\ &- \left(2\,M^{\mathrm{t3}} + F^{\mathrm{a2}}\,L^{f}\,\cos\left(\theta^{a}+\theta^{h}+\theta^{k}\right) + F^{\mathrm{pedal2}}\,L^{f}\,\cos\left(\theta^{a}+\theta^{h}+\theta^{k}\right)\right) = 0 \quad \text{(sum of moments on the foot)} \end{aligned} \tag{10}$$

APPENDIX B
PARAMETERS

Symbol	Description	Value	Units
l^G	Gluteals Length	0.271	m
l^I	Iliopsoas Length	0.248	m
l^H	Hamstrings Length	0.383	m
l^R	Rectus Femoris Length	0.474	m
l^C	Gastrocnemius Length	0.487	m
l^T	Tibialis Anterior Length	0.381	m
L^t	Thigh Length	0.46	m
L^s	Shank Length	0.44	m
L^f	Foot Length	0.07	m
m^t	Thigh Mass	7.5	kg
m^s	Shank Mass	3.49	kg
m^f	Foot Mass	1.09	kg
x^c	Crank Center in X	-0.1	m
y^c	Crank Center in Y	-0.7	m
R	Crank Radius	0.17	m
$F^{pedal,x}$	Applied Force from the Pedal in X	see AC	N
$F^{pedal,y}$	Applied Force from the Pedal in Y	see AC	N

APPENDIX C
EXPERIMENTAL DATA

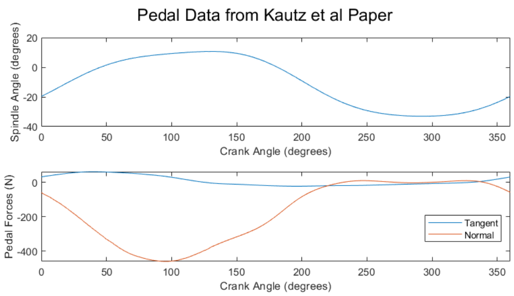


Fig. 9: Experimental Data from Kautz et al [4]. Figure displays the Spindle angle, F^{Pedal} in the tangent and normal directions.

APPENDIX D
MATLAB OPTIMIZATION CODE

This is the code that executes the optimization to produce the activation profiles, using the torques computed in Appendix A below.

```
%% Parameters
frmax = 1200*5; % Rectus femoris
fimax = 5*1500; % Iliopsoas
fgmax = 3000*5; % gluteals
fhmax = 3000*5; % Hamstring Fhmax rh
ftmax = 2500; % Tibialis Anterior
fgamax = 3000; % Gastrocneius
rh = 0.081; %radius of the hip joint
rk = 0.035; %radius of knee joint
rankle = 0.052; %radius of the ankle joint

%% Setting up the problem
% x1 = activation Rectus femoris
% x2 = activation Iliopsoas
% x3 = activation gluteals
% x4 = activation hamstring
% x5 = activation Tibialis Anterior
% x6 = activation gastrocneius
load('ForceTorque.mat')
objective = @(x) x(1)^2 + x(2)^2 + x(3)^2 + x(4)^2 + x(5)^2 + x(6)^2;
% initial guess
x0 = [0.2 0.2 0.2 0.2 0.2 0.2];
% variable bounds
lb = [0.1 0.1 0.1 0.1 0.1 0.1];
ub = [1 1 1 1 1 1];
% show initial objective
disp(['Initial Objective: ' num2str(objective(x0))])

%% linear constraints
% Mh3 = - Fhmax*x4*rh - Fgmax*x3*rh + Fimax*x2*rh + Frmax*x1*rh
% Mk3 = -Fhmax*x4*rk + Frmax*x1*rk
% Ma3 = -Fgamax*x6*ra + Ftmax*x5*ra
A = [];
b = [];
Aeq = [frmax*rh fimax*rh -fgmax*rh -fhmax*rh 0 0; frmax*rk 0 0 -fhmax*rk 0 0; 0 0 0 0 ftmax*rankle -fgmax*
    rankle];
% nonlinear constraints
nonlincon = @nlcon;
%options
options = optimoptions(@fmincon,'Algorithm','interior-point');
%moments
mh3 = totalTable.("M_h3");
mk3 = totalTable.("M_k3");
ma3 = totalTable.("M_a3");

%% Managing the output table
x1 = [];
x2 = [];
x3 = [];
x4 = [];
x5 = [];
x6 = [];
totalActivation = [];
```

```
eflag = [];

%% Running the optimizer

for i = 1:1:361
    beq = [mh3(i); mk3(i); ma3(i)];

    % optimize with fmincon
    %[X,FVAL,EXITFLAG,OUTPUT,LAMBDA,GRAD,HESSIAN]
    % = fmincon(FUN,X0,A,B,Aeq,Beq,LB,UB,NONLCON,OPTIONS)
    [X, FVAL, EXITFLAG, OUTPUT, LAMBDA, GRAD, HESSIAN] = fmincon(objective,x0,A,b,Aeq,beq,lb,ub,nonlincon,
        options);
    x1(end+1) = X(1);
    x2(end+1) = X(2);
    x3(end+1) = X(3);
    x4(end+1) = X(4);
    x5(end+1) = X(5);
    x6(end+1) = X(6);
    totalActivation(end+1) = FVAL;
    eflag(end+1) = EXITFLAG;
end

%% Displaying the results
% show final objective
X = [x1; x2; x3; x4; x5; x6];
figure();
tiledlayout(6,1)

for k = 1:6
    ax=nexttile;
    plot(ax,0:360,X(k,:))

end

figure();
tiledlayout(3,1)
ax=nexttile
plot(ax,0:360,totalTable.M_h3)
ax=nexttile
plot(ax,0:360,totalTable.M_k3)
ax=nexttile
plot(ax,0:360,totalTable.M_a3)


figure();
tiledlayout(2,1)
ax=nexttile
plot(ax,0:360,totalTable.f_x)
ax=nexttile
plot(ax,0:360,totalTable.f_y)

figure();
plot(0:360,totalActivation)

function [c,ceq] = nlcon(x)
  c = [];
  ceq = [];
```

APPENDIX E
PRE-PROCESSING CODE

This is the code used to generate the equations of motion for each of the leg segments, and calculates the torques at the hip, knee and ankle used in Appendix F above.

```
clc
clearvars
syms t thetaT

lin_accel_O = sym('lin_accel_O',[3 1]); %acceleration of point O %change all these to symfunmatrix of
    variable t in order to perform time differentiation
alpha_OP = sym('alpha_OP',[3 1]); %angular acceleration of link OP
r_O_P = sym('r_O_P',[3 1]); %vector from O to point P on link OP
omega_OP = sym('omega_OP',[3 1]); %vector from O to point P on link OP
theta_OP = sym('theta_OP'); %Angle of link OP with frame B centered at point O, with respect to frame A.
l_OP = sym('l_OP'); %length of the link from O to P.
```

```
lin_accel = lin_accel_O + cross(alpha_OP,r_O_P) + cross(omega_OP,cross(omega_OP,r_O_P));

HomogenousTrans = [cos(thetaT),-sin(thetaT),0,l_OP;
                   sin(thetaT),cos(thetaT),0,0;
                   0, 0, 1, 0;
                   0, 0, 0, 1]; %homogenous transform transform vector from frame B to frame A when frame B
                        rotates relative to Frame A with angle theta_OP

RotTrans =  [cos(thetaT),-sin(thetaT);
                   sin(thetaT),cos(thetaT)]; %rotation transform transform vector from frame B to frame A
                       when frame B rotates relative to Frame A with angle theta_OP

frame_abbrev = {"h","k","a","t","p","c"}; %h=hip, k = knee, a = ankle, t = toe, p = pedal, c = center
link_length = {"h_k","k_a","a_t","t_p","p_c"}; % h_k = length from hip to knee, k_a = knee to ankle, a_t =
    ankle to toe, t_p = toe to pedal, p_c = pedal to crank center
numLinks = length(link_length);

thetaVec = sym(zeros(numLinks,3));
omegaVec = sym(zeros(numLinks,3));
alphaVec = sym(zeros(numLinks,3));
linAccelVec = sym(zeros(numLinks,3)); %need to initialize to zero because the first frame (in this case the
     hip) is assumed to not be moving.
linAccelCOMVec = sym(zeros(numLinks,3));
H_Mat = sym(repmat(eye(4),1,1,numLinks)); %homogenous transform to transform from frame{k} to the base
    frame at the origin
Rot_Mat = sym(repmat(eye(2),1,1,numLinks)); %rotation transform to transform from frame{k} to the base
    frame at the origin
rVec = sym(zeros(numLinks,3)); %vector from the base of the link to the end of the link
rCOMVec = sym(zeros(numLinks,3)); %vector from the base of the link to the COM of the link
I_mat = sym(repmat(zeros(3),1,1,numLinks)); %for inertia matrix

gVec = sym([0;-str2sym("g");0]);
sumFvec = sym(zeros(numLinks,3));
sumMoments = sym(zeros(numLinks,3));

symVec=sym(zeros(1,numLinks*2*3));
thetaSymVec_t=sym(zeros(1,numLinks)); %function of time
thetaSymVec=sym(zeros(1,numLinks)); %not a function of time

for k=1:numLinks
    thetaVec(k,3) = str2sym("theta"+"_"+frame_abbrev{k}+"(t)"); %this gives the angular position of frame{k
        } relative to the frame {k-1}
    thetaSymVec_t(k) = thetaVec(k,3);
    thetaSymVec(k) = str2sym("theta"+"_"+frame_abbrev{k});

    omegaVec(k,:) = diff(thetaVec(k,:),t,1);
    omegaVec(k,:) = sum(omegaVec(max(k-1,1):k,:),1); %omega for the frame is the sum of all the omegas of
        frame {k} and all previous frames
    alphaVec(k,:) = diff(omegaVec(k,:),t);   %simplyy differentiate because there are no additional
        acceleration terms resulting from movement of coordinate frame as the motion is assumed to happen
        in the plane
    I_mat(:,:,k) = sym("I_"+link_length{k},[3 3]);

    H_T = subs(HomogenousTrans,{thetaT, l_OP},[thetaVec(k,3),sym("L_"+link_length{k})]);
    H_Mat(:,:,k) = H_Mat(:,:,max(k-1,1))*H_T;
    rot_T = subs(RotTrans,{thetaT},[thetaVec(k,3)]);
    Rot_Mat(:,:,k) = Rot_Mat(:,:,max(k-1,1))*rot_T;


    rVec_H = Rot_Mat(:,:,k)*[H_T(1,4);0];
    rVec(k,:)=[rVec_H(:,1).',0];
    linAccelVec(k,:) = subs(lin_accel,[lin_accel_O,alpha_OP,omega_OP,r_O_P],[linAccelVec(max(k-1,1),:).',
        alphaVec(k,:).',omegaVec(k,:).',rVec(k,:).']);

    rVecCOM_H = Rot_Mat(:,:,k)*[H_T(1,4)/2;0];
    rCOMVec(k,:)=[rVecCOM_H(:,1).',0];
    linAccelCOMVec(k,:) = subs(lin_accel,[lin_accel_O,alpha_OP,omega_OP,r_O_P],[linAccelVec(max(k-1,1),:)
        .',alphaVec(k,:).',omegaVec(k,:).',rCOMVec(k,:).']);



    %% Compute moment and sum of forces given that we've computed the linear and angular acceleration
    F_reaction_joint1 = sym("F_"+frame_abbrev{k},[3,1]) ;
    M_reaction_joint1 = sym("M_"+frame_abbrev{k},[3,1]) ;
    F_reaction_joint2 = -sym("F_"+frame_abbrev{k+1},[3,1]);
    M_reaction_joint2 = -sym("M_"+frame_abbrev{k+1},[3,1]) ;
```

```matlab
    massSym =  sym("m_"+link_length{k});
    F_ext = sym("F_ext_"+frame_abbrev{k},[3,1]);
    M_ext = sym("M_ext_"+frame_abbrev{k},[3,1]);

    Fsum = F_reaction_joint1 + F_reaction_joint2 + massSym*gVec + F_ext ==linAccelVec(k,:).';
    Fsum(3)=0;%No forces in z direction because the motion is planar
    sumFvec(k,:) = Fsum.';

    Msum = I_mat(:,:,k)*alphaVec(k,:).' + cross(omegaVec(k,:).',I_mat(:,:,k)*omegaVec(k,:).') == cross(-rCOMVec(k,:).',F_reaction_joint1) + cross(rVec(k,:).'-rCOMVec(k,:).',F_reaction_joint2) + M_ext + M_reaction_joint1 + M_reaction_joint2;
    Msum(1)=0;%only moments around the z axis for planar motion
    Msum(2)=0;%only moments around the z axis for planar motion
    sumMoments(k,:) = Msum.';

    symVec((k-1)*6+1:(k-1)*6+3)=F_reaction_joint1.';
    symVec((k-1)*6+4:(k-1)*6+6)=M_reaction_joint1.';




end
totaleqs=[reshape(sumFvec,[15,1]);reshape(sumMoments,[15,1])];
%% get static equilibria
staticEqs = subs(totaleqs,thetaSymVec_t,thetaSymVec);
textStr="";
for jj=1:length(staticEqs)

    textStr = textStr+newline+"& "+latex(simplify(staticEqs(jj,1))) + " \\";

end


%writelines(textStr,"StaticEqEquations.txt")

%%print
%latex(simplify(staticEqs(30,1)))

%% equations for muscle forces and torques
createVarNameFcn = @(prefix,suffix) sprintf('%s_%s',prefix,suffix);

muscleSuffix ={'RF' 'IP' 'G' 'H' 'TA' 'GA'};
%muscleRadii = str2sym(cellfun(@(r)sym(createVarNameFcn('r',r),[6,1]),muscleSuffix,'UniformOutput',false));
muscleRadii = sym("r_",[3,6]);
muscleForce =  str2sym(cellfun(@(r)createVarNameFcn('F',r),muscleSuffix,'UniformOutput',false));
muscleActivation = str2sym(cellfun(@(r)createVarNameFcn('a',r),muscleSuffix,'UniformOutput',false));
MuscleTorques = sym({'M_h','M_k','M_t'});

maxMuscleForces = [1200*5, 1500*5, 3000*5, 3000*5, 2500, 3000];
MuscleTorqueEquations = MuscleTorques.'-[muscleRadii(1,1) muscleRadii(1,2) -muscleRadii(1,3) -muscleRadii(1,4) 0 0;muscleRadii(2,1) 0 -muscleRadii(2,3) 0 0 0;0 0 0 0 muscleRadii(3,5) -muscleRadii(3,6) ]*(muscleActivation.*muscleForce).';
MuscleTorqueEquations_P = subs(MuscleTorqueEquations,muscleRadii,repmat([0.081; 0.035;0.052],[1,6]));  %use same radii for now
MuscleTorqueEquations_P = subs(MuscleTorqueEquations_P,muscleForce, maxMuscleForces);

if true  %set to true if you want to regenerate the constraint functions for muscle optimization
    currdir = [pwd filesep]; % You might need to use currdir = pwd
    filename_cons = [currdir,'constraint_MuscleOptimization.m'];
    matlabFunction([],MuscleTorqueEquations_P,'file',filename_cons,'vars',{[MuscleTorques,muscleActivation]},'outputs',{'c','ceq'});
end

%% create table to input parameters
letterDictStruct.FirstLetter=containers.Map(["F","L","M","g","m","theta"],["Force","Length","Moment","gravity","mass","angle"]);
letterDictStruct.SecondLetter=containers.Map(["a","c","h","k","p","t","ext","1","2","3"],["ankle","crank center","hip","knee","pedal center","toe","external","x axis","y axis", "z axis"]);
svar_arr = symvar(staticEqs);
if false  %don't overwrite table
    createVariableTable("StaticVarTable.csv",svar_arr, letterDictStruct);
end

%% solve equations for the reaction forces and torques
parmTable = readtable("StaticVarTable.csv");

pTable = parmTable((parmTable.SolveSymbolically=="Y"),:);
```

```

StaticEqs_solved = solve(staticEqs,str2sym(pTable.VarName));

pTable2 = parmTable((parmTable.SolveSymbolically=="N" & ~isnan(parmTable.Value)),:); %for parameters
subEqs = subs(StaticEqs_solved,str2sym(pTable2.VarName).',pTable2.Value.');

%create matlab function from symbolic equations
paramOutputs = fieldnames(subEqs);  %names of the param outputs
totEqs = sym(zeros(length(paramOutputs),1));

for j= 1:length(paramOutputs)

    totEqs(j) = subEqs.(paramOutputs{j});


end

paramNames = parmTable.VarName(parmTable.SolveSymbolically=="P")
JointForceTorque_Fcn = matlabFunction(totEqs,'Vars',{'F_t1','F_t2','theta_h','theta_k','theta_a'})

if false %set to true if you want to regenerate a separate file for calculating joint forces and torques
    JointForceTorque_Fcn = matlabFunction(totEqs,'Vars',{'F_t1','F_t2','theta_h','theta_k','theta_a'},'File','JointForceTorque_FileFcn')
end

%% read table of parameters and solve equations using the simplified structure subEqs


%convert matrix to table
KautzDataStruct = load("output.mat");
KautzData = KautzDataStruct.combined;
KautzData(:,3) = -KautzData(:,1) -KautzData(:,2)+KautzData(:,3);  %convert the ankle angles in the convention defined above
KautzData = array2table(KautzData,'VariableNames',{'theta_h','theta_k','theta_a','f_x','f_y','x','y'});
KautzData.f_x_negated=-KautzData.f_x; %need to negate because the negative sign is assumed in the EOMs. Recall that these forces that are read in were converted to be in the global x and y frames
KautzData.f_y_negated=-KautzData.f_y; %need to negate because the negative sign is assumed in the EOMs.
parmTable = readtable("StaticVarTable.csv");

%get variables that are non-zero
pTable = parmTable((parmTable.SolveSymbolically=="P"),:); %for parameters

resultStruct=struct();
for vv=1:length(paramOutputs)

    resultStruct.(paramOutputs{vv}) = zeros(size(KautzData,1),1);

end

for vv=1:length(muscleSuffix)

    resultStruct.("Activation_"+muscleSuffix{vv}) = zeros(size(KautzData,1),1);

end


exitflags = zeros(size(KautzData,1),1)


for j=1:size(KautzData,1)

    %substitue F_t1 and F_t2 to the f_x and f_y_respectively.  Substitute
    %theta_h, theta_k,theta_a
    resSolution = JointForceTorque_Fcn(KautzData.f_x_negated(j),KautzData.f_y_negated(j),KautzData.theta_h(j),KautzData.theta_k(j),KautzData.theta_a(j));

    for vv=1:length(paramOutputs)

        resultStruct.(paramOutputs{vv})(j) = resSolution(vv);

    end

    %%perform optimization to get the activations
    MuscleTorques = [resSolution(paramOutputs=="M_h3"),resSolution(paramOutputs=="M_k3"), resSolution(paramOutputs=="M_a3")] ;
    objFunc = @(x) sum(x.^2) + sum(x);
    x0=zeros(1,6);
```

```
    options = optimoptions('fmincon','Algorithm','sqp');
    lb=zeros(1,6)+0.1;
    ub=ones(1,6);
    [xval_activation,fval,exitflags(j),output,lambda,grad]=fmincon(objFunc,x0,[],[],[],[],lb,ub,@(x)
        constraint_MuscleOptimization([MuscleTorques,x]),options);
    for vv=1:length(muscleSuffix)

        resultStruct.("Activation_"+muscleSuffix{vv})(j) = xval_activation(vv);

    end

    %check exit flag


end

disp("Finished")
resultTable = struct2table(resultStruct);

totalTable = [KautzData,resultTable];
writetable(totalTable,"results_w_Activation.xlsx")



%% plots

thighL = parmTable.Value(parmTable.VarName=="L_h_k");
shankL = parmTable.Value(parmTable.VarName=="L_k_a");
footL =parmTable.Value(parmTable.VarName=="L_a_t");

Ho_func = matlabFunction(HomogenousTrans,'vars',[thetaT,l_OP]);



close all
numMuscles = 6;
fig=figure();
set(gcf, 'Units', 'normalized');
set(gcf, 'Position', [0 0.1 0.8 0.8]);
set(gcf,'color','w');
set(0, 'DefaultAxesFontName', 'Arial')
tiledlayout(numMuscles,2);

leg_ax = nexttile([numMuscles,1]);
set(leg_ax,'XColor', 'none','YColor','none');

axColl = {};
animatedLineCol = {}

MuscleAVid = VideoWriter('MuscleActivation'); %open video file
MuscleAVid.FrameRate = 30;
open(MuscleAVid)


muscleNames = {'Rectus Femoris (RF)' 'Iliopsoas (IP)' 'Gluteals (G)' 'Hamstrings (H)' 'Tibialis Anterior (
    TA)' 'Gastrocnemius (GA)'};

for vv = 1:numMuscles
    axCol1{vv} = nexttile;
    set(axCol1{vv},'FontSize',12)
    animatedLineCol{vv} = animatedline(axCol1{vv});
    xlim(axCol1{vv},[0,360]);
    ylim(axCol1{vv},[0,max(totalTable.(sprintf("Activation_%s",muscleSuffix{vv})))*maxMuscleForces(vv)]);
    %ylim(axCol1{vv},[0,inf]);
    title(muscleNames{vv});
    ylabel('Force (N)','FontSize',14,'FontWeight','bold');


end


aLine_thigh = animatedline(leg_ax,'LineWidth',3,'Marker','o','MarkerFaceColor','r','MarkerSize',9);
aLine_shank = animatedline(leg_ax,'LineWidth',3,'Marker','o','MarkerFaceColor','b','MarkerSize',9);
aLine_foot = animatedline(leg_ax,'LineWidth',3,'Marker','o','MarkerFaceColor','g','MarkerSize',9);
xlim(leg_ax,[-.6,.6]);
ylim(leg_ax,[-1,.2]);

```

```
xlabel(axCol1{6},'Crank Angle of Bicycle (deg)', 'FontSize',14,'FontWeight','bold');


angles=[0:360];


for j=1:length(angles)
    for v=1:length(muscleSuffix)

        activationV = totalTable.(sprintf("Activation_%s",muscleSuffix{v}));
        %plot(axCol1{v},angles(1:j),activationV(1:j));
        addpoints(animatedLineCol{v},angles(j),activationV(j).*maxMuscleForces(v)   );
    end


    %plot cyclist
    posH    = [0;0];
    posK = Ho_func(totalTable.theta_h(j),0)*Ho_func(totalTable.theta_k(j),thighL)*[0;0;0;1];
    posA = Ho_func(totalTable.theta_h(j),0)*Ho_func(totalTable.theta_k(j),thighL)*Ho_func(totalTable.theta_a(j),shankL)*[0;0;0;1];
    posT = Ho_func(totalTable.theta_h(j),0)*Ho_func(totalTable.theta_k(j),thighL)*Ho_func(totalTable.theta_a(j),shankL)*Ho_func(0,footL)*[0;0;0;1];

    clearpoints(aLine_thigh);
    clearpoints(aLine_shank);
    clearpoints(aLine_foot);

    addpoints(aLine_thigh,[posH(1);posK(1)],[posH(2);posK(2)]);
    addpoints(aLine_shank,[posK(1);posA(1)],[posK(2);posA(2)]);
    addpoints(aLine_foot,[posA(1);posT(1)],[posA(2);posT(2)]);

    drawnow limitrate;
    pause(0.0027);
    frame = getframe(gcf); %get frame
    writeVideo(MuscleAVid, frame);

    %clf(fig);

end

close(MuscleAVid);



%% Plot torque

numMuscles = 6;
fig=figure();
set(gcf, 'Units', 'normalized');
set(gcf, 'Position', [0 0.1 0.8 0.8]);
set(gcf,'color','w');
set(0, 'DefaultAxesFontName', 'Arial')
tiledlayout(numMuscles,2);

ax=nexttile(1);
plot(ax,angles,totalTable.M_h3);
ylabel('(Nm)','FontSize',12,'FontWeight','bold');
title('Hip Torque','FontSize',12,'FontWeight','bold');

ax=nexttile(3);
plot(ax,angles,totalTable.M_k3);
ylabel('(Nm)','FontSize',12,'FontWeight','bold');
title('Knee Torque','FontSize',12,'FontWeight','bold');

ax=nexttile(5);
plot(ax,angles,totalTable.M_a3);
ylabel('(Nm)','FontSize',12,'FontWeight','bold');
title('Ankle Torque','FontSize',12,'FontWeight','bold');

ax=nexttile(7);
plot(ax,angles,totalTable.f_x);
ylabel('(N)','FontSize',12,'FontWeight','bold');
title('Foot x-force','FontSize',12,'FontWeight','bold');

ax=nexttile(9);
plot(ax,angles,totalTable.f_y);
ylabel('(N)','FontSize',12,'FontWeight','bold');
```

```
title('Foot y-force (N)','FontSize',12,'FontWeight','bold');
xlabel('Crank Angle of Bicycle (deg)', 'FontSize',12,'FontWeight','bold');


axColl = {};




muscleNames = {'Rectus Femoris (RF)' 'Iliopsoas (IP)' 'Gluteals (G)' 'Hamstrings (H)' 'Tibialis Anterior (
    TA)' 'Gastrocnemius (GA)'};

for vv = 1:numMuscles
    axCol1{vv} = nexttile((vv)*(2));



    activationV = totalTable.(sprintf("Activation_%s",muscleSuffix{vv}));
        %plot(axCol1{v},angles(1:j),activationV(1:j));
    plot(axCol1{vv},angles,activationV,'k-');
    ylabel(axCol1{vv} ,'Activation','FontSize',14,'FontWeight','bold');
    title(muscleNames{vv});

    set(axCol1{vv},'FontSize',12)
    xlim(axCol1{vv},[0,360]);
    ylim(axCol1{vv},[0,1]);


end
xlabel(axCol1{6},'Crank Angle of Bicycle (deg)', 'FontSize',14,'FontWeight','bold');

%% plot the activations side by side

PrilutskyData = readtable("PrilutskyData_AdjustedActivation.csv");

numMuscles = 6;
fig=figure();
set(gcf, 'Units', 'normalized');
set(gcf, 'Position', [0 0.1 0.8 0.8]);
set(gcf,'color','w');
set(0, 'DefaultAxesFontName', 'Arial')
tiledlayout(numMuscles,2);
muscleNames = {'Rectus Femoris (RF)' 'Iliopsoas (IP)' 'Gluteals (G)' 'Hamstrings (H)' 'Tibialis Anterior (
    TA)' 'Gastrocnemius (GA)'};

PrilutskyPos =[1,5,7,9,11];
PrilutskyMuscles={"RF","GLM","HA","TA","GA"};
numMuscles_Pri = 5;

axColl = {};

for vv = 1:numMuscles_Pri
    axCol1{vv} = nexttile(PrilutskyPos(vv));
    pT = PrilutskyData(strcmp(PrilutskyData.Muscle,PrilutskyMuscles{vv}),:);
    activationV_P =interp1(pT.angle,pT.AdjustedActivation,angles,'linear') ;
        %plot(axCol1{v},angles(1:j),activationV(1:j));
    plot(axCol1{vv},angles,activationV_P/100,'k-');
    ylabel(axCol1{vv} ,'Activation','FontSize',14,'FontWeight','bold');
    title(PrilutskyMuscles{vv});

    set(axCol1{vv},'FontSize',12)
    xlim(axCol1{vv},[0,360]);
    ylim(axCol1{vv},[0,max(activationV_P/100)*1.1]);


end
xlabel(axCol1{5},'Crank Angle of Bicycle (deg)', 'FontSize',14,'FontWeight','bold');




axColl = {};

for vv = 1:numMuscles
    axCol1{vv} = nexttile((vv)*(2));


```

```

    activationV = totalTable.(sprintf("Activation_%s",muscleSuffix{vv}));
        %plot(axCol1{v},angles(1:j),activationV(1:j));
    plot(axCol1{vv},angles,activationV,'k-');
    ylabel(axCol1{vv} ,'Activation','FontSize',14,'FontWeight','bold');
    title(muscleNames{vv});

    set(axCol1{vv},'FontSize',12)
    xlim(axCol1{vv},[0,360]);
    ylim(axCol1{vv},[0,max(activationV)]*1.1);


end
xlabel(axCol1{6},'Crank Angle of Bicycle (deg)', 'FontSize',14,'FontWeight','bold');



function[outTable] = createVariableTable(fnamestr,sym_vars, letterDictStruct)

    numVars = length(sym_vars);
    out_s.Var=sym(zeros(numVars,1));
    out_s.VarName=cell([numVars,1]);
    out_s.definition=cell([numVars,1]);
    out_s.Value=NaN([numVars,1]);
    for j=1:length(sym_vars)
        s_var =(sym_vars(j));
        out_s.Var(j)=s_var;
        out_s.VarName{j}=string(s_var);
        s_var=string(s_var); %convert to string
        spVar = split(s_var,"_");
        DefS="";
        for vv=1:length(spVar)
            if vv ==1
                DefS = DefS+letterDictStruct.FirstLetter(spVar(vv))+":";
            else
                %get number which represents component at end
                returnv=regexp(spVar(vv),".*(\d)","tokens");
                if ~isempty(returnv)
                    St1=strsplit(spVar(vv),returnv{1});
                    DefS = DefS + letterDictStruct.SecondLetter(St1{1})+"_";
                    DefS = DefS + letterDictStruct.SecondLetter(returnv{1});
                else
                    DefS = DefS+ letterDictStruct.SecondLetter(spVar(vv)) +"_";
                end



            end

        end

        out_s.definition{j} = DefS;

    end

    outTable = struct2table(out_s);
    writetable(outTable,fnamestr,'Delimiter',',');

end
```

Appendix F
MATLAB Optimization code

All authors contributed to this work. **Risheekesh Kesavan** and **Pratiksha Ganesan** coded, debugged and analyzed the optimization code and performed the analysis of the Lagrange multipliers. **Ravesh Sukhnandan** created the dynamics pipeline for determining the forces and torques at the hip, knee and ankle from the pedal force data [4] and produced visualizations. **Mikayla Schneider** created the pipeline for determining the angles of the hip and knee, and performed the sensitivity analysis and the implementation of the non-linear force-length relationships.

References

[1] B. Prilutsky and R. Gregor, "Analysis of muscle coordination strategies in cycling," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 3, pp. 362–370, Sep. 2000, conference Name: IEEE Transactions on Rehabilitation Engineering.

[2] T. Lencioni, I. Carpinella, M. Rabuffetti, A. Marzegan, and M. Ferrarin, "Human kinematic, kinetic and EMG data during different walking and stair ascending and descending tasks," *Scientific Data*, vol. 6, no. 1, p. 309, Dec. 2019, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41597-019-0323-z

[3] F. C. Anderson and M. G. Pandy, "Static and dynamic optimization solutions for gait are practically equivalent," *Journal of Biomechanics*, vol. 34, no. 2, pp. 153–161, Feb. 2001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002192900000155X

[4] S. A. Kautz, M. E. Feltner, E. F. Coyle, and A. M. Baylor, "The pedaling technique of elite endurance cyclists: Changes with increasing workload at constant cadence," *Journal of Applied Biomechanics*, vol. 7, no. 1, pp. 29–53, Feb. 1991, publisher: Human Kinetics, Inc. Section: Journal of Applied Biomechanics. [Online]. Available: https://journals.humankinetics.com/view/journals/jab/7/1/article-p29.xml

[5] J. M. Winters and C. , Patrick E, *Biomechanics and neural control of posture and movement*. New York: Springer, 2000, oCLC: 1256378326. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-2104-3