MYOSUITE
A CONTACT-RICH SIMULATION SUITE FOR MUSCULOSKELETAL MOTOR CONTROL

Vittorio Caggiano*
Meta AI Research, New York, NY, USA
caggiano@gmail.com

Huawei Wang*
University of Twente, the Netherlands
h.wang-2@utwente.nl

Guillaume Durandau
University of Twente, the Netherlands
g.v.durandau@utwente.nl

Massimo Sartori
University of Twente, the Netherlands
m.sartori@utwente.nl

Vikash Kumar
Meta AI Research, Pittsburgh, PA, USA
vikashplus@gmail.com

May 22, 2022

ABSTRACT

Embodied agents in continuous control domains have been traditionally exposed to tasks with limited opportunity to explore musculoskeletal details that enable agile and nimble behaviors in biological beings. The sophistication behind bio-musculoskeletal control not only poses new challenges for the learning community but realizing agents embedded in the same perception-action loop that the human sensory-motor system solves can also have a far-reaching impact in fields of neuro-motor disorders, rehabilitation, assistive technologies, as well as collaborative-robotics.

Human biomechanics is a complex multi-joint-multi-actuator musculoskeletal system. The sensory-motor system relies on a range of sensory-contact rich and proprioceptive inputs that define and condition motor actuation required to exhibit intelligent behaviors in the physical world. Current frameworks for studying musculoskeletal control do not include at the same time the needed physiological sophistication of the musculoskeletal systems and support physical world interaction capabilities. In addition, they are neither embedded in complex and skillful motor tasks nor are computationally effective and scalable to study motor learning in the timescale that current learning paradigms require.

To realize a platform where physiological detail and challenges behind human motor control can be investigated, we present MyoSuite – a suite of physiologically accurate biomechanical models of elbow, wrist, and hand, with physical contact capabilities which allow complex and skillful contact-rich real-world tasks. The implemented motor tasks provide a great variability of control challenges: from simple postural control to skilled hand-object interactions involving tasks like turning a key, twirling a pen, rotating two balls in one hand, etc. Finally, by supporting physiological alterations in musculoskeletal geometry (tendon transfer), assistive devices (exoskeleton assistance), and muscle contraction dynamics (muscle fatigue, sarcopenia), we present real-life tasks with temporal changes, thereby exposing realistic non-stationary conditions in our tasks which most continuous control benchmarks lack.
Project Webpage: https://sites.google.com/view/myosuite
Github: https://github.com/facebookresearch/myosuite
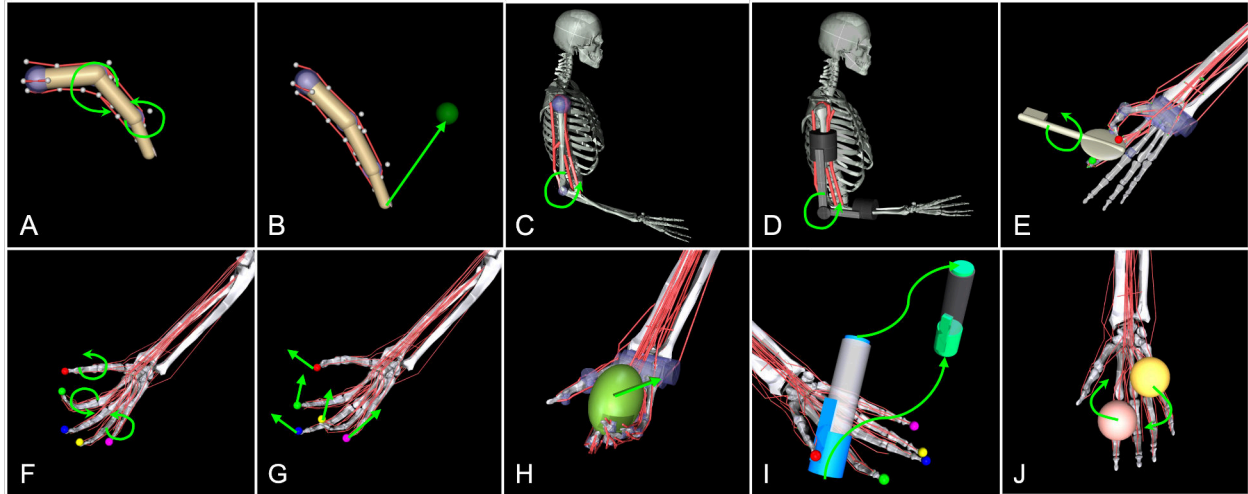
1   Introduction

Figure 1: MyoSuite tasks – **Finger**: joint pose (A), tip reach (B). **Elbow**: joint pose (C), exo-assisted (D). **Hand**: key turn (E), joint pose (F), tips reach (G), object reposition (H), pen-twirl (I), baoding balls (J).

Data-driven learning paradigms have enabled rapid progress in multiple fields such as vision ([15]), natural language processing ([71]), speech ([50]), among others. While there has been progress, the field of embodied AI is still awaiting its breakthrough moments. As opposed to fields in which decision making is one step open loop, embodied agents operate decisions in a close action-perception loop over time, which adds complexities to the problem and the solutions. Traditionally this has been approached exposing agents to simplified problems, that, although have pushed the field forward do not translate into real-world advancements. One of the root causes is that real-world problems are complex and non-stationary. The human central nervous system, on the other side, can easily handle non-stationary and complex tasks ([36]) by building control schemes that integrate and model proprioceptive and sensory information and transform them into optimal ([61, 58]) and adaptable ([75]) control policies. Such neuro-control challenges that the human central nervous system can seamlessly handle can pose as the next frontier for continuous control algorithmic paradigms. Furthermore, any development in such a critical problem space can translate into meaningful advancements in critical fields like neuromechanics, physiotherapy, rehabilitation, as well as robotics.

The musculoskeletal system presents more complex and challenging problems for embodied agents as opposed to virtual and robotic systems. Muscles can only generate forces in one direction (pulling, not pushing); and undergo changes in their force-generating capacity depending on their operating length, contractile velocity, or fatigue state ([4]). Moreover, muscles may undergo structural changes in response to aging e.g. sarcopenia, or to exercise, thereby further altering their contractile properties across time ([31]). Surgical interventions and assistive devices e.g. exoskeleton can also alter the muscle actuation. In addition, the muscle control space is also high dimensional (the number of muscles exceeds the number of human joints - about 600 muscles to control about 300 joints), redundant (multiple muscles act on the same joint), and multi-articular (muscles very often act on multiple joints) ([25, 67, 21]). As a result, the overall system suffers from the 'curse of dimensionality' ([5]) as well as non-stationarity challenges, which are less common in joint level control, which is typical in robotics ([75]).

The machine learning community has made major advancements in the embodied AI field by defining benchmarks and baselines. OpenAI-Gym ([7]) and DmControl ([66]) are the *de-facto* benchmarks for behavior synthesis in continuous control domains. These benchmarks however are not suitable for real-world problems. First, they consists mostly of simple problems[1] and are largely already solved. Second, those benchmarks have very limited capabilities to test the adaptability of an agent in response to non-stationary environment changes. Usually, those environment changes have been investigated via contrived examples - removing links ([47]) or adding spurious joints ([22, 16]). There is a dire need for new benchmarks embedded closely in the real world which, at the same time, create new challenges for algorithmic paradigms and translate into real-world impact.

---

[1]With some exceptions like *HandManipulateEgg* and *HandManipulatePen* ([51]) or in-hand manipulation of real-world objects ([48, 26, 9])

Previous attempts at establishing realistic musculoskeletal tasks as benchmarks ([64]) were narrowly defined. They favored bigger and functionally relevant muscle groups e.g. legs ([24, 55, 74, 1]) and arms ([14, 59, 57, 44, 38, 56]). These attempts relied on physics-based musculoskeltal simulation engines such as OpenSim ([59]), AnyBody ([12]) and [63] that although are widely used for simulating human neural-mechanical control, human robot interaction, and rehabilitation are computationally expensive (simulating large number of muscle is intractable) and provide limited support for contact rich interactions with their environments (see Table 1). While, physics engines used in the robotic field (PyBullet [10], MuJoCo [2] [68], IsaacGym [43], RaiSim [27], and Dart [37])[3] are relatively more efficient and support contact interactions,

	Contact Dynamics	Muscle Model	Speed
MuJoCo(MyoSuite)	✓	✓	↑
OpenSim		✓	↓
AnyBody		✓	↓
SIMM		✓	↓
PyBullet	✓		↑
Dart	✓	3	↑
RaiSim	✓	3	↑

Table 1: Physic simulators. MyoSuite provides physiologically realistic and computationally efficient musculoskeletal models in the MuJoCo physics engine.

but lack adequate support for modeling anatomical and functionally validated musculoskeletal models (see Table 1).

In order to expose the community to the exciting challenges presented by the musculoskeletal control of biological systems, we present a physiologically realistic and computationally efficient framework: **MyoSuite**.

Our contributions:

- A fully automated model-agnostic pipeline which reduce the tedious manual effort often required in creating new models by optimizing existing and validated (OpenSim) musculoskeletal models. The resulting models are numerically equivalent to the original and can be simulated two orders of magnitude faster than the original models

- We developed a set of musculoskeletal models (from simple one joint to complex full hand) in MuJoCo that are physiologically accurate, and supports full contact dynamics.

- We designed a family of 9 realistic dexterous manipulation tasks (from simple posing to simultaneous manipulation of two Baoding balls) using these models. These task families take inspiration from the state of art robotic dexterous manipulation results ([53, 46, 3]).

- MyoSuite supports physiological alterations in musculoskeletal geometry (tendon transfer, exoskeleton assistance) and muscle contraction dynamics (muscle fatigue, sarcopenia) to expose real-life tasks with temporal changes, thereby subjecting self-adapting algorithms to the realistic non-stationary challenges under continuous control settings.

- In its present form MyoSuite consists of 204 tasks: 9 task-families x 2 difficulty level (easy, hard) x 3 reset conditions (fixed, random, and none[4]) x 8 (or 4) combinations of non-stationarity variations. We present some baseline results on MyoSuite outlining its features and complexities.

2 Preliminaries

Behavior synthesis for embodied agents can be formulated as a sequential decision-making problem where the goal is to generate a coordinated sequence of trajectories that allows the agent to achieve desired outcomes. We first provide a brief overview of the musculoskeletal system before outlining how its behavior synthesis can be formulated using Markov Decision Processes (MDP) formulations under the Reinforcement Learning (RL) paradigm.

2.1 Musculoskeletal models

Musculoskeletal systems consist of bones of various lengths connected together via a redundant system of skeletal muscles and tendons. Unlike robotic systems, which can be viably investigated by directly controlling and analyzing the resulting samples (e.g. parameter identification is solved comprehensively), mathematical models for biological systems are based, however, on sparse data available from cadavers (e.g. not all parameters can be identified leading to inherent uncertainty). Owing to the limitations of live experimentation, musculoskeletal models have been the cornerstone of

---

[2]Contains a basic implementation of a muscle model that MyoSuite builds upon and it has been used to develop musculoskeletal models of animals e.g. [34]

[3]Efforts have been made to add muscle models in both DART [39] and RaiSim [78]

[4]To facilitate investigation in reset-free algorithms such as [23]

most investigation and understanding behind biological motor control. Musculoskeletal models are commonly modeled as a 3rd order system which contains first or second-order muscle activation and contractile dynamics as well as the second-order body dynamics.

A typical muscle-tendon unit consists of contractile fibers and series-elastic tendons (see Figure 2A). Contractile fibers are the active actuating elements of a muscle-tendon unit that are controlled by the signals generated from alpha motor neurons (α -MNs) in the spinal cord. A widely used mathematical model for this is the Hill-type model (see Figure 2B). It typically contains three elements: a contractile element (CE, Figure 2B), a parallel elastic component (fiber stiffness, PEE, Figure 2B), and a series elastic component (tendon stiffness, SEE, Figure 2B). Muscle forces are functions of both the muscle lengths and velocities, due to the force-length and force-velocity properties (see Figure 2B, Equation 1a,b). In MuJoCo, tendons were assumed infinitely stiff. A first order dynamics is normally used to represent the muscle twitch response (e.g. activation and and deactivation dynamics, see Equation 1c). Muscle-tendon units project mechanical forces onto skeletal bones. Geometrically, muscle-tendon units normally originates and insert with attachments points spanning different segments/bones, thereby contributing to move the joints in between by pulling segments/bones closer to each other (see Figure 2A). Dynamics of the skeleton system follows the Newton's laws and it is normally written in the format of dynamic equations (see Equation 1d). All together, a complete musculoskeletal model can be defined as a 3rd order system.
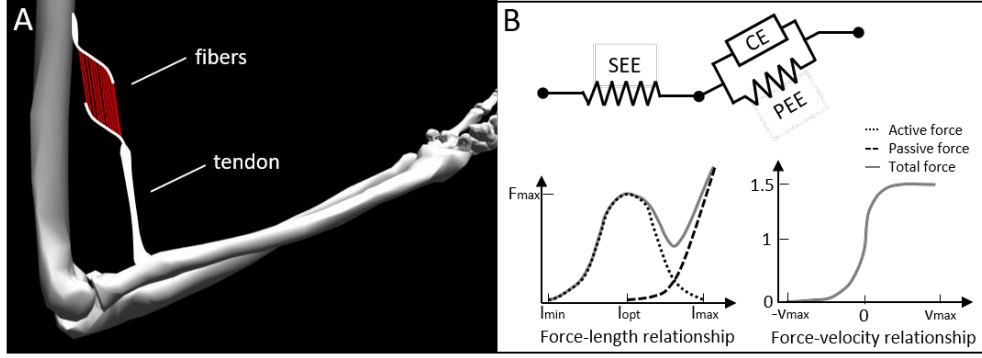

Figure 2: Example of muscle structure (A) and the Hill-Type model (B).

$$F_{mtu} = F_m \cos \alpha \tag{1a}$$

$$F_m = a \cdot f_a(l) \cdot f_v(\dot{l}) + f_p(l) \tag{1b}$$

$$\dot{a} = (u - a)/\tau(u, a) \tag{1c}$$

$$M \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q) = F_{mtu} \cdot r \tag{1d}$$

where, F_{mtu} represents the muscle-tendon unit force; α represents the pennation angle between tendon and muscle fibers; F_m represents the muscle force; a represents the muscle activation; f_a is the active muscle force-length function; f_v is the muscle force-velocity function; f_p is the passive muscle force-length function; l represents the muscle length; $\dot{l}$ represents the muscle velocity; u represents the neural control signal; $\tau(u, a)$ is the time constant of the muscle activation and the deactivation dynamics; M is the Mass/inertia matrix of the skeleton system; $q, \dot{q}, \ddot{q}$ are the joint angles, velocities, and accelerations; $C(q, \dot{q})$ is the Coriolis force term; $G(q)$ is the gravity force term; r represents the moment arm of the muscle at the contributing joint.

2.2 Markov decision process (MDP) formulation

In RL paradigms, we operate in a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho, \gamma)$. Per usual notation, $\mathcal{S} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{R}^m$ represent the continuous state and action spaces respectively. The unknown transition dynamics is described by $s' \sim \mathcal{T}(\cdot|s, a)$. $\mathcal{R} : \mathcal{S} \to [0, R_{\max}]$, $\gamma \in [0, 1)$, and ρ represents the reward function, discount factor, and initial state distribution respectively. Policy is a mapping from states to a probability distribution over actions, i.e. $\pi : \mathcal{S} \to P(\mathcal{A})$, which is parameterized by θ. The goal of the agent is to learn a policy $\pi_\theta^*(a|s) = argmax_\theta[J(\pi, \mathcal{M})]$, where $J = \max_\theta \mathbb{E}_{s_0 \sim \rho(s), a \sim \pi_\theta(a_t|s_t)}[\sum_t R(s_t, a_t)]$ i.e. the expected sum of discounted rewards in an episodic setting. Policy gradient algorithms ([62]), TD-learning based algorithms, such as Q-learning ([73]), SARSA ([65]), actor-critic based methods ([32]), etc. can be leveraged to optimize J to generate behaviors.

2.3 MDP formulation for Robotic systems vs. Musculoskeletal systems

Robotic systems: Under our MDP characterization $\mathcal{M}$, state space $\mathcal{S}$ consists of $\{joint\ position,\ joint\ velocity\}$, action space $\mathcal{A}$ consists of actuator's $\{position/velocity/torque\ demands\}$ and samples for policy optimization, are gathered either directly from the real world transition $\mathcal{T}_{real}$, or via physics simulation engines $\mathcal{T}_{sim}$ ([68, 10, 37]).

Musculoskeltal systems: Under the MDP characterization $\mathcal{M}$, state space $\mathcal{S}$ consists of $\{muscle - tendon\ length,\ muscle - tendon\ velocity,\ muscle\ activations\}$, action space $\mathcal{A}$ consists of actuator's $\{\alpha - Motoneurons\ signals\}^5$ samples for policy optimization, are gathered from the physics simulation engines $\mathcal{T}_{sim}$ controlled via muscle actuators.

3 MyoSuite

Unlike robotic counterparts, which are double acting and usually have one-to-one relationships between joints and actuators, tendons and muscles in musculoskeletal systems act via contraction (pull-only) and span multiple joints inducing strong coupling between them. Muscles become fatigued with extended usage. Tendons transfer muscle forces to bones while serving as temporary energy storage units for motion efficiency. These details, while complex, conceal within themselves the ingredients of effective motor control in biological systems. To facilitate investigations in these details, we present the *"MyoSuite"* (see Figure 3) which consists of a collection of physiologically accurate musculoskeletal models (sec 3.1) and physically realistic contact-rich tasks (sec 3.2) of varying complexity.

Figure 3: MyoSuite offers a playground to test several physiologically accurate musculoskeletal models challenged to perform skillful and contact-rich tasks.

3.1 Models

Our models are developed using a thorough investigation of well studied existing models ([14, 59, 57, 44, 38, 56]) and functional studies ([76]). We started from OpenSim models ([38, 44, 57]) of the arm and hand which are widely used in fields of human neural-mechanical control, human-robot interaction, and rehabilitation, among others. In order to implement those models in MuJoCo, we developed a pipeline (see Appendix A) to perform geometry transformations of bones and muscles attachment, moment arm optimization, and muscle force optimization.

A pipeline - MyoSim (see Appendix A) - was developed to convert OpenSim musculoskeletal models into an equivalent MuJoCo model. This pipeline - based on three steps - enabled us to maximize the similarities between the newly developed MuJoCo models with the referencing OpenSim models, despite the difference in the wrapping surfaces, and the muscle dynamic definitions. It is worth noting that, these three steps are not specifically for the models presented in this benchmark, but are rather generalizable, i.e. then can be translated to other models. In addition, the proposed

---

[5]Motorneurons are the final neuronal stage that connects the central nervous system to the muscles.

pipeline can operate based on data derived from cadaver data or from functional tasks, thereby allowing flexibility in building accurate musculoskeletal models directly in MuJoCo. These three steps are:

1. **Geometry Transfer:** The transformation of segment (bone) geometries, muscle attachment points, and wrapping surfaces, from an OpenSim model to a MuJoCo model. This step was based on extensions of previous work from [28].

2. **Moment Arm Optimization:** The optimization of the 3D positions of side sites for the wrapping surfaces in MuJoCo, so that matching moment arms can be achieved with respect to the reference OpenSim model.

3. **Muscle Force Optimization:** The optimization of the muscle parameters in MuJoCo, so that matching muscle force generating-capacity can be achieved with respect to the reference OpenSim model.

After rigorous modeling and validation, we built three models of varying complexities.
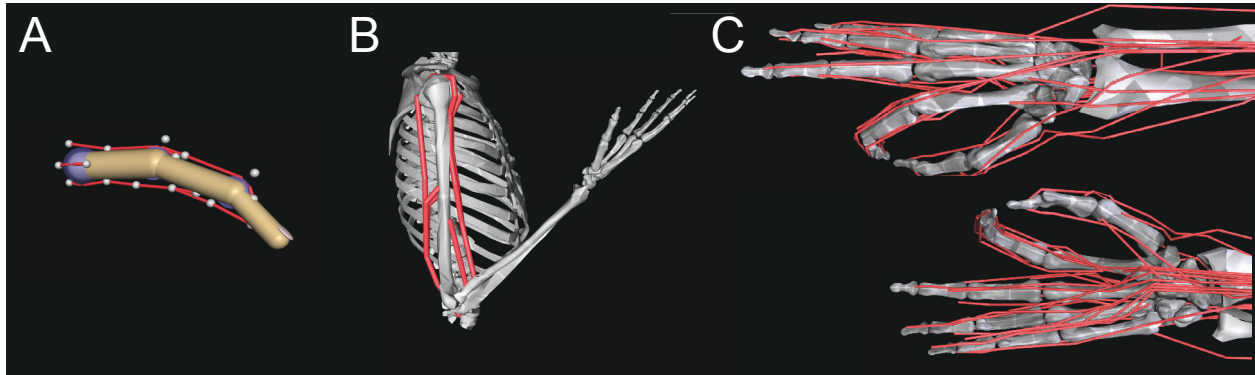

Figure 4: Musculoskeletal models included in MyoSuite. A - MyoFinger (4 joints - 5 muscles), B - MyoElbow: (1 joint - 6 muscles), C - MyoHand: (23 joints - 39 muscles).

MyoFinger: First, we implemented a simplified and intuitive model (based on [77]) of a 4 Degree of Freedom (DoF) finger (MyoFinger, Figure 4A), which is actuated through a series of 5 simplified antagonistic muscle-tendon units. We also provide its robotic counterpart with simple torque actuators to facilitate the comparative investigation.

MyoElbow: A model of 1 DoF human elbow joint (Figure 4B). MyoElbow model is based on OpenSim's default testing arm model ([14, 59]) and actuated using multiple agonist/antagonist pairs (3 flexors and 3 extensors).

MyoHand: The dexterous human hand requires coordination of multiple highly redundant muscles, which have complementary and antagonistic effects on various joints. This more complex musculoskeletal model is comprised of 29 bones, 23 joints, and 39 muscles-tendon units (see section and Appendix 3 for a detailed description of the muscles). This forearm-wrist-hand model (MyoHand, Figure 4C) was based on 2 popular OpenSim models: MoBL - human upper extremity model ([57] [44]) - and the 2nd-Hand - for hand and fingers models ([38]). Muscle attachments and range of operation in the original model were based on cadaveric studies [2, 6, 33, 35, 40, 45, 49]. Additionally, (a) our model also includes an Opponens Pollicis muscle for the critical role that it has in manual dexterity [30], and (b) leverages functional studies [76] to recover the force-length characteristics of the Opponens Pollicis, Radial Interosseous, and the Lumbrical Palmar (or Ulnar Interosseous), while defaulting to MoBL and 2nd-Hand for others.

3.2 Tasks

Leveraging these musculoskeletal models, we built a series of tasks (see Figure 1) of varying difficulty. The task difficulty is varied along two axes: task-complexity, and task-non-stationarity. Task-difficulty has two variations (*complexity* - easy/hard, and *Reset* - Fix/Random/None), and task-non-stationarity has 8 (or 4 if tendon-transfer and exoskeleton assistance are not possible) variations. In total, in its current form MyoSuite consists of 204 tasks (Table 2): 9 task-families x 2 difficulties-levels x 3 Resets x 8 (or 4) combinations of non-stationarity variations.

The definition of the tasks follows the ones presented in the github repository https://github.com/facebookresearch/myosuite.

Finger Joint Pose: In the finger joint pose task, the MyoFinger model (Section 3.1, Figure 4A) needed to be controlled to reach a given pose. In the easy version (*myoFingerPoseFixed-v0*) the target joint pose was fixed, while in the hard condition (*myoFingerPoseRandom-v0*) the target pose was produced by a random configuration of the joints.

Finger Tip Reach: In the finger tip reach task, the MyoFinger (Section 3.1, Figure 4B) tip needed to reach a fixed position - easy (*myoFingerReachFixed-v0*) - or a random position - hard (*myoFingerReachRandom-v0*) - in the working space.

Elbow Flexion: The MyoElbow model (Section 3.1, Figure 1C) was simplified to have only elbow rotations. Although it is not a physiologically accurate model it can be a very simple model for troubleshooting initial control schemes. There are 2 versions of this task: reaching a random posture either with one muscle (*myoElbowPose1D6MFixed-v0*) or with 6 muscles (*myoElbowPose1D6MRandom-v0*)

Key Turn: A key turn task (Figure 1E) where MyoHand finger movements need to be coordinated to rotate a key. There are two versions of this task of different difficulty: easy, to reach a half rotation of the key (*myoHandKeyTurnFixed-v0*), and difficult, to reach full rotation of the key from different random position and rotations (*myoHandKeyTurnRandom-v0*)

Hand Tips Pose: In this task, the MyoHand model (Section 3.1, Figure 1G) needed to be controlled to have the joint reach either a fixed pose - easy (*myoHandPoseFixed-v0-v0*) - or a random pose - hard (*myoHandPoseRandom-v0-v0*).

Hand Tips Reach: The fingers of the MyoHand model (Section 3.1, Figure 1F) needed to be controlled to have the fingers tips reach either a fixed position - easy (*myoHandReachFixed-v0*) - or a random position in their workspace - hard (*myoHandReachRandom-v0*).

Hand Object Hold: The MyoHand model (Section 3.1, Figure 1H) was controlled to maintain an object balanced in the hand without letting it fall. The object could be placed in a fixed position in the palm - easy (*HandObjHoldFixed-v0*) - or a random position - hard (*myoHandObjHoldRandom-v0*).

Pen Twirling: The MyoHand needed to be controlled to rotate a pen in the hand to reach a given orientation (indicated by the green object in the scene) a given orientation without making if falling off (Figure 1I). The complexity of this task is due to the intermittent contacts between the object and multiple fingers while trying to stabilize the object. There are two versions of this task of different difficulty: easy, target in a fixed orientation of the target (*myoHandPenTwirlFixed-v0*), and difficult, target in a random orientation (*myoHandPenTwirlRandom-v0*).

Baoding balls: Finally, a highly dexterous baoding ball task (Figure 1J) involving simultaneous relative rotation around each other of two free-floating spheres over the palm. There are two versions of this task of different difficulty: easy, swap the positions of the ball (*myoHandBaodingFixed-v1*) - and difficult - achieve continuous rotations of the balls (*myoHandBaodingRandom-v1*).

	Complexity		Non-Stationarity				
	Easy/Hard	Reset	None	Sarcopenia	Fatigue	Tendon-transf.	Exo.
Finger Joint Pose	✓/✓	F/R/N	✓	✓	✓		
Finger Tip Reach	✓/✓	F/R/N	✓	✓	✓		
Elbow Joint Pose	-/✓	F/R/N	✓	✓	✓		✓
Hand Key Turn	✓/✓	F/R/N	✓	✓	✓	✓	
Hand Joints Pose	✓/✓	F/R/N	✓	✓	✓	✓	
Hand Tips Reach	✓/✓	F/R/N	✓	✓	✓	✓	
Hand Object Hold	✓/✓	F/R/N	✓	✓	✓	✓	
Hand Pen Twirl	✓/✓	F/R/N	✓	✓	✓	✓	
Hand Baoding Balls	✓/✓	F/R/N	✓	✓	✓	✓	

Table 2: List of different tasks in the MyoSuite with both the complexity (easy/hard and Reset - **F**ix, **R**andom, **N**one) and the non-stationarities (None, Sarcopenia, Fatigue, Tendon-transfer and exoskeleton) for each of them.

Next we detail various non-stationarities that are supported in MyoSuite tasks.

3.3   Realistic non-stationary task-variations

Muscle properties are constantly changing. These changes can be instantaneous - like for musculoskeletal injury or surgery - or can vary over a short time frame - like muscle fatigue or exoskeleton assistance. To study neuromuscular adaptation to non-stationarities due to these changes during the real work-life scenario, four different variations in muscle properties have been included: Sarcopenia, tendon transfer, Fatigue, and exoskeleton assistance.

Sarcopenia: Sarcopenia is a muscle disorder that occurs commonly in the elderly population ([11]) and characterized by a reduction in muscle mass or volume. The peak in grip strength can be reduced up to 50% from age 20 to 40 ([17]). We modeled sarcopenia for each muscle as a reduction of 50% of its maximal isometric force.
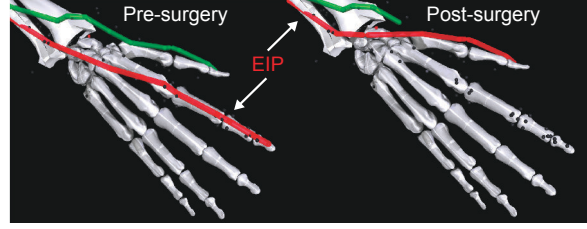
Figure 5: Tendon transfer. Left, EIP (in red - control the Index extensor) and EPL (in green - control the thumb extensor) muscles on an intact hand are shown. Right, the muscle path of EIP is re-routed on the EPL to control the activity of the thumb.

Effect of Fatigue: Muscle Fatigue is a short-term (second to minutes) effect that happens after sustained or repetitive voluntary movement and it has been linked to traumas e.g. cumulative trauma disorder ([8]). A dynamic muscle fatigue model ([42]) was integrated into the modeling framework. This model was based on the idea that different types of muscle fibers have different contributions and resistance to fatigue ([70]). The implemented model mainly considers the current maximal force production of the muscle and is updated using the following equation:

$$F_{upd}^{Max}(t) = F^{Max} \exp\left(k_{fatigue} * \int_0^t -\frac{F_m^{Act}(u)}{F^{Max}} du\right) \qquad (2)$$

With $F_{upd}^{Max}(t)$ the current updated maximal muscle force, F^{Max} the maximal without fatigue, F_m^{Act} the active part of the muscle force and $k_{fatigue}$ a fatigue coefficient with a value of 1.

Tendon transfer pre/post surgery: Contrary to muscle fatigue or sarcopenia that occurs to all muscles, tendon transfer surgery can target a single muscle-tendon unit. Tendon transfer surgery allows redirecting the application point of muscle forces from one joint DoF to another (see Figure 5). It can be used to regain functional control of a joint or limb motion after injury. One of the current procedures in the hand is the tendon transfer of Extensor Indicis Proprius (EIP) to replace the Extensor Pollicis Longus (EPL) ([20]). Rupture of the EPL can happen after a broken wrist and create a loss of control of the Thumb extension. We introduce a physical tendon transfer where the EIP application point of the tendon was moved from the index to the thumb and the EPL was removed (see Figure 5).

Exoskeleton assistance: Human-Machine interaction simulations were generated as a use-case to illustrate how musculoskeletal model response to the exoskeleton assistance can be studied. An elbow soft exoskeleton (Cable driven for example) was modeled as an ideal torque actuator perfectly aligned with the elbow joint with a weight of 0.101 Kg for the upper arm and 0.111 Kg on the forearm (Fig. 1-D). The assistance given by the exoskeleton was a percentage of the biological joint torque, which represents a ideal version of the controller presented in [19, 18].

4   Experiments

In this section, we first present results of the conversion pipeline to convert models from OpenSim to MuJoCo. Then, we present some baseline results on training reinforcement learning agents to solve some of the available tasks. Finally, we show some results when we train agents to solve tasks in presence of intrinsic non-stationary perturbation and while co-learning via human-robot interactions.

4.1   Models validation

A 2 joints 6 muscles (TRILong - Triceps Long, TRILat - Triceps Lateral, TRImed - TricepsMedial, BIClong - Biceps Long, BICshort - Biceps Short , BRA - Brachialis) OpenSim elbow model [14] is used to validate the conversion pipeline. The three steps validation is shown in Figure 6. The first column shows the model appearances in OpenSim (top) and in MuJoCo (bottom). Second column indicates the matches of markers in forward kinematics checks. As shown in the plot, two selected markers have identical locations in both models, which means the match of geometry and joint definitions. The third column shows the results of moment arm validation. Compared to the OpenSim model, the converted MuJoCo model after Cvt1 (Mjc_Cvt1) has large differences (the orange dash line with triangle markers), however after the second conversion step Cvt2, the moment arm differences between the OpenSim model and the MuJoCo model (Mjc_Cvt2) is greatly reduced (the green dashed lines are overlapping on the blue lines). The fourth column shows the validation of muscle forces. A large reduction of differences of muscle force was achieved also by applying the third step of the pipeline. Results of this last step are indicated by the matches between the blue solid line (OpenSim model), and the green dash line (converted MuJoCo model after the third step Mjc_Cvt3). The yellow dash
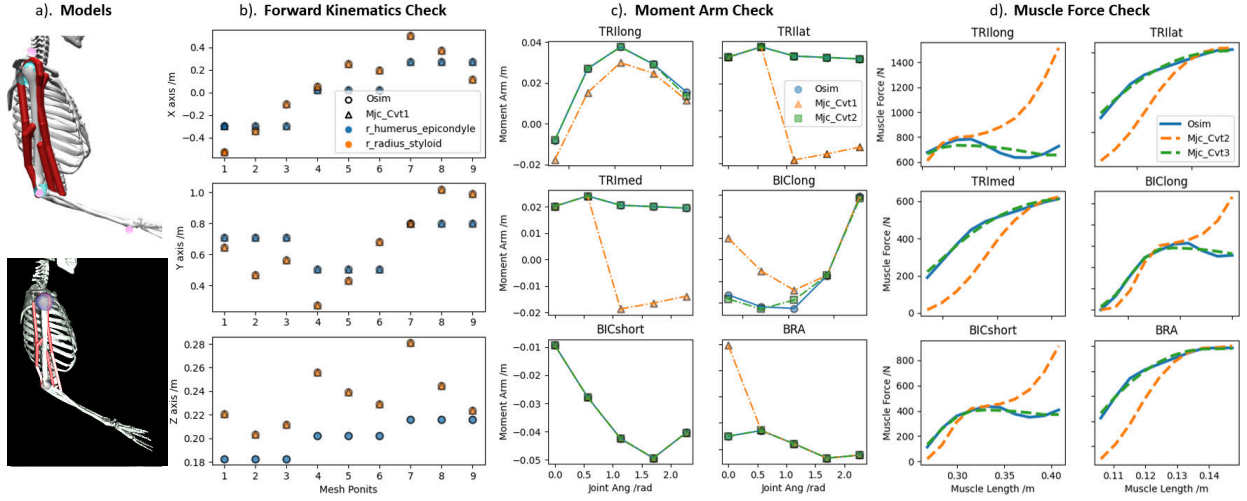
Figure 6: Validation between the converted MuJoCo and reference OpenSim models. a) The two models, top: OpenSim; bottom: MuJoCo. b) The forward kinematics validation. c) The moment arm validation. d). The muscle force validation. Muscle acronyms TRIlong - Triceps Long, TRIlat - Triceps Lateral, TRImed - TricepsMedial, BIClong - Biceps Long, BICshort - Biceps Short , BRA - Brachialis
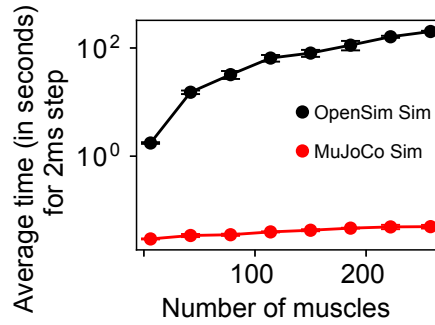

Figure 7: Performance comparison between MuJoCo and OpenSim with respect to the complexity of the model being simulated.

line (converted MuJoCo model shows the intermediate step Mjc_Cvt2). Similar validation metrics and trends were observed in the full hand model as well.

We compared the MuJoCo models against the correspondent OpenSim implementation. Muscle moment arm root mean square (RMS) differences between the MuJoCo model with respect to the Opensim model were $0.044 \pm 0.09\%$ for the elbow model (see Figure 6) and $0.38 \pm 0.57\%$ for the hand model. Also, the RMS error in forces was $2.2 \pm 1.4\%$ F_{max} (OpenSim peak force) for the elbow model and $4.1 \pm 2.0\%$ F_{max} for the hand model (see Figure 6). Those errors indicate that the MuJoCo models are anatomically and dynamically similar to the SOTA OpenSim model. Forward simulations showed that MuJoCo models can be several orders of magnitude faster than OpenSim (see Figure 7, from 60x to 4000x faster). By simulating an elbow model (6 muscles) where we iteratively replicated all muscles, it was possible to observe that the OpenSim computing time increased exponentially while the MuJoCo did not (see Figure 7). This increase in efficiency is mostly the result of a simplified implementation of the muscle actuator in MuJoCo, which allows faster and more stable simulations.

4.2 Baselines

We present some baseline results obtained using Natural Policy Gradient (NPG, [29] for a subset of the conditions available i.e. stationary easy and hard with Fix reset complexity. We chose this algorithm as it has been showing recently SOTA results in dexterous manipulation tasks ([54]). In Figure 8, we show success rates for the different tasks up to 5M samples. More complex tasks e.g. Baoding ball can be solved, albeit with much higher sample complexity. This success rate was reflected also in behavioral relevant movements. For example, in Figure 9A, it is shown a sequence of snapshots of the solution of the key turning task. It is possible to see how the index and thumb activation is functional

to the effective rotation of the key. Also, in Figure 9B a sequence of snapshots of the Pen Twirl task is shown. Indeed, in this task, it is visible how the coordination of wrist and hand joints make it possible to rotate the pen in the hand to reach the desired position. Finally, in Figure 9C a sequence of snapshots of the Baoding tasks show the complex coordination between wrist and fingers to move and rotate the balls against each other. It has to be noticed that solving the Baoding ball tasks was achieved with a greater number of samples i.e. more than 70M, and only occasionally i.e. one seed out of 5.
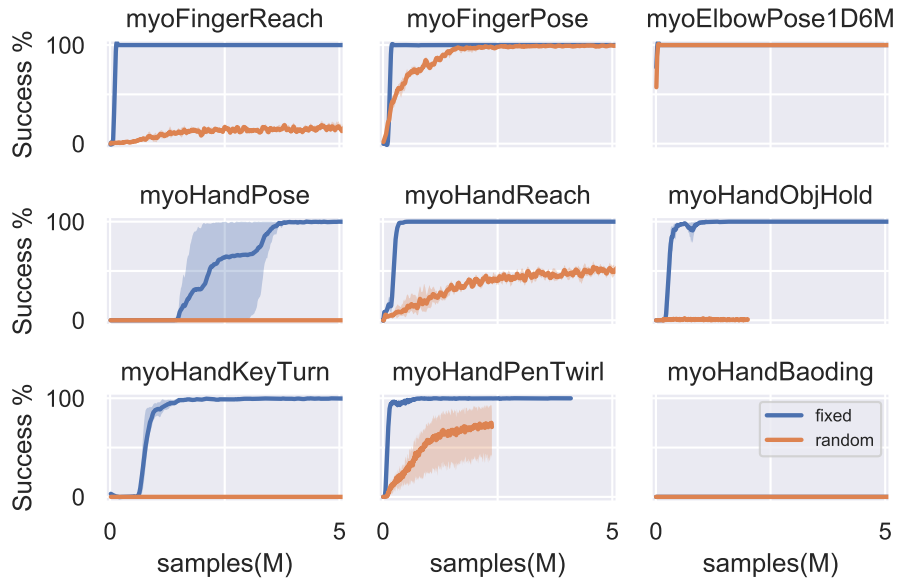

Figure 8: Tasks performance in the easy (fixed) and the hard (random) conditions with fixed reset.
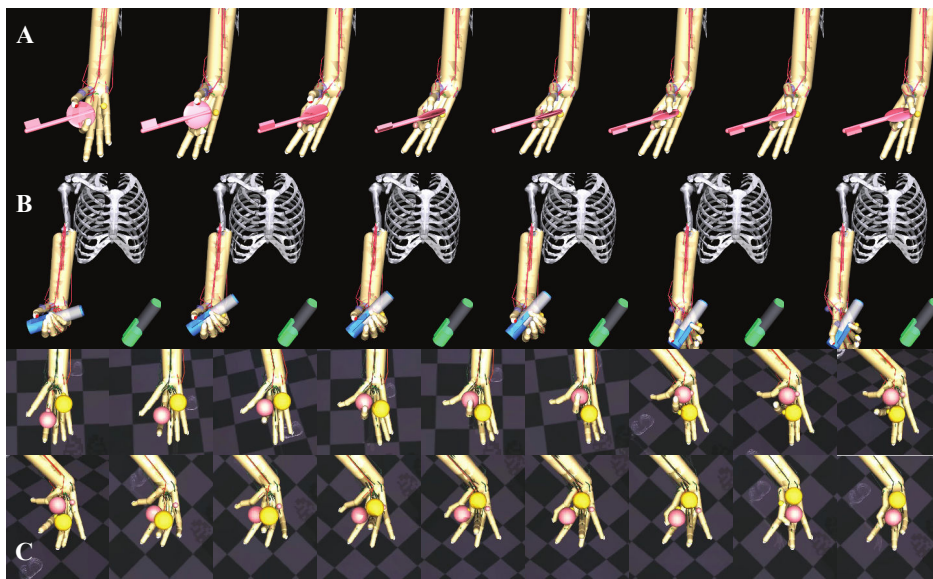

Figure 9: Frames from a selection of tasks. Complex wrist and finger coordination to solve the Key task (A) the Pen Twirl task (B) and the Baoding Balls task (C).

4.3   Intrinsic non-stationary perturbations

While extrinsic perturbation can be easily added to each task, we show results testing the above policies with a series of intrinsic non-stationary perturbations compatible with real-life scenarios.

4.3.1 Control in presence of intrinsic non-stationary perturbation: synergistic muscles contributions

We tested a policy trained on a 1D elbow flexion task to reach random targets in its operational space on alternated movements between points A and B (see Figure 10A).

First, we study how sarcopenia (muscle weakness) affects the control of movement. In presence of Sarcopenia (Figure 10B) the Brachioradialis - which in normal conditions can solve the task without the support of other muscles - needs stronger activations and the support of synergistic muscles (BICLong - biceps longus - and BICShort - biceps short) to solve the task.

Second, we investigated the effect of muscle fatigue (introduced in section 3.3). In this case, the loss of muscle power is progressive over time. In the same alternated movements elbow tasks, we observe a gradual contribution of synergistic muscles to compensate for the muscle force loss (Figure 10C).
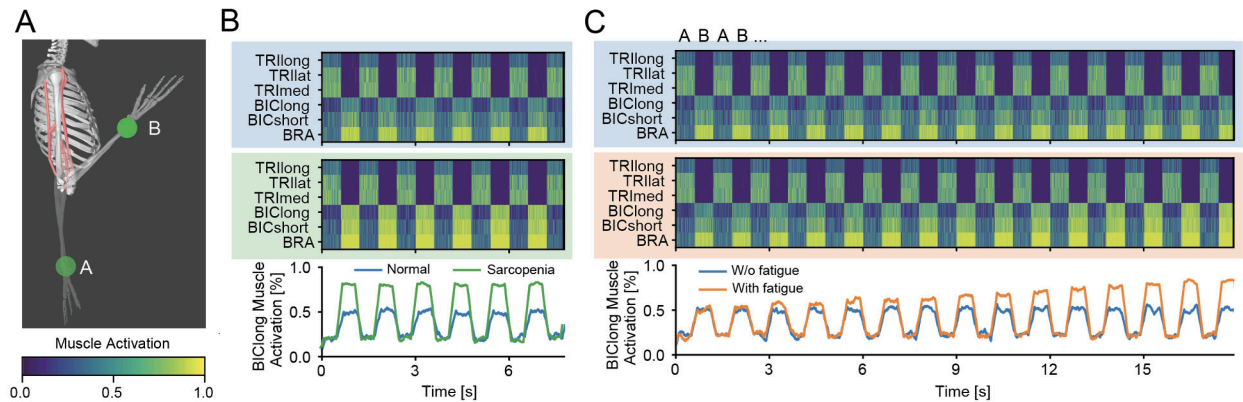

Figure 10: Control Policy with muscle sarcopenia (weakness) and fatigue. (A) Experiment design: Train a policy to reach random position of the wrist and test with a task of making alternate reaching movements between two elbow angle. (B) Test of the effects of Sarcopenia. Top-row: muscle activations for 6 arm muscles (3 agonists and 3 antagonists). Middle row: muscle activations in presence of sarcopenia. All synergistic flexor muscles (BRA - brachioradialis, BICLong - Biceps Long, BICShort - Biceps short) increase their activation to compensate for the reduced force. Bottom row: a trace of the activation only for the Biceps Long muscle. (C) Fatigue: Without fatigue weaker activation of the Biceps Long and Biceps Short where enough to reach the B-posture, in presence of fatigue, stronger activations from the three synergistic muscles are needed.

4.3.2 Accidents (Tendon tear)

Next, in order to study the effects of tendon tear, a policy was first trained to solve the Key-Turn task and then challenged with the selective damage(tear) of different thumb muscles (Figure 11). While there are redundancies (not every tear crippled ability), we found that FPL and OP are critical for the correct solution of the key task. In absence of OP the FPL is able to compensate. But when OP is torn FPL cannot compensate resulting in reduced key rotations.

4.3.3 Control in presence of tendon transfer

Finally, a tendon transfer - EIP muscle was routed to replace the EPL muscle - was performed to test the ability of the policies to compensate for action re-mapping due to tendon surgery. After the surgery, the major thumb muscles needed to compensate for the different activation space (Figure 12 and, a previously trained policy was unable to solve the task. Indeed it was necessary an extensive additional training to control the thumb after tendon transfer (Online Appendix/Solutions). This is typically observed in patient undergoing extensive physiotherapeurical sessions to re-learn the control of the thumb ([72]).

4.4 Human Robot interaction

In the MyoElbow model, we locked shoulder movements and we added an exoskeleton (see Appendix 3.3). Simulation of human robot interaction were realized for two experiments: I) flexion extension (between 0 - 30, 0 - 60 and 0 - 80 degrees with 2 sec hold time) with an healthy model with different weights on the hand (task inspired by [41]) and II) static with the elbow at 90 degrees with an healthy, Sarcopenia (see section 3.3) and fatigue (see section 3.3) conditions. The trials were repeated for the standard arm model with and without the exoskeleton support presented in section 3.3.
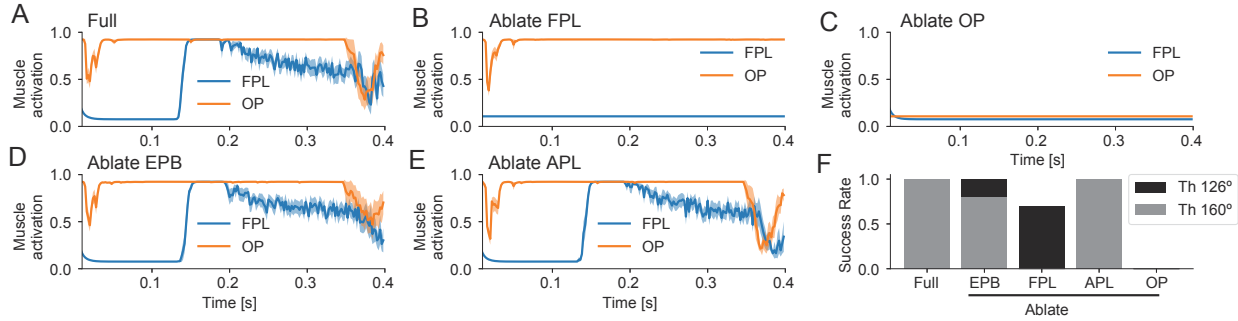
Figure 11: Effect of Tendon tear on the key-turning task. In this task a policy was trained to solve the key turning task. After training, selectively a different thumb muscle was removed for each experiment: 2 flexors (Flexor Pollicis Longus (FPL), Opponens Pollicis (OP)) and 1 extensor (Extensor Pollicis Brevis (EPB)) and 1 abductor (Abductor Pollicis Brevis (APL)), and the task was repeated 10 times. Normal thumb (Panel A). When FPL was ablated (Panel B), OP muscle continued its activation to accomplish the task while during ablation of OP (Panel C), FPL muscle did not show any activation. Ablation of Extensors muscles did not have an effect on the other muscles activation patterns (D-E). Finally, summarized in Panel F are the success rates in turning the key to reach a threshold of $126°$ or $160°$ rotation.
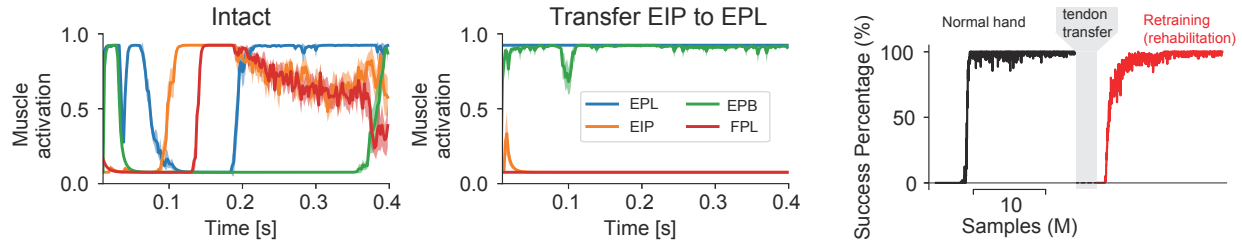

Figure 12: Index to thumb tendon transfer. In this experiment we first test a key-turning task on an intact hand and then we operated a tendon transfer: activity of the muscle Extensor Indicis Proprius (EIP) was redirected to activate muscle Extensor Pollicis Longus (EPL). In the left panel, thumb muscles (EPL, EIP, Extensor Pollicis Brevus (EPB), and (Flexor Pollicis Longus (FPL)) are shown (average over 10 repetitions) as the are orderly recruited to perform the task. After EIP to EPL remapping - center - the normal ordered recruitment is lost and the task cannot be any longer performed. The policy needs to be retrained in order to solve the task (right).

We trained a joint policy of exo and muscles by means of a Natural Policy Gradient (NPG) [29] algorithm in the natural (healthy) condition without any weight on the hand to reach random targets in the range of elbow flexions between 0 and 130 degrees of the elbow.

Figure 13 shows the effects of the reaching task (Experiment I) without and with exoskeleton assistance when a load is applied on the hand. As expected the reaching is minimally impacted when the exoskeleton is not functional but there is no additional perturbation. When weight is applied the exoskeleton is able to recover the original position with about 5 degrees error (distance of joint angles, mean $\pm$ std, 0 Kg: 4.95 ± 3.42, 1 Kg: 4.54 ± 3.12, 2 Kg: 4.34 ± 3.43). This is obtained both with a 60% reduction of both the muscle activation (Figure 13A, BIClong 60%, BICshort 54%, BRA 66%) and forces (Figure 13B, BIClong 60%, BICshort 58%, BRA 67%) at the weigth of 2 Kg.

Figure 14 and Figure 15 show the effects of holding a static position without and with exoskeleton in the presence of intrinsic perturbation to the muscle models in the form of sarcopenia (muscle weakness) and fatigue. As expected the holding is minimally impacted when the exoskeleton is not functional (distance of joint angles, mean $\pm$ std: 4.12 ± 2.97) but there is no additional intrinsic perturbation (first row of Figure 14). When intrinsic perturbations are present, the use of the exoskeleton is able to partially recover for those conditions (mean $\pm$ std, Sarcopenia 6.03 ± 2.97, Fatigue 11.89 ± 4.03). This is obtained both with a reduction of the muscle activation (Figure 15A, reduction without vs with exoskeleton of Normal, Sarcopenia and Fatigue for BIClong: 40%, 52%, 61%, BICshort: 30%, 44%, 56%; BRA: 41%, 54%, 62%) and forces (Figure 15B, reduction without vs with exoskeleton of Normal, Sarcopenia and Fatigue for BIClong: 41%, 55%, 98%; BICshort 34%, 50%, 98%, BRA: 43%, 56%, 88%).
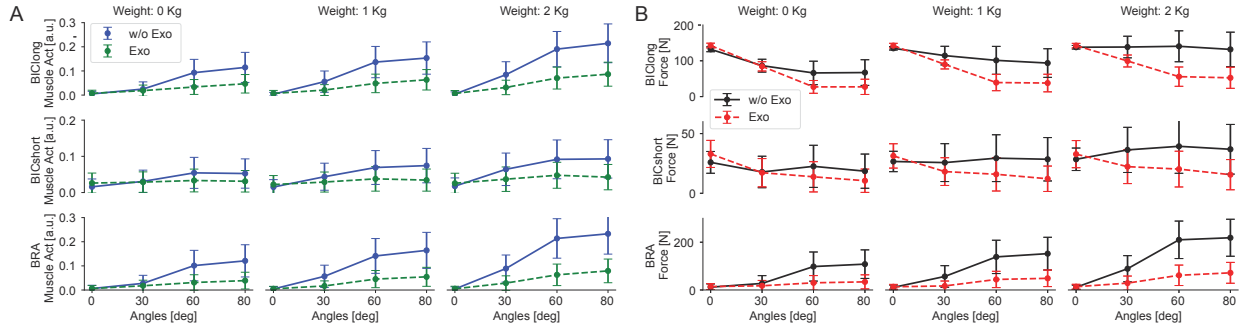
Figure 13: Effects of loads on Exoskeleton assisted reaching. A policy was trained in a reaching task of exoskeleton assisted elbow flexion to reach targets at 0 deg, 30 deg, 60 deg, 80 deg and hold the position for 2 seconds. A - Flexor muscles activations without (blue) and with (green) exoskeleton assistance at different target positions. With exoskeleton assistance the muscle activation required to reach the target angles was less. B - Same analysis as for A but on forces.
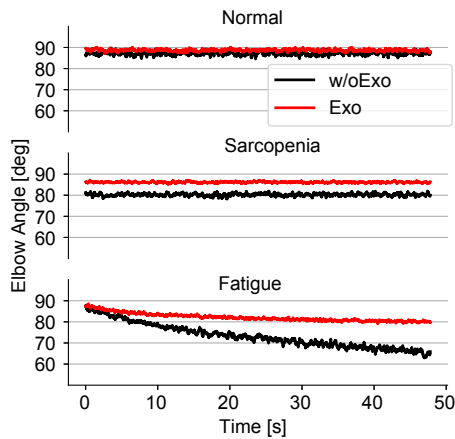

Figure 14: Effects of Sarcopenia and Fatigue on Exoskeleton assisted reaching. The same policy trained for exoskeleton assisted reaching was tested on holding a 90 deg angle elbow posture. The three rows show the effect of Normal, Sarcopenia and Fatigue respectively on holding the position. During Sarcopenia the muscle loss of force (weakness) is almost completely recovered with the exoskeleton. Likewise for fatigue, the exoskeleton compensate partially for the loss of force. Nevertheless, since the policy was not trained to compensate for those conditions, as results it is not able to drive the exoskeleton to a complete recovery of the function.
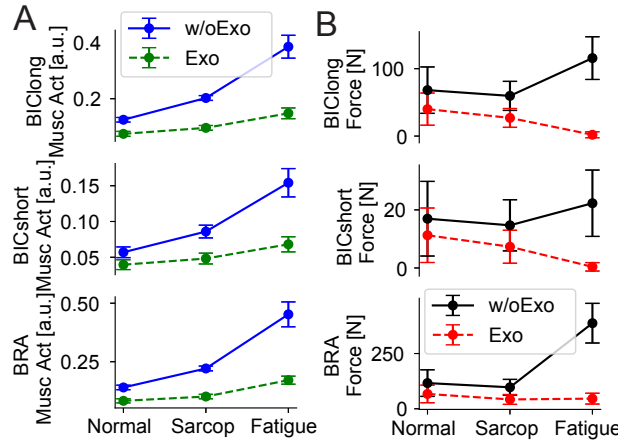
Figure 15: Effects of Sarcopenia and Fatigue on Exoskeleton assisted holding. Same task and policy as for 14. A - Flexor muscles activations without (blue) and with (green) exoskeleton assistance with the different conditions. With exoskeleton assistance the muscle activation required to compensate for the functional deficit is decreased. B - Same analysis as for A but on the forces.

5   Discussion and Conclusions

Here, we have proposed a new framework of physiologically realistic and computationally efficient models and tasks to study human motor control. The proposed models include realistic non-stationarities to simulate real-life scenarios such as muscle fatigue, sarcopenia, and tendon transfer. The suite of tasks entails realistic highly skilled manipulations. We hope that the wide range of complex and realistic challenges that the proposed platform affords, provides benchmarks that catalyze robot control in the same way that challenging datasets and benchmarks did for computer vision and natural language. This benchmark will provide biologically relevant problems where task success and physiological representations might differ. Actually, because activation of the musculoskeletal models can be directly related to a (normalized) intramuscular activity recorded in human subjects, it will be possible to validate experimentally the learning solutions.

The basic policies trained already showed physiologically relevant behaviors. First, we observe an automatic adaptation of muscles to leverage antagonistic effects of flexors and extensors. This was clearly visible with the elbow model where those effects could be more easily isolated. Second, co-contractions evolve naturally to compensate for changes

in muscle properties e.g. sarcopenia and fatigue. Nevertheless, those compensations are not enough to replace muscles like the Opponens Pollicis which has unique functions for hand manipulations.

Finally, it is worth mentioning that the implemented models are only the first iteration of an approximation of the musculoskeletal system that will need further development and validation. Indeed, both the tasks and physiological changes in muscle properties are a subset of the possible changes that can be considered.

All in all, we would like this work to facilitate cross-pollination and catalyzation of new ideas between different communities. The ML community will have more challenging tasks with 3rd order dynamics (which are uncommon in ML/RL benchmarks) and the proposed non-stationarity provides a more physiological realistic use-case to challenge their algorithms. The biomechanics community will have the possibility to afford a new platform where contact-rich interactions at scale can be studied. Finally, the robotic community will have a biological system to develop strategies for. Overall, this will allow for the creation of realistic simulation opening the ways to *in silico* trials of humans, robots, and their interaction.

Acknowledgments

References

[1]   Marko Ackermann and Antonie J Van den Bogert. "Optimality principles for model-based prediction of human gait". In: *Journal of biomechanics* 43.6 (2010), pp. 1055–1060.

[2]   Kai-Nan An et al. "Tendon excursion and moment arm of index finger muscles". In: *Journal of biomechanics* 16.6 (1983), pp. 419–425.

[3]   OpenAI: Marcin Andrychowicz et al. "Learning dexterous in-hand manipulation". In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20.

[4]   Edith M. Arnold and Scott L. Delp. "Fibre operating lengths of human lower limb muscles during walking". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 366 (2011), pp. 1530–1539.

[5]   Nikolai Bernstein. "The co-ordination and regulation of movements". In: *The co-ordination and regulation of movements* (1966).

[6]   PW Brand and AM Hollister. *Clinical mechanics of the hand 2 nd ed. St. Louis, MO*. 1993.

[7]   Greg Brockman et al. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[8]   Don B Chaffin, Gunnar B J Andersson, and Bernard J Martin. *Occupational biomechanics*. John wiley \& sons, 2006.

[9]   Tao Chen, Jie Xu, and Pulkit Agrawal. "A System for General In-Hand Object Re-Orientation". In: *arXiv preprint arXiv:2111.03043* (2021).

[10]   Erwin Coumans and Yunfei Bai. "Pybullet, a python module for physics simulation for games, robotics and machine learning". In: (2016).

[11]   Alfonso J. Cruz-Jentoft and Avan A. Sayer. "Sarcopenia". In: *The Lancet* 393.10191 (June 2019), pp. 2636–2646. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(19)31138-9.

[12]   Michael Damsgaard et al. "Analysis of musculoskeletal systems in the AnyBody Modeling System". In: *Simulation Modelling Practice and Theory* 14.8 (2006), pp. 1100–1111.

[13]   Scott L Delp et al. "OpenSim: open-source software to create and analyze dynamic simulations of movement". In: *IEEE transactions on biomedical engineering* 54.11 (2007), pp. 1940–1950.

[14]   Scott L. Delp et al. "OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement". In: *IEEE Transactions on Biomedical Engineering* 54.11 (2007), pp. 1940–1950. DOI: 10.1109/TBME.2007.901024.

[15]   Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[16]   Coline Devin et al. "Learning modular neural network policies for multi-task and multi-robot transfer". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 2169–2176.

[17]   Richard M. Dodds et al. "Global variation in grip strength: a systematic review and meta-analysis of normative data". In: *Age and Ageing* 45.2 (Mar. 2016), pp. 209–216. ISSN: 0002-0729. DOI: 10.1093/AGEING/AFV192. URL: https://academic.oup.com/ageing/article/45/2/209/2195303.

[18]   Guillaume Durandau, Dario Farina, and Massimo Sartori. "Robust real-time musculoskeletal modeling driven by electromyograms". In: *IEEE transactions on biomedical engineering* 65.3 (2017), pp. 556–564.

[19]   Guillaume Durandau et al. "Voluntary control of wearable robotic exoskeletons by patients with paresis via neuromechanical modeling". In: *Journal of neuroengineering and rehabilitation* 16.1 (2019), pp. 1–18.

[20]   R. I. Gelb. "Tendon Transfer for Rupture of the Extensor Pollicis Longus". In: *Hand Clinics* 11.3 (Aug. 1995), pp. 411–422. ISSN: 0749-0712. DOI: 10.1016/S0749-0712(21)00062-7.

[21]   Friedl De Groote et al. "Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem". In: *Annals of Biomedical Engineering* 44 (2016), pp. 2922–2936.

[22]   Abhishek Gupta et al. "Learning invariant feature spaces to transfer skills with reinforcement learning". In: *arXiv preprint arXiv:1703.02949* (2017).

[23]   Abhishek Gupta et al. "Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention". In: *arXiv preprint arXiv:2104.11203* (2021).

[24]   Samuel R Hamner, Ajay Seth, and Scott L Delp. "Muscle contributions to propulsion and support during running". In: *Journal of biomechanics* 43.14 (2010), pp. 2709–2716.

[25]   Masaya Hirashima and Tomomichi Oya. "How does the brain solve muscle redundancy? Filling the gap between optimization and muscle synergy hypotheses". In: *Neuroscience Research* 104 (2016). Body representation in the brain, pp. 80–87. ISSN: 0168-0102. DOI: https://doi.org/10.1016/j.neures.2015.12.008. URL: https://www.sciencedirect.com/science/article/pii/S0168010215002990.

[26] Wenlong Huang et al. "Generalization in Dexterous Manipulation via Geometry-Aware Multi-Task Learning". In: *arXiv preprint arXiv:2111.03062* (2021).

[27] Jemin Hwangbo, Joonho Lee, and Marco Hutter. "Per-contact iteration method for solving contact dynamics". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 895–902. URL: www.raisim.com.

[28] Aleksi Ikkala and Perttu Hämäläinen. *Converting Biomechanical Models from OpenSim to MuJoCo*. 2020. arXiv: 2006.10618 [q-bio.QM].

[29] Sham M Kakade. "A natural policy gradient". In: *Advances in neural information processing systems* 14 (2001).

[30] Fotios Alexandros Karakostis et al. "Biomechanics of the human thumb and the evolution of dexterity". In: *Current Biology* 31.6 (2021), 1317–1325.e8. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2020.12.041. URL: https://www.sciencedirect.com/science/article/pii/S0960982220318935.

[31] Donald T. Kirkendall and William E. Garrett. "The Effects of Aging and Training on Skeletal Muscle". In: *The American Journal of Sports Medicine* 26.4 (1998). PMID: 9689386, pp. 598–602. DOI: 10.1177/03635465980260042401. eprint: https://doi.org/10.1177/03635465980260042401. URL: https://doi.org/10.1177/03635465980260042401.

[32] Vijay R Konda and John N Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems*. 2000, pp. 1008–1014.

[33] David K Kuechle et al. "Shoulder muscle moment arms during horizontal flexion and elevation". In: *Journal of Shoulder and Elbow Surgery* 6.5 (1997), pp. 429–439.

[34] Vittorio La Barbera et al. *OstrichRL: A Musculoskeletal Ostrich Simulation to Study Bio-mechanical Locomotion*. 2021. DOI: 10.48550/ARXIV.2112.06061. URL: https://arxiv.org/abs/2112.06061.

[35] Joseph Langenderfer et al. "Musculoskeletal parameters of muscles crossing the shoulder and elbow and the effect of sarcomere length sample size on estimation of optimal muscle length". In: *Clinical biomechanics* 19.7 (2004), pp. 664–670.

[36] Mark L Latash. "The bliss (not the problem) of motor abundance (not redundancy)". In: *Experimental brain research* 217.1 (2012), pp. 1–5.

[37] Jeongseok Lee et al. "Dart: Dynamic animation and robotics toolkit". In: *Journal of Open Source Software* 3.22 (2018), p. 500.

[38] Jong Hwa Lee et al. "Finger Muscle Attachments for an OpenSim Upper-Extremity Model". In: *PLOS ONE* 10.4 (Apr. 2015), pp. 1–28. DOI: 10.1371/journal.pone.0121712. URL: https://doi.org/10.1371/journal.pone.0121712.

[39] Seunghwan Lee et al. "Scalable Muscle-Actuated Human Simulation and Control". In: *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3322972. URL: https://doi.org/10.1145/3306346.3322972.

[40] Richard L Lieber, Babak M Fazeli, and Michael J Botte. "Architecture of selected wrist flexor and extensor muscles". In: *The Journal of hand surgery* 15.2 (1990), pp. 244–250.

[41] Nicola Lotti et al. "Adaptive model-based myoelectric control for a soft wearable arm exosuit: A new generation of wearable robot control". In: *IEEE Robotics & Automation Magazine* 27.1 (2020), pp. 43–53.

[42] Liang Ma et al. "A new simple dynamic muscle fatigue model and its validation". In: *International Journal of Industrial Ergonomics* 39.1 (2009), pp. 211–220. ISSN: 01698141. DOI: 10.1016/j.ergon.2008.04.004. URL: http://dx.doi.org/10.1016/j.ergon.2008.04.004.

[43] Viktor Makoviychuk et al. "Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021. URL: https://openreview.net/forum?id=fgFBtYgJQX_.

[44] Daniel C. McFarland et al. "Spatial Dependency of Glenohumeral Joint Stability During Dynamic Unimanual and Bimanual Pushing and Pulling". In: *Journal of Biomechanical Engineering-Tansation of the ASME* 141.5 (May 2019). DOI: 10.1115/1.4043035.

[45] Wendy M Murray, Scott L Delp, and Thomas S Buchanan. "Variation of muscle moment arms with elbow and forearm position". In: *Journal of biomechanics* 28.5 (1995), pp. 513–525.

[46] Anusha Nagabandi et al. "Deep dynamics models for learning dexterous manipulation". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1101–1112.

[47] Anusha Nagabandi et al. "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning". In: *arXiv preprint arXiv:1803.11347* (2018).

[48] OpenAI et al. "Solving Rubik's Cube with a Robot Hand". In: *arXiv preprint arXiv:1910.07113* (2019).

[49] James C Otis et al. "Changes in the moment arms of the rotator cuff and deltoid muscles with abduction and rotation." In: *The Journal of bone and joint surgery. American volume* 76.5 (1994), pp. 667–676.

[50] Vassil Panayotov et al. "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5206–5210. DOI: `10.1109/ICASSP.2015.7178964`.

[51] Matthias Plappert et al. *Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research*. 2018. arXiv: `1802.09464 [cs.LG]`.

[52] Kenneth V Price. "Differential evolution". In: *Handbook of optimization*. Springer, 2013, pp. 187–214.

[53] Aravind Rajeswaran et al. "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations". In: *Proceedings of Robotics: Science and Systems (RSS)*. 2018.

[54] Aravind Rajeswaran et al. "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations". In: *Proceedings of Robotics: Science and Systems (RSS)*. 2018.

[55] Massimo Sartori et al. "A musculoskeletal model of human locomotion driven by a low dimensional set of impulsive excitation primitives". In: *Frontiers in computational neuroscience* 7 (2013), p. 79.

[56] Katherine R Saul et al. "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model". In: *Computer methods in biomechanics and biomedical engineering* 18.13 (2015), pp. 1445–1458.

[57] Katherine R. Saul et al. "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model". In: *Computer Methods in Biomechanics and Biomedical Engineering* 18.13 (2015). PMID: 24995410, pp. 1445–1458. DOI: `10.1080/10255842.2014.916698`. eprint: `https://doi.org/10.1080/10255842.2014.916698`. URL: `https://doi.org/10.1080/10255842.2014.916698`.

[58] Stephen H. Scott. "The computational and neural basis of voluntary motor control and planning". In: *Trends in Cognitive Sciences* 16.11 (2012), pp. 541–549. ISSN: 1364-6613. DOI: `https://doi.org/10.1016/j.tics.2012.09.008`. URL: `https://www.sciencedirect.com/science/article/pii/S1364661312002240`.

[59] Ajay Seth et al. "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement". In: *PLOS Computational Biology* 14 (July 2018), pp. 1–20. DOI: `10.1371/journal.pcbi.1006223`. URL: `https://doi.org/10.1371/journal.pcbi.1006223`.

[60] Ajay Seth et al. "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement". In: *PLoS computational biology* 14.7 (2018), e1006223.

[61] Reza Shadmehr and John W Krakauer. "A computational neuroanatomy for motor control". In: *Experimental brain research* Mar;185(3):359-81 (2008).

[62] David Silver et al. "Deterministic policy gradient algorithms". In: *International conference on machine learning*. PMLR. 2014, pp. 387–395.

[63] *SIMM: Software for Interactive Musculoskeletal Modeling*. `https://motionanalysis.com/simm/`.

[64] Seungmoon Song et al. "Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation". In: *bioRxiv* (2020). DOI: `10.1101/2020.08.11.246801`. eprint: `https://www.biorxiv.org/content/early/2020/08/12/2020.08.11.246801.full.pdf`.

[65] Richard S Sutton. "Generalization in reinforcement learning: Successful examples using sparse coarse coding". In: *Advances in neural information processing systems* (1996), pp. 1038–1044.

[66] Yuval Tassa et al. "Deepmind control suite". In: *arXiv preprint arXiv:1801.00690* (2018).

[67] Lena H. Ting et al. "Review and perspective: neuromechanical considerations for predicting muscle activation patterns for movement". In: *International Journal for Numerical Methods in Biomedical Engineering* 28.10 (2012), pp. 1003–1014. DOI: `https://doi.org/10.1002/cnm.2485`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.2485`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2485`.

[68] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[69] Peter JM Van Laarhoven and Emile HL Aarts. "Simulated annealing". In: *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[70] Nina K Vøllestad. "Measurement of human muscle fatigue". In: *Journal of neuroscience methods* 74.2 (1997), pp. 219–227.

[71] Alex Wang et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *CoRR* abs/1804.07461 (2018). arXiv: `1804.07461`. URL: `http://arxiv.org/abs/1804.07461`.

[72] Johanna Wangdell et al. "Early active rehabilitation after grip reconstructive surgery in tetraplegia". In: *Archives of physical medicine and rehabilitation* 97.6 (2016), S117–S125.

[73] Christopher JCH Watkins and Peter Dayan. "Q-learning". In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[74] Scott C White, H John Yack, and David A Winter. "A three-dimensional musculoskeletal model for gait analysis. Anatomical variability estimates". In: *Journal of biomechanics* 22.8-9 (1989), pp. 885–893.

[75] Daniel M. Wolpert and Zoubin Ghahramani. "Computational principles of movement neuroscience". In: *Nature Neuroscience* 3 Suppl 1 (2000), pp. 1212–1217.

[76] John Z Wu et al. "Analysis of musculoskeletal loading in an index finger during tapping". In: *Journal of biomechanics* 41.3 (2008), pp. 668–676.

[77] Zhe Xu et al. "Design of an anthropomorphic robotic finger system with biomimetic artificial joints". In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2012, pp. 568–574.

[78] Kim Younguk et al. "Similarities and differences between musculoskeletal simulations of OpenSim and AnyBody modeling system". In: *Journal of Mechanical Science and Technology* 32 (Dec. 2018), pp. 6037–6044. DOI: 10.1007/s12206-018-1154-0.

Appendix

A   MyoSim: A pipeline to generate MuJoCo musculoskeletal models

In this section, the musculoskeletal model conversion tool is explained in detail.

A.1   Musculoskeletal model conversion tool

We used physiologically validated and widely adopted OpenSim musculoskeletal models as reference. Then, a layer-by-layer automatic conversion pipeline was developed to convert those OpenSim models into MuJoCo. The pipeline consisted of three major conversion (Cvt) steps, as shown in Figure 16. A validation (Vlt) module followed each conversion step to match the converted MuJoCo model and the reference OpenSim model. A short description of them is listed below, followed by the detail explanations in the subsections.
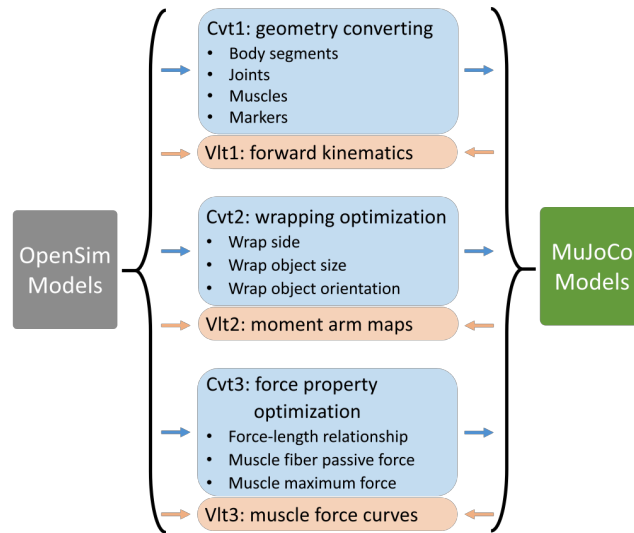

Figure 16: Three-steps conversion pipeline

1. **Cvt1: geometry conversion.** Parse kinematics details (body segment lengths, joints, muscles-tendon pathways, etc)[28] and fiducial markers from OpenSim model into MuJoCo model.

   Vlt1: forward kinematics. Check that salient positions along the kinematic chain are matched between the two models at different joint configurations.

2. **Cvt2: wrapping optimization.** Optimize wrapping constraints to guarantee that the muscle pathways are physiologically reasonable.

   Vlt2: moment arm maps. Check that moment arms of each muscle at each joint angle are matched between models.

3. **Cvt3: force property optimization.** Optimize muscle force parameters in MuJoCo to approximate OpenSim model's force curves.

   Vlt3: muscle force curves. Check force-length-activation differences between the two models.

Geometry Conversion:
OpenSim and MuJoCo geometry structures are simply defined based on their relative position in a similar hierarchical (XML) format. An open-access conversion code was created by Ikkala et al [28] that parsed the bodies, joints, and muscles into the MuJoCo modelling format. We extended their code by adding in the conversion 1) markers and 2) wrapping surfaces for muscles to avoid collisions with bones during movements. The forward kinematics module (*Vlt1*) checks the differences of the selected marker locations at different joint configurations (see the Results section 4.1).

Wrapping Optimization:
OpenSim and MuJoCo have different ways of defining how muscles wrap over wrapping surfaces. In OpenSim, three

wrapping methods were defined: MidPoint, Axial, Hybrid [13, 60]. Nevertheless, in MuJoCo, muscles can wrap over or can be forced to pass through a surface. In order to guarantee similar forces produced by a muscle at each joint, an optimization was developed. We adapted both the location where a muscle wraps on the surface in MuJoCo i.e. *'side site'* and the dimensions of the wrapping objects (surfaces such as ellipsoids and torus, which are missing in MuJoCo, were approximated with other surfaces).

The following optimization was developed:

Find: wrapping sides, sizes & orientations:
$$X = \{[L_1, ..., L_n], [S_1, ..., S_p], [R_1, ..., R_p]\}$$

To minimize: objective function:
$$F = \sum_{m=1,j=1}^{M,J} (d_{m,j}^{osim}(Q) - d_{m,j}^{mjc}(Q, X))^2$$

When: iterate all joint angle meshes

Subject to: model kinematics; parameter bounds (3)

where, $L = (x, y, z)$ represents the location of side site; $S = (r, l)$ represents the size of wrapping e.g. cylinder dimensions; $R = (\alpha, \beta, \gamma)$ are the Euler angles of wrapping surface orientation; d represents the moment arms; $Q = [q_1, ...q_J]$ represents joint angles; M represents the total number of muscles; J represents the total number of joints

Simulated annealing (SA) [69] was used as the optimization method to minimize the differences of moment arms i.e. errors, between the converted MuJoCo model and the referencing OpenSim model for all muscles at all joints angles. Details of this can be found in the validation section 4.1.

Force Property Optimization:
The OpenSim and the MuJoCo platforms also use different methods for modelling muscle force properties. MuJoCo does not consider elastic tendons, fiber pennation angles, while OpenSim has both of them. OpenSim uses physical lengths for the definition of optimal fiber and tendon slack lengths, whereas, MuJoCo uses normalized value. Even though the MuJoCo muscle model is a simplified version of the OpenSim muscle model, it is still possible to generate similar force-length-velocity relationships. However, the muscle property parameters needs to be optimized, instead of directly mapping from the OpenSim model. The following optimization was developed to match the muscle force properties:

Find: muscle force parameters:
$$X = \{l_{min}, l_{max}, fp_{max}, f_{max}\}$$

To minimize: objective function:
$$F = \sum_{m=1}^{M} (f_m^{osim}(l_m, a) - f_m^{mjc}(l_m, a, X))^2$$

When: iterate all muscle length l_m and activation a

Subject to: muscle dynamic; parameter bounds (4)

where, l_{min} is the minimal normalized muscle length when active fiber force reaches zero; l_{max} is the maximum normalized muscle length when active fiber force reaches zero; fp_{max} is the normalized passive fiber force when muscle length equal to l_{max}; f_{max} is the absolute value of maximum muscle force; $f(l_m, a)$ represents the muscle forces when the muscle has an activation of a and at the length of l_m.

Differential evolution (DE) [52] was used as optimization method. Muscle activation dynamics and the force-velocity relationship are identical between the OpenSim and MuJoCo models, as a result, they do not need to be optimized, but can be directly transferred. Validation of this step is the errors of force-length-activation maps between the OpenSim and MuJoCo models. More details of it can be found in the validation section 4.1.

Label	Muscle	Number of muscles
ECRL	Extensor Carpis Radialis Longus	1x
ECRB	Extensor Carpis Radialis Brevis	1x
ECU	Extensor Carpi Ulnaris	1x
FCR	Flexor Carpi Radialis	1x
FCU	Flexor Carpi Ulnaris	1x
PL	Palmaris longus	1x
PT	Pronator teres	1x
PQ	Pronator	1x
EIP	Extensor Indicis Proprius	1x
EPL	Extensor Pollicis Longus	1x
EPB	Extensor Pollicis Brevis	1x
FPL	Flexor Pollicis Longus	1x
APL	Abductor Pollicis Longus	1x
OP	Opponens Pollicis	1x
FDS	Flexor Digitorum Superficialis	4x
FDP	Flexor Digitorum Profundus	4x
EDC	Extensor Digitorum Communis	4x
EDM	Extensor digiti minimi	1x
RI	Radial Interosseous	4x
LU-RB	Lumbrical	4x
UI-UB	Palmar or Ulnar Interosseous	4x

Table 3: List of muscles that are included in the MyoHand model.