Musculoskeletal specialization for sprinting and distance running

# Methods

Differentiable musculoskeletal simulator

The state of a musculoskeletal model is determined by the activations of the 92 included lower limb muscles, the fiber lengths of these muscles , the activation of the eight torque actuators of the upper limb degrees of freedom , the generalized positions and velocities that include the six degrees-of-freedom of the pelvis and 25 joint angles:

The state derivatives are described by muscle activation dynamics:

muscle-tendon dynamics, which also determine tendon force:

torque actuator activation dynamics:

and skeleton dynamics:

with muscle excitations, tendon forces, muscle-tendon lengths, muscle-tendon velocities, the skeleton parameters, the muscle parameters, the torque actuator excitations, joint torques, the mass matrix, the vector of gravitational forces, the vector of Coriolis and centrifugal forces. We collect these equations into the system dynamics:

The joint torques are the result of the biological joint torques generated by the muscles , the torque actuators for the upper limbs , passive joint torques , and joint torques that result from contact :

with the 92x33 matrix of moment arms of the muscles with respect to the joints and the function describing the joint torques that result from contact. Contact is modelled using eight Hunt-Crossley contact spheres attached to the feet and the ground. The location and properties of these contact spheres are as in [ref]. The passive joint torques consist of joint limit torques for the lower limb joints as in Anderson et al. and as damper joint torques for the upper limb joints with a damping constant of 0.1 Nm.s/rad. Finally, for the metatarsophalangeal joint a spring-damper joint torque was added that had different parameters depending on whether sprinting (stiffness: 40 Nm/rad, damping: 0.4 Nm.s/rad) [Haralabidis et al.] or marathon running (stiffness: 25 Nm/rad, damping: 1.9 Nm.s/rad) [Falisse et al.] was simulated.

The skeleton parameters consist of three scaling factors for each body that scales that body in the three dimensions. As such the skeleton parameters change segmental geometrical and inertial properties and thus .

The computation of the muscle tendon lengths and the moment arm matrix is implemented as a neural network that takes the relative joint positions and skeleton parameters as input. OpenSim performs two non-differentiable operations to compute muscle tendon lengths and moment arms. The first step is to scale the musculoskeletal geometry. The second step is the actual computation which is an iterative, computationally costly and non-differentiable operation. We replaced this two-step process by a neural network:

The muscle tendon velocities are computed by applying the chain rule:

Finally, the Hill-type muscle parameters consist of: the muscle physiological cross sectional area (), the specific tension of muscle fibers (), tendon slack length (), optimal fiber length (), pennation angle () and tendon stiffness () [37]. The cross-sectional area and specific tension determine the muscle maximal isometric force:

From the PCSA and the optimal fiber length, the muscle volume can be calculated:

which is an input to several other computations such as for example the metabolic energy consumption.

When scaling the skeleton, the tendon slack length and optimal fiber length are adapted as well depending on the total length change of the muscle tendon unit length when the model is place in the anatomical pose **:**

To mimic strength training, we allow the physiological cross-sectional area of individual muscles to be increased.

Skeleton dynamics are derived from SimBody using an adapted version of the differentiable implementation by Falisse et al. [38]. The adaptation is needed to allow for differentiation with respect to .

Figure - Differentiable musculoskeletal simulator

Trajectory optimization

Each predictive simulation was solved as a trajectory optimization problem. We are simulating steady-state gait (running and sprinting) and assume musculoskeletal symmetry. As such we only need to simulate half a gait cycle if we impose symmetry and periodicity constraints for all states.

For every trajectory optimization problem we optimize at least the muscle and torque excitations and the initial state of the musculoskeletal system

To develop the four new musculoskeletal models we solved a trajectory optimization problem where the either the skeleton scaling parameters or the scaling of muscle physiological cross-sectional areas () were added to the optimization variables. For example, to generate SPRINT\_SKEL, we added to the optimization variables and optimized for the sprinting objective function. Altogether, we solve 10 trajectory optimization problems:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *SIM* | *Objective function* | *Optimization variables* |  |  | *Model* |
| 1 | sprint |  | 1 | 1 | GEN |
| 2 | marathon |  | 1 | 1 | GEN |
| 3 | sprint |  | [0.8; 1.2] | 1 | SPRINT\_SKEL |
| 4 | marathon |  | [0.8; 1.2] | 1 | MARATHON\_SKEL |
| 5 | sprint |  | SIM 4 | 1 | MARATHON\_SKEL |
| 6 | marathon |  | SIM 3 | 1 | SPRINT\_SKEL |
| 7 | sprint |  | 1 | [1; 1.2] | SPRINT\_MUSC |
| 8 | marathon |  | 1 | [1; 1.2] | MARATHON\_MUSC |
| 9 | sprint |  | 1 | SIM 8 | MARATHON\_MUSC |
| 10 | marathon |  | 1 | SIM 7 | SPRINT\_MUSC |

When we simulate the task of marathon running with the model optimized for sprinting, we impose the scaling parameters found to be optimal for sprinting while optimizing muscle and torque actuator excitations for marathon running. For example, in SIM 6 we impose the resulting of SIM 3 while optimizing for the marathon running objective.

**Direct collocation and implicit dynamics**

To improve numerical conditioning, we formulated muscle and skeleton dynamics with implicit rather than explicit differential equations. For the muscle contraction and skeleton dynamics, we introduced muscle velocities and coordinate accelerations as additional controls, and we imposed the nonlinear dynamic equations describing muscle contraction and skeleton dynamics as algebraic constraints in their implicit form.

We used direct collocation to transcribe each trajectory optimization problem into a large sparse nonlinear program. We used a third-order Radau quadrature collocation scheme with 50 mesh intervals per half gait cycle and solved the resulting NLP with the solver IPOPT.

Because we fix the number of mesh intervals for every simulation problem we make the mesh interval length a variable to accommodate for different possible stride lengths at a given speed.

**Constraints and bounds**

Muscle and torque excitations are bound to be between zero and one. We added bounds for the joint range of motion that were not reached for most degrees of freedom. The upper limb degrees of freedom were exceptions where the coordinates did reach their bounds as we did not model muscular or ligamentous structures to limit these. Similarly, hip inward (-10°) and outward rotation (+10°) as well as knee extension (0°) bounds were reached. For the remaining variables (muscle lengths, joint velocities, muscle velocities, joint accelerations) we added generous bounds that where not reached in any solution.

For the simulations of distance running we impose the average speed to be 3.33m/s.

We added path constraints to avoid penetration between body segments.

For simulations where the skeleton scaling parameters, are optimized, these are bounded between 0.8 and 1.2. When simulating strength training, the increase of individual muscle physiological cross-sectional areas was limited to 20% and the total increase in muscle volume summed over all muscles was limited to 5%.

**Objective function**

The minimal energy objective for marathon running was from Falisse et al. and consists of five main contributions:

with the muscle metabolic energy based on the Bhargava model of metabolic energy expenditure, the sum of squared muscle activations modelling muscle fatigue, the sum of squared joint accelerations modelling motion smoothness, the sum of squared limit joint torques modelling avoidance of ligament strain and the sum of squared upper limb torque actuator activations modelling upper body fatigue and energy expenditure.

A straightforward sprint objective to maximize the average velocity was chosen:

With a small contribution of the marathon energy term to improve the numerical condition of the optimization problem.