EMG-tracking for the Rapid Muscle Redundancy solver

Implementation & evaluation

by

F.J. van Melis

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended on Tuesday August 26th, 2025 at 11:30.

Student number:	4590384	
Project duration:	June, 2024 – August, 2025	
Thesis committee:	A. Seth,	TU Delft, main supervisor;
	I. Belli,	TU Delft, daily supervisor;
	L. Peternel,	TU Delft

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

Here it is. A few pages of text and figures to conclude my 9 year study journey. But all joking aside, I am quite proud of what it has become! While the length of my study career and the duration of this graduation project may be longer than conventional, I think it has made up for it by all the skills I've learned and knowledge I've gained during this time. Reflecting on this thesis in particular, I feel like I really know how musculoskeletal simulation works now. The heavy emphasis on programming has also made me significantly more proficient at Python and object-oriented programming. Other notable skills that have improved are managing long individual projects, scientific attitude and writing.

Of course, I have received plenty of support during this challenging project. First, I want to thank my supervisors, Ajay and Italo, for their patience during the whole project, which took longer to complete than I first anticipated. Ajay, you are a tremendous source of information and inspiration. Italo, you've been very supportive throughout the project and you have helped me get back on track several times.
I would also like to thank the CBL and PT group, from which I received valuable feedback during all phases of the project. In particular I want to thank Sabrina. Besides providing me with support concerning the knee simulations, you also gave me very extensive feedback on my last drafts of this thesis. Without you, the quality of the thesis would not have been what it is now.
Shout-out to all my partners-in-crime (fellow graduating students) as well: Guiomar, Albert, Ted, Tom, Evert and Simon.
Guiomar, thanks for taking the time to talk about musculoskeletal graduation projects, which convinced me to go for a thesis in this field. Best of luck with your PhD!
Albert, thank you for being my best friend for over 14 years. I know I can always share any of my struggles with you. I am sure you'll graduate in no-time as well!
Ted, Tom, thanks for listening to any problems or dilemmas I was facing. Your familiarity with the project meant that I could go into as much depth as with any of my supervisors and receive valuable advice.
Thank you Evert and Simon for sitting with me throughout the summer months, and thanks Ruben for supplying caffeinated beverages free of charge.
Besides my fellow graduating students, I am also grateful to my housemates, my brother and all my friends for giving me energy in life.
Finally, I would like to thank my parents for their support during the whole of my life and study career. I've never felt the pressure to do better or faster. This and their trust in my decisions and capabilities is exactly what has made me excel.

F.J. van Melis
Delft, August 2025

Summary

Many are affected by musculoskeletal disorders, for example rotator cuff tears or knee osteoarthritis. Disorders affect muscle coordination, which can negatively impact treatment of these disorders. In order to offer personalized treatment options, muscle recruitment needs to be predicted accurately in these patients. Musculoskeletal modeling can be used to estimate muscle coordination non-invasively, but established methods like Static Optimization (SO) underestimate co-contraction of muscles. Predicting co-contraction is important, because it is associated with musculoskeletal disorders through joint stability, joint stiffness and compensatory muscle control strategies, e.g. to minimize pain. State-of-the-art methods used to predict co-contraction include assistance from electromyography (EMG) measurements or joint stability constraints. The existing Rapid Muscle Redundancy (RMR) solver utilizes a glenohumeral stability constraint to predict co-contraction in the deep rotator cuff muscles. The addition of EMG-assistance to this solver, might improve it's capability to predict co-contraction in superficial muscles as well.

Therefore, the main goal of this thesis is to equip the existing RMR solver with an EMG-assisted cost function. Along with this extension, the RMR solver is revised to be class-based in order to improve modularity, scalability and user interaction. The new implementation is verified and is able to utilize EMG-tracking alongside the glenohumeral joint stability constraint.

Because EMG is difficult to normalize reliably and is less practical to acquire outside lab environments, this study focuses on evaluating the effect of providing one EMG signal at the time. Thus, the second goal involves investigating the effect of tracking single EMG signals on muscle activation using as shoulder model and data. It is revealed that the impact of EMG-tracking single EMG signals on muscle recruitment is minimal, with Mean Absolute Error (MAE) and Zero-normalized Cross-Correlation (ZNCC) changes that are generally less than 0.005 and 0.03, respectively.

Since the role of muscle coordination on joint loading is crucial in pathologies like knee osteoarthritis, the third goal is to evaluate the effect of EMG-tracking on knee joint contact force (JCF). To this end estimations are performed on gait data, which includes in-vivo knee implant reaction forces. EMG-tracking tasks generally cause higher knee JCF during stance. Tracking the most anterior element of the Gluteus Medius improves knee JCF accuracy, with respect to ground truth in-vivo data, the most on average over three subjects and multiple trials, with the Root Mean Squared Error (RMSE) decreasing with 8% and the Zero-Normalized Cross-Correlation (ZNCC) increasing with 4%, although unreliable EMG-normalization raises questions regarding the legitimacy of these results. The potential of EMG-tracking is especially demonstrated with tracking the Rectus Femoris muscle in challenge 6, in which more co-contractionis present. Through an increase of the the Rectus Femoris activity as well as its antagonist, Semitendinosus, the knee load estimation is increased at the first peak during stance, which was previously underestimated without EMG-assistance. However, compared to results from SO and CEINMS, our EMG-assisted RMR solver only achieves lower RMSE in 2 out of 14 trials, although the average difference is only 0.03 body weight.

While the new implementation has not proved itself definitively, it seems promising in predicting co-contraction using EMG-assistance from few EMG signals. In the future this or similar methods might play a crucial role in investigating changes in muscle coordination due to rotator cuff tears, knee osteoarthritis or other musculoskeletal pathologies. This can contribute to more personalized prevention, intervention or rehabilitation.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
BW	Body weight
CEINMS	Calibrated, EMG-informed neuromusculoskeletal modeling (toolbox)
CMC	Computed muscle control (method)
EMG	Electromyography
GH	Glenohumeral (joint)
GRF	Ground reaction force
JCF	Joint contact force
JRF	Joint reaction force
KGC	Knee Grand Challenge (data set)
RMR	Rapid muscle redundancy (solver)
SO	Static optimization (method)
ZNCC	Zero normalized cross-correlation

Symbols

Symbol	Definition	Unit
$\mathbf{a}, a$	Muscle activation (vector, scalar)	-
$\mathbf{e}, e$	Pre-processed EMG (vector, scalar)	-
$\mathbf{F}, F$	Estimated joint reaction/contact force (vector, scalar)	N, Nm or BW
L	Measured joint contact force	N or BW
J	Objective/cost function	-
$\mathbf{q}, q$	Generalized coordinate (vector, scalar)	m or rad
t	Time	s
τ	Time constant	-
$\mathbf{w}, w$	Weight factor (vector, scalar)	-
θ	Angle of the GH-JRF vector with respect to the center line	$^\circ$ or rad

1
Introduction

1.1. Relevance of musculoskeletal studies

Musculoskeletal disorders affect up to a fourth of population of working age [1]. Among secondary industry workers, the prevalence is even around half [2]. In both groups a significant part of disorders affects the shoulder [1], [2], which include tendinitis and muscle-tendon tears. These are especially prominent in the rotator cuff muscles [3], [4], i.e. Infraspinatus, Subscapularis, Supraspinatus and Teres Minor, which are crucial for the stability in the glenohumeral joint [5]. Here, muscle coordination strategies play a key role. For example, rotator cuff tears are related to muscle overloading and overusage [6]. As a consequence of a tear, muscle recruitment might change to relief the teared muscle, which can lead to increased loads on other muscles, possibly causing more tears. In rehabilitation a focus on minimizing loads on the teared muscles as well as preventing overloading of other muscles may be necessary to improve patient outcomes.

A second musculoskeletal disorder is knee osteoarthritis, which affects 10-13% of people of 60 years old and up [7] or even 50% of people 80 years old and up in the UK [8] and is associated with self-reported knee joint instability [9]. Early signs of knee osteoarthritis include increased activity of muscles that wrap around the knee, e.g. the hamstrings and quadriceps [10]. This can cause increased knee loading, which is suspected to be related to the progression of knee osteoarthritis [8], [11], [12]. If increased joint loading due to osteoarthritis increases the progression of osteoarthritis, early treatment could be crucial to slow its progression.

Both rotator cuff tears and knee osteoarthritis can be caused by increased muscle activity, but also might result in increased muscle activity, creating a loop of cause and effect. Therefore, we would like to investigate muscle coordination in patients in order to provide personalized treatment, i.e. preventative measures, interventions or rehabilitation. Measuring biomechanical quantities, such as muscle force or joint contact force, directly is highly invasive, thus musculoskeletal simulation is used to estimate them.

1.2. Muscle redundancy problem

In musculoskeletal simulation, the human musculoskeletal system is modeled as a multi-body system, which enables dynamic simulations. If the kinematics and external forces are measured, muscle forces can be estimated, but this creates a muscle redundancy problem. Because the human body has more muscles than necessary to provide all joint torques, an infinite set of muscle force inputs exist that will result in the measured kinematics. Therefore we need to choose a single muscle recruitment strategy from an infinite possible set. The goal of a muscle redundancy solver is to choose a solution that best approximates human muscle control during the measured movement.

Muscle redundancy is often solved using optimization, such as Static Optimization (SO) [13], in which the best muscle coordination is chosen based on minimizing some cost function, which can be evaluated at any time instant. These methods rely on the idea that humans will learn to use their muscles in predictable way [14] and that this muscle recruitment is optimal in some way, for example lowest in metabolic energy cost. Several studies compare cost functions to find the ones that predict muscle activation the best [15]–[20]. Minimization of the sum of squared or cubed muscle activations provides accurate estimations for most muscles in healthy subjects during low-effort activities, such as normal walking [20], [21], simple shoulder movements [22] or even spine movements [15]. These cost functions will always predict muscle recruitment with the least amount of antagonistic activity as possible.

This is because antagonistic muscles will oppose the joint torques that are necessary to perform the given task, therefore increasing the demand on agonistic muscles. This simultaneous activation of agonist and antagonist muscles is called co-contraction and results in increased overall muscle activity. In a cost function that depends on the sum of muscle activation to some power, muscle recruitment with co-contraction will be of higher cost, thus be perceived as sub-optimal. The same applies to many other cost functions, such as those minimizing the sum of muscle forces, muscle stress [18], [23] or metabolic energy [24], [25]. The problem is that methods using these cost functions cannot accurately predict muscle recruitment in coordination strategies with co-contraction, for example in knee osteoarthritis patients [26].

1.3. Occurrence of co-contraction

It is important to be able to predict co-contraction, because it occurs in many situations, i.e. to stabilize joints, to increase joint stiffness or adapt to impairment or pain.

A stable joint requires compressive forces between bone contact surfaces, which can be provided by co-contracting muscles. When a joint is subject to tensile loading, muscles that span that joint may contract to create a net compressive force again. An example are the rotator cuff muscles in the shoulder which co-contract to push the humeral head into the glenoid fossa of the scapula [5]. Joint instability is especially relevant during fast movements, due to inertial effects, or during loaded movements. In a simulation study by Raikova [27] it has been shown that co-contraction can reduce tensile loads in the elbow joint during flexion, especially fast flexion.

Another reason to use co-contraction is to increase joint or end-point stiffness. Because joint stiffness increases with muscle co-contraction [28], [29], performing a movement or holding a pose with increased antagonistic activity contributes to better disturbance resistance. It has been shown that this strategy plays a key role in motor learning [30], [31], in which co-contraction starts high and is decreased during learning, given disturbances are predictable [32]. For example, subjects asked to walk faster than their preferred speed, use more co-contraction in gait [10], possible because they are not used to walking at the increased speed.

Similarly, muscle recruitment can change if a patient needs to adapt to impairments that impact the musculoskeletal system mechanically, for example in the case of rotator cuff tears which were mentioned earlier. Also changed kinematics after joint replacement [33] or decreased muscle capacity after interventions [34] can change muscle recruitment to contain more co-contraction. The change in mechanical capacity requires relearning, during which co-contraction would be higher until new low-effort muscle control strategies are learned. Besides mechanical change due to impairment, pathologies can also impact the neurological control. For example, individuals affected by cerebral palsy, a neurological disorder, co-contract more during gait than healthy individuals [35]. Also stroke patients use more co-contraction compared to healthy subjects, according to several studies [36], [37]. Many impairments also cause pain, which can lead to adapted movements or muscle recruitment in order to minimize this pain. According to the pain-adaptation theory, muscles that contribute to pain are activated less [38], while at the same time antagonistic muscles are recruited more, which leads to movements with reduced amplitude [39]. In knee osteoarthritis patients, increased co-contraction is present, to increase knee stability and reduce pain [26]. This example also demonstrates that lack of joint stability can induce co-contraction through pain, meaning causes of co-contraction can be related.

In addition to these three main reasons for co-contraction, a small amount of co-contraction might occur in fast movements due to muscle activation dynamics, which causes a delay between excitation change and activation change [40]. When the agonist is deactivating, while the antagonist is activating, momentary co-contraction can be present, because the deactivation rate is lower than the activation rate.

1.4. Muscle redundancy solvers that account for co-contraction

Established methods, such as SO with a lowest effort cost function, will probably underestimate muscle activity in agonists and antagonists in all co-contraction cases described in the section 1.3. Moreover, SO ignores passive forces, possibly causing underestimation of minimum muscle force, and activation

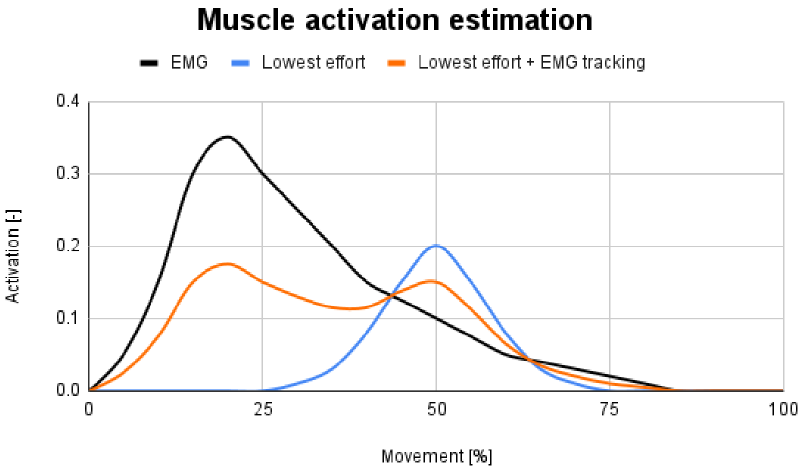


Figure 1.1: An example of EMG-assisted muscle activation estimation. In **black** is pre-processed EMG, in **blue** is activation estimated based on a lowest effort objective and in **orange** is activation estimated based on a combination of lowest effort and EMG-tracking objective. This example demonstrates that EMG-assistance can cause higher activation in case EMG is higher than lowest effort estimations, or lower activation if EMG is lower than the lowest effort activation. Note that this effect is not guaranteed and will depend on the musculoskeletal model, data and solver.

dynamics, potentially causing discontinuities in activation patterns. These limitations are addressed by the Computed Muscle Control method (CMC) by Thelen and Anderson [41], which combines SO and forward dynamic simulation in a PID-loop, which ensures continuous muscle excitation and activation patterns, and includes passive force-length relations. However, CMC still relies on SO with a lowest effort cost function, meaning it will probably predict minimal co-contraction.

While the accuracy of the estimations depends not only on the solver, but also on the musculoskeletal model and the recorded data, it seems probable that optimization algorithm is the main bottleneck. Therefore, a literature study has been performed to find existing methods that claim to predict co-contraction [42], as preparation for this thesis. In this study, four main components to predict co-contraction were identified, which are based on incorporating electromyography (EMG) tracking, muscle synergies, joint stability or joint stiffness.

EMG-tracking

Methods that are based on EMG-tracking use EMG as additional information to feed to the optimizer. EMG measures the electrical excitation of muscles, which reveals when muscles are 'turned on'. Additionally, the magnitude of EMG signal can also be related to the magnitude of muscle excitation, if normalized to the maximum EMG measured during maximum voluntary contraction (MVC) [43], which should correspond to a muscle excitation of 1. Tracking EMG can be done with a cost function that minimizes the difference between predicted muscle excitation and pre-processed EMG, which is called EMG-assisted optimization. Several methods base their muscle recruitment estimations entirely on this EMG-tracking objective [44]–[46]. However, another set of methods uses additional objectives in their cost functions, such as the sum of squared muscle excitation [47]–[53], sum of squared muscle stress [54] or sum of joint contact forces [16]. An example of EMG-assisted muscle activation estimation is given in figure 1.1. The advantage of a multi-objective function over a pure EMG-tracking cost function is that unmeasured muscles, i.e. deep muscles, can still be predicted using the non-EMG-tracking part of the cost function. In multi-objective functions, a balance must be achieved between both objectives, which is controlled through the weight factors for different terms of the cost function [55], which can be muscle-, subject- or experiment-specific. For example, if EMG measurements are trusted, higher weights might be applied to the EMG-tracking objective.

In other methods EMG is used to drive a musculoskeletal model in forward simulation, which is called EMG-driven simulation. In this case, the pre-processed EMG is used to set the excitation of muscle elements. However, in EMG-driven forward simulations the resulting kinematics are often different than

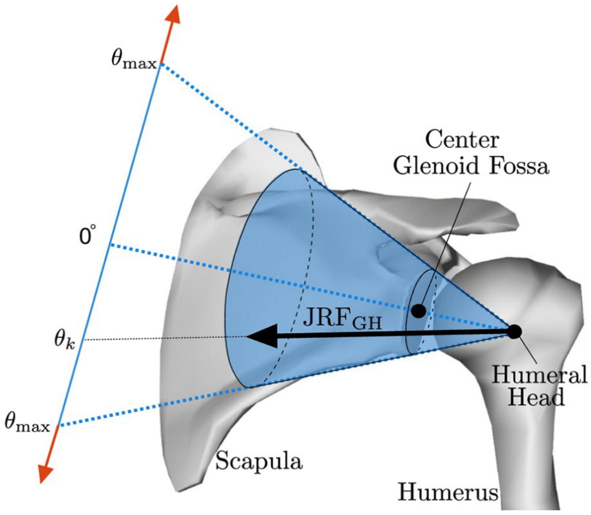


Figure 1.2: Conic glenohumeral (GH) joint reaction force (JRF) constraint as used by the Rapid Muscle Redundancy solver [59]. The angle θ_k of the JRF vector with respect to the center line is enforced to be at most the maximum angle θ_{max}. This maximum angle is set to approximately intersect with the outer edge of the glenoid. This figure is part of Figure 4 from Belli *et al.* [59].

the kinematics that were measured, e.g. using markers or inertial measurement units. Therefore, it is more often used for calibration. For example the optimal set of muscle parameters might be found for which the EMG-driven joint torques approximate the measured joint torques with the smallest error [47].

Muscle synergy

Another set of methods assumes that groups of muscles will have their own synergistic behavior, i.e. certain groups of muscles share activation behavior. This behavior is captured in a constraint, which is enforced during solving of the muscle redundancy problem. The synergy constraint is described as a function of parameters such as joint angles or loads [56]–[58]. The coefficients of the relations are calibrated based on EMG from a set of trials. After calibration, the muscle recruitment estimation routine no longer requires EMG data.

Joint stability

Methods that enforce joint stability also enable the prediction of co-contraction. In practice, it involves setting requirements for the magnitude and direction of the joint reaction force. Several studies constrain the direction of the glenohumeral (GH) joint reaction force (JRF) to a cone aimed at the glenoid fossa [54], [59], [60], as shown in figure 1.2. The rationale behind this constraint is that there should be sufficient compressive force between the humeral head and the glenoid fossa to ensure stability. Some degree of stability is important to prevent strain on ligaments and potential dislocation [5].

Joint stiffness

Similarly to joint stability, joint stiffness requirements can also be used to induce co-contraction in the muscle recruitment solution. Targets for joint stiffness can be used as part of the cost function [61] or be set as a constraint [62].

State-of-the-art

In state-of-the-art methods to focus is on EMG-tracking and joint stability components, which is reflected by two of the most accurate methods found in the literature research.

The first method is the Calibrated, EMG-Informed Neuromusculoskeletal modeling toolbox (CEINMS) by Pizzolato *et al.* [47]. CEINMS uses EMG from calibration trials for model parameter and objective function calibration. This involves an optimization routine with the objective to minimize the error between inverse dynamic and EMG-driven torques, and with muscle parameters and objective weights as optimization parameters. After calibration, muscle recruitment can be predicted for new trials using an objective function that tracks EMG and minimizes squared excitations. Excitation to activation dynamics is accounted for. CEINMS is able to more accurately predict joint contact forces during cycling (hip and knee) [50] and during gait (knee) [63] than conventional methods such as SO. However, the extensive dependency on EMG is also a limitation, because several aspects can make EMG less reliable. EMG measurements can be noisy and pick-up signals from other nearby muscles (cross-talk), especially for surface EMG, which is more accessible and less invasive [43], [64]. Additionally, EMG requires careful placement of electrodes, to which the signal is very sensitive, which complicates experiments. Beside the EMG signal, EMG normalization can also be less accurate, because proper normalization requires an MVC measurements from each measured muscle, which can be hard to obtain. Because CEINMS relies on EMG during calibration as well as estimation, it is especially sensitive to inaccurate muscle excitations derived from EMG. Another limitation with CEINMS is the high demand on computational resources for calibration, which takes multiple hours for each model and cost function [16].

The second method is the Rapid Muscle Redundancy (RMR) solver by Belli *et al.* [59], which successfully implements a joint stability constraint for the glenohumeral joint and accounts for activation dynamics as well as passive muscle forces. Muscle recruitment is predicted based on a cost function that minimizes the sum of squared muscle activation and a conic GH-JRF constraint, depicted in figure 1.2. For simple shoulder movements, the RMR solver is able to accurately predict activation of superficial muscles, with most Mean Absolute Errors (MAE) below 0.1, as well as co-contraction in the rotator cuff muscles, which is necessary for GH joint stability [59]. A limitation of the RMR solver is that it focuses on predicting co-contraction related to GH joint stability in the rotator cuff muscles, not on co-contraction for other reasons, i.e. end-point stiffness, or in other muscles, i.e. the Trapezius.

1.5. Research gap

If we want to improve the accuracy of co-contraction prediction related to the shoulder, we can consider combining the GH-JRF constraint with EMG-tracking. These components could be a good combination, since the GH-JRF constraint predicts co-contraction in the deep rotator cuffs, while EMG-tracking can predict co-contraction in surface agonist/antagonists. Only one study by Sarshari *et al.* [54] combines both a GH-JRF constraint and EMG-tracking, but in this study the EMG-assistance is implemented as a hard constraint. Alternatively, including EMG-tracking as an objective would softly constrain the activation estimations to EMG, which seems more appropriate, because of the possible EMG normalization inaccuracy. The RMR solver [59] already utilizes a GH-JRF constraint, which is verified to predict co-contraction in the rotator cuffs. It also predicts superficial muscle activation well for low co-contraction levels and the algorithm has a low computational demand. The addition of EMG-tracking to this algorithm could increase the accuracy of muscle activation prediction, especially in cases with significant co-contraction.

Since EMG can provide valuable information about muscle recruitment, it seems sensible to use as many EMG signals as possible to increase the accuracy of predictions. However, preferable we do not rely on EMG too much, given the concerns stated in *State-of-the-art* in section 1.4. The EMG signal of one primary agonist might reveal information on the activation level of it's antagonists or fellow agonists. Thus it is hypothesized that tracking a few informative signals could already improve the accuracy of prediction greatly by accounting for co-contraction in those muscles. If this is the case, experiments would require less EMG electrodes, which would be especially helpful during data acquisition outside the lab. However, studies that use EMG-tracking mostly involve many EMG signals at the same time [47]–[52], [54], while few studies have investigated the effect of tracking just one or a few EMG signals. In Romanato *et al.* [53] only one signal of each of four muscle groups is tracked using CEINMS and this method is already more accurate at predicting muscle recruitment of unused EMG signals than SO. Nevertheless, CEINMS also relies on EMG for model and cost function calibration. In order to rely less on EMG, we would like to use the EMG-assisted cost function without the EMG-informed calibration process.

Thus, the first and main goal of this thesis is to implement an EMG-assisted cost function for the

RMR solver and verify that is able to track EMG signals. Existing musculoskeletal models and data are chosen, such that subject specific scaling, marker allocation and data acquisition can be omitted. Using a shoulder model and data by Seth *et al.* [22], the combination of EMG-tracking and the GH-JRF constraint can be tested. **As a second objective of this thesis, the effect of tracking one EMG signal at the time on muscle activation estimations will be evaluated, using the same shoulder model and data.** Unfortunately, the effect on joint reaction forces in the GH joint cannot be validated, since no ground truth joint loads are available in this data set. Therefore, we also run estimations with EMG-tracking using existing lower extremity models by Princelle *et al.* [65] and normal gait data from the *Grand Challenge to Predict In Vivo Knee Loads* [66], which includes in-vivo knee load data. **The third goal of this thesis is to investigate the effect of tracking one EMG signal at the time on knee joint contact force.** Using the same models and data as Princelle *et al.* [63] also allows us to compare estimations from the RMR solver to results from SO, an established method for muscle force prediction, and from CEINMS, a state-of-the-art EMG-assisted method. **Thus, the fourth and final goal is to verify if the knee joint contact force predictions from the RMR solver with and without EMG-tracking are similar to results using SO and CEINMS, respectively, by Princelle *et al.* [63].**

2
Methodology

The Rapid Muscle Redundancy (RMR) solver from Belli *et al.* [59] is re-implemented into Python using classes and EMG-tracking functionality is added. The algorithm is verified using a shoulder model and shoulder movement data from Seth *et al.* [22], which was also used by Belli *et al.* [59]. The same model and data is also used to investigate the effect of EMG-tracking on muscle activation estimations for tracking one muscle at the time. Three different weights for EMG-tracking are briefly investigated and one is chosen for muscle recruitment analysis. A lower extremity model by Princelle *et al.* [65] in combination with normal gait movement data from the *Grand Challenge to Predict In Vivo Knee Loads* (abbreviated to Knee Grand Challenge or KGC) [66] is used to evaluate the effect of single EMG-tracking tasks on knee joint contact force during stance. This data is acquired from subjects with an instrumented knee prosthesis, with force sensors to measure in-vivo knee loads to validate our estimations. The accuracy of knee load estimations by the RMR solver, with and without EMG-assistance, are compared to predictions from SO and CEINMS by Princelle *et al.* [63] using the same models and data.

2.1. Rapid Muscle Redundancy solver

2.1.1. Original RMR solver algorithm

The method proposed in this thesis is an extension of the Rapid Muscle Redundancy (RMR) solver by Belli *et al.* [59]. The RMR solver is able to efficiently optimize muscle activation while taking into account active as well as passive muscle forces, dynamic activation behavior and constraints on joint reaction forces. It requires a musculoskeletal model, generalized coordinate data and any external force data that is not yet embedded in the musculoskeletal model. With this data as input, the RMR solver performs a constrained minimization of the objective function to find muscle activations at every time instant.

In short, the RMR solver minimizes muscle effort, while (1) matching experimental accelerations, (2) ensuring continuous activation and (3) constraining the GH-JRF vector to a cone. The objective function and each of the constraints will be explained in more detail in the following paragraphs.

Minimal muscle effort cost function

The cost function is based on a lowest effort criterion which minimizes the sum of squared muscle actuator activation and sum of squared reserve actuator activation:

$$\min_{\mathbf{a}_k} J_k = \sum_{i=0}^{N_{\text{act}}} w_i (a_{k,i})^2 \ , \tag{2.1}$$

where:

- J_k is the cost of the solution at time step k;
- w_i is the weight attributed to actuator i;
- $a_{k,i}$ is the activation of actuator i at time step k.
- N_{act} is the total number of actuators.

The reserve actuators can provide direct force or torque to a generalized coordinate, thus helping to fulfill the acceleration constraint in case the muscles are too weak. In order to penalize the use of reserve actuators, their contribution to the cost function must be higher than that of the muscle actuators. In equation 2.1, all actuators are lumped together and w_i is used to attribute different weights to muscles and reserve actuators. For the RMR solver, the reserve actuators should be embedded in the musculoskeletal model as `Opensim::CoordinateActuator` types. There needs to be at least one coordinate actuator per unlocked coordinate, which is equal to the number of degrees of freedom.

Experimental acceleration constraint

The first constraint makes sure that the joint acceleration imposed by the muscle recruitment matches that of the kinematic data at every time step [59]:

$$\mathbf{A}_k \mathbf{a}_k = \ddot{\mathbf{q}}_{k,\,\text{ind}} \ , \tag{2.2}$$

where:

- $\mathbf{a}_k$ is a vector containing the activation of all N_{act} actuators at time step k;
- $\ddot{\mathbf{q}}_{k,\,\text{ind}}$ is a vector of actively generated generalized accelerations of all N_q coordinates at time step k, which is equal to the inverse kinematics (experimental) accelerations $\ddot{\mathbf{q}}_{k,\,\text{exp}}$ minus the passive, state induced accelerations $\ddot{\mathbf{q}}_{k,0}$ (gravity, external forces and passive muscle forces);
- $\mathbf{A}_k$ is a $N_q \times N_{\text{act}}$ matrix with each entry (j,i) the effect of each actuator i (activation of 1) on the acceleration of each coordinate j at time step k.

If this constraint fails, the found muscle recruitment solution will not result in the joint torques necessary for the movement.

Activation dynamics constraint

The second constraint enforces muscle activation dynamics. Based on the activation of each muscle at the last time step and the size of the time step itself, boundaries for the muscle activation at the current step can be calculated [59]:

$$\begin{aligned} a_{i,k,\,\text{lb}} &= \max\left[0, \left(a_{i,k-1} - a_{i,k-1}\left(\frac{1}{2} + \frac{3}{2}a_{i,k-1}\right)\frac{t_k - t_{k-1}}{\tau_{\text{deact}}}\right)\right] \ , \\ a_{i,k,\,\text{ub}} &= \min\left[1, \left(a_{i,k-1} - \frac{1 - a_{i,k-1}}{\frac{1}{2} + \frac{3}{2}a_{i,k-1}}\frac{t_k - t_{k-1}}{\tau_{\text{act}}}\right)\right] \ , \end{aligned} \tag{2.3}$$

where:

- $a_{i,k,\,\text{lb}}$ and $a_{i,k,\,\text{ub}}$ are the lower and upper activation bounds (respectively) for actuator i at time step k;
- t_k is the time (in seconds) at time step k;
- τ_{act} and τ_{deact} are the activation and deactivation time constants, respectively, also in seconds.

These activation boundaries are only applied to all actuators that are muscles. The max and min operators are used to ensure boundaries never go below 0 or above 1. By enforcing these boundaries, the muscle recruitment solution will account for muscle activation dynamics.

Glenohumeral stability constraint

The final constraint is the glenohumeral stability constraint, which is used only for the shoulder model in this thesis. It dictates that the GH-JRF vector must have an angle θ_k which is less than θ_{max}. The angles are with respect to center line, which is defined to pass through the middle of the humeral head and the center of the glenoid fossa, as shown in figure 1.2. The joint reaction force vector F_k (in cartesian space) at any joint can be calculated as follows [59]:

$$\mathbf{F}_k = \mathbf{A}_{F,k}\mathbf{a}_k + \mathbf{F}_{0,k} \ , \tag{2.4}$$

where:

- $\mathbf{A}_{F,k}$ is a $3 \times N_{\text{act}}$ matrix with every column $(:,i)$ the effect of actuator i (activation of 1) on the JRF vector for time step k;

- $\mathbf{F}_{0,k}$ is the JRF vector of passively generated forces (gravity, external forces and passive muscle forces) for time step k.

The stability constraint, which concerns the direction of GH-JRF vector, is then formulated as [59]:

$$\left(\frac{\theta_k(\mathbf{a}_k)}{\theta_{\text{max}}}\right)^2 - 1 \leq 0 \ , \tag{2.5}$$

where:

- $\theta_k(\mathbf{a}_k)$ is the angle of the GH-JRF vector at time step k, which is effected by actuator activation;
- θ_{max} is the maximum angle for which the GH-JRF still intersects with the glenoid fossa.

2.1.2. New EMG-tracking extension

EMG-tracking will be established using an EMG-assisted cost function, in which EMG data is associated with muscle activation. It is important to note that 'EMG-tracking' does not necessarily mean unconstrained tracking of EMG. In this thesis any 'EMG-tracking' or 'tracking tasks' are always performed together with a lowest effort objective, namely the minimization of squared muscle activation. This lowest effort criterion is chosen based on accurate predictions using the original RMR solver in Belli *et al.* [59].

EMG-assisted objective function

For the extension, a new EMG-assisted cost function is implemented to replace the original (equation 2.1):

$$\min_{\mathbf{a}_k} J_k = \sum_{i=0}^{N_{\text{act}}} w_{1,i}(a_{i,k})^2 + w_{2,i}(a_{i,k} - e_{i,k})^2 \ , \tag{2.6}$$

where:

- $w_{1,i}$ and $w_{2,i}$ are the weights for minimal effort and EMG-tracking (respectively) for actuator i;
- $a_{i,k}$ and $e_{i,k}$ are actuator activation and pre-processed EMG (respectively) for actuator i at time step k.

The reason for taking the square of the tracking error, is to penalize large errors exponentially, which also applied in several other studies [50]–[53], [67].

Pre-processed EMG (e) would more accurately represent excitation than activation. Taking into account activation dynamics, the excitation signal would precede activation with around 10 to 50 ms [68]. Because the RMR solver uses activation as an optimization parameter, the decision is made to track EMG with activation and ignore activation dynamics for the tracking task. The pre-processing of EMG is described in the *EMG pre-processing* subsection.

In the original cost function described by Belli *et al.* [59], a distinction was made between muscles and reserve actuators. Here, this distinction is removed and instead the weight vectors are used as a general way to differentiate between actuators, which can be muscles as well as reserves actuators. Thus, the weight vectors allow balancing between actuators as well as between objectives. In the same way, this means that not all actuators need to be tracked, because weights in $\mathbf{w_2}$ can be set to zero. It should be noted that both activation and tracking error have similar magnitudes (up to 1), thus weights of similar magnitudes would approximately make both objectives equally important. However, when picking weights, the number of actuators in the model and the number of actuators that track must be taken in consideration. For example, take a model with 100 muscles in which we want to track only a single muscle. If we take $w_1 = 1$ for all muscles, $w_2 = 1$ for the tracking muscle and $w_2 = 0$ for all other muscles, we end up with a large emphasis on lowest effort, with a total cumulative weight of $|\mathbf{w}_1| = 100$, compared to EMG-tracking, with a total cumulative weight of $|\mathbf{w}_2| = 1$.

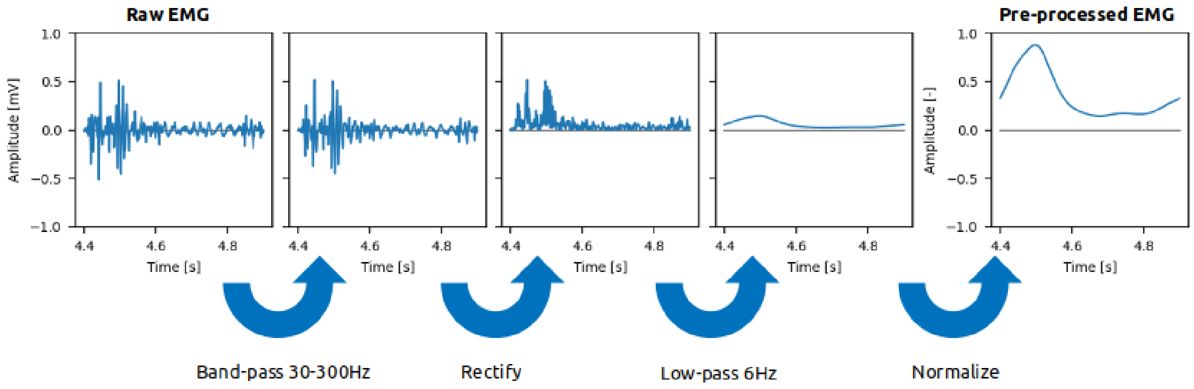


Figure 2.1: Example of EMG pre-processing. The data is from the Gastrocnemius Medialis EMG signal from trial 1 of KGC 5. See section 2.3.2 for more on Knee Grand Challenge data.

EMG pre-processing

In the EMG-assisted cost function presented in equation 2.6 requires pre-processed EMG, because raw EMG signals are not yet suitable for tracking, since they are noisy and their magnitude cannot be related to a non-zero excitation level. Therefore, raw EMG needs to be filtered, rectified and normalized. For filtering, a combination of high-pass filtering, to remove long-term drift, and low-pass filtering, to remove noise, is used. This can be done subsequently or simultaneously, i.e. by using a band-pass filter. In this thesis zero-lag 2nd or 4th order Butterworth filters are used with varying cut-off frequencies, see section 2.2.3 and 2.3.3 for detailed descriptions for the shoulder and gait simulations, respectively. Rectification and subsequent normalization of filtered EMG is crucial in order to relate the EMG values in millivolts to muscle excitation on a scale of 0 to 1. This can be achieved by dividing each EMG signal by the value associated with a muscle excitation of 1. Commonly, this value is determined by taking the maximum recorded signal during a maximum voluntary contraction (MVC) trial of the targeted muscle. Alternatively, the maximum value across all available trials could be used to make sure no EMG signal represents an excitation exceeding a value of 1. Figure 2.1 shows an example of EMG pre-processing.

2.1.3. Class-based revision

Before the EMG-assisted cost function is added to the RMR solver, the base code for the RMR solver needs to made suitable for this adaptation. The original RMR solver by Belli *et al.* [59] was written in MATLAB. Beck *et al.* [69] have re-implemented the RMR solver in Python, in order to “facilitate the integration of the RMR solver with the proposed active strain map method and a robotic system” ([69], p. 4). For this thesis, a new Python implementation is made, which is inspired by the implementation of Beck *et al.* [69]. The key features of the revision is that it is generalized and class-based.

Motivation for revision

The advantage of a generalized implementation is that it is more practical to use with different models and different settings. Since the RMR solver was developed for a shoulder model, the previous versions made assumptions that could clash when using different musculoskeletal models. The new version makes less assumptions, e.g. about the order of a list containing model objects, and features more settings to deal with model differences and user preferences.

Using classes in the new implementation also has several advantages: modularity, scalability and ease of use. Every class can be seen as a module that is developed to fulfill certain functions and which has specific interfaces. If certain functions are not necessary, corresponding modules can easily be removed or replaced by other modules. With this modularity also comes the advantage that the solver (`RMRsolver` class) can be used in a live estimation process as well as for pre-recorded data analysis. `RMRsolver` allows this by solving the muscle redundancy problem one time step at the time, without internally looping over data. Live estimation can be used for live muscle strain maps during robot-assisted shoulder therapy [69]. The version of Beck *et al.* [69] only worked for pre-recorded data, because the looping was integrated, requiring the total number of time steps to be known at setup. In order to work in a live estimation loop, the previous implementation needs several modifications throughout the script. A separate, adjusted version can be used in live estimation, but this version would no longer work on pre-recorded data.

This ties into the second main advantage of the class based implementation, which is improved feature scalability. The `RMRsolver` class only needs a single version and within the class script the amount of duplicate lines of code is minimized, through the use of subclasses and functions. This means that when a developer wants to add or refine a feature of the RMR solver, it can be done in one place. If there would be multiple versions or many repeat lines of code, changing a single feature requires the same edits in multiple locations, increasing the risk to introduce bugs.

A final important advantage of the new implementation is ease of use. In the previous implementation, user input was required in several places inside the main script which counts hundreds of lines of code, which made it hard to set up for new users. The class-based implementation requires only top-level interaction with classes, making it more intuitive to use. With basic programming knowledge a personalized script can be set up to employ the RMR solver. This top-level script would be similar to setting options in a GUI (e.g. Opensim [70]), such as selecting file locations and setting filtering cut-off frequencies. An example of a top-level script can be found in appendix B.1.

Realization of revision

The algorithm used for solving the muscle redundancy problem, as described in section 2.1.1, is implemented into a single class called `RMRsolver`. However, several more classes are created to assist this class. Two classes are mandatory to set up the `RMRsolver` class: `EMGtrack` for cost function evaluation and `ModelSpecs` for musculoskeletal model information. These three classes could be integrated in a live estimation loop, depicted in figure 2.2. All other classes provide extra functionality for estimations on pre-recorded data, as shown in figure 2.3. This includes up to four 'manager' classes for experimental data management, the `MotionAnalysis` class for looping data and plotting and the `Result` class for storage of simulation results. In this thesis, only the estimation routine for pre-recorded data is used.

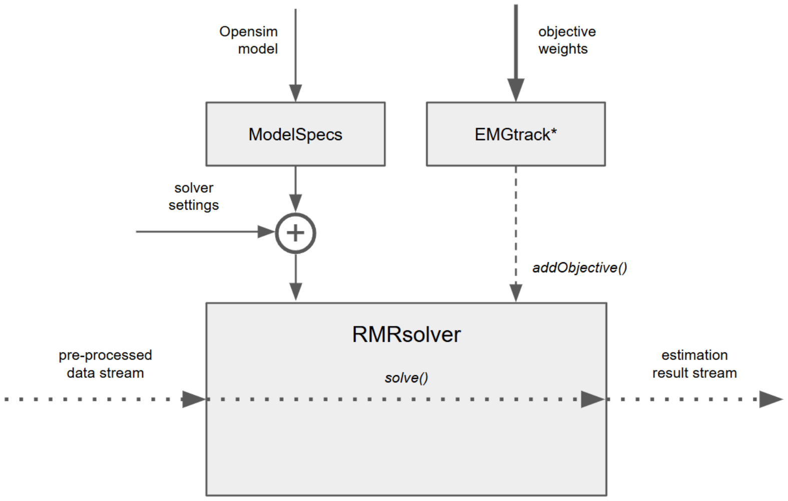


Figure 2.2: Block diagram of class interaction for estimation on live data. Blocks represent classes. **Solid arrows** represent class initialization, e.g. to initialize an instance of `RMRsolver`, the solver settings and an instance of `ModelSpecs` need to be provided. **Dashed arrows** represent adding class instances to another class as members, e.g. an instance of `EMGtrack` can be added to an instance of `RMRsolver` by invoking the `addObjective()` member function of `RMRsolver`. The **dotted arrows** represent estimation of muscle recruitment, which is done per time step by invoking the `solve()` member function of `RMRsolver`. * `EMGtrack` is the objective class used in this thesis, but could be replaced by another objective class.

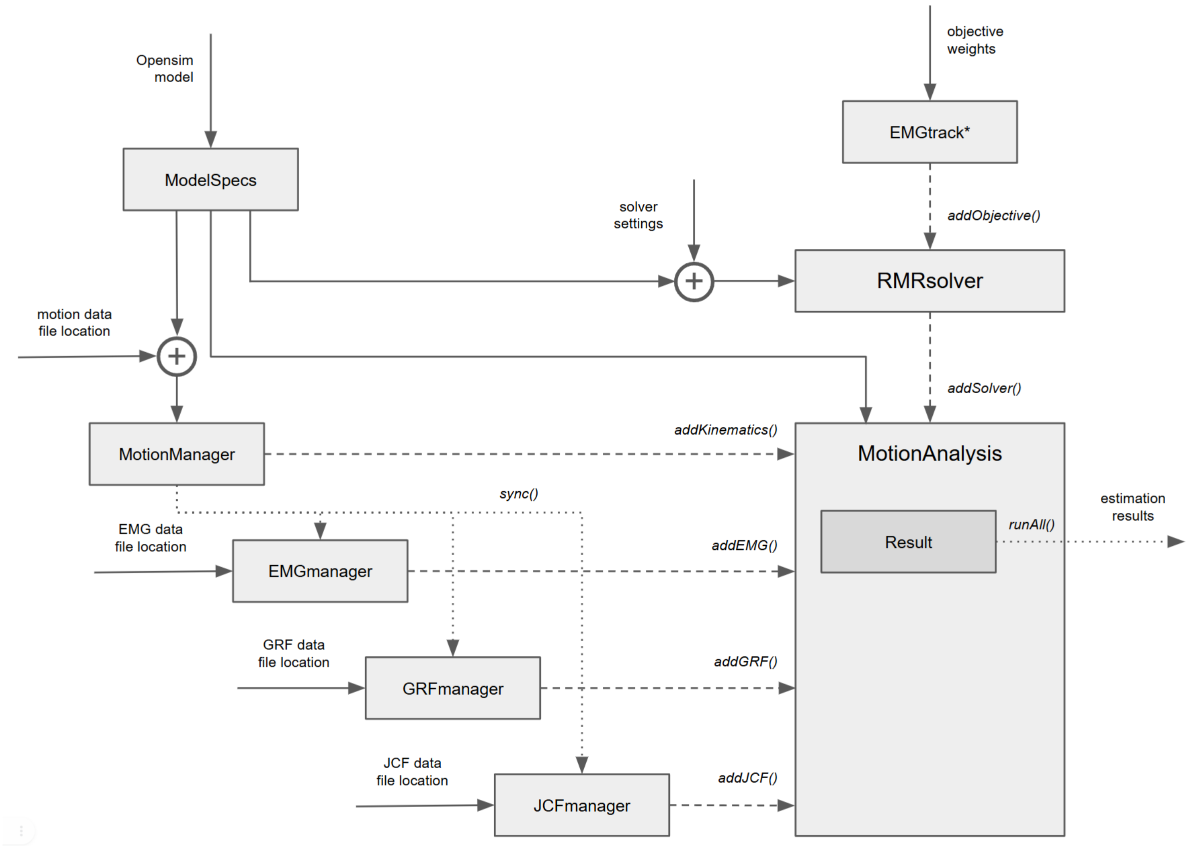


Figure 2.3: Block diagram of class interaction for estimation on pre-recorded data. Blocks represent classes. **Solid arrows** represent class initialization, e.g. to initialize an instance of `MotionManager`, the motion data file location and an instance of `ModelSpecs` need to be provided. **Dashed arrows** represent adding class instances to another class as members, e.g. an instance of `MotionManager` can be added to an instance of `MotionAnalysis` by invoking the `addKinematics()` member function of `MotionAnalysis`. **Dotted arrows** represent other operations, e.g. time-synchronizing the EMG data series of an instance of `EMGmanager` based on the kinematic data series of an instance of `MotionManager` by invoking the `sync()` member function of `EMGmanager`. * `EMGtrack` is the objective class used in this thesis, but could be replaced by another objective class.

`ModelSpecs` has been created to extract and store Opensim model information, e.g. generalized coordinate names or muscle optimal fiber lengths. Extracting this information to the Python environment takes many Opensim Python API commands, which are performed on initialization of this class. After initialization, it functions as a data class to be passed to other classes that need model information (`RMRsolver`, `MotionManager` and `MotionAnalysis`).

The `EMGtrack` class is used as a configurable cost function. The `scipy.minimize()` function used in `RMRsolver` requires a callable function as an objective function. This function must take at least optimization parameters as an input and return only solution cost as an output. If a static function was used, the cost function weight factors would need to be hard-coded or provided at every function evaluation. Since we set weight factors to be constant over time for estimation routine, it would be preferred to store the weights inside the function. This is possible with classes, where the weights can be set on initialization (`__init__` method) and the cost function evaluation is performed on call (`__call__` method). Moreover, because the objective is formulated in a separate class, it can easily be swapped for alternative cost functions in future studies.

`RMRsolver` needs an instance of `ModelSpecs` to initialize. In addition, values for several settings can be passed at initialization, such as acceleration constraint tolerance or reserve actuator bounds. All settings that are not customized are assigned a default value, which makes setting up the solver easier. Before `RMRsolver` can be used to solve muscle redundancy, an objective class (i.e. `EMGtrack`) needs to be added using `addObjective()`. Most computational work is done within the `solve()` class method, which solves the muscle redundancy problem for a single time step. `solve()` requires generalized position, velocity and acceleration. If applicable, external forces or EMG data can be passed as well. The output includes actuator activation, joint reaction forces, optimization result info and muscle forces (active and passive). Because `RMRsolver` needs to make many repeat computations, several functions are implemented, which are collected in a separate file (`utilsRMRsolver`).

Many studies run musculoskeletal simulations after experimental data is collected. In this thesis, we also only use the extended RMR solver on pre-recorded data. Therefore a class, `MotionAnalysis`, has been implemented to bring experimental data together and loop it through the `RMRsolver` using `runAll()`. For initialization only an instance of `ModelSpecs` is required, but for looping access to kinematic data is also necessary. This access is given by adding the kinematic manager class `MotionManager` to `MotionAnalysis` using the `addKinematics()` method. Since this class has already access to all the data, it has also been equipped with plotting functionality, which can be used to quickly interpret results after estimation.

Pre-recorded data is loaded and pre-processed in dedicated manager classes: `MotionManager` for kinematic data, `EMGmanager` for EMG data, `GRFmanager` for ground reaction forces and `JCFmanager` for joint contact forces measured with implants. All manager classes share similar functions. For initialization, each class needs to be provided with the appropriate data file location, in order to load the data. Then, data can be filtered (`filter()`) or cropped and resampled (`setTime()`). In `EMGmanager`, EMG can be normalized using `norm()` by providing values for normalization, as well as rectified using `rect()`. When bringing all this data together, it is important that it is time-synchronized. Thus, a method `sync()` can be used to synchronize the data from any manager class instance with respect to data from another `MotionManager` instance.

Finally, the `Result` class is used to store the processed experimental data (manager classes) together with the estimation for activation, JRFs and muscle forces. Instances of this class can be saved to a file to use for analysis later.

See appendix A for a link to repository with the revised Python RMR solver code, which also includes more detailed descriptions for all classes.

2.2. Shoulder movement simulation

For verification of the revised RMR solver and the EMG-assisted cost function, the same shoulder model and data are used as Belli *et al.* [59]. This model and data are also used to test the sensitivity of the EMG-tracking weight and to evaluate the effect of EMG-tracking on muscle activation predictions. An overview of the simulation routine used is shown in figure 2.4.

2.2.1. Shoulder model

The thoracoscapular shoulder model, shown figure 2.5, is an Opensim [70] model originally by Seth *et al.* [22] and adapted by Belli *et al.* [59] to work with the RMR solver. This model includes the elbow, shoulder/glenohumeral, acromiumclavicular, sternumclavicular and scapulothoracic joints. In total, it has 17 degrees of freedom and 33 *Millard2012EquilibirumMuscle* actuators, listed in table C.1, based on the muscle model by Millard *et al.* [71]. Seth *et al.* [22] scaled the generic model to fit the subject of the experiment. The bones, joint locations and muscle insertion points were scaled “linearly based on marker-based distances between the subject and the base (generic) model” ([22], p. 2). Additionally, the optimal fiber length and tendon slack length from each muscle element are scaled in such a way that the ratio between the two is consistent to the unscaled model. The ellipsoid that presents the thorax for the scapulothoracic joint was scaled to result in the lowest marker errors for inverse kinematics. All the scaling was done by Seth *et al.* [22] and the scaled model, including minor modifications by [59], was kindly provided for use in this thesis. No further modifications to the model were made.

2.2.2. Shoulder data

The shoulder movement data is from a single subject experiment performed by Seth *et al.* [22]. It consists of three trials for each of three movements (forward flexion, abduction and shrugging) for two different conditions (unloaded, 2 kg load in hand). For this thesis, only the 9 loaded trials are considered, because these will involve higher overall muscle activation. The subject is a 26-year-old

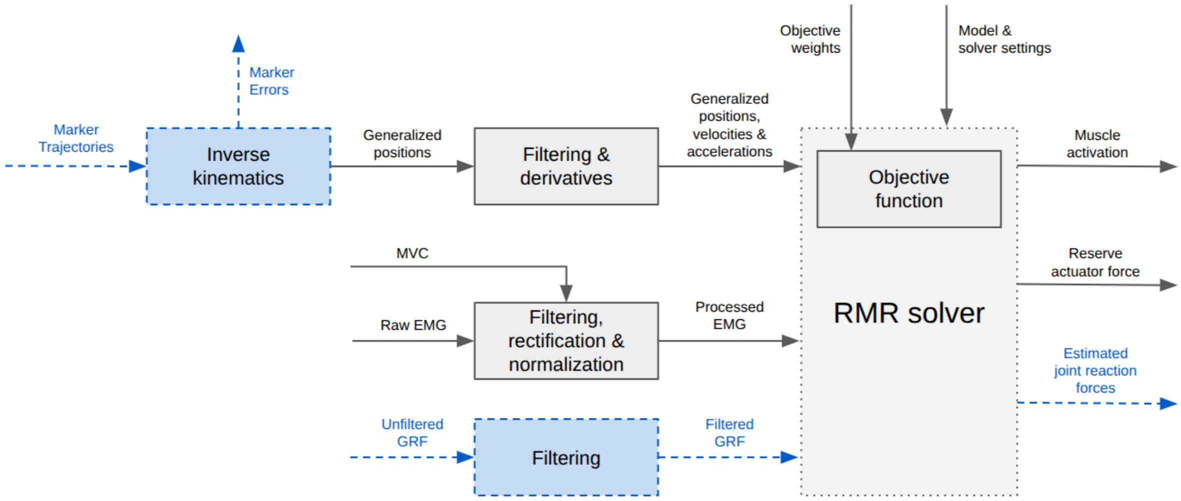


Figure 2.4: Flowchart of the simulation routine used in this thesis. Blocks and arrows in dotted blue are only relevant for the normal gait simulations.

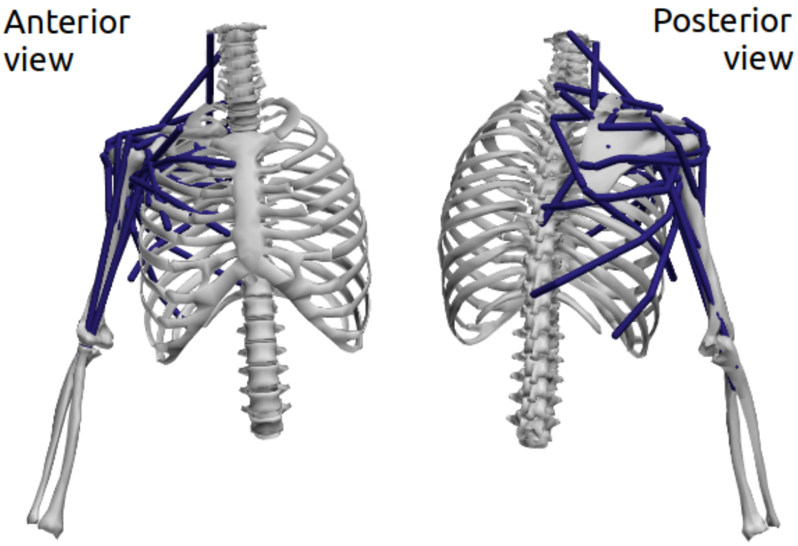


Figure 2.5: The thoracoscapular shoulder model, created by Seth *et al.* [22].

female (mass 52 kg, height 1.62 m) which performs all movements with her dominant (right) shoulder. All trials start and end in resting position, i.e. with the arm alongside torso. Flexion movements reach a peak flexion of 140 degrees, while the abduction get to an angle of 160 degrees with respect to torso. Markers positions are recorded as well as the EMG of 11 muscles, which are listed table C.1. For this thesis, the motion files containing generalized joint angle data were kindly provided, such that it was not necessary to perform inverse kinematics. Seth *et al.* [22] have reported that the marker RMSE is below 1cm for all motions. The provided EMG and motion data are also time-synchronized.

2.2.3. Shoulder data processing and estimation

The generalized coordinate data, saved in *.mot* files, is low-pass filtered with a cut-off at 3 Hz (4th-order Butterworth). EMG is first high-pass filtered at 100 Hz (4th-order Butterworth), then rectified and low-pass filtered at 4 Hz (2nd-order Butterworth). All filters are zero-lag. Finally, EMG is also normalized with respect to the highest values found during maximum voluntary contraction (MVC), which are displayed in table C.1. For the *Infraspinatus_S* and *LatissimusDorsi_M* static offset is removed, based on the lowest value across the whole trial. Each trial is cropped to a time range in which the motion is performed, see table C.2. Motion and EMG data is resampled at 10 Hz, before it is evaluated by the RMR solver.

All muscle actuators and the reserve actuators receive weights of 1 and 10, respectively. The coordinate actuators that are set as reserves have an optimal force of 1 N(m), meaning they produce a force/torque of 1 N(m) for an activation of 1. Reserve actuators that control coordinates which cannot be fully actuated by muscles get weights of zero, making them free to use by the optimizer. Refer to table C.3 for all weights used. The activation dynamics constraint (with time constants $\tau_{\text{act}} = 10$ ms and $\tau_{\text{deact}} = 40$ ms) and the GH-JRF constraint (with $\theta_{\text{max}} = 19°$) are active. Other solver settings were set to default, which is described in appendix B.2.

Everything, except for the resampling, is done to replicate the method used by Seth *et al.* [22] and Belli *et al.* [59]. The reason to subsample the 100 Hz motion data to 10 Hz is to speed up computation, which takes about 0.2-0.3 seconds per time step. Provided an average trail is around 5 seconds, the reduced sampling rate saves around 90-135 seconds per trial or 8-12 hours total for all 9 trails, 12 cost function conditions and 3 tracking weight factors computed for this thesis.

EMG-tracking weights

To investigate the effect of tracking, simulations are run in which a single muscle receives a non-zero weight for the EMG-tracking objective. This is performed for each muscle that has a corresponding EMG signal. To determine a balanced tracking weight, a rudimentary sensitivity test is performed. All shoulder movement trials and tracking tasks are performed with tracking weights of 1, 3 or 5 and results are assessed using Mean Absolute Error and Zero-Normalized Cross-Correlation metrics, as described in section 2.2.4. The lowest tracking weight that shows significant impact on the tracking muscle for all tracking tasks is chosen for all subsequent analysis of shoulder movements.

2.2.4. Muscle activation estimation analysis

Estimated muscle activation is compared to processed EMG using Mean Absolute Error (MAE) and Zero-Normalized Cross-Correlation (ZNCC) at zero time shift, see equations 2.7 and 2.8, respectively. A drawback of MAE is that muscles with low activation and EMG will always have a low MAE, regardless of their trend over time. At the same time, muscles with similar trends for activation and EMG might show a high MAE. These trends over time are important, since magnitude error also depends on the reliability of the normalization to maximum voluntary contraction. Therefore, the Zero-Normalized Cross-Correlation is used as a secondary metric to compare muscle activation to EMG. ZNCC is a score between 1 and -1, with 1 meaning perfectly matching trends, -1 meaning perfectly opposite trends and 0 meaning uncorrelated.

Besides comparisons between estimations and EMG, differences between estimations can also be presented in MAE or ZNCC, substituting EMG for a second estimation.

$$\text{MAE} = \frac{1}{N_k} \sum_{k=0}^{N_k} |a_k - e_k| \tag{2.7}$$

$$\text{ZNCC} = \frac{1}{N_k} \sum_{k=0}^{N_k} x_k y_k \tag{2.8}$$

$$\text{with} \quad x_k = \frac{a_k - a_{\text{mean}}}{a_{\text{std}}}$$

$$\text{and} \quad y_k = \frac{e_k - e_{\text{mean}}}{e_{\text{std}}}$$

- N_k : number of time steps
- a_k : muscle activation at time step k
- e_k : normalized EMG at time step k
- $_{\text{mean}}$: mean over the whole time series
- $_{\text{std}}$: standard deviation over the whole time series

2.3. Gait simulation

Because the shoulder data has no joint contact forces available, we also perform estimations using lower extremity models and normal gait data, which includes in-vivo joint load measurements. This allows use to investigate the effect of EMG-tracking on knee joint loading. Princelle *et al.* [63] has used the same models and data for estimation used SO and CEINMS, which enables us to compare the prediction of RMR to these methods.

2.3.1. Lower extremity model

The lower extremity model used is an Opensim [22] model by Princelle *et al.* [65]. It consists of the hip, knee, ankle, subtalar and metatarsophalangeal joints of only one leg (the instumented side of the corresponding subject). The model has 13 degrees of freedom and 40 *Millard2012EquilibriumMuscle* actuators, which are listed in table C.4. The skeletal system is generated from CT scans using the *INSIGNEO* pipeline. A patellofemoral joint is added based on the *Rajagopal2015* model [72]. The muscle tendon units from that same *Rajagopal2015* model are scaled and mapped onto the skeletal system, with some adjustments to insertion points and additions of extra wrapping shapes based on the CT scans if necessary. Muscle parameters are scaled from the *Rajagopal2015* model as well, then morphologically optimized. In this way Princelle *et al.* [65] have created three scaled models to fit the three different participants of challenges 4, 5 and 6 from the Knee Grand Challenge, described in section 2.3.2.

In order to make the models suitable for the RMR solver some modifications need to be made. For each coordinate, a coordinate actuator is added to act as a reserve actuator during simulation. Additionally, the patella constraint, which ensures that the patella rotates with respect to the femur based on the knee angle, is removed for muscle recruitment estimation. Because its position will already be constrained though generalized coordinate data during estimation, it is not required anymore and must be removed to avoid conflicts. For inverse kinematics a separate version *with* patella constraint is used to acquire correct generalized coordinate data for the patella.

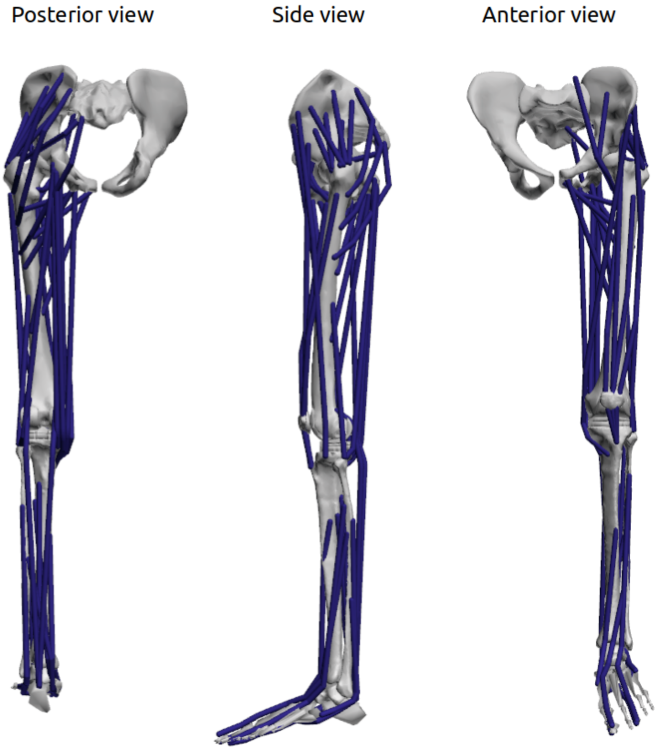


Figure 2.6: The lower extremity model of subject PS from challenge 5, created by Princelle *et al.* [65].

Passive forces

One problem that arises when using these models with the RMR solver are high passive forces, for example of more than one times the maximum isometric force for the Gastrocnemius Lateralis (*gaslat*). The effect of passive forces is increased due to the use of rigid tendons in the RMR solver. With rigid tendons the muscle fiber length is increased more during stretching than with compliant tendons, increasing passive fiber force, which is demonstrated in figure C.1. In the study by Princelle *et al.* [63], passive forces are not considered, because SO and CEINMS do not include them, as opposed to the RMR solver.

For solving the problem of unrealistically high passive forces, we consider two different options, which are both tested in simulation. Option A is to disable passive forces altogether, which can be done with a simple setting for the RMR solver. Option B is to adjust the model to have more realistic passive forces. To this end, a *B-specification* of each model is made by adjusting the passive fiber force curves in an optimization process to match experimental passive joint torques, using a publicly available MATLAB script by Uhlrich *et al.* [73]. The optimization adjusts two parameters of the passive fiber force curve, as shown in figure 2.7, within a predefined range to change passive muscle force onset. *strainAtZeroForce* (ε_0) determines the start of passive force generation and *strainAtOneNormForce* (ε_1) the point at which the passive force reaches one times the normalized fiber force. An objective function evaluates the fit of estimated passive joint torque curves to experimental curves from Silder *et al.* [74] and penalizes solutions based on the deviation from the starting values. This ensures that the optimizer only changes the parameters if the fit of passive joint torque curve improves significantly. The default values for ε_0 and ε_1 in *Millard2012EquilibriumMuscle* elements is 0 and 0.7, respectively. In the original application to the *Rajagopal2015* model [72], Uhlrich *et al.* [73] allowed ε_0 and ε_1 to deviate by 0.2 from the starting value, in order to keep the change physiological (according to the muscle model). However, this resulted in several muscles reaching an optimum at this parameter limit, which indicates that a better fit may exist beyond this limit. Therefore we allow broader parameter ranges for calibration in this study: $\varepsilon_0 \in [-0.5, 2]$ and $\varepsilon_1 \in [0, 3]$.

	Challenge 4	Challenge 5	Challenge 6
Sex	Male	Male	Male
Code	JW	PS	DM
Height [m]	1.68	1.80	1.72
Weight [kg]	66.7	75	70
Intrumented side	Right	Left	Right
Trials (normal gait)	2, 3, 4, 5, 7	1, 8, 9, 11	4, 5, 6, 7, 9

Table 2.1: Information on subjects [76] and trials used for each challenge.

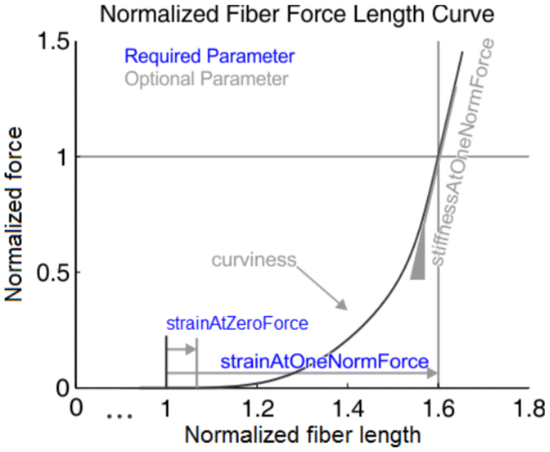


Figure 2.7: Force-length curve for the passive fiber force contribution of a *Millard2012EquilibriumMuscle* muscle element [75]. *strainAtZeroForce* = ε_0, *strainAtOneNormForce* = ε_1.

2.3.2. Gait data

Gait data comes from the publicly available set of the *Grand Challenge Competition to Predict In Vivo Knee Loads* [66]. Each challenge features a single participant with an instrumented knee prothesis. Several different exercises are performed, while marker motion data, ground reaction forces, EMG signals and implant force data are collected.

For this thesis, the normal gait trials of challenges 4, 5 and 6 are used. Table 2.1 shows information about the participants as well as the trials evaluated for each challenge. The trials evaluated in this thesis are the same as evaluated in the study of Princelle *et al.* [63]. Each normal gait trial consists of four to five steps, corresponding to two to three seconds of data. Positions of 52 (challenge 4) or 48 (challenge 5 and 6) markers are recorded at 120 Hz. Ground reaction forces are available for the middle three steps (right stance, left stance, right stance) from three different force plates, sampled at 3840 Hz. EMG from 15 muscles of the right leg, listed in table C.4, were measured at 1000 Hz. In challenge 4, the instrumented implant has recorded four uniaxial forces for the medial-anterior, medial-posterior, lateral-anterior and lateral-posterior contact points on the prosthetic tibia at approximately 50 Hz. For challenges 5 and 6, forces and torques acting on the tibia are available for all cartesian directions. For this thesis, the synchronized data set was used, in which all marker, GRF, EMG and implant force data are time-synchronized. Figure 2.8 shows an overview of the time-synchronization process, in which the data is filtered and resampled using cubic spline interpolation [66].

The sample frequencies in the downloaded data from challenge 4, 5 and 6 are different than described in available documentation [66], [76]. The downloaded data frequencies, which are used in this thesis, are 120 Hz instead of 100 Hz for marker data and implant force data, and 1200 Hz instead of 1000 Hz for GRF data and EMG data.

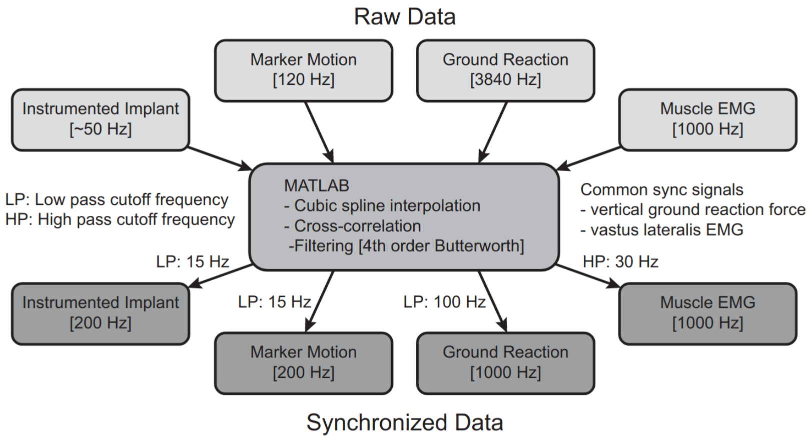

Raw Data
Marker Motion [120 Hz]
Ground Reaction [3840 Hz]
Instrumented Implant [~50 Hz]
Muscle EMG [1000 Hz]
MATLAB
- Cubic spline interpolation
- Cross-correlation
-Filtering [4th order Butterworth]
LP: Low pass cutoff frequency
HP: High pass cutoff frequency
Common sync signals
- vertical ground reaction force
- vastus lateralis EMG
LP: 15 Hz
HP: 30 Hz
LP: 15 Hz
LP: 100 Hz
Instrumented Implant [200 Hz]
Muscle EMG [1000 Hz]
Marker Motion [200 Hz]
Ground Reaction [1000 Hz]
Synchronized Data

Figure 2.8: "Flowchart describing filtering and synchronization of raw experimental data" from Figure 3 from Fregly *et al.* [66] [66].

2.3.3. Gait data processing and estimation

Inverse Kinematics (IK) is performed on marker data to derive generalized coordinate data. Markers that do not have a valid position, i.e. appearing at $x, y, z = 0, 0, 0$ due to obscuration, during a large portion of the motion are omitted from IK tracking objectives, as indicated in table D.1. Many markers were only obscured at the start or end of the motion. If this was the case, the motion was cropped to the part were none were obscured, as reported in table D.2. Marker errors were calculated, in order to evaluate the quality of the IK solution.

The motions resulting from IK are low-pass filtered with a cut-off at 8 Hz (4th-order Butterworth). EMG is band-pass filtered between 30 and 300 Hz (2nd-order Butterworth), then rectified and filtered again with a low-pass at 8 Hz (4th-order Butterworth). All filtering is zero-lag. Finally, EMG is also normalized with respect to the highest value found during MVC trials or any of the normal gait trials available, as shown in table C.4. Motion and EMG data is resampled at 60 Hz, before it is evaluated by the RMR solver. All filter settings are chosen to be as similar as possible to the work of Princelle *et al.* [63] [63]. The sampling frequency of 60 Hz is chosen as a compromise between accuracy and computation time.

Ground reaction forces on the instrumented side are applied to the foot, presented by the *calc* body inside the model. Since the models include only the instrumented leg, the ground reaction forces on the non-instrumented side are applied to the pelvis instead, with the point of application still at the center of pressure. Because analysis will focus on the stance phase of the instrumented leg, the impact of the simplification of GRF application for the other leg is deemed minimal, as this GRF will be zero for most of this time, because the leg is in swing. Ground reaction forces are also used to determine the time ranges for stance phases. These ranges are detected automatically for each force plate, by finding the first and last times at which the measured vertical force exceeds 100 N, then increasing the range with 30 ms at the start and end. The stance phase detection is performed before the low-pass filter is applied, because filtering causes oscillatory artifacts before force onset (i.e. the filtered force has a negative peak before unfiltered force sets in). The 100 N threshold allows to ignore measurement noise up to 100 N, while the 30 ms shift will compensate to include the very start and end of the stance phase (with GRF between 0 and 100 N).

All muscle actuators and the reserve actuators receive weights of 1 and 10, respectively. The coordinate actuators that are set as reserves have an optimal force of 1 N(m). Reserve actuators that control coordinate which cannot be fully actuated by muscles get weights of zero. Table C.5 presents the cost function weights for all actuators. The activation dynamics constraint is active, but the GH-JRF constraint is disabled because the lower extremity model does not contain a glenohumeral joint.

For the gait simulations, no tracking weight sensitivity analysis is performed. Instead the best tracking weight found in shoulder simulations, as described in section 2.2.3, is used, after it is scaled linearly with respect to total number of muscle actuators.

As described in *Passive forces* in section 2.3.1, we consider the models without passive forces (specification A) as well as those with adjusted passive fiber force curves (specification B). Baseline simulations, without EMG-tracking and only using the lowest effort objective, will be run on both specifications for all trials and all subjects. Then, the most suitable specification will be used to investigate EMG-tracking.

2.3.4. Knee load estimation analysis

Estimated joint contact force is compared to measured joint contact force using Root Mean Square Error (RMSE) and coefficient of determination (R^2), which are computed as shown in equations 2.9 and 2.10, respectively. Also ZNCC at zero time shift is calculated, as presented in equation 2.8, but substituting activation and EMG with estimated and measured joint contact forces. RMSE and R^2 are calculated in order to compare to the results from Princelle *et al.* [63]. ZNCC is chosen for improved trend-over-time analysis. Differences between estimations can also be presented in RMSE or ZNCC, substituting measured joint contact force with another estimation.

$$\text{RMSE} = \sqrt{\frac{1}{N_k}\sum_{k=0}^{N_k}\left[F_k - L_k\right]^2} \tag{2.9}$$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{2.10}$$

$$\text{with} \quad SS_{\text{res}} = \sum_{k=0}^{N_k}\left[F_k - L_k\right]^2$$

$$\text{and} \quad SS_{\text{tot}} = \sum_{k=0}^{N_k}\left[L_k - L_{\text{mean}}\right]^2$$

- N_k : number of time steps
- F_k : estimated joint contact force at time step k
- L_k : measured joint load at time step k
- $_{\text{mean}}$: mean over the whole time series
- $_{\text{std}}$: standard deviation over the whole time series

3 Results

3.1. Verification

RMR solver revision

The revised Python implementation achieves similar results compared to the previous Python implementation by Beck *et al.* [69]. For a loaded shoulder flexion trial (*Flx21*), the MAE between muscle activation predictions of the two implementations is below 0.001 for all muscles, as can be concluded from figure D.1.

EMG-tracking and the effect of tracking weight

For all eleven tracking tasks, MAE of the tracked muscle decreases and ZNCC increases, with respect to the estimations without tracking. For higher tracking weights, this effect is stronger. Using a tracking weight of $w_2 = 1$, *LatissimusDorsi_M* still has a MAE of 0.10 and *TeresMajor* has a ZNCC of -0.54. For a tracking weight of $w_2 = 3$, all muscles have a MAE < 0.10 and a ZNCC > 0.50, which is deemed as sufficient tracking strength. Therefore $w = 3$ is used for subsequent analysis of the shoulder model. A full overview of the effect of EMG-tracking on the muscle activation of the tracked muscles for different weights is given by figures D.2 and D.3.

Effect of sampling frequency

Simulation of a single loaded forward flexion trial (*Flx21*) at 10 and 100 Hz did not produce significantly different muscle activation patterns (MAE < 0.005), justifying the use a sampling frequency 10 Hz instead of 100 Hz for shoulder movement simulations.

3.2. EMG-tracking in shoulder movement simulations

Validity

The optimizer of the RMR solver converged at almost all time instances and never failed for many consecutive time steps. Penalized reserve actuators are rarely used. The computation speed for the RMR solver using the shoulder model is around 3-5 Hz.

Muscle activation estimations

Table 3.1 summarizes the effect of EMG-tracking a single muscle on the estimation accuracy of other muscles. The best improvement in MAE that occurs in another muscle for any tracking tasks is -0.02 for the *DeltoideusClavicle_A* while tracking *DeltoideusScapula_P*. The maximum improvement in ZNCC, for any muscle that has an activation higher than 0.1, is +0.28 for *DeltoideusScapula_P* while tracking *DeltoideusClavicle_A*. However, in general muscle tracking tasks do not impact muscle recruitment of other muscles. The absolute values for MAE and zero-normalized cross-correlation without using the tracking objective can be found in figure D.4 and D.5, respectively. Figures D.6-D.11 show the full effect of tracking tasks on activation estimations of other muscles, upon which table 3.1 is based.

Tracking task	**Best improvement MAE**			**Best improvement XCORR**		
	Flexion	**Abduction**	**Shrugging**	**Flexion**	**Abduction**	**Shrugging**
DeltoideusClavicle_A	0	**-0.01**	0	**+0.10**	**+0.28**	**+0.26**
DeltoideusScapula_M	0	0	0	0	**+0.02**	0
DeltoideusScapula_P	**-0.01**	**-0.02**	0	**+0.05**	**+0.08**	**+0.02**
TrapeziusScapula_S	0	0	0	**+0.01**	**+0.01**	**+0.11**
TrapeziusScapula_M	0	0	0	**+0.05**	**+0.03**	**+0.17**
TrapeziusScapula_I	0	0	0	**+0.01**	**+0.02**	**+0.06**
Infraspinatus_S	0	0	0	**+0.01**	**+0.02**	0
SerratusAnterior_M	0	0	0	**+0.06**	**+0.01**	0
LattissimusDorsi_M	0	0	0	**+0.02**	**+0.02**	**+0.01**
PectoralisMajorClavicle_S	0	0	0	**+0.03**	0	**+0.05**
TeresMajor	0	0	0	**+0.06**	**+0.02**	**+0.07**

Table 3.1: Best absolute improvement (compared to no tracking) for any other muscle besides the one tracked. Muscles with EMG peak activation below 0.1 are not considered for best improvement of cross-correlation. Values of 0 are 0.00 and non-zero values are highlighted in bold.

3.3. EMG-tracking in gait simulations

Inverse kinematics

Inverse kinematic mean marker RMSE are all below 20mm, see table D.2. For the trials for challenge 5 these errors are all below 10mm. All marker tracking tasks have equal weights. Table D.1 shows which markers are tracked.

EMG-tracking weight

The best tracking weight found for shoulder simulations is linearly scaled for use with the lower extremity model, which has 40 muscle actuators instead of 33. Thus, a weight of $w = 3 \times \frac{40}{33} = 3.64$ is used for the gait simulations.

Model specification choice

Calibration of passive fiber force curve parameters have lead to a model for which the onset and progression of passive forces is shifted to higher muscle strains. Figure D.12 gives an overview of the optimized curve parameters. As suspected the original model causes extreme passive joint torque. Through calibration, the passive torques of the *B-specification* model inhibit more realistic passive joint torques, which is presented in figure D.13.

Figure 3.1 shows a comparison for knee JCF between estimations from both specifications against knee implant ground truth measurements. It considers the stance phase of the instrumented leg, averaged over all trials. The knee JCF is normalized to bodyweight (BW) of the subject.

When we look at the metrics, RMSE is lower for estimations without passive forces for challenge 4 (0.50 instead of 0.78 BW) and 5 (0.27 instead of 0.44 BW). However, ZNCC for challenge 4 with calibrated passive forces is 0.91, compared to only 0.60 without passive forces. Based on lower RMSE scores, it is decided that further evaluations will be performed without passive forces.

In all curves in figure 3.1 we can recognize two peaks, one at around 20-30% of stance and another at 70-80%, with a valley in between. Considering the blue lines, which represent simulation without passive forces, we see that for challenge 4 estimations without passive forces show peaks within 0.2 BW, but underestimate forces at the interval in between by almost 0.9 BW. The second peak in challenge 5 is within 0.1 BW, while the first peak is overestimated with 0.3 BW and the valley is underestimated with 0.1 BW. In Challenge 6, knee loads are underestimated by 0.3-0.8 BW across the whole stance. This indicates that there is room for improvement of the accuracy of knee JCF estimations.

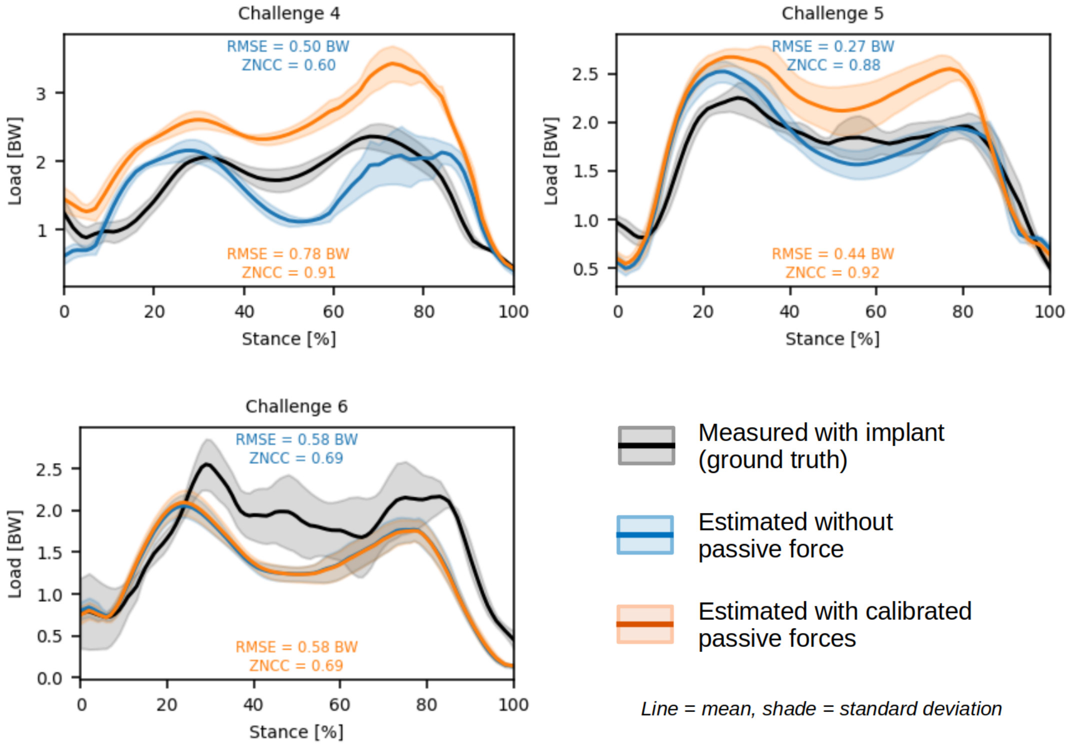


Figure 3.1: Comparison of knee joint contact force estimations from different passive force models. In **black** the implant measured knee joint contact force (ground truth), in **blue** the estimations without passive muscle forces and in **orange** the estimations with calibrated passive forces, as described in 2.3.1. Forces are normalized for bodyweight. One stance of the instrumented leg is taken from each trial. RMSE and ZNCC with respect to the ground truth are reported as well.

Validity

The optimizer within the RMR solver converged for the majority of the stance phase for all challenges and trials. The penalized reserve actuators are also rarely used. Only in challenge 4 around the second peak (75% stance), some convergence issues occurred, but only ever for less than 0.1s, thus these trials were not invalidated. In the situations where the solver did come to a solution, it often included reserve actuators usage (hip and knee 2-5 Nm, ankle 40 Nm). Around this same time in the simulation, many muscles are estimated to fully activate, see figure D.15. The computation speed for the RMR solver using the lower extremity models is around 5-10 Hz.

Knee joint contact force estimations

Figure 3.2 and 3.3 show the effect of tracking tasks on knee JCF prediction accuracy. The depicted change is with respect to the RMSE and ZNCC for the estimations without tracking (baseline estimations). The mean change is calculated as follows: for each trial a single RMSE and ZNCC value is calculated. Then, the relative change of this value with respect to the baseline estimation is taken. All these relative changes are averaged over trials within a challenge. Finally these averages (depicted as markers in figures 3.2 and 3.3) are averaged with all challenges weighed equally.

On average tracking *glmed1* increases the accuracy of the knee JCF estimation the most (RMSE -8%, ZNCC +4%). *glmed1* is the most anterior of three Gluteus Medius elements and is responsible for hip abduction, see figure 3.4. Figure 3.5 shows the improved estimations per challenge over the stance phase. Most of the improvement is made for challenge 6, in which the tracking task estimates higher knee loads at the first peak, which was initially underestimated. All other challenges show only minor improvements. The tracking of *glmed1* for each challenge is shown in figure 3.6.

If we consider the means of individual challenges, tracking *vasmed* (RMSE -20%, ZNCC +3%), *semimem* (RMSE -10%, ZNCC +9%) and *recfem* (RMSE -6%,ZNCC -9%) perform well for challenge 6, see figures 3.7. In challenge 6, knee JCFs are underestimated without tracking. Tracking *vasmed* increases estimates for the second knee load peak as well as the interval between peaks, tracking *semimem* mainly increases the first peak estimate, while tracking *recfem* increases the first peak and interval estimates. However, it must be noted that EMG for *vasmed* and *semimem* are normalized to maximum values found in gait trials, not MVC trails, meaning that is likely that these EMG values overestimate muscle excitation.

This means tracking *recfem* might provide the most legitimate results. Figure 3.8 shows that in estimations without tracking, *recfem* is inactive. However, EMG from Rectus Femoris indicates significant activation around 20% stance phase, which corresponds to the first peak in knee loading. Some amount of this activation remains up till 60% of stance. Through tracking Rectus Femoris EMG with *recfem*, the estimated activation of *recfem* as well as *semimem* is increased between 15 and 60% of stance.

Tracking all muscles with available EMG significantly overestimates knee JCF: RMSE increases with 147% while ZNCC decreases with 27%. If we only track the muscles that Princelle *et al.* [63] also tracks, RMSE increases with 144% and ZNCC decreases with 29%.

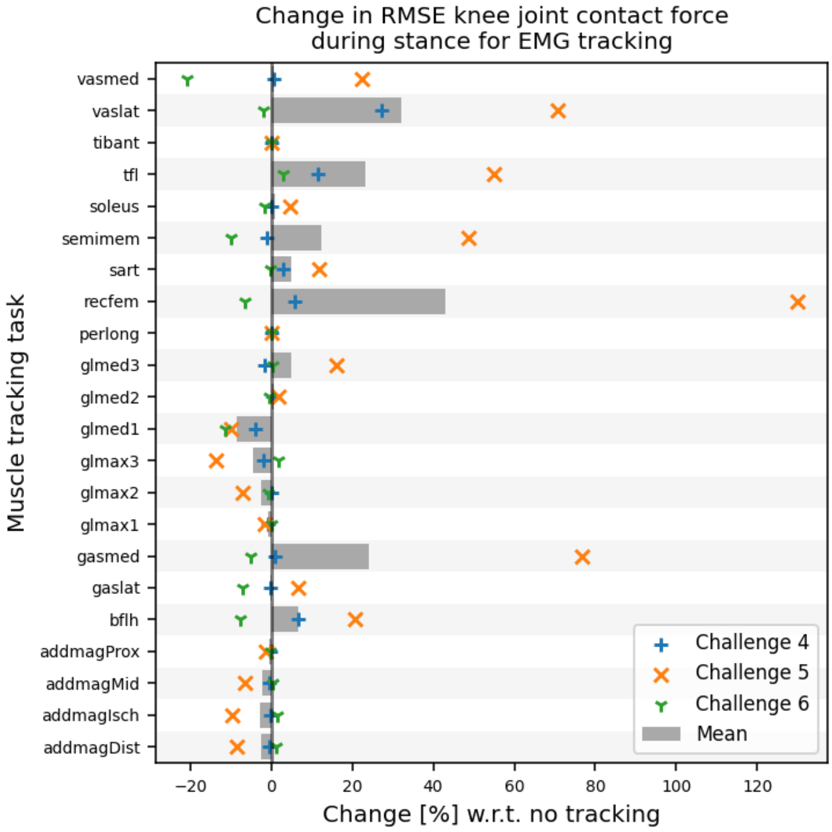


Figure 3.2: Percentage change in knee joint contact force RMSE when tracking individual muscles (compared to no tracking). Markers indicate mean change for all trials within the same challenge, while the gray bar represents the mean across all challenges. Only the stance phase is considered.

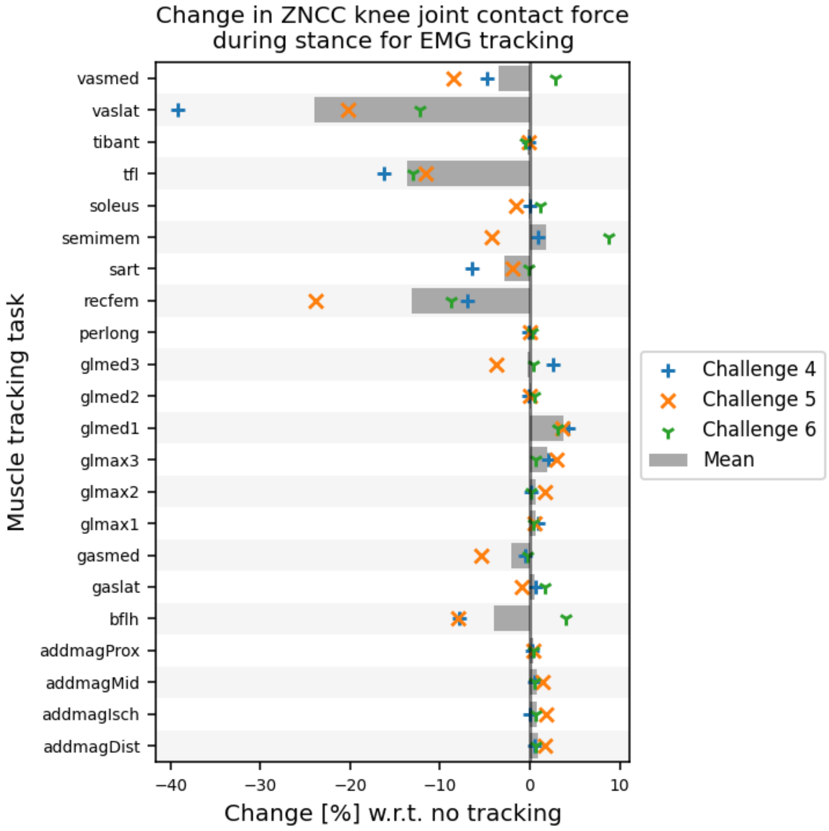


Figure 3.3: Percentage change in knee joint contact force ZNCC when tracking individual muscles (compared to no tracking). Markers indicate mean change for all trials within the same challenge, while the gray bar represents the mean across all challenges. Only the stance phase is considered.

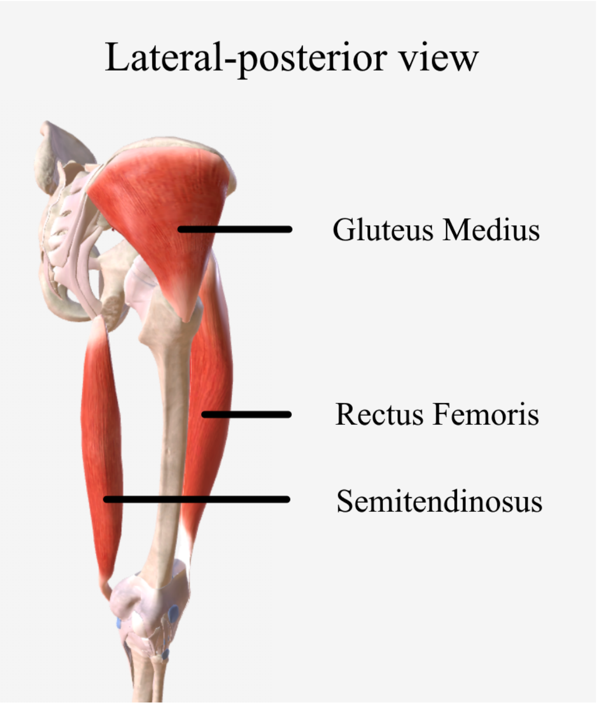


Figure 3.4: The location of a few important leg muscles. Gluteus Medials is used for hip abduction, the Rectus Femoris for hip flexion and knee extension, and the Semitendinosus for hip extension and knee flexion.

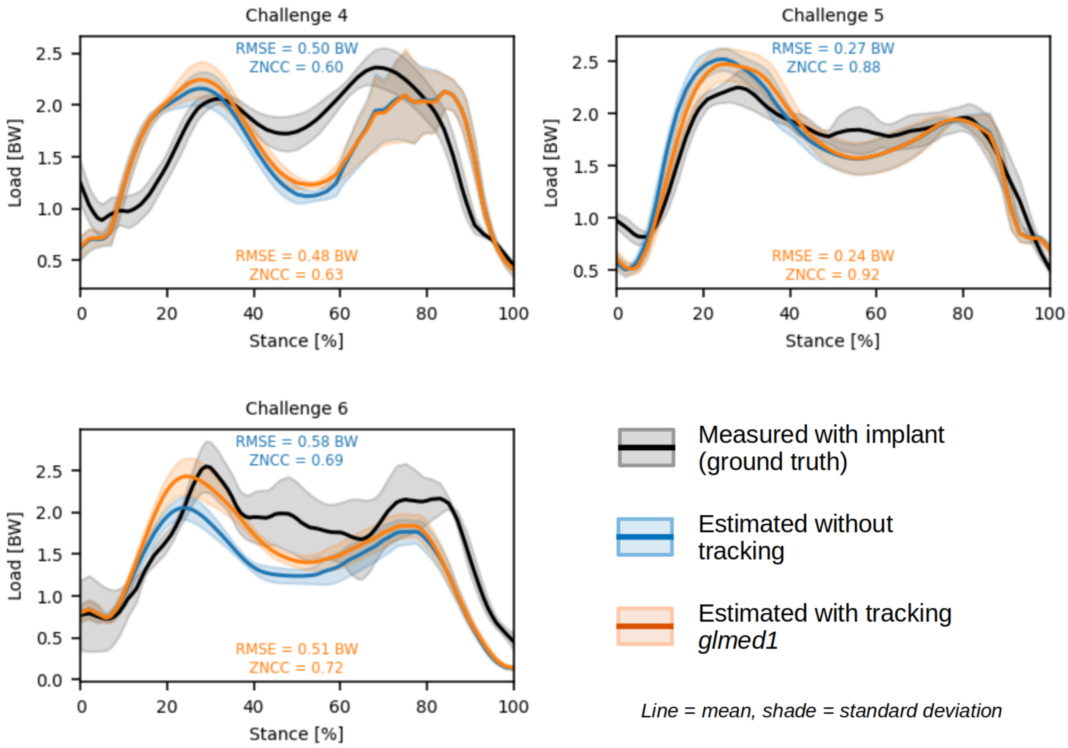


Figure 3.5: Comparison of knee joint contact force estimations with and without tracking *glmed1*. In **black** the implant measured knee contact force (ground truth), in **blue** the estimations without tracking and in **orange** the estimations with tracking of *glmed1*. Both estimations ignore passive forces. Forces are normalized for bodyweight. One stance of the instrumented leg is taken from each trial. RMSE and ZNCC with respect to the ground truth are reported as well.

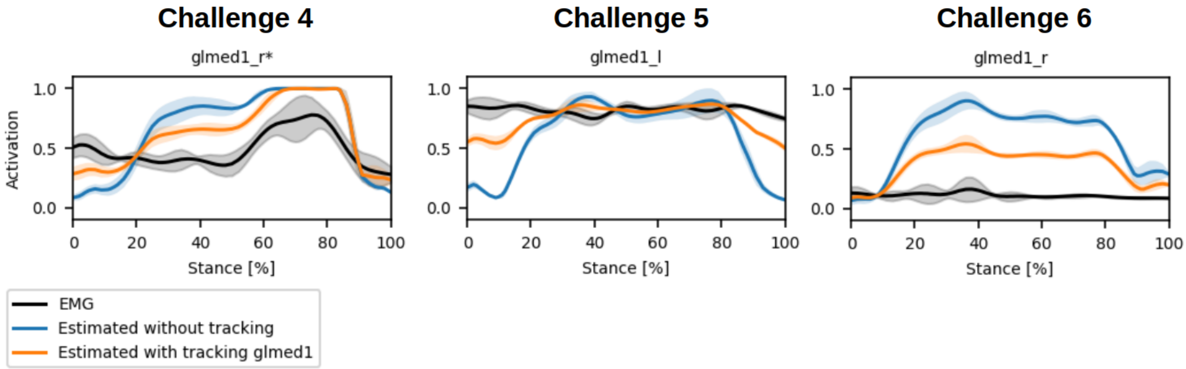


Figure 3.6: Comparison of muscle activation estimations and EMG with and without tracking *glmed1*. In **black** is EMG, in **blue** the estimations without tracking and in **orange** the estimation with tracking *glmed1*. Estimations are means over all trials for the indicated challenge. One stance of the instrumented leg is taken from each trial. * indicates that EMG of *glmed1* for challenge 4 is normalized with a value found in a gait trial.

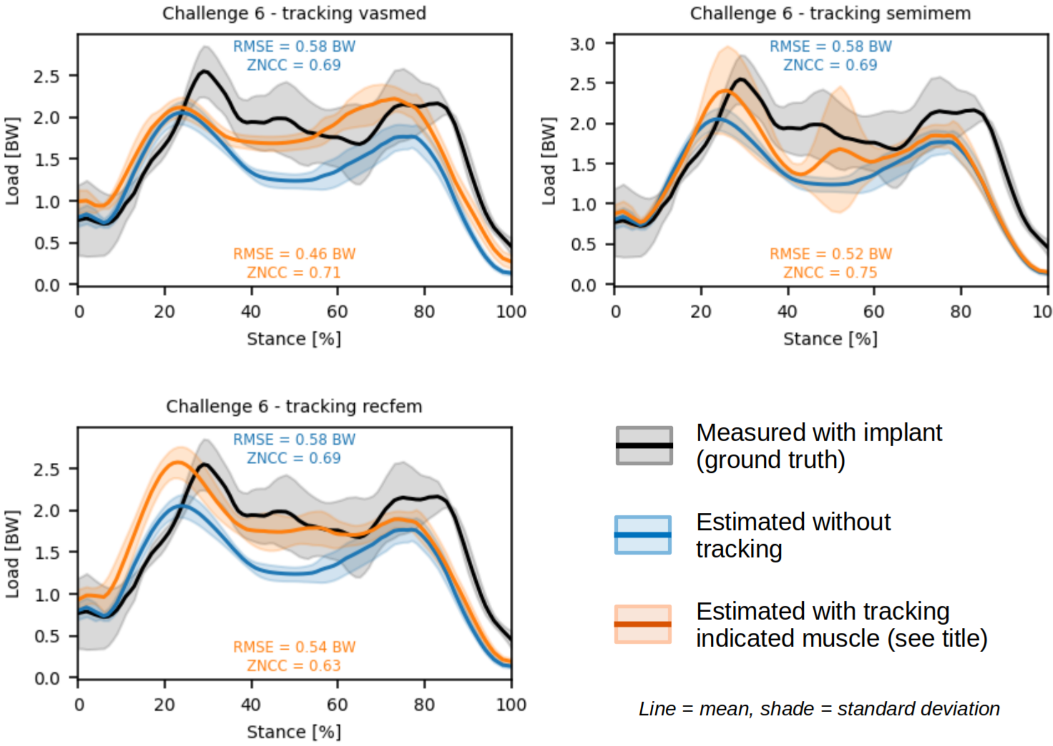


Figure 3.7: Comparison of knee joint contact force estimations with and without tracking *vasmed*, *semimem* or *recfem* for challenge 6. In **black** the implant measured knee contact force (ground truth), in **blue** the estimations without tracking and in **orange** the estimations with tracking the muscle indicated in the titles. All estimations ignore passive forces. Forces are normalized for bodyweight. One stance of the instrumented leg is taken from each trial. RMSE and ZNCC with respect to the ground truth are reported as well.

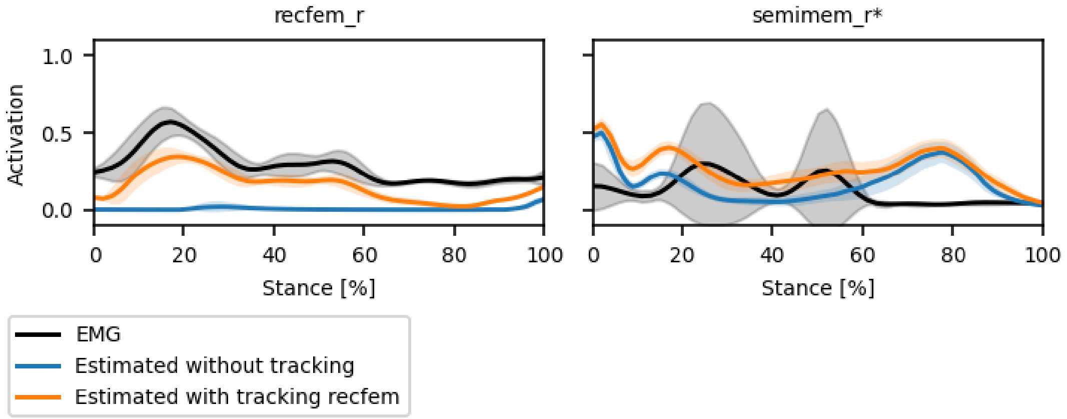


Figure 3.8: Comparison of muscle activation estimations with and without tracking *recfem*. In **black** is EMG, in **blue** the estimations without tracking and in **orange** the estimation with tracking *recfem*. Estimations are means over all trials for challenge 6. One stance of the instrumented leg is taken from each trial. * indicates that EMG of *semimem* is normalized with a value found in a gait trial.

Participant	Trial	RMSE [BW]				
		SO	CEINMS	RMR	RMR	RMR
		[63]	[63]	No tracking	Tracking all	Tracking *glmed1*
KGC 4	og2	0.41	0.43	**0.38**	0.66	**0.38**
	og3	**0.46**	0.50	0.64	0.95	0.62
	og4	0.51	**0.35**	0.43	0.59	0.38
	og5	**0.45**	-	0.52	0.83	0.51
	og7	0.52	**0.37**	0.52	0.90	0.50
KGC 5	og1	**0.27**	-	0.31	1.24	0.28
	og8	0.34	**0.21**	0.26	1.13	0.26
	og9	0.33	0.25	0.25	1.17	**0.21**
	og11	0.23	**0.21**	0.26	1.31	0.22
KGC 6	og4	**0.28**	0.46	0.39	0.68	0.36
	og5	**0.40**	**0.40**	0.57	0.67	0.54
	og6	0.67	**0.53**	0.77	1.00	0.62
	og7	0.56	**0.51**	0.57	0.66	0.53
	og9	**0.47**	0.62	0.58	0.74	0.51

Table 3.2: Comparison of knee joint contact force estimation accuracy between methods, measured using RMSE. Only the stance phase is considered. The most accurate estimation per trial (row) is highlighted in bold.

Participant	Trial	R^2				
		SO	CEINMS	RMR	RMR	RMR
		[63]	[63]	No tracking	Tracking all	Tracking *glmed1*
KGC 4	og2	0.70	**0.93**	0.40	-0.82	0.40
	og3	0.71	**0.74**	-0.07	-1.35	-0.01
	og4	0.47	**0.79**	0.29	-0.34	0.45
	og5	**0.58**	-	-0.06	-1.74	-0.03
	og7	0.67	**0.83**	0.32	-1.08	0.36
KGC 5	og1	**0.89**	-	0.51	-7.08	0.60
	og8	0.88	**0.93**	0.71	-4.39	0.71
	og9	0.88	**0.94**	0.75	-4.73	0.82
	og11	0.88	**0.90**	0.66	-7.33	0.76
KGC 6	og4	**0.88**	0.78	0.47	-0.58	0.57
	og5	**0.81**	0.76	-0.23	-0.71	-0.12
	og6	0.50	**0.67**	-0.43	-1.44	0.06
	og7	0.66	**0.73**	0.37	0.16	0.46
	og9	**0.80**	0.60	0.25	-0.20	0.43

Table 3.3: Comparison of knee joint contact force estimation accuracy between methods, measured using R^2. Only the stance phase is considered. The most accurate estimation per trial (row) is highlighted in bold.

Comparison to SO and CEINMS

Table 3.2 and 3.2 show comparisons of knee JCF estimation accuracy for different methods, including RMR without tracking, RMR with tracking *glmed1*, RMR with tracking all muscles, CEINMS [63] and SO [63]. Compared to CEINMS and SO, the RMR solver without tracking only achieves lower RMSE for 1 of 14 trials and with tracking *glmed1* only for 2 of 14 trials. However, if we consider the averages across trials per challenge, disregarding CEINMS calibration trials, RMR with tracking *glmed1* has the second lowest RMSE score (after CEINMS) for challenges 4 and 5. For challenge 4, CEINMS has a RMSE of 0.41 BW, RMR tracking glmed1 of 0.47 BW, SO of 0.48 BW and RMR of 0.59 BW. For challenge 5 the respective scores are 0.22, 0.23, 0.30 and 0.26 BW. For challenge 6 SO has the lowest RMSE (0.48 BW), followed by CEINMS (0.50 BW), RMR tracking *glmed1* (0.51 BW) and RMR (0.58 BW). RMR with tracking all muscles has significantly higher RMSE than all other methods for all trails.

All RMR solver variants have a lower R^2 scores than SO and CEINMS for all trials.

4 Discussion

4.1. RMR solver and EMG-assisted cost function implementation

The main goal of this thesis is to add an EMG-assisted cost function to the RMR solver and verify its functionality. A revised RMR solver is created to ease implementation of this new functionality. The revised RMR solver is verified by comparing it to the original Python implementation by Beck *et al.* [69], which yields virtually the same results for the same simulation conditions (MAE < 0.001). The EMG-assisted cost function also works as intended: estimations for tracked muscles show decreased MAE and increased ZNCC when compared to EMG. As expected, this effect is stronger for higher tracking weights.

One other study by Sarshari *et al.* [54] also implements an EMG-assisted cost functions together with a glenohumeral stability constraint. However, there are some notable differences between that existing method and the implementation in this thesis. Firstly, in Sarshari *et al.* [54] a lowest effort criterion is based on the sum of squared muscle stress is used instead of the sum of squared muscle activation. Secondly, the EMG assistance is implemented as a constraint instead of an objective. This constraint enforces estimated activation to be within a certain margin of EMG, thus reducing the solution space for the optimizer. The EMG-assisted cost function itself is not novel and similar ones have been used, for example by Sartori *et al.* [55]. However, the integration of the EMG-assisted cost function alongside the GH-JRF constraint novel, to the knowledge of the author.

Limitations of implementation

Regarding the RMR solver revision, several limitations apply. The EMG-tracking objective implemented in this thesis compares EMG to activation, without accounting for muscle activation dynamics. Activation dynamics causes a delay between excitation and activation change of 10 to 50 ms [68]. The impact of this delay is minimal for slow movements, including the movements in this thesis, but is more significant for faster movements. For simplification, activation dynamics are disregarded for the EMG-tracking objective in this thesis. However, other methods such as CEINMS [47] or a method by Assila *et al.* [49] do account for activation dynamics for the EMG-tracking objective. They do this by defining the EMG tracking error as the difference between EMG and estimated excitation, instead of activation. Every set of excitations that the optimizer comes up with needs to be processed to activation, in order to know the impact on joint moments. There are several options for implementing activation dynamics in the EMG-tracking objective of the RMR solver. One option is to process the EMG data further to an signal that represents muscle activation, which is also done in an EMG-driven method by Tahmid *et al.* [77]. This would also be possible live, since EMG change would precede activation change. Another option is to optimize for muscle excitation instead, similar to above-mentioned methods [47], [49]. This would involve replacing the optimization parameter bounds from activation bounds (equation 2.3) to excitation bounds, as well as modifying the acceleration constraint (equation 2.2) to use muscle excitation. The activation dynamics equations used in the RMR solver [59], [78] would then be used to construct a new $\mathbf{A_k}$ which captures the effect of muscle excitation, instead of activation, on joint moments.

Another limitation of the current implementation is the lack of flexibility of mapping EMG signals to multiple muscle actuators in the model. This mapping is done on initialization of `EMGmanager` and can only be changed by adding a new case inside `__init__`. CEINMS features a generalized way, called *Neural Mapping*, to map any linear combination of EMG signals to any muscle element. The RMR solver could also use such a generalized implementation of neural mapping, which could be done by extension of the `EMGmanager` class.

A more general limitation to the revised RMR solver, is that all manager classes share some of the same functionality, such as filtering, sampling and cropping, which leads to unnecessary duplicate code. The cause is a long development process, in which this shared functionality was not foreseen. A new revision might collect these functions in a manager superclass, with data-specific functionality implemented in subclasses. This will reduce duplicate code, improving maintenance and modularity, which are the main advantages of a well made class-based code base.

4.2. Shoulder movements results

The secondary objective of this thesis is to investigate the effect of EMG-tracking on muscle activation. To this end, muscle recruitment is estimated for simple shoulder movements while tracking a single EMG signal. With EMG as ground truth, MAE and ZNCC of all tracking tasks is compared to the MAE and ZNCC without tracking. While the effect of tracking only a single muscle at the time is expected to be low, results indicate very minimal impact. Most tracking tasks do not improve MAE (change < 0.005) of any other muscle. Similarly, ZNCC is only improved by 0.03 or less for any muscle, most of the time. It seems that redundancy of this model is so high, that changes in muscle force of one muscle can be compensated by only slight changes in muscles force of multiple other muscles. Because single tracking tasks have minimal impact, it is expected that many tracking tasks at the same time are necessary for a significant impact, given this model and data. Combinations of multiple tracking tasks were not simulated, because many combinations exist. No attempt was made to qualitatively inspect the lowest effort estimations together with the available EMG signals, to come to a selection of EMG signals to track.

It must be noted that simulations using only the lowest effort objective already give good estimations for muscle recruitment using this model and data, with most MAEs below 0.01 as can be seen in figure D.4. This indicates that co-contraction levels are low in the measured muscles, in which case EMG-tracking seems less likely to significantly impact muscle recruitment estimations. Contrary, the MAEs of estimations without tracking found in this study are slightly different, up to 0.03, to the ones reported by Belli *et al.* [59], despite using the same solver (verified), the same model, the same data and replicating the same filter settings. Based on verification of the first Python RMR solver by Beck *et al.* [69] and of the revised Python RMR solver in this thesis, the expected MAE deviations of this implementation to the original MATLAB implementation should be less than 0.01, but this is not the case. One possible explanation is that movement time was normalized in different ways, which could alter the contributions of movement phases to the MAE.

While the RMR solver did not converge for around 2% of the time steps, this never happened for many consecutive steps or for many steps in the same trial, therefore the impact on the results is deemed minimal.

Limitations shoulder simulations

The data that was used includes only one subject, three movement types and a low levels of co-contraction in superficial muscles, based on predictions within 0.1 MAE by the lowest effort criterion. The original purpose of the data was to investigate muscle contributions in the shoulder using a new model in combination with SO and CMC, both with lowest effort objectives [22]. This experiment therefore did not focus on co-contraction, but rather on the biomechanics of the scapular and glenohumeral joints, hence the inclusion of the shrugging motion in the experiment. Because of the low trial count as well as the small changes between tracking conditions, no significance was computed for these results.

Another limitation in this study is that the effect of tracking multiple EMG signals at the same time is not explored. Simulating all combinations of two, three or even more EMG-tracking tasks at the same time, would take a considerable amount of time and generate a large set of results. Only testing certain sets of tracking tasks could introduce bias, thus it would require good intuition or qualitative assessment to find sensible combinations. It seems likely that the effect of tracking tasks would stack when used simultaneously. Since the positive effect of single tracking tasks found in this thesis is so small, we expect many signals (e.g. four) need to be tracked for significant impact.

Only tracking weights of 1, 3 or 5 were investigated (considering all other weights as in table C.3) and a weight of '3' was chosen for all tracking tasks based on tracking performance. However, different weights for different EMG signals should be investigated, as well as a finer selection of weights, e.g. 1-10 with intervals of 0.5, because EMG signals that were already close to estimations based on lowest effort require less weight to track appropriately.

4.3. Gait results

The third goal of this thesis is to analyze the effect of EMG-tracking on knee joint contact forces. Data from Knee Grand Challenges 4, 5 and 6 are used together with models from Princelle *et al.* [65]. Passive forces are ignored to prevent unrealistically high muscle forces, e.g. in the Gastrocnemius Medialis and Lateralis, and simulations are run while tracking one EMG signal at the time. For all tracking tasks MAE and ZNCC for knee load estimations, with measured implant force as ground truth, are compared to estimations without tracking. Relative changes in MAE and ZNCC indicate some significant effects of EMG-tracking on knee JCF estimations. Various tracking tasks decrease RMSE or ZNCC, although usually not for all challenges.

Tracking the Gluteus Medius EMG signal with *glmed1* delivers the most improvement on average over all challenges. When inspecting the knee JCF curves (figure 3.5), the most significant improvement is made for challenge 6, where tracking glmed1 increases the first peak estimation from 2.05 BW to 2.45 BW, with ground truth being 2.55 BW. In order to understand how this increase in accuracy is achieved, we look into the tracked signals with more detail, shown in figure 3.6. The estimated activation of *glmed1* across all three challenges is around 0.8 or higher without tracking, which does not seem realistic for a low intensity activity. In challenge 6, the Gluteus Medius EMG signal is around 0.1, which is more as expected. This causes EMG-tracking to reduce *glmed1* activation, which indirectly causes higher knee JCF. Consequently the estimation improves, because the lowest effort knee JCF estimation underestimated knee loads in challenge 6. However, the legitimacy of the improvements for challenge 4 and 5 is questionable. The processed Gluteus Medial EMG signal is surprisingly high for challenge 4 and 5 (0.5-0.7 and 0.9 respectively), while challenge 6 suggests that *glmed1* remains inactive during stance. Visual inspection of the raw EMG for challenge 5, reveals a constant signal with noise, which also suggests muscle inactivity. It seems likely that bad normalization causes the processed EMG signal of challenge 4 and 5 to be high. Therefore, any effect of tracking *glmed1* in these challenge is likely artificial.

Knee load estimation in challenge 6 can also be improved by tracking other muscles, such as *vasmed*, *semimem* or *recfem*, see figure 3.7. All these tracking tasks increase load estimations. Because the lowest effort estimations in challenge 6 underestimate knee JCF, increased estimations can lead to lower RMSE. This suggests that subject DM in challenge 6 uses more co-contraction during gait. The high standard deviation in implant force data during stance also indicates that subject DM does not have a steady gait pattern. These ongoing adaptations of gait are likely to be related to increased co-contraction.

An example of legitimate EMG-assistance is tracking *recfem* in challenge 6. The Rectus Femoris EMG signal causes increased *recfem* activation during the first knee load peak, which would otherwise be inactive, see figure 3.8. Consequently, the activation of the antagonist *semimem* is increased, to balance the induced torque around the knee. Because both muscle elements wrap the knee joint (figure 3.4), their increased activation directly contributes to increased knee JCF at the first peak. This increased knee load estimation is closer to the measured knee JCF as can be seen figure 3.7. This example shows how EMG-tracking can predict co-contraction and corresponding changes in knee loading.

In challenge 4 and 5 there is less room for improvement of knee JCF estimations. In challenge 4 only mid-stance and the second peak are underestimated. For challenge 5, the RMSE of the lowest effort estimation without tracking is already low (0.27 BW) and only the first peak overestimated. Higher knee load estimations for these challenges likely lead to higher RMSE. The combination of low RMSE and slight overestimation for challenge 5 leads to large relative increases in RMSE for some tracking tasks, e.g. tracking *recfem* (+130%), *gasmed* (+75%) or *vaslat* (+70%). Since EMG-assisted optimization always increases total muscle activation with respect to the lowest effort solution, it usually also increases knee JCF RMSE for challenge 4 and 5.

EMG-normalization seems crucial when it comes to the effectiveness of EMG-tracking. As is the case with the Gluteus Medius signal in challenge 5, unreliable MVC values can impact the magnitude of normalized EMG immensely. Often, EMG is visually inspected in order to exclude any suspected noisy or faulty EMG signals [16]. Preferably, there should be a way to either get more accurate MVC data for normalization or have EMG-assisted methods be less dependent on normalization. The first might be achieved through better experimental protocols, but this cannot be applied on existing data sets. Less dependency on MVC might be achieved by methods that rely more on EMG trend than absolute magnitude. Even with a badly normalized EMG signal, we know that muscle excitation will increase when EMG increases (disregarding other effects, e.g. bad electrode placement and cross-talk); only the amount of increase will be unclear. However, it might be hard to implement EMG-assistance like this by optimizing statically. Instead optimization on a window of time might be necessary.

Muscle passive forces

Per default, the RMR solver includes muscle passive forces. However, it turns out that unadjusted passive forces leads to enormous overestimation of knee JCF, mainly caused by the Gastrocnemius muscles. This has lead to the consideration of two options: (A) disabling passive forces all together, or (B) calibrating passive forces to produce realistic joint torques. The calibration process adjusted the muscle fiber force curve to an unphysiological level with most passive forces starting at a strain of 0.45, instead of 0, and reaching one normalized muscle force at 1.05 strain, instead of 0.7, as shown in figure D.12. This does not invalidate the adjusted *B-specification* model, because the adjustments do result in realistic passive joint torques, as is demonstrated in figure D.13. The passive fiber force curve is merely used to enhance the macro properties of the model. It seems likely that adjusting other muscle properties would have been more appropriate, although this would require careful considerations. Since SO and CEINMS do not model passive muscle force, Princelle *et al.* [63] will not have accounted for them while making the models.

When choosing which model to use for EMG-tracking analysis, accuracy measured with RMSE was given priority. This lead to the choice to estimate without using passive forces. However, the ZNCC of estimations using the adjusted passive forces were strictly better, especially for challenge 4 (ZNCC 0.91 vs. 0.60 without passive forces). This could indicate that passive forces are playing an important role, but that this role is not tuned right in this model. However, the results from challenge 6 weaken this statements. Because both specifications result in virtually the same estimations, it seems that passive forces do not impact knee JCF at all for this subject.

Comparison to other studies

The fourth goal of this thesis is to verify if the performance of the EMG-assisted RMR solver can achieve similar estimations as Princelle *et al.* [63] using SO and CEINMS. All RMR solver variants have lower R^2 than CEINMS and SO, as can be concluded from table 3.3. R^2 penalizes occasional larger errors more than consistent smaller errors, because it squares the prediction error. This sensistivity is probably why the RMR solver has such low R^2 values, while the RMSEs are similar to SO and CEINMS.

As can derived from table 3.2, the RMR solver with tracking *glmed1* generally has a higher RMSE than CEINMS, but lower than SO for challenges 4 and 5. This indicates that the RMR solver with EMG-tracking has potential for improving knee load estimation compared to the established SO method, although tracking *glmed1* in challenge 4 and 5 does not seem entirely legitimate, as was described in the second paragraph of section 4.3. Surprisingly, SO achieves lower average RMSE (0.48 BW) than both CEINMS (0.50 BW) and RMR with tracking *glmed1* (0.51 BW) for challenge 6. It is expected that SO would underestimate knee loads for this challenge, similar to RMR without tracking, because it likely contains more co-contraction. However, considering individual trials reveals that SO only has the lowest RMSE in 2 of 5 trials in challenge 6. Given that the measured knee loads for challenge 6 also have the highest standard deviation of all challenges, it may be plausible that SO predicts only the trials with low co-contraction and lower knee loads accurately. The CEINMS calibration trial *og3* was not used in this thesis, but according to Princelle *et al.* [63] SO has a RMSE of 0.53 BW for that trial. This is on the higher side for challenge 6, indicating possible co-contraction. Since CEINMS was calibrated on trial with more suspected co-contraction, it probably has more trouble accurately prediction the low co-contraction trails at which SO excels.

The RMR solver without tracking also has significantly higher average RMSE (+0.10 BW) for challenge

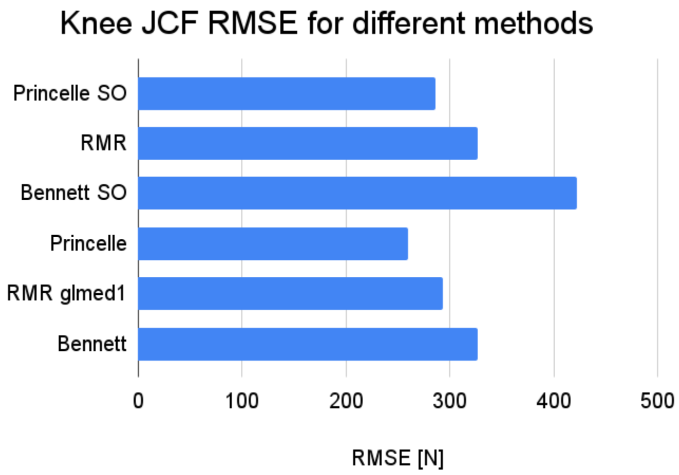


Figure 4.1: Comparison of knee joint contact force RMSE for different methods. The average RMSE of Knee Grand Challenges 4, 5 and 6 is displayed for stance phases only. Methods used in this thesis are compared to those used in Bennett *et al.* [16] and Princelle *et al.* [63]. Note that this comparison has no statistical significance and is not normalized to bodyweight.

4 and 6 compared to SO. Based on results by Belli *et al.* [59], the RMR solver should be at least be on par with SO, since it uses the same muscle model complexity and even considers muscle activation dynamics, similar to CMC. One possible explanation is that Princelle *et al.* [63] have adjusted the models further, thus that the models used in this thesis [65] are not exactly the same after all.

This brings up the importance of musculoskeletal models and their personalization, which may contribute more to estimation accuracy than EMG-tracking. Assila *et al.* [49] shows that using SO with an EMG-calibrated model estimates more stable GH JRFs, even when using SO for solving, suggesting that CEINMS may be able to compensate for models deficiencies. However, in Bennett *et al.* [16] estimations using CEINMS are worse than using SO in Princelle *et al.* [63] (RMSE 342 N vs. 286 N), see also figure 4.1. This difference might be because Bennett *et al.* [16] uses a lowest joint load cost function, while Princelle *et al.* [63] optimizes for lowest effort plus lowest EMG-tracking error. However, it seems likely that the model used also plays as significant role. Bennett *et al.* [16] uses a scaled version of the *gait2392* model, which uses older *Thelen2003Muscle* elements, a hinging knee joint and out-dated muscle properties. Contrary, Princelle *et al.* [63] has created a personalized model sharing properties to the Rajagopal2015 model [72], using more recent *Millard2012EquilibirumMuscle* elements and a more realistic rolling knee joint with moving patella. Knee JCF estimations with SO in Princelle *et al.* [63], in which the models are suspected to be more personalized than the ones used in this thesis, also seem to perform better than the best results achieved in this thesis, in particular when considering the R^2 metric, for which SO has higher scores than RMR in all 14 trials. This suggests that there is more to gain in model personalization than with EMG-assisted cost functions and that EMG-calibration, as part of the CEINMS routine, might be able to help automate personalization.

Another observation is that tracking all muscles at the same time using the RMR solver, overestimates knee JCF at the first peak with 1.0 up to 2.0, even though CEINMS also tracks all these muscles without overestimation. It must be considered that CEINMS has more elaborate mapping of EMG signals to muscles, which will help. Authors are also selective regarding which EMG to include in tracking. For example, for challenge 4, Bennett *et al.* [16] and Princelle *et al.* [63] ignored EMG for *vasmed*, *glmax* and *perlong* because of noise, as well as for *soleus* and *sart* because they were deemed faulty. Still, discarding those same muscles and only tracking remaining ones using the EMG-assisted RMR solver, results in similar knee JCF overestimation. Likely, EMG-tracking only works adequately when the model is calibrated to similar EMG. Calibration might even be experiment specific. While EMG-tracking is not performed on the same trials as EMG-calibration, it is still the same experiment. It would be interesting to investigate if CEINMS calibration for one experiment would be effective for subsequent experiments with the same subject. If this is the case, it would confirm that CEINMS calibration enhances subject specificity and not merely experiment specificity.

Limitations gait simulations

Some of the same limitations mentioned for the shoulder simulations also apply for the gait simulations. This includes the low number of trials, meaning statistical significance is not considered. While this data includes three subjects instead of one, only challenge 6 seems to contain signification co-contractions, based on underestimation of knee loads with the lowest effort criterion. The increased muscle activity is probably because of the old age of the subject and its medical history concerning the knee replacement. Similarly to the shoulder simulations, tracking multiple EMG signals at the same time is not explored and weight factors are taken the same for every tracking task.

Because the models include only the instrumented leg, the GRF from the other leg is applied to the pelvis. The point-of-application still lies at the center of pressure, which should account for any moments that the GRF applies. Since we only consider the estimations of the stance phase of the instrumented leg, during which the GRF for the other leg would be mostly zero, the impact of the simplification is deemed small. Nevertheless, a two-legged model with both GRF applied at the feet may result in more accurate estimations.

Muscle recruitment, and consequently knee JCF, estimations for challenge 4 are less valid, because reserve actuator forces are needed to generate the inverse dynamic accelerations. In addition, several muscles are maximally activated during the second half of the stance phase. This indicates that the model for the challenge 4 subject is likely not strong enough.

Models of both specifications exhibit unexpected jumps in muscle fiber length in some muscle elements. For example the *gasmed* in the model for challenge 5 has visible wrapping error around -5° knee angle, but also invisible errors around -32° and -56°, see figure D.16. During the visible wrapping error, the muscle element wraps around the wrong, i.e. unintended, way of the wrapping surface, which can be caused by a modeling mistake or bug in the wrapping function. The other jumps, however, cannot be explained. The impact of these muscle wrapping discontinuities could be significant, but further analysis would be required to capture the full scope of its impact.

4.4. Implications and recommendations

The revised RMR solver with EMG-assisted cost function created in this thesis is meant to be a new tool to use in state-of-the-art musculoskeletal simulations. The combination of an EMG-assisted cost function together with the GH stability constraint makes the tool especially suited for shoulder stability studies. The revised RMR solver has already been adopted by fellow researches, because it is more practical to set up and customize than the previous implementation, even if EMG-assistance is not used. Therefore, the revised RMR solver promises to aid future musculoskeletal research.

Recommendations regarding the RMR solver are related to improving the modularity and expanding function flexibility. The manager classes might be fused into one, in order to reduce code redundancy, which can be achieved by creating one manager super class. Plotting functionality could be improved and moved from `MotionAnalysis` into a separate class in order keep classes clear in their purpose. More elaborate EMG mapping functionality could be added to allow muscles to track any combination of EMG signals. For example, a muscle might track half of signal 1 and half of signal 2 ($a = 0.5e_1 + 0.5e_2$). Or an EMG signal can be tracked by the sum of three muscles ($a_1 + a_2 + a_3 = e$).

While EMG-tracking can reduce MAE and increase ZNCC of other muscles, or reduce RMSE and increase ZNCC of knee loads, none of these findings are significant yet. However, the potential of EMG-tracking few signals in knee JCF estimation is demonstrated using examples, which is especially promising in gait with suspected co-contraction (challenge 6), meaning it might be of great help in research with knee osteoarthritis patients. But factors such as EMG normalization and model personalization are also crucial for success.

Several interesting research topics might be pursued to improve estimations more. Integrating activation dynamics with the EMG-tracking might allow more accurate tracking, by accounting for EMG delay. Tracking multiple EMG signals at the same time can be investigated. It is expected that the impact of tracking more signals at the same time is stronger, but which combinations of tracking might increase accuracy in muscle coordination is still unclear, as well as how it would compare to CEINMS.
Applying EMG-tracking to different movement types as well as movements with more co-contraction,

could also reveal a more significant impact. Forcing different levels of co-contraction through experiment design may offer a way to prove if the accuracy gain of EMG-tracking is stronger for higher co-contraction.

A more extensive sensitivity analysis of EMG-tracking weight factors in the cost function might lead to weights that result in more accurate estimations. We would expect that increased accuracy can be achieved, although it might be hard to find a universal set of weights. It would also be interesting to see how the resulting weights compare to calibrated weights from CEINMS.

The implementation of normalization-independent EMG-tracking could be researched. It is difficult to predict what results such a method would produce, but if a method can be found that results in increased accuracy, while being insensitive to potentially unreliable EMG-normalization, it could be promising.

Finally, using the RMR solver in combination with a CEINMS calibrated model, might reveal how important the CEINMS calibration step is compared to EMG-assistance at simulation.

5 Conclusion

The main goal of this thesis is to equip the existing Rapid Muscle Redundancy (RMR) solver, which already featured a GH stability constraint, with an EMG-assisted cost function. Along with this extension, the RMR solver is revised to be class-based in order to improve modularity, scalability and user interaction. The new implementation is verified to produce identical estimations without EMG-assistance compared to the previous implementation and is successful at tracking EMG using the EMG-assisted cost function.

Simulations using a shoulder model and data, reveal that the impact of EMG-tracking single EMG signals on muscle recruitment is minimal, which is attributed to the lack of co-contraction used by the subject.

Simulations using a lower extremity model and gait data, result in a more significant impact of EMG-tracking tasks, which generally cause higher knee joint contact forces (JCFs) during stance. Tracking the most anterior element of the Gluteus Medius improves knee JCF accuracy, with respect to ground truth in-vivo data, the most on average over three subjects and multiple trials, with the Root Mean Squared Error (RMSE) decreasing with 8% and the Zero-Normalized Cross-Correlation (ZNCC) increasing with 4%, although unreliable EMG-normalization may invalidate these results to some extent.

Compared to results from the conventional Static Optimization (SO) method and the state-of-the-art Calibrated, EMG-Informed Neuromusculoskeletal Modeling toolbox (CEINMS), our EMG-assisted RMR solver only achieves lower RMSE in 2 out of 14 trials, although the average difference is only 0.03 BW.

The potential of EMG-tracking is especially demonstrated with tracking the Rectus Femoris muscle in challenge 6, in which more co-contraction is present. Through an increase of the the Rectus Femoris activity as well as its antagonist, Semitendinosus, the knee load estimation is increased at the first peak during stance, which was previously underestimated without EMG-assistance.

While the new implementation has not proved itself definitively, it seems promising in predicting co-contraction using EMG-assistance from few EMG signals. In the future this or similar methods might play a crucial role in investigating changes in muscle coordination due to rotator cuff tears, knee osteoarthritis or other musculoskeletal pathologies, which can contribute to more personalized prevention, intervention or rehabilitation.

References

[1] K. Walker□Bone, K. T. Palmer, I. Reading, D. Coggon, and C. Cooper, “Prevalence and impact of musculoskeletal disorders of the upper limb in the general population,” en, *Arthritis Care & Research*, vol. 51, no. 4, pp. 642–651, Aug. 2004.

[2] R. Govaerts *et al.*, “Prevalence and incidence of work-related musculoskeletal disorders in secondary industries of 21st century Europe: A systematic review and meta-analysis,” en, *BMC Musculoskeletal Disorders*, vol. 22, no. 1, p. 751, Dec. 2021.

[3] A. Yamamoto *et al.*, “Prevalence and risk factors of a rotator cuff tear in the general population,” en, *Journal of Shoulder and Elbow Surgery*, vol. 19, no. 1, pp. 116–120, Jan. 2010.

[4] H. Minagawa *et al.*, “Prevalence of symptomatic and asymptomatic rotator cuff tears in the general population: From mass-screening in one village,” en, *Journal of Orthopaedics*, vol. 10, no. 1, pp. 8–12, Mar. 2013.

[5] J. E. Labriola, T. Q. Lee, R. E. Debski, and P. J. McMahon, “Stability and instability of the glenohumeral joint: The role of shoulder muscles,” en, *Journal of Shoulder and Elbow Surgery*, vol. 14, no. 1, S32–S38, Jan. 2005.

[6] H. B. Park, J. Y. Gwark, and J.-B. Na, “Risk factors of chronic subscapularis tendon tear,” en, *Clinics in Shoulder and Elbow*, vol. 25, no. 4, pp. 257–264, Dec. 2022.

[7] B. Heidari, “Knee osteoarthritis prevalence, risk factors, pathogenesis and features: Part I,” eng, *Caspian Journal of Internal Medicine*, vol. 2, no. 2, pp. 205–212, 2011.

[8] G. S. Dulay, C. Cooper, and E. Dennison, “Knee pain, knee injury, knee osteoarthritis & work,” en, *Best Practice & Research Clinical Rheumatology*, vol. 29, no. 3, pp. 454–461, Jun. 2015.

[9] S. Farrokhi, M. O’Connell, A. B. Gil, P. J. Sparto, and G. K. Fitzgerald, “Altered Gait Characteristics in Individuals With Knee Osteoarthritis and Self-Reported Knee Instability,” en, *Journal of Orthopaedic & Sports Physical Therapy*, vol. 45, no. 5, pp. 351–359, May 2015.

[10] J. A. Zeni, K. Rudolph, and J. S. Higginson, “Alterations in quadriceps and hamstrings coordination in persons with medial compartment knee osteoarthritis,” en, *Journal of Electromyography and Kinesiology*, vol. 20, no. 1, pp. 148–154, Feb. 2010.

[11] D. Felson, “Osteoarthritis as a disease of mechanics,” en, *Osteoarthritis and Cartilage*, vol. 21, no. 1, pp. 10–15, Jan. 2013.

[12] H. X. Hoang, L. E. Diamond, D. G. Lloyd, and C. Pizzolato, “A calibrated EMG-informed neuromusculoskeletal model can appropriately account for muscle co-contraction in the estimation of hip joint contact forces in people with hip osteoarthritis,” en, *Journal of Biomechanics*, vol. 83, pp. 134–142, Jan. 2019.

[13] J. L. Hicks, T. K. Uchida, A. Seth, A. Rajagopal, and S. L. Delp, “Is My Model Good Enough? Best Practices for Verification and Validation of Musculoskeletal Models and Simulations of Movement,” *Journal of Biomechanical Engineering*, vol. 137, no. 2, p. 020 905, Feb. 2015.

[14] Y. Sheng, J. Zeng, J. Liu, and H. Liu, “Metric-Based Muscle Synergy Consistency for Upper Limb Motor Functions,” en, *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[15] W. Wang, D. Wang, and G. Li, “Towards improving the accuracy of musculoskeletal simulation of dynamic three-dimensional spine rotations with optimizing model and algorithm,” en, *Medical Engineering & Physics*, vol. 110, p. 103 916, Dec. 2022.

[16] K. J. Bennett *et al.*, “EMG-Informed Neuromusculoskeletal Models Accurately Predict Knee Loading Measured Using Instrumented Implants,” en, *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 7, pp. 2268–2275, Jul. 2022.

[17] R. T. Raikova and B. I. Prilutsky, “Sensitivity of predicted muscle forces to parameters of the optimization-based human leg model revealed by analytical and numerical analyses,” en, *Journal of Biomechanics*, vol. 34, no. 10, pp. 1243–1255, Oct. 2001.

[18] N. Arjmand and A. Shirazi-Adl, “Sensitivity of kinematics-based model predictions to optimization criteria in static lifting tasks,” *Medical Engineering and Physics*, vol. 28, no. 6, pp. 504–514, 2006.

[19] F. Michaud, U. Lugris, Y. Ou, J. Cuadrado, and A. Kecskemethy, “Influence of muscle recruitment criteria on joint reaction forces during human gait,” 2015, pp. 1024–1031.

[20] A. Zargham, M. Afschrift, J. De Schutter, I. Jonkers, and F. De Groote, “Inverse dynamic estimates of muscle recruitment and joint contact forces are more realistic when minimizing muscle activity rather than metabolic energy or contact forces,” en, *Gait & Posture*, vol. 74, pp. 223–230, Oct. 2019.

[21] F. C. Anderson, “Static and dynamic optimization solutions for gait are practically equivalent,” *Journal of Biomechanics*, vol. 34, pp. 153–161, 2001.

[22] A. Seth, M. Dong, R. Matias, and S. Delp, “Muscle Contributions to Upper-Extremity Movement and Work From a Musculoskeletal Model of the Human Shoulder,” en, *Frontiers in Neurorobotics*, vol. 13, p. 90, Nov. 2019.

[23] R. D. Crowninshield and R. A. Brand, “A physiologically based criterion of muscle force prediction in locomotion,” en, *Journal of Biomechanics*, vol. 14, no. 11, pp. 793–801, Jan. 1981.

[24] M. Praagman, E. Chadwick, F. Van Der Helm, and H. Veeger, “The relationship between two different mechanical cost functions and muscle oxygen consumption,” en, *Journal of Biomechanics*, vol. 39, no. 4, pp. 758–765, 2006.

[25] F. C. Anderson and M. G. Pandy, “Dynamic Optimization of Human Walking,” vol. 123, no. 5, pp. 381–390, Oct. 2001.

[26] Y. Wakimoto, M. Anan, K. Tanimoto, H. Hattori, and K. Shinkoda, “Factors associated with greater co-contraction during walking and quadriceps exercises in patients with knee osteoarthritis,” en, *Physiotherapy Practice and Research: The Official Journal of The Irish Society of Chartered Physiotherapists*, p. 22 130 683 251 337 340, Apr. 2025.

[27] R. T. Raikova, “Investigation of the influence of the elbow joint reaction on the predicted muscle forces using different optimization functions,” *Journal of Musculoskeletal Research*, vol. 12, no. 1, pp. 31–43, 2009.

[28] T. E. Milner, C. Cloutier, A. B. Leger, and D. W. Franklin, “Inability to activate muscles maximally during cocontraction and the effect on joint stiffness,” en, *Experimental Brain Research*, vol. 107, no. 2, Dec. 1995.

[29] T. E. Milner, “Adaptation to destabilizing dynamics by means of muscle cocontraction,” en, *Experimental Brain Research*, vol. 143, no. 4, pp. 406–416, Apr. 2002.

[30] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, and T. E. Milner, “Adaptation to Stable and Unstable Dynamics Achieved By Combined Impedance Control and Inverse Dynamics Model,” en, *Journal of Neurophysiology*, vol. 90, no. 5, pp. 3270–3282, Nov. 2003.

[31] J. B. Heald, D. W. Franklin, and D. M. Wolpert, “Increasing muscle co-contraction speeds up internal model acquisition during dynamic motor learning,” en, *Scientific Reports*, vol. 8, no. 1, p. 16 355, Nov. 2018.

[32] E. Burdet, D. W. Franklin, and T. E. Milner, “Motor Learning under Unstable and Unpredictable Conditions,” in *Human Robotics: Neuromechanics and Motor Control*, The MIT Press, Sep. 2013, p. 175.

[33] M. Febrer-Nafría, M. J. Dreyer, A. Maas, W. R. Taylor, C. R. Smith, and S. H. Hosseini Nasab, “Knee kinematics are primarily determined by implant alignment but knee kinetics are mainly influenced by muscle coordination strategy,” en, *Journal of Biomechanics*, vol. 161, p. 111 851, Dec. 2023.

[34] R. Brookham and C. Dickerson, “Comparison of humeral rotation co-activation of breast cancer population and healthy shoulders,” *Journal of Electromyography and Kinesiology*, vol. 29, pp. 100–106, 2016.

[35] D. Van Roon, B. Steenbergen, and R. G. Meulenbroek, “Trunk use and co-contraction in cerebral palsy as regulatory mechanisms for accuracy control,” en, *Neuropsychologia*, vol. 43, no. 4, pp. 497–508, Jan. 2005.

[36] J. Luo *et al.*, “Characterization of the coordination of agonist and antagonist muscles among stroke patients, healthy late middle-aged and young controls using a myoelectric-controlled interface,” *Journal of Neural Engineering*, vol. 15, no. 5, 2018.

[37] T. Stoeckmann, K. Sullivan, and R. Scheidt, “Elastic, viscous, and mass load effects on poststroke muscle recruitment and co-contraction during reaching: A pilot study,” *Physical Therapy*, vol. 89, no. 7, pp. 665–678, 2009.

[38] T. Graven-Nielsen, P. Svensson, and L. Arendt-Nielsen, “Effects of experimental muscle pain on muscle activity and co-ordination during static and dynamic motor function,” en, *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, vol. 105, no. 2, pp. 156–164, Apr. 1997.

[39] P. W. Hodges and K. Tucker, “Moving differently in pain: A new theory to explain the adaptation to pain,” en, *Pain*, vol. 152, no. 3, S90–S98, Mar. 2011.

[40] R. R. Neptune and S. A. Kautz, “Muscle Activation and Deactivation Dynamics: The Governing Properties in Fast Cyclical Human Movement Performance?” en,

[41] D. G. Thelen and F. C. Anderson, “Using computed muscle control to generate forward dynamic simulations of human walking from experimental data,” en, *Journal of Biomechanics*, vol. 39, no. 6, pp. 1107–1115, Jan. 2006.

[42] F. J. Van Melis, “Computational Methods for Muscle Co-contraction Prediction in Upper Extremity,” en, *Literature Review to obtain MSc degree (unpublished)*, 2025.

[43] E. R. Avila, S. E. Williams, and C. Disselhorst-Klug, “Advances in EMG measurement techniques, analysis procedures, and the impact of muscle mechanics on future requirements for the methodology,” en, *Journal of Biomechanics*, vol. 156, p. 111 687, Jul. 2023.

[44] F. Bailly, A. Ceglia, B. Michaud, D. Rouleau, and M. Begon, “Real-Time and Dynamically Consistent Estimation of Muscle Forces Using a Moving Horizon EMG-Marker Tracking Algorithm—Application to Upper Limb Biomechanics,” *Frontiers in Bioengineering and Biotechnology*, vol. 9, 2021.

[45] C. Bélaise, B. Michaud, F. Dal Maso, K. Mombaur, and M. Begon, “Which data should be tracked in forward-dynamic optimisation to best predict muscle forces in a pathological co-contraction case?” en, *Journal of Biomechanics*, vol. 68, pp. 99–106, Feb. 2018.

[46] C. Engelhardt *et al.*, “Comparison of an EMG-based and a stress-based method to predict shoulder muscle forces,” en, *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 18, no. 12, pp. 1272–1279, Sep. 2015.

[47] C. Pizzolato *et al.*, “CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks,” en, *Journal of Biomechanics*, vol. 48, no. 14, pp. 3929–3936, Nov. 2015.

[48] M. Sartori, M. Reggiani, D. Farina, and D. G. Lloyd, “EMG-Driven Forward-Dynamic Estimation of Muscle Force and Joint Moment about Multiple Degrees of Freedom in the Human Lower Extremity,” en, *PLoS ONE*, vol. 7, no. 12, P. L. Gribble, Ed., e52618, Dec. 2012.

[49] N. Assila, C. Pizzolato, R. Martinez, D. Lloyd, and M. Begon, “EMG-Assisted Algorithm to Account for Shoulder Muscles Co-Contraction in Overhead Manual Handling,” *Applied Sciences (Switzerland)*, vol. 10, no. 10, 2020.

[50] C. B. Crossley *et al.*, “A calibrated EMG-informed neuromusculoskeletal model can estimate hip and knee joint contact forces in cycling better than static optimisation,” en, *Journal of Biomechanics*, vol. 182, p. 112 586, Mar. 2025.

[51] M. J. Hambly, A. C. C. De Sousa, D. G. Lloyd, and C. Pizzolato, “EMG-Informed Neuromusculoskeletal Modelling Estimates Muscle Forces and Joint Moments During Electrical Stimulation,” en, in *2023 International Conference on Rehabilitation Robotics (ICORR)*, Singapore, Singapore: IEEE, Sep. 2023, pp. 1–6.

[52] E. Meinders, C. Pizzolato, B. A. Gonçalves, D. G. Lloyd, D. J. Saxby, and L. E. Diamond, “Electromyography measurements of the deep hip muscles do not improve estimates of hip contact force,” en, *Journal of Biomechanics*, vol. 141, p. 111 220, Aug. 2022.

[53] M. Romanato, L. Zhang, Z. Sawacha, and E. Gutierrez-Farewik, “Influence of different calibration methods on surface electromyography-informed musculoskeletal models with few input signals,” en, *Clinical Biomechanics*, vol. 109, p. 106 074, Oct. 2023.

[54] E. Sarshari, M. Mancuso, A. Terrier, A. Farron, P. Mullhaupt, and D. Pioletti, “Muscle co-contraction in an upper limb musculoskeletal model: EMG-assisted vs. standard load-sharing,” en, *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 24, no. 2, pp. 137–150, Jan. 2021.

[55] M. Sartori, D. Farina, and D. G. Lloyd, “Hybrid neuromusculoskeletal modeling to best track joint moments using a balance between muscle excitations derived from electromyograms and optimization,” en, *Journal of Biomechanics*, vol. 47, no. 15, pp. 3613–3621, Nov. 2014.

[56] C. Pontonnier and G. Dumont, “Inverse dynamics method using optimization techniques for the estimation of muscles forces involved in the elbow motion,” *International Journal on Interactive Design and Manufacturing*, vol. 3, no. 4, pp. 227–236, 2009.

[57] R. Brookham, E. Middlebrook, T.-J. Grewal, and C. Dickerson, “The utility of an empirically derived co-activation ratio for muscle force prediction through optimization,” *Journal of Biomechanics*, vol. 44, no. 8, pp. 1582–1587, 2011.

[58] J. Chopp-Hurley, R. Brookham, and C. Dickerson, “Identification of potential compensatory muscle strategies in a breast cancer survivor population: A combined computational and experimental approach,” *Clinical Biomechanics*, vol. 40, pp. 63–67, 2016.

[59] I. Belli *et al.*, “Does enforcing glenohumeral joint stability matter? A new rapid muscle redundancy solver highlights the importance of non-superficial shoulder muscles,” en, *PLOS ONE*, vol. 18, no. 11, A. F. Shaheen, Ed., e0295003, Nov. 2023.

[60] P. Favre, R. Sheikh, S. Fucentese, and H. Jacob, “An algorithm for estimation of shoulder muscle forces for clinical use,” *Clinical Biomechanics*, vol. 20, no. 8, pp. 822–833, 2005.

[61] J. Cashaback and T. Cluff, “Increase in joint stability at the expense of energy efficiency correlates with force variability during a fatiguing task,” *JOURNAL OF BIOMECHANICS*, vol. 48, no. 4, pp. 621–626, Feb. 2015.

[62] J. Niemi, H. Nieminen, E.-P. Takala, and E. Viikari-Juntura, “A static shoulder model based on a time-dependent criterion for load sharing between synergistic muscles,” *Journal of Biomechanics*, vol. 29, no. 4, pp. 451–460, 1996.

[63] D. Princelle, M. Viceconti, and G. Davico, “EMG-Informed Neuromusculoskeletal Simulations Increase the Accuracy of the Estimation of Knee Joint Contact Forces During Sub-optimal Level Walking,” en, *Annals of Biomedical Engineering*, Mar. 2025.

[64] S. Day, “EMG Introduction,” en,

[65] D. Princelle, G. Davico, and M. Viceconti, “Comparative validation of two patient-specific modelling pipelines for predicting knee joint forces during level walking,” en, *Journal of Biomechanics*, vol. 159, p. 111 758, Oct. 2023.

[66] B. J. Fregly *et al.*, “Grand challenge competition to predict in vivo knee loads,” en, *Journal of Orthopaedic Research*, vol. 30, no. 4, pp. 503–513, Apr. 2012.

[67] A. Ceglia, F. Bailly, and M. Begon, “Moving Horizon Estimation of Human Kinematics and Muscle Forces,” en, *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5212–5219, Aug. 2023.

[68] J. M. Winters, “An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models,” en, *Annals of Biomedical Engineering*, vol. 23, no. 4, pp. 359–374, Jul. 1995.

[69] I. Beck, I. Belli, L. Peternel, A. Seth, and J. M. Prendergast, “Real-Time Tendon Strain Estimation of Rotator-Cuff Muscles during Robotic-Assisted Rehabilitation,” en, in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, Austin, TX, USA: IEEE, Dec. 2023, pp. 1–8.

[70] A. Seth *et al.*, “OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement,” en, *PLOS Computational Biology*, vol. 14, no. 7, D. Schneidman, Ed., e1006223, Jul. 2018.

[71] M. Millard, T. Uchida, A. Seth, and S. L. Delp, "Flexing Computational Muscle: Modeling and Simulation of Musculotendon Dynamics," en, *Journal of Biomechanical Engineering*, vol. 135, no. 2, p. 021 005, Feb. 2013.

[72] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, and S. L. Delp, "Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait," en, *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 10, pp. 2068–2079, Oct. 2016.

[73] S. D. Uhlrich, R. W. Jackson, A. Seth, J. A. Kolesar, and S. L. Delp, "Muscle coordination retraining inspired by musculoskeletal simulations reduces knee contact force," en, *Scientific Reports*, vol. 12, no. 1, p. 9842, Jul. 2022.

[74] A. Silder, B. Whittington, B. Heiderscheit, and D. G. Thelen, "Identification of passive elastic joint moment–angle relationships in the lower extremity," en, *Journal of Biomechanics*, vol. 40, no. 12, pp. 2628–2635, 2007.

[75] M. Millard. "Opensim::fiberforcelengthcurve class reference." (2024), [Online]. Available: `https://simtk.org/api_docs/opensim/api_docs/classOpenSim_1_1FiberForceLengthCurve.html` (visited on 07/30/2025).

[76] B. Fregly, D. D'Lima, and T. Besier. "Grand challenge competition to predict in vivo knee loads." (2019), [Online]. Available: `https://simtk.org/projects/kneeloads` (visited on 08/01/2025).

[77] S. Tahmid, J. M. Font-Llagunes, and J. Yang, "Upper Extremity Muscle Activation Pattern Prediction Through Synergy Extrapolation and Electromyography-Driven Modeling," en, *Journal of Biomechanical Engineering*, vol. 146, no. 1, p. 011 005, Jan. 2024.

[78] D. G. Thelen, "Adjustment of Muscle Mechanics Model Parameters to Simulate Dynamic Contractions in Older Adults," en, *Journal of Biomechanical Engineering*, vol. 125, no. 1, pp. 70–77, Feb. 2003.

A

Data availability

Shoulder model & data

The shoulder data and scaled shoulder model by Seth *et al.* [22] are available at:
`https://simtk.org/projects/thoracoscapular`.

Gait data

Data from the Grand Challenge Competition to Predict In Vivo Knee Loads is available at:
`https://simtk.org/projects/kneeloads`.

Lower extremity models

The scaled lower extremity models from Princelle *et al.* [65] for use with the Knee Grand Challenge data is available at:
`https://doi.org/10.6092/unibo/amsacta/7528`.

Python code

The Python code used for this thesis is available at a private repository:
`https://github.com/PTbot-TUDelft/live-active-strainmaps-florian`.

Access may be requested

B
Revised RMR solver details

B.1. Script example

The example below shows a top-level Python script, in which the RMR solver is employed on a normal gait trial of KGC 6, with and without EMG-assistance. For the code of all classes, functions and other run files, please refer to appendix A.

```python
"""
Example script demonstrating the revised Python RMR solver.

In this script the RMR solver without tracking is compared to the RMR solver with tracking.
Data: Grand Knee Challenge 6
Trial: DM_ngait_og4
Tracking: recfem (Rectus Femoris)
"""

### Import libraries, classes and utility files ###
import os
import warnings
import numpy as np

from Classes.ModelSpecs import ModelSpecs
from Classes.MotionManager import MotionManager
from Classes.EMGmanager import EMGmanager
from Classes.RMRsolver import RMRsolver
from Classes.MotionAnalysis import MotionAnalysis

import Utilities.utilsObjectives as utilsObj # objective function classes
import Utilities.utilsLoadFile as utilsLoad # loading Results
warnings.filterwarnings('ignore', category=RuntimeWarning) # suppress scipy.minimize warnings

### Set script flags ###
run = True
save = True
plot = False
track = 'recfem'

### Select file locations ###
# Main folder containing all models & data:
mainFolder = "Grand-Knee-Challenge"

# Musculoskeletal model:
modelFolder = os.path.join(mainFolder,"Model","Princelle")
modelName = "STAPLE_KGC6_RMR"

# Generalized coordinate data:
motionFolder = os.path.join(mainFolder,"Result-IK","Motion","Princelle")
motionName = "DM_ngait_og4_new_conv"

# EMG data:
emgFolder = os.path.join(mainFolder,"Experimental","EMG")
emgName = "DM_ngait_og4_emg"
mvcName = "DM_MVC"

# GRF data:
grfFolder = os.path.join(mainFolder,"Experimental","GRF")
grfName = "DM_ngait_og4_grf"

```

```python
# JCF data:
jcfFolder = os.path.join(mainFolder,"Experimental","JCF")
jcfName = "DM_ngait_og4_knee_forces"

# Result data (will be created if run=True):
resultFolder = os.path.join(mainFolder,"Result-RMR")
resultName = "DM_ngait_og4_track_none"
result_trackName = "DM_ngait_og4_track_" + track

### Setup simulation ###
# Load model:
spec = ModelSpecs(modelFolder,modelName) # load model and its specifications

# Setup motion:
motion = MotionManager(motionFolder,motionName) # load motion data
motion.filter(cutoff=8) # low-pass filter
motion.setTime(freq=60) # crop & resample

# Setup EMG:
emg = EMGmanager(emgFolder,emgName) # load EMG data
emg.norm(emgFolder,mvcName) # normalize with MVC
emg.filter(cutoff=[30,300],bandType='bandpass',order=2,filterType='sosfilt') # band-pass filter
emg.rect() # rectify
emg.filter(cutoff=8,bandType='lowpass',order=4,filterType='filtfilt') # low-pass filter
emg.sync(motion) # synchronize with motion data (crop & resample)

# Setup GRF:
grf = GRFmanager(grfFolder,grfName) # load GRF data
grf.sync(motion)

specs.addPrescribedForce(body='pelvis',name='GRF_left1') # add each GRF to the model
specs.addPrescribedForce(body='calcn_r',name='GRF_right1')
specs.addPrescribedForce(body='calcn_r',name='GRF_right2')

# Setup JCF (for result only):
jcf = JCFmanager(jcfFolder,jcfName) # load JCF data
jcf.sync(motion)

# Setup solvers:
effortWeights = np.concatenate( (1*np.ones(40),
                                 0*np.ones(6),
                                 np.array([10, 10, 10, 10, 0, 10, 0, 0])) )
    # weights for lowest effort objective: muscle = 1, penalized reserve = 10, free reserve = 0

objective = utilsObj.EMGtrack(specs) # without EMG tracking
solver = RMRsolver(specs,outputJoint=['knee_r'],prescribedForceIndex=[54,55,56], musclePassiveForce=False)
    # output for right knee JRF, no passive forces
solver.addObjective(objective) # add objective to solver

objective_track = utilsObj.EMGtrack(specs,track=track,trackWeights=3) # with EMG tracking
solver_track = RMRsolver(specs,outputJoint=['knee_r'],prescribedForceIndex=[54,55,56], musclePassiveForce=False)
solver_track.addObjective(objective_track)

analysis = MotionAnalysis(specs)
analysis_track = MotionAnalysis(specs)

# Run solver:
if run:  # run simulations
    result = analysis.runAll()
    result_track = analysis_track.runAll()

    if save: # save result
        result.save(resultFolder,resultName)
        result_track.save(resultFolder,result_trackName)

# Setup plotting
if plot:
    if not run: # load result
        result = utilsLoad.loadResult(resultFolder,resultName)
        result_track = utilsLoad.loadResult(resultFolder,result_trackName)

analysis.addPlot('activation') # add muscle activation plot
analysis.addPlot('reserve') # add reserve actuator activation plot
analysis.addPlot('jointContactForce',plotRange=['knee_r']) # knee joint contact force plot

analysis.plotResult([result, result_track],labels=['No␣track','track' + track]) # show all added plots
```

B.2. `RMRsolver` default settings

The following default settings are used for the `RMRsolver` class in this thesis:

- For the very first time step, `RMRsolver` uses an initial guess of 0.1 and 0 activation of muscle and reserve actuators, respectively. In subsequent time steps, the solution of the last time step is used as an initial guess.
- Reserve actuators have activation bounds of `[-1e6, 1e6]`.
- The experimental acceleration constraint tolerance scales with a constraint tolerance factor (δ). The expression for the constraint is as follows: $\ddot{q}_{i,\text{tol}} = \max[\delta, \delta\ddot{q}_i]$, with $\ddot{q}_i$ the ith entry of the induced acceleration vector, which needs to be matched, and $\ddot{q}_{i,\text{tol}}$ the ith entry of the acceleration constraint tolerance. As can be seen, the constraint tolerance assumes a value equal to δ for induced accelerations below 1 N(m)/s^2 and scales with δ for induced accelerations above 1. Within the constraint tolerance, the constraint is deemed satisfied. For the first optimization attempt at each time step, a tolerance of 0.001 is used. If this first attempts fails to converge, a second attempt is done using a tolerance of 0.005. If the second attempt fails, `RMRsolver` fails to find a solution for that time step.
- For the actual optimization process the `scipy.signal.minimize()` function is used with the Sequential Least Squares Programming method. The following settings are used: `maxiter=10000, ftol= 1e-6, disp= False, eps= 1e-8` (see also `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`).

C
Additional method information

C.1. Shoulder movement simulation

Full name	Abbreviation	EMG signal	MVC value
Trapezius	TrapeziusScapula_M	TM	4.175
	TrapeziusScapula_I	TI	3.202
	TrapeziusScapula_S	sum = TS	0.547
	TrapeziusCalvicle_S		
Serratus Anterior	SerratusAnterior_I	mean = SA	1.862
	SerratusAnterior_M		
	SerratusAnterior_S		
Rhomboid	Rhomboideus_S		
	Rhomboideus_I		
Levator Scapulae	LevatorScapulae		
Coracobrachialis	Coracobrachialis		
Deltoid	DeltoideusClavicle_A	AD	2.225
	DeltoideusScapula_P	PD	0.706
	DeltoideusScapula_M	MD	1.378
Latissimus Dorsi	LatissimusDorsi_S	LatD	0.448
	LatissimusDorsi_M		
	LatissimusDorsi_I		
Pectoralis Major	PectoralisMajorClavicle_S	PM_Clav	0.564
	PectoralisMajorThorax_I		
	PectoralisMajorThorax_M		
Teres Major	TeresMajor	Tmajor	1.155
Infraspinatus	Infraspinatus_I		
	Infraspinatus_S	Infra	0.173
Pectoralis Minor	PectoralisMinor		
Teres Minor	TeresMinor		
Subscapularis	Subscapularis_S		
	Subscapularis_M		
	Subscapularis_I		
Supraspinatus	Supraspinatus_P		
	Supraspinatus_A		
Triceps Brachii Longhead	TRIlong		
Bisceps Brachii Longhead	BIC_long		
Biceps Brachii Shorthead	BIC_brevis		

Table C.1: Muscles in the shoulder model and mapping to experimental EMG signals.

Movement type	Trial name	Start time [s]	End time [s]
Loaded forward flexion	Flx21	0.032	6.972
	Flx22	0.967	9.354
	Flx23	1.774	9.201
Loaded abduction	Abd21	0.639	8.536
	Abd22	0.706	8.942
	Abd23	0.618	8.768
Loaded shrugging	Shrug21	0.544	2.303
	Shrug22	0.477	2.924
	Shrug23	0.739	2.895

Table C.2: Trials and time ranges used for the shoulder data set. All loaded movements are performed using 2kg handheld weights. Time ranges are extracted from the `PlotMuscleResult` MATLAB script in the supplementary materials from Seth *et al.* [22], see appendix A for data availability.

$w_1 = 0$	$w_1 = 1$	$w_1 = 10$
ground_thorax_rot_x_actuator	*All muscles,*	scapula_elevation_actuator
ground_thorax_rot_y_actuator	*see table C.1*	clav_prot_actuator
ground_thorax_rot_z_actuator		clav_elev_actuator
ground_thorax_tx_actuator		scapula_abduction_actuator
ground_thorax_ty_actuator		scapula_upward_rot_actuator
ground_thorax_tz_actuator		scapula_winging_actuator
elbow_actuator		plane_elv_actuator
pro_sup_actuator		shoulder_elv_actuator
		axial_rot_actuator

Table C.3: Weights for vector $\mathbf{w}_1$ (lowest effort objective) for the shoulder model.

C.2. Normal gait simulation

Full name	**Abbreviation**	**EMG max value**		
		Challenge 4	Challenge 5	Challenge 6
Adductor Brevis	addbrev			
Adductor Longus	addlong			
Adductor Magnus	addmagDist addmagIsch addmagMid addMagProx	MVC	Gait	Gait
Biceps Femoris Longhead	bflh	MVC	Gait	MVC
Biceps Femoris Shorthead	bfsh			
Extensor Digitorum Longus	edl			
Externsor Hallucis Longus	ehl			
Flexor Digitorum Longus	fdl			
Flexor Hallucis Longus	fhl			
Gastrocnemius Lateralis	gaslat	MVC	MVC	MVC
Gastrocnemius Medialis	gasmed	MVC	MVC	MVC
Gluteus Maximus	glmax1 glmax2 glmax3	MVC	MVC	Gait
Gluteus Medius	glmed1 glmed2 glmed3	Gait	MVC	MVC
Gluteus Minimus	glmin1 glmin2 glmin3			
Gracilis	grac			
Iliacus	iliacus			
Peronius Brevis	perbrev			
Peronius Longus	perlong	Gait	Gait	Gait
Piriformis	piri			
Psoas Major	psoas			
Rectus Femoris	recfem	MVC	MVC	MVC
Sartorius	sart	Gait	Gait	MVC
Semimembranosus	semimem	MVC	Gait	Gait
Semitendinosus	semiten			
Soleus	soleus	MVC	Gait	Gait
Tractus Iliotibial	tfl	Gait	Gait	MVC
Tibialis Anterior	tibant	MVC	Gait	MVC
Tibialis Posterior	tibpost			
Vastus Intermedius	vasint			
Vastus Lateralis	vaslat	Gait	MVC	MVC
Vastus Medialis	vasmed	Gait	MVC	Gait

Table C.4: Muscles in the lower extremity model and the trial type in which the maximum value of EMG is found. This maximum value is used for normalization of EMG.

$w_1 = 0$	$w_1 = 1$	$w_1 = 10$
pelvis_tilt	*All muscles,*	hip_flexion
pelvis_list	*see table C.4*	hip_adduction
pelvis_rotation		hip_rotation
pelvis_tx		knee_angle
pelvis_ty		ankle_angle
pelvis_tz		
knee_angle_beta		
subtalar_angle		
mtp_angle		

Table C.5: Weights for vector w_1 (lowest effort objective) for the lower extremity model.

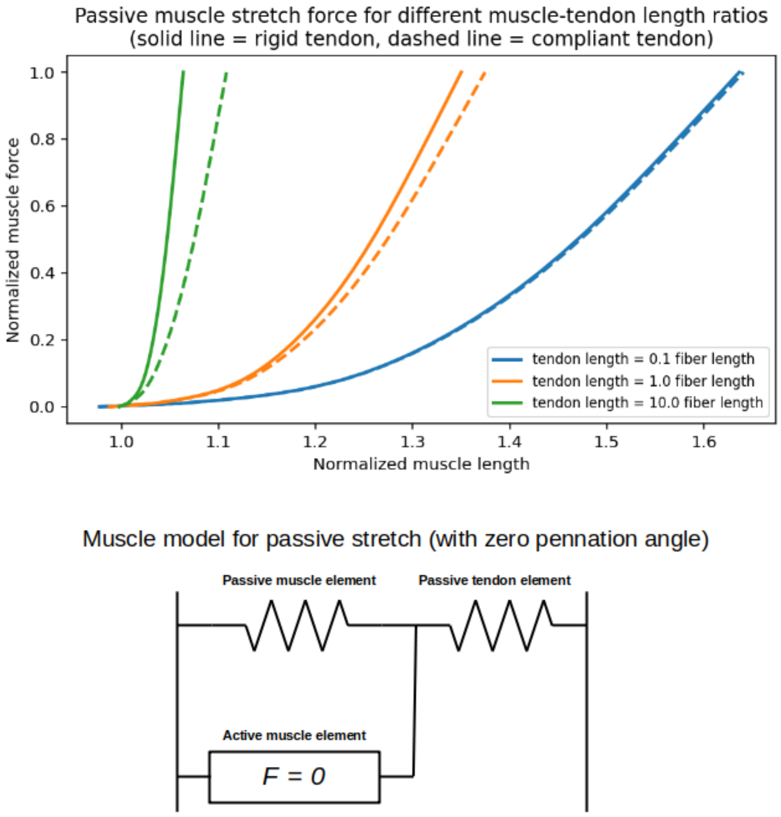


Figure C.1: Passive muscle force of rigid vs. compliant tendons for different muscle-tendon length ratios. Without active muscle forces and a pennation angle of $0°$, a model with rigid tendons causes higher passive forces than a model with compliant tendons. In the compliant tendon model, two spring elements act in series, which decreases the net muscle-tendon-unit stiffness. At the same time, increasing the tendon to muscle length ratio, increases passive muscle-tendon-unit stiffness, because tendon stiffness is significantly higher than muscle stiffness (one normalized force at 5% and 70% strain, respectively). Increased stiffness from a rigid tendon model is more pronounced at high tendon to muscle length ratios.

D Additional result information

D.1. Verification

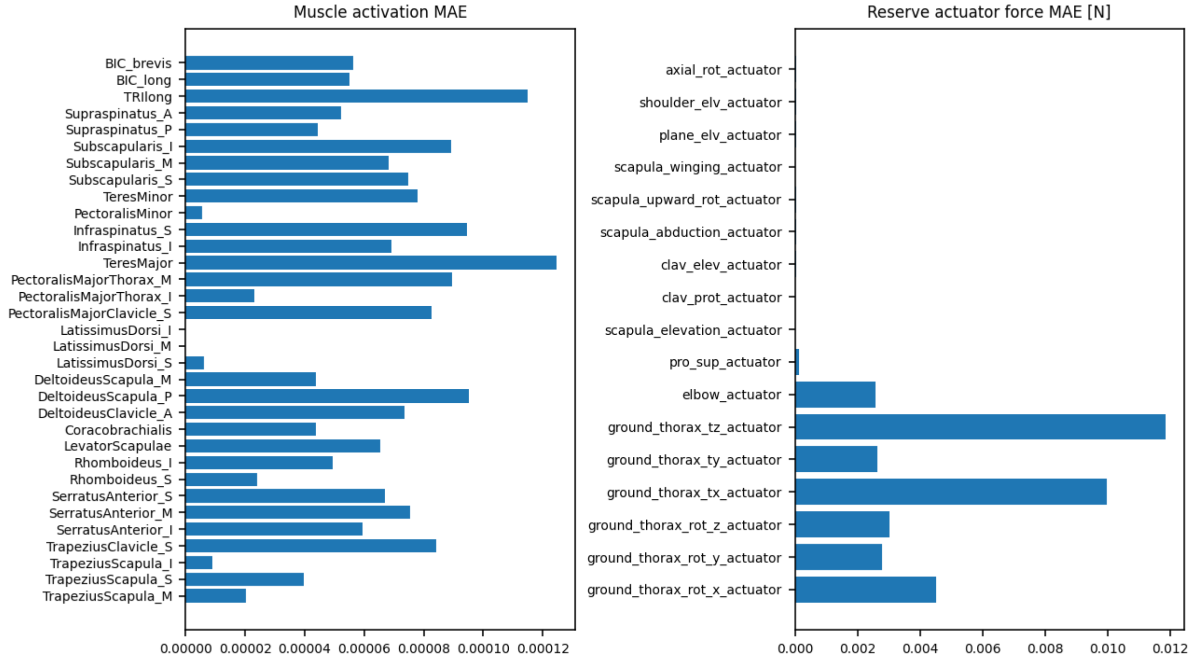


Figure D.1: MAE between actuator recruitment of the original Python implantation [69] and the revised implementation (this thesis).

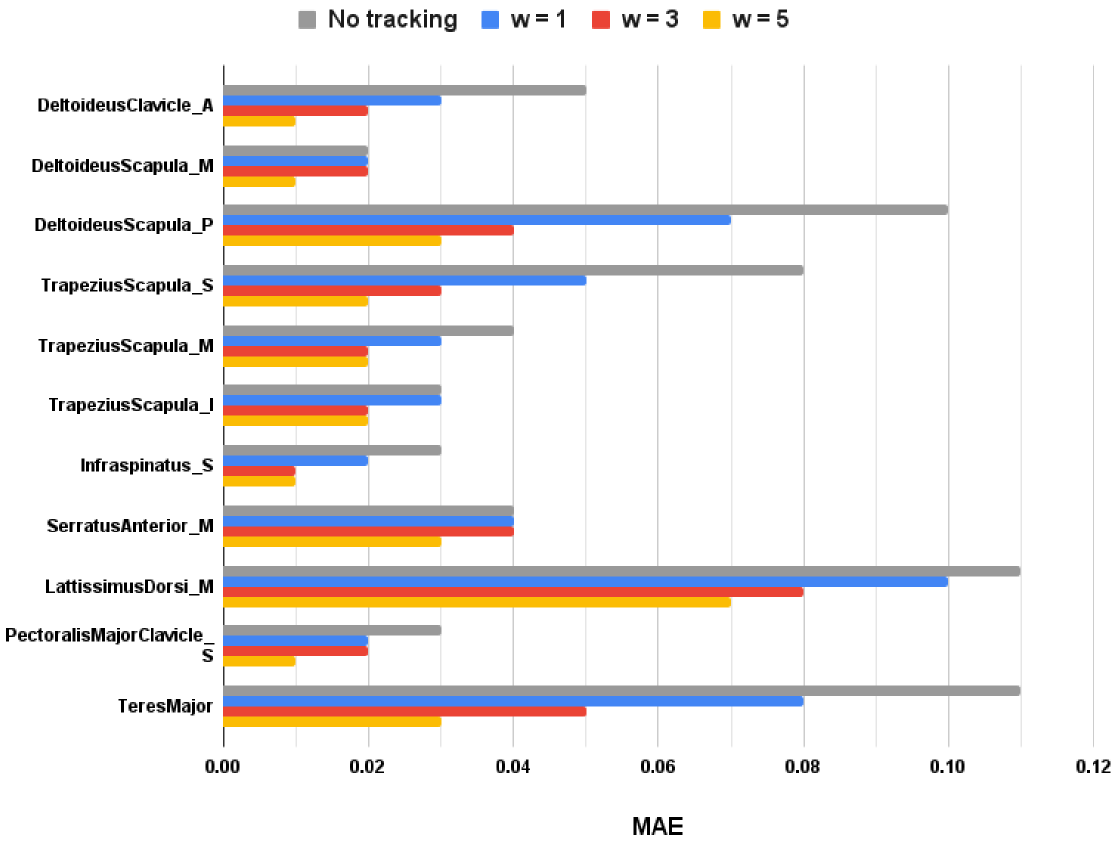


Figure D.2: MAE between muscle activation and EMG for different tracking tasks. Each row presents the tracking task as well as the muscle evaluated. When the named muscle is tracked, its tracking weight is set 1, 3 or 5, while all other muscles are not tracked.

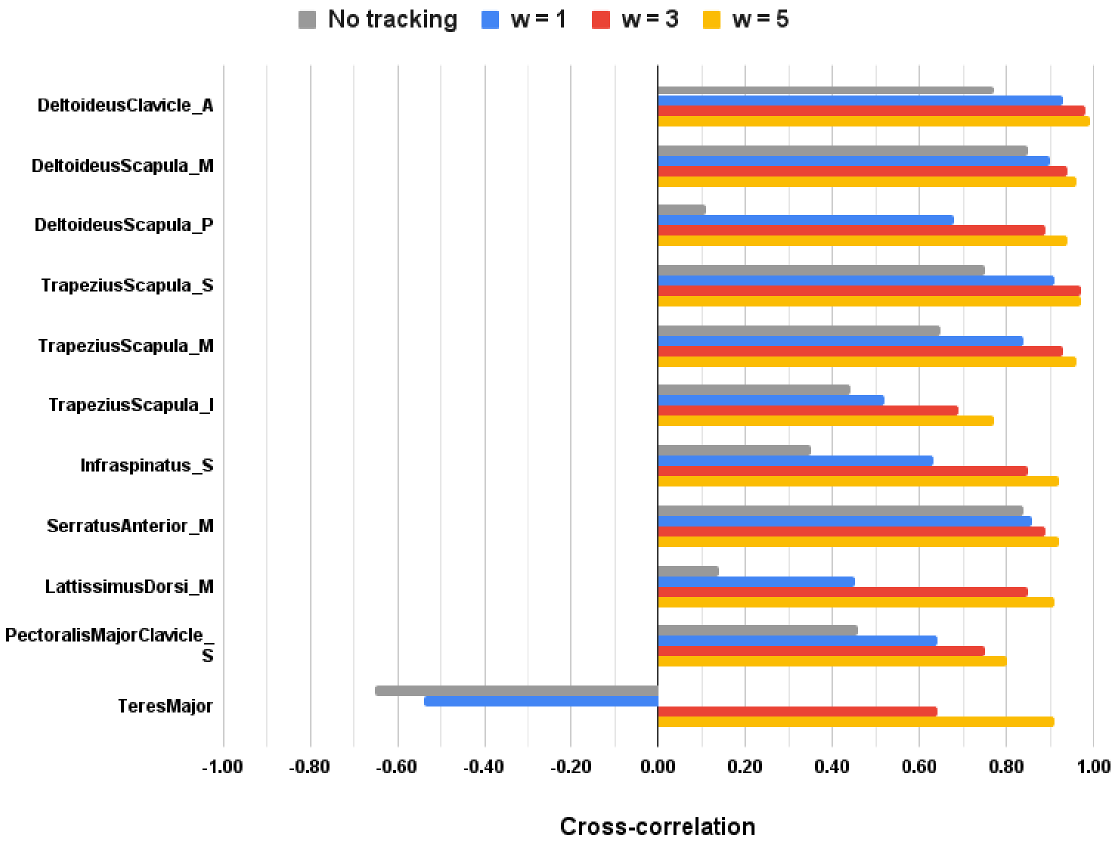


Figure D.3: Zero-normalized cross-correlation between muscle activation and EMG for different tracking tasks. Each row presents the tracking task as well as the muscle evaluated. When the named muscle is tracked, its tracking weight is set 1, 3 or 5, while all other muscles are not tracking.

D.2. Shoulder movement simulation

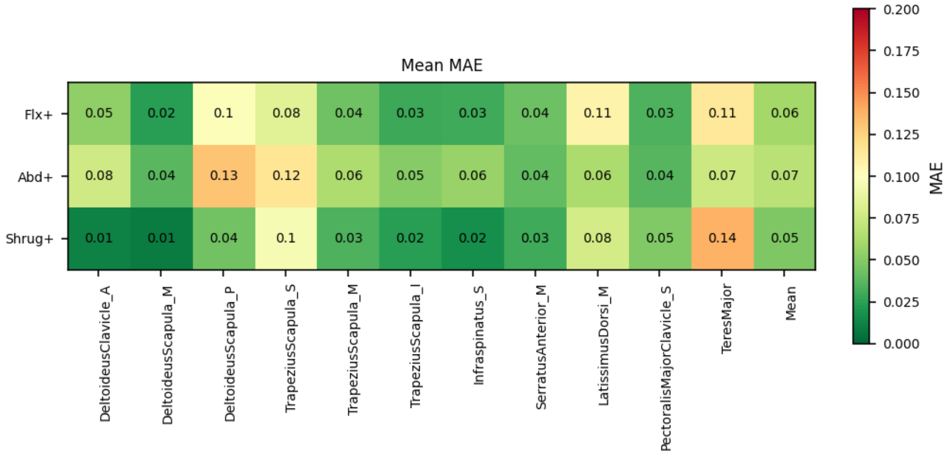


Figure D.4: MAE for shoulder movements without any EMG-tracking. For each movement type, an average is taken across three trials.

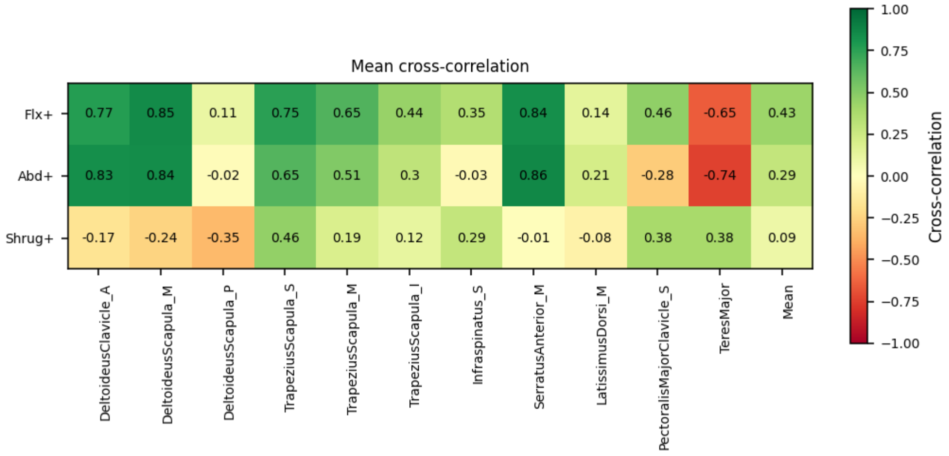


Figure D.5: Zero-normalized cross-correlation for shoulder movements without any EMG-tracking. For each movement type, an average is taken across three trials.

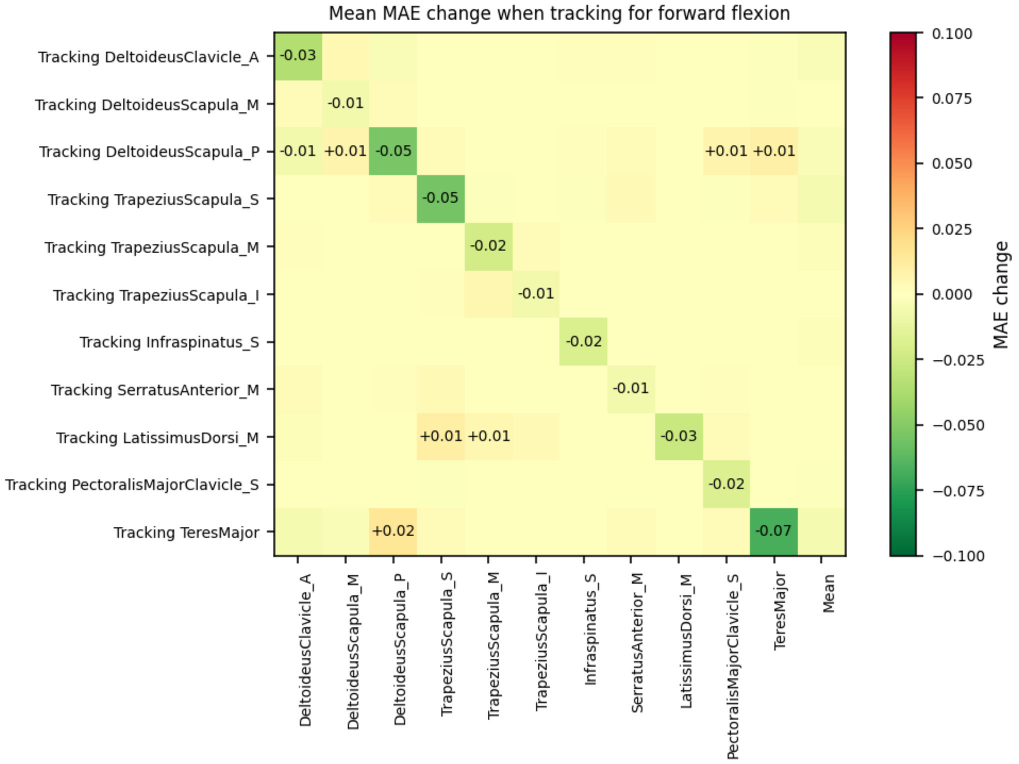


Figure D.6: The effect of tracking task on MAE of the tracked muscles for loaded forward flexion. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00.

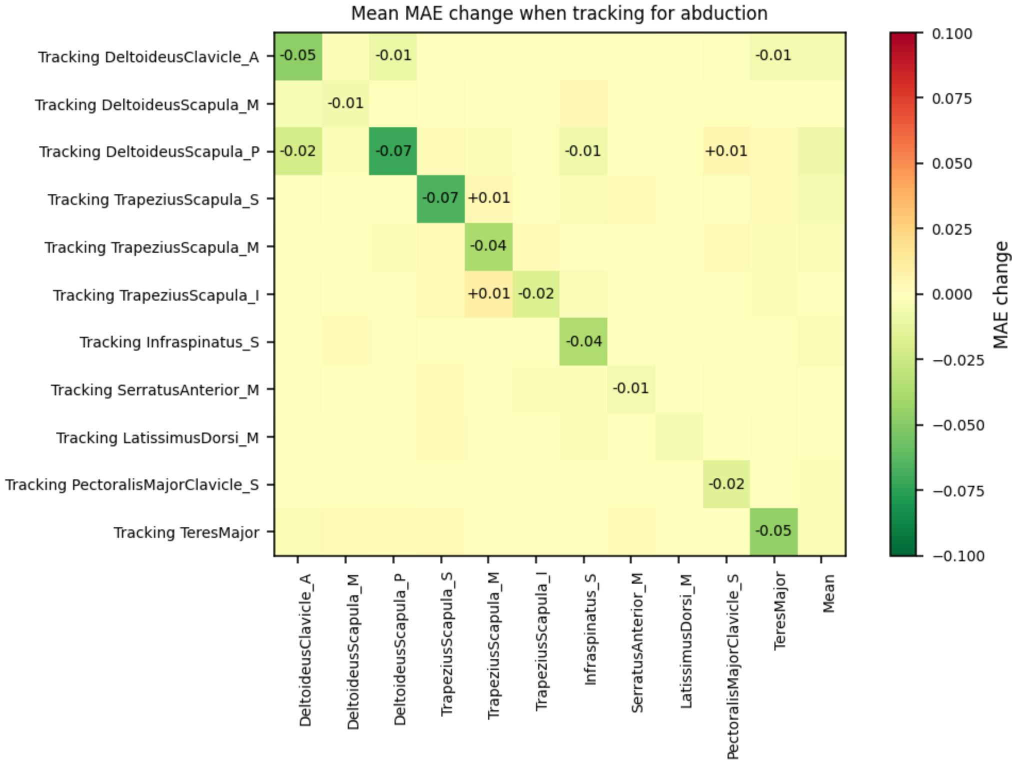


Figure D.7: The effect of tracking task on MAE of the tracked muscles for loaded abduction. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00.

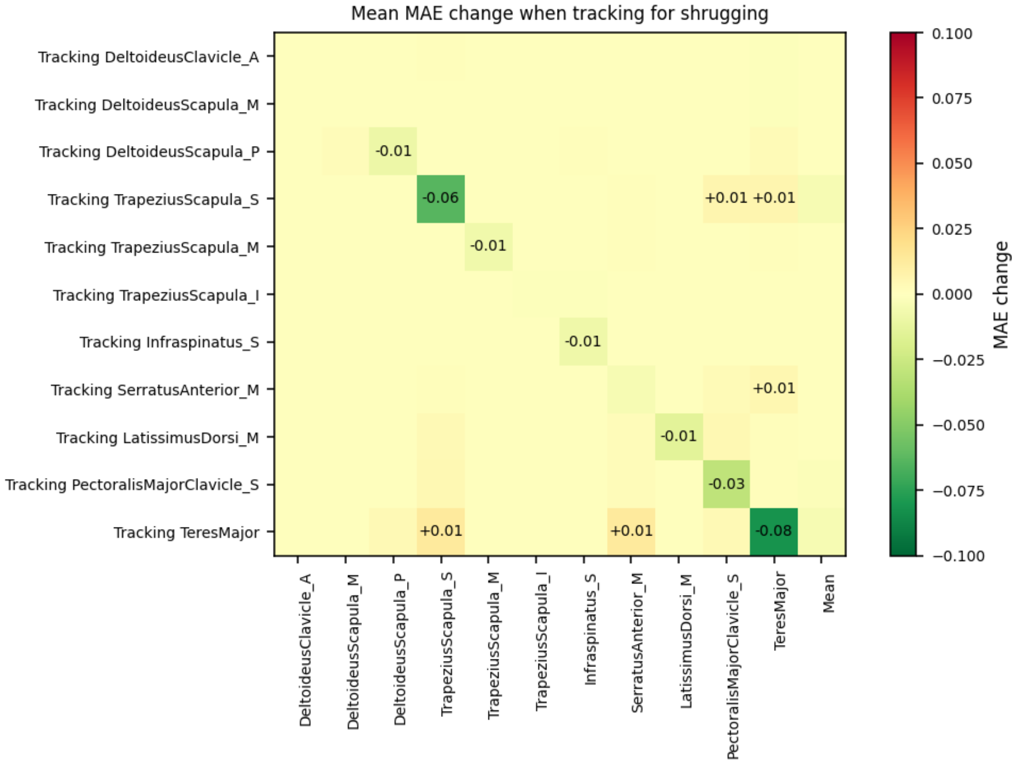


Figure D.8: The effect of tracking task on MAE of the tracked muscles for loaded shrugging. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00.

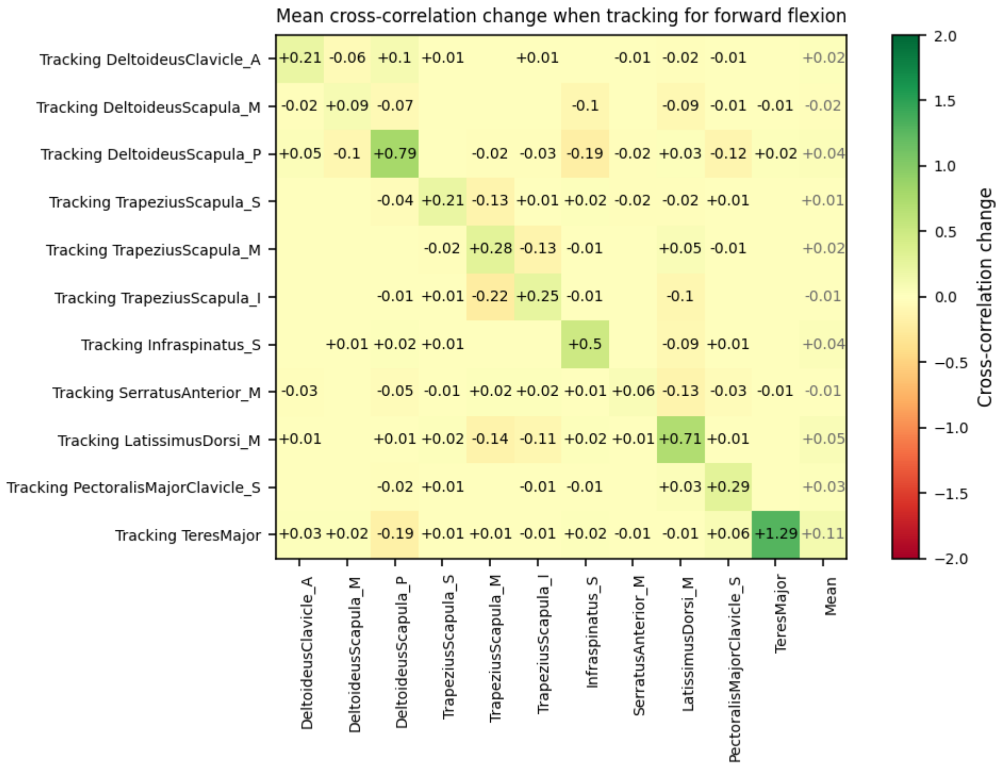
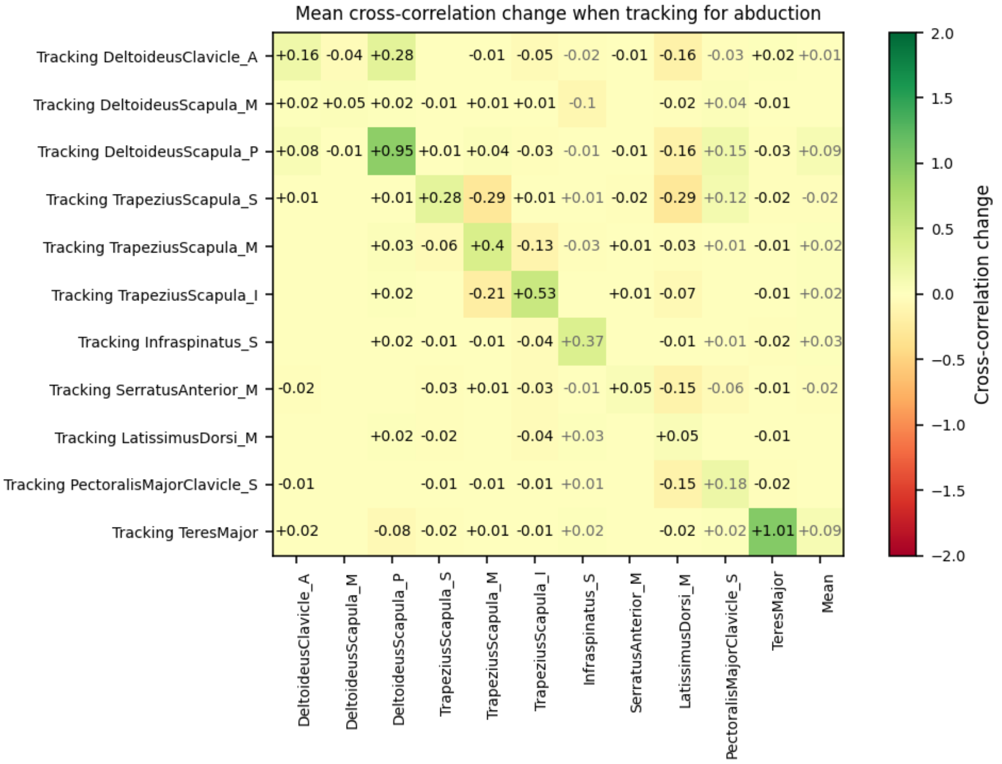
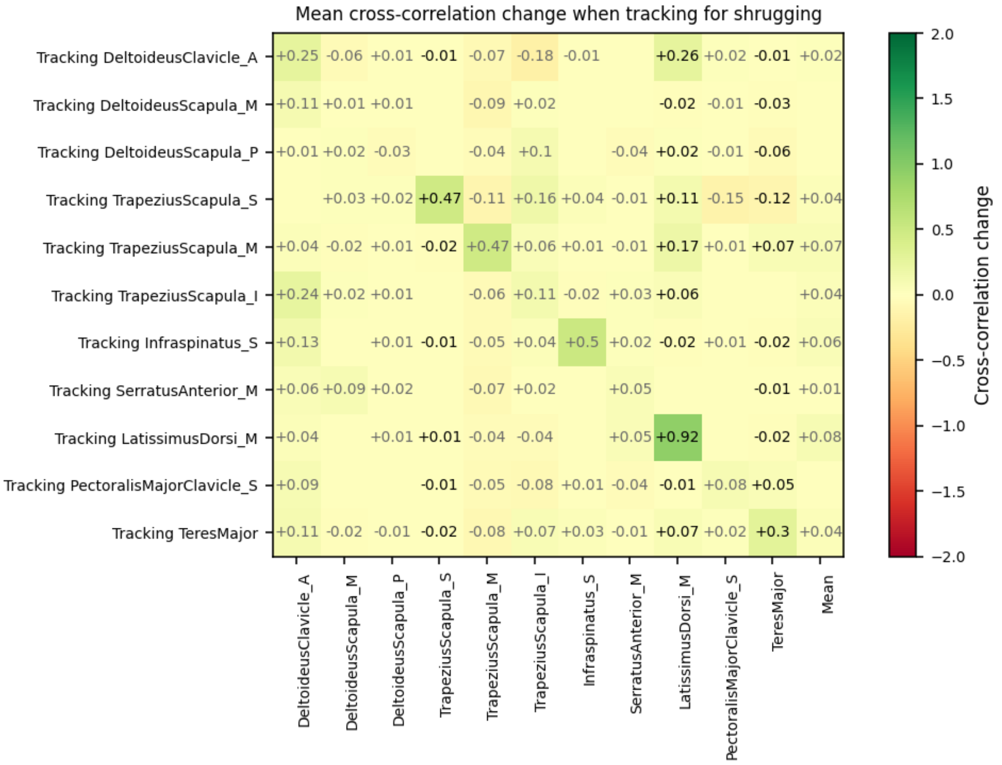


Figure D.9: The effect of tracking task on ZNCC of the tracked muscles for loaded forward flexion. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00. Grey values indicate that the maximum muscle activation is below 0.1.



	DeltoideusClavicle_A	DeltoideusScapula_M	DeltoideusScapula_P	TrapeziusScapula_S	TrapeziusScapula_M	TrapeziusScapula_I	Infraspinatus_S	SerratusAnterior_M	LatissimusDorsi_M	PectoralisMajorClavicle_S	TeresMajor	Mean
Tracking DeltoideusClavicle_A	+0.16	-0.04	+0.28		-0.01	-0.05	-0.02	-0.01	-0.16	-0.03	+0.02	+0.01
Tracking DeltoideusScapula_M	+0.02	+0.05	+0.02	-0.01	+0.01	+0.01	-0.1		-0.02	+0.04	-0.01	
Tracking DeltoideusScapula_P	+0.08	-0.01	+0.95	+0.01	+0.04	-0.03	-0.01	-0.01	-0.16	+0.15	-0.03	+0.09
Tracking TrapeziusScapula_S	+0.01		+0.01	+0.28	-0.29	+0.01	+0.01	-0.02	-0.29	+0.12	-0.02	-0.02
Tracking TrapeziusScapula_M			+0.03	-0.06	+0.4	-0.13	-0.03	+0.01	-0.03	+0.01	-0.01	+0.02
Tracking TrapeziusScapula_I			+0.02		-0.21	+0.53		+0.01	-0.07		-0.01	+0.02
Tracking Infraspinatus_S			+0.02	-0.01	-0.01	-0.04	+0.37		-0.01	+0.01	-0.02	+0.03
Tracking SerratusAnterior_M	-0.02			-0.03	+0.01	-0.03	-0.01	+0.05	-0.15	-0.06	-0.01	-0.02
Tracking LatissimusDorsi_M			+0.02	-0.02		-0.04	+0.03		+0.05		-0.01	
Tracking PectoralisMajorClavicle_S	-0.01			-0.01	-0.01	-0.01	+0.01		-0.15	+0.18	-0.02	
Tracking TeresMajor	+0.02		-0.08	-0.02	+0.01	-0.01	+0.02		-0.02	+0.02	+1.01	+0.09

Figure D.10: The effect of tracking task on ZNCC of the tracked muscles for loaded abduction. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00. Grey values indicate that the maximum muscle activation is below 0.1.



	DeltoideusClavicle_A	DeltoideusScapula_M	DeltoideusScapula_P	TrapeziusScapula_S	TrapeziusScapula_M	TrapeziusScapula_I	Infraspinatus_S	SerratusAnterior_M	LatissimusDorsi_M	PectoralisMajorClavicle_S	TeresMajor	Mean
Tracking DeltoideusClavicle_A	+0.25	-0.06	+0.01	-0.01	-0.07	-0.18	-0.01		+0.26	+0.02	-0.01	+0.02
Tracking DeltoideusScapula_M	+0.11	+0.01	+0.01		-0.09	+0.02			-0.02	-0.01	-0.03	
Tracking DeltoideusScapula_P	+0.01	+0.02	-0.03		-0.04	+0.1		-0.04	+0.02	-0.01	-0.06	
Tracking TrapeziusScapula_S		+0.03	+0.02	+0.47	-0.11	+0.16	+0.04	-0.01	+0.11	-0.15	-0.12	+0.04
Tracking TrapeziusScapula_M	+0.04	-0.02	+0.01	-0.02	+0.47	+0.06	+0.01	-0.01	+0.17	+0.01	+0.07	+0.07
Tracking TrapeziusScapula_I	+0.24	+0.02	+0.01		-0.06	+0.11	-0.02	+0.03	+0.06			+0.04
Tracking Infraspinatus_S	+0.13		+0.01	-0.01	-0.05	+0.04	+0.5	+0.02	-0.02	+0.01	-0.02	+0.06
Tracking SerratusAnterior_M	+0.06	+0.09	+0.02		-0.07	+0.02		+0.05			-0.01	+0.01
Tracking LatissimusDorsi_M	+0.04		+0.01	+0.01	-0.04	-0.04		+0.05	+0.92		-0.02	+0.08
Tracking PectoralisMajorClavicle_S	+0.09			-0.01	-0.05	-0.08	+0.01	-0.04	-0.01	+0.08	+0.05	
Tracking TeresMajor	+0.11	-0.02	-0.01	-0.02	-0.08	+0.07	+0.03	-0.01	+0.07	+0.02	+0.3	+0.04

Figure D.11: The effect of tracking task on ZNCC of the tracked muscles for loaded shrugging. Values represent the mean over all three trials. Empty fields indicate that the change is less than 0.00. Grey values indicate that the maximum muscle activation is below 0.1.

D.3. Gait simulation

Bodypart	**Markers**	**Participant**		
		KGC4	KGC5	KGC6
Hip	Asis	L, R	L, R	L, R
	Psis	L, R	L, R	L, R
Thigh	KneeLateral	R		
	KneeMedial	R		
	ThighInferior	R	L	R
	ThighLateral	R*	L	R
	ThighSuperior	R	L	R
Patella	Patella	R*	L	R
Shank	AnkleLateral	R		
	AnkleMedial	R		
	ShankInferior	R	L	R
	ShankLateral	R	L	R
	ShankSuperior	R	L	R
Foot	Heel	R	L	R
	MidfootLateral	R	L	R
	MidfootMedial	R	L	R
	MidfootSuperior	R	L	R
	Toe	R	L	R
	ToeLateral	R	L	R
	ToeMedial	R	L	R

Table D.1: Tracked markers for each challenge, grouped per body part. 'L' and 'R' indicate if the left and/or right marker is used. Markers with * are omitted for trial 7, because of obscuration.

Participant	**Trial**	**Adjusted start time [s]**	**Adjusted end time [s]**	**Mean RMSE [mm]**
KGC 4	og2			12.8
	og3			13.1
	og4			13.2
	og5	2.14		13.0
	og7			13.3
KGC 5	og1			9.8
	og8			9.3
	og9			8.9
	og11			9.2
KGC 6	og4			18.5
	og5		3.90	17.8
	og6		5.00	18.3
	og7		3.15	16.4
	og9			17.8

Table D.2: Inverse kinematic mean marker RMSE for each trial for the given time range. Unadjusted start or end times indicate that the first or last available time in the trajectory data is used.

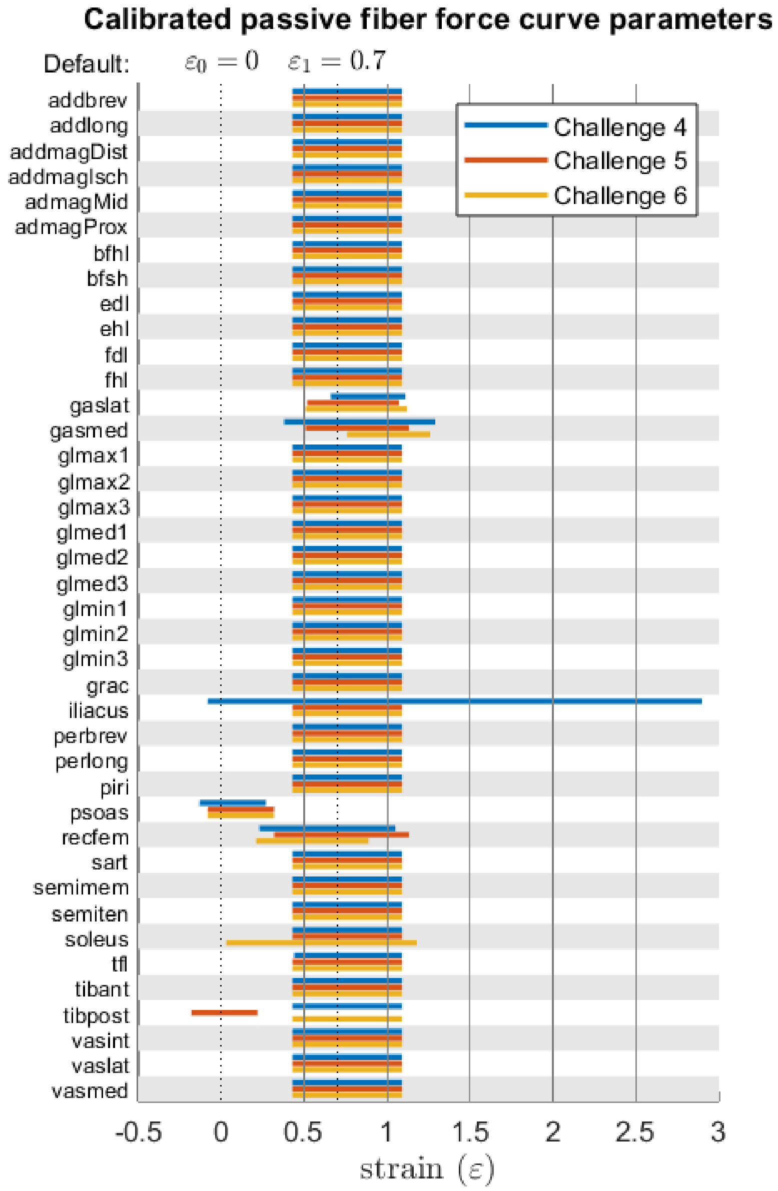


Figure D.12: Visualization of the calibrated passive fiber force curve parameters per muscle and challenge. The start of each bar represents the onset of passive fiber force (ε_0), while the end of each bar indicates the point of one times the max isometric force of the respective muscle (ε_1). The y-axis represents muscle strain (ε). The black dotted lines indicate the default curve parameters ($\varepsilon_0 = 0$ and $\varepsilon_1 = 0.7$), which are the starting values for the optimization. Calibration and plotting was done with help from the MATLAB script by Uhlrich *et al.* [73].

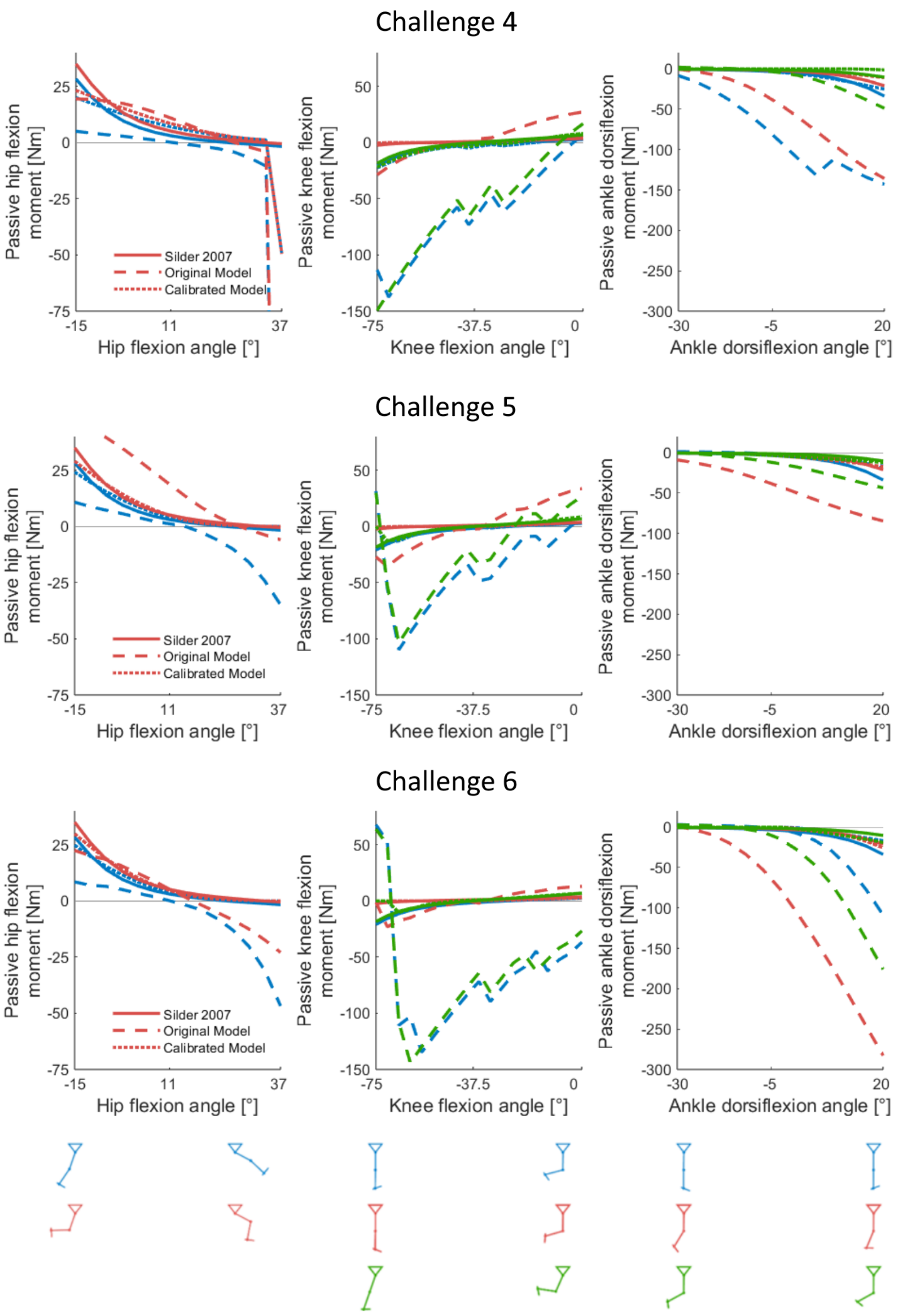


Figure D.13: Comparison of passive joint torques from experimental data and model specifications for each challenge. Solid lines represent experimental data from Silder *et al.* [74], dashed lines represent the original, uncalibrated model (specification A) and dotted lines represent the calibrated model (specification B). Line colors represent different poses for adjacent joints, according to the similar colored stick figures. It is clear that the original model exhibits unrealistically high passive joint moments, which have been corrected through calibration. Calibration and plotting was done with help from the MATLAB script by Uhlrich *et al.* [73].

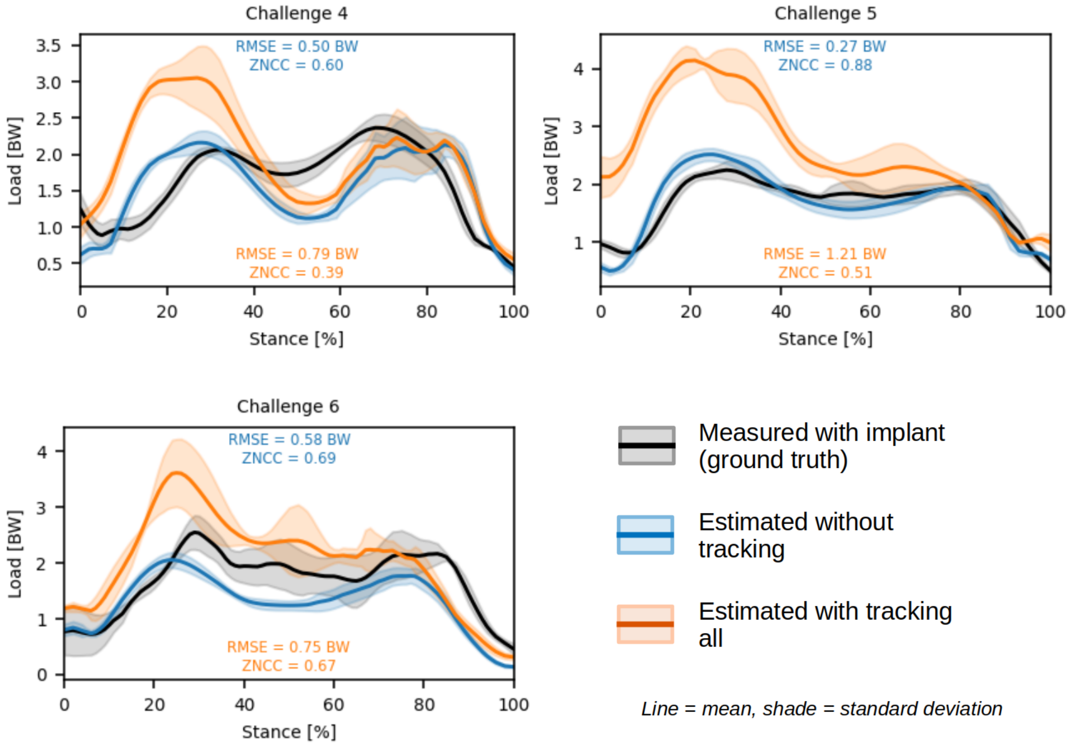


Figure D.14: Comparison of knee joint contact force estimations with and without tracking all available EMG. In **black** the implant measured knee contact force (ground truth), in **blue** the estimations without tracking and in **orange** the estimations with tracking of all muscles with available EMG. Both estimations ignore passive forces. Forces are normalized for bodyweight. One stance of the instrumented leg is taken from each trial. RMSE and ZNCC with respect to the ground truth are reported.

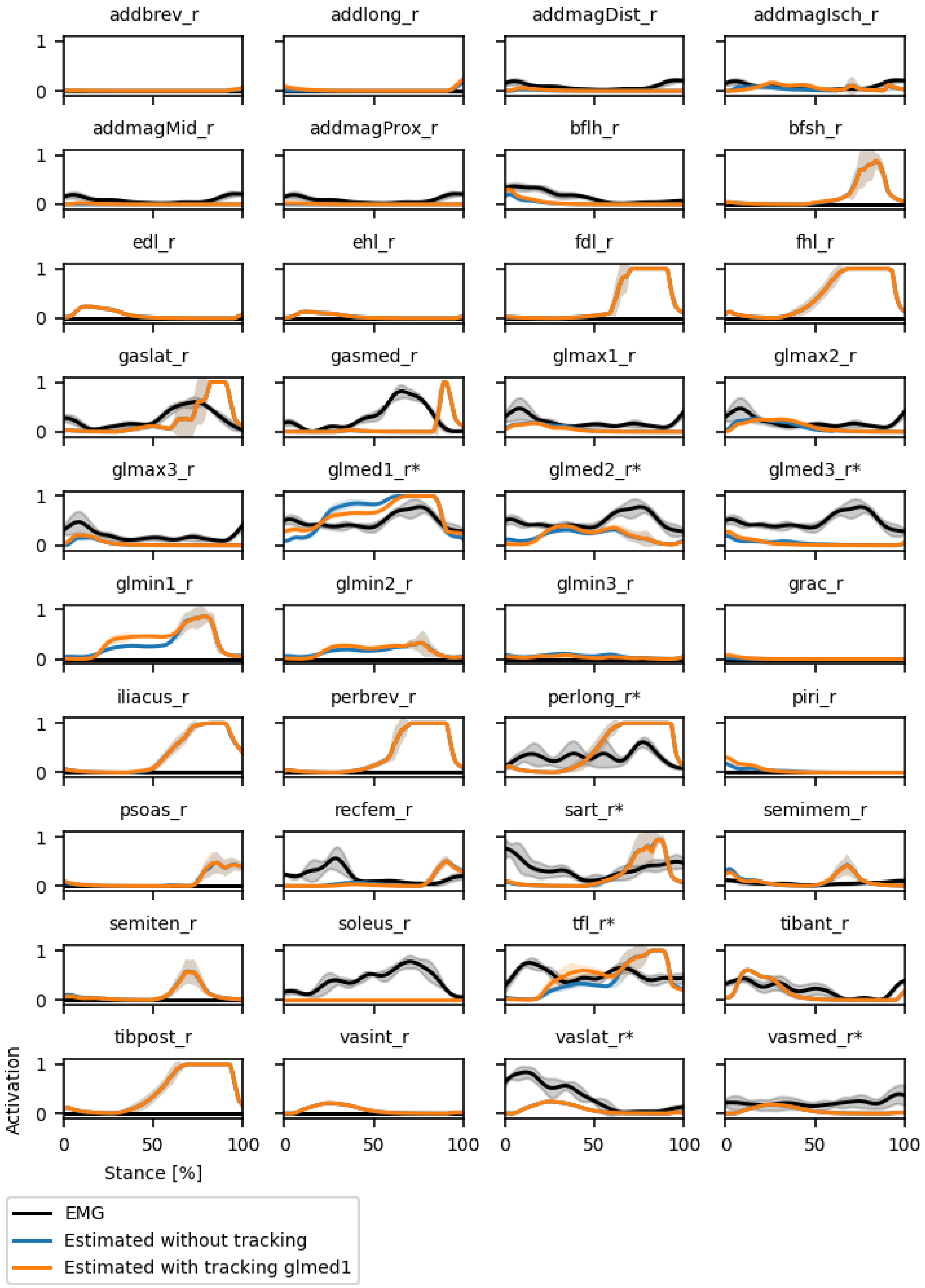


Figure D.15: Comparison of muscle activation estimations with and without tracking *glmed1* for challenge 4. In **black** pre-processed EMG (ground truth), in **blue** the estimations without tracking and in **orange** the estimations with tracking of *glmed1*. One stance of the instrumented leg is taken from each trial. Muscles indicated with * have been normalized to the highest value found in normal gait trials, see also table C.4.

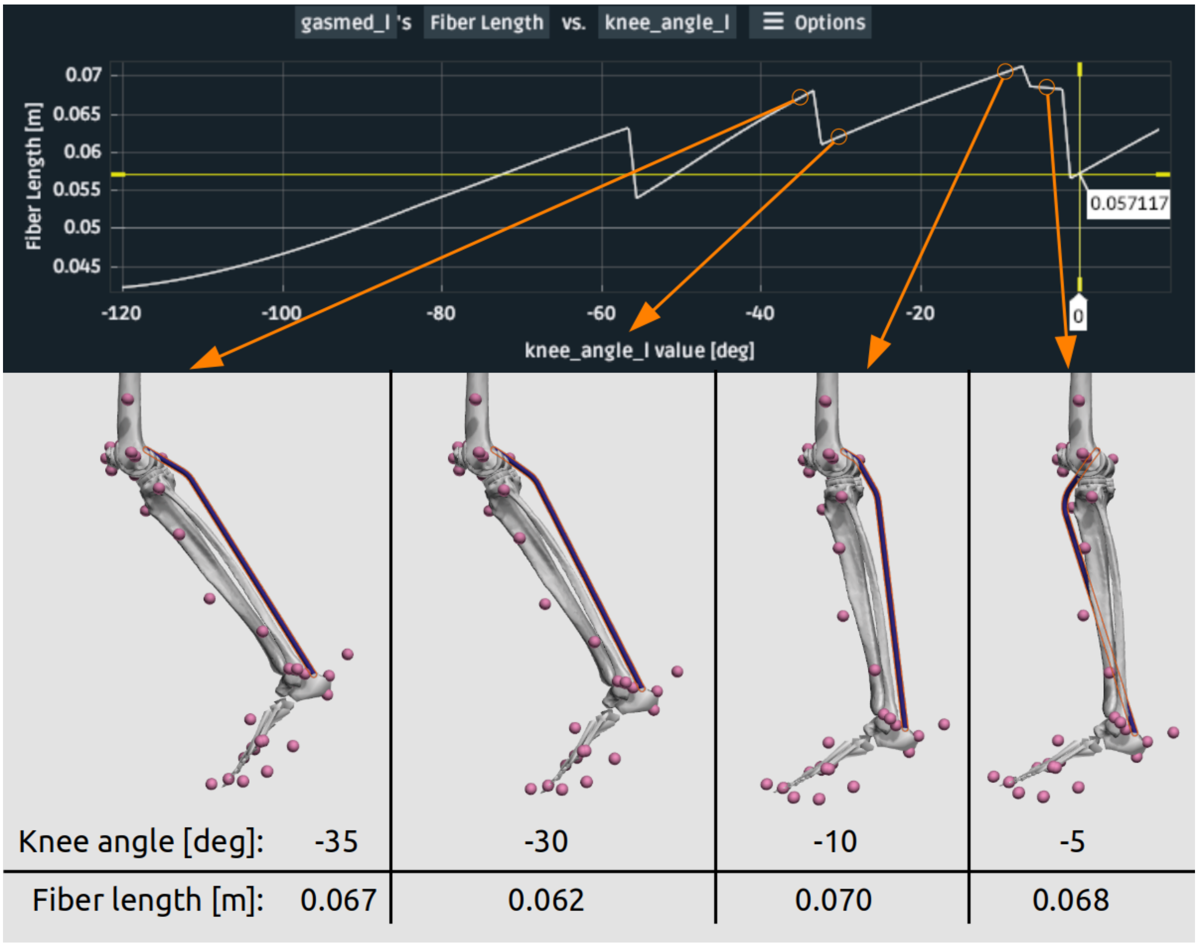


Figure D.16: The effect of knee angle on the fiber length of *gasmed* for the challenge 5 model. The jumps between 0 and -10deg are caused by a flip in the wrapping direction. The cause of the jump between -30 and -35deg is unknown (similarly for the jump around -55deg). Ankle angle is set to zero.