

Self Model for Embodied Intelligence: Modeling Full-Body Human Musculoskeletal System and Locomotion Control with Hierarchical Low-Dimensional Representation

Chenhui Zuo*, Kaibo He*, Jing Shao, Yanan Sui

Abstract—Modeling and control of the human musculoskeletal system is important for understanding human motor functions, developing embodied intelligence, and optimizing human-robot interaction systems. However, current human musculoskeletal models are restricted to a limited range of body parts and often with a reduced number of muscles. There is also a lack of algorithms capable of controlling over 600 muscles to generate reasonable human movements. To fill this gap, we build a musculoskeletal model (MS-HUMAN-700) with 90 body segments, 206 joints, and 700 muscle-tendon units, allowing simulation of full-body dynamics and interaction with various devices. We develop a new algorithm using low-dimensional representation and hierarchical deep reinforcement learning to achieve state-of-the-art full-body control. We validate the effectiveness of our model and algorithm in simulations with real human locomotion data. The musculoskeletal model, along with its control algorithm, will be made available to the research community to promote a deeper understanding of human motion control and better design of interactive robots.

Project page: <https://lmsgroup.cc/research/MS-Human-700>

I. INTRODUCTION

The human musculoskeletal system is a complex dynamic system with high spatio-temporal degrees of freedom. While current technological advancements cannot effectively measure the dynamic characteristics of the human body as a whole, modeling and simulation are very important for the understanding of human motion control and human factors in human-robot interaction. Dynamic modeling can also facilitate the design of humanoid robots and provide a self-model of human for embodied intelligence. A musculoskeletal model is built from anatomical and biomechanical data of the human body. By simulating human motion with a musculoskeletal model, it is possible to gain insight into the dynamics of the human body and the factors that influence motion, such as muscle activation patterns, joint stability, and limb coordination. We can use simulation data instead of hard-to-obtain real data to design and optimize robots, such as humanoids, exoskeletons, prosthetics, and other human-interactive devices.

Many of our daily movements require the use of the entire body. For instance, walking and running involve the coordinated movement of the lower limbs for propulsion, complemented by the swinging motion of the upper limbs for

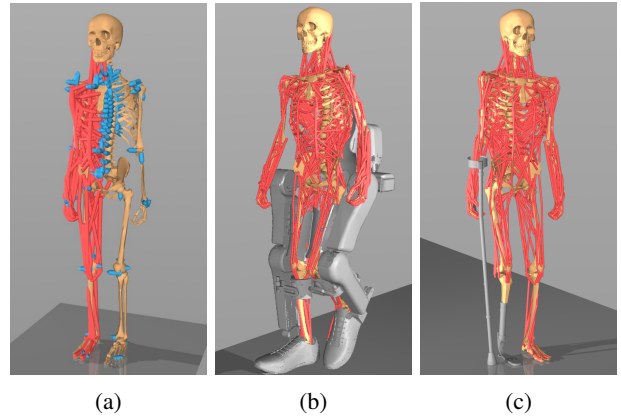


Fig. 1: (a) MS-HUMAN-700: full-body musculoskeletal model. Blue arrows represent joint axes, red lines represent muscle-tendon units (right half). (b) The model interacting with an exoskeleton in simulation. (c) The model with a prosthetic leg and a crutch.

balance, while the trunk muscles contribute to torso stability. However, there is no musculoskeletal model that represents the entire human body. Constructing a full-body human musculoskeletal model enables comprehensive characterization of the action space and state space in human movement simulations. The dynamics of the model improves our understanding of human motor intelligence. Furthermore, current algorithms struggle to address the highly dimensional control problem associated with whole-body human musculoskeletal motion.

The modeling and control of the musculoskeletal model of full human body is profoundly difficult due to the following challenges. First, building a human musculoskeletal model requires detailed quantitative knowledge of anatomy and physiology. Second, the muscle control parameter space is complex, where over 600 skeletal muscles control hundreds of joints, leading to high dimensionality and redundancy of the system [1], [2]. Third, the dynamical property of neuro-muscle actuators is non-linear and inconstant [3], [4].

Here, we develop MS-HUMAN-700, a full-body MusculoSkeletal Human model with 90 body segments, 206 joints, and 700 muscle-tendon units for neuromuscular-driven dynamic simulation, as shown in Fig.1. The model can be imported into open-source physics simulation engines like MuJoCo [5] and OpenSim [6], establishing a framework

*These authors contributed equally to this work.

The authors are with the National Engineering Research Center of Neuromodulation, School of Aerospace Engineering, Tsinghua University, Beijing, China 100084. Correspondence to Yanan Sui. E-mail: ysui@tsinghua.edu.cn

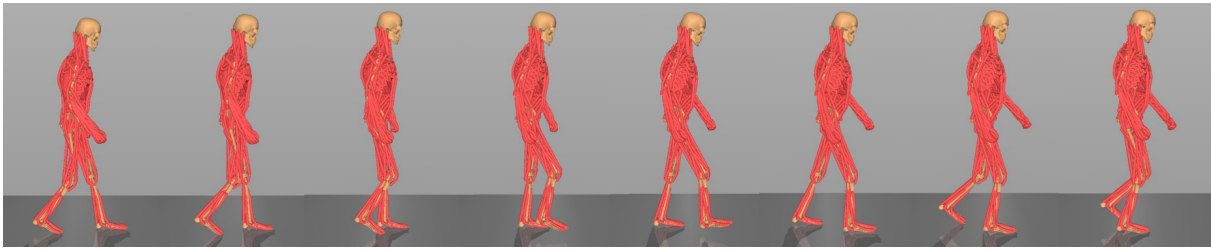


Fig. 2: Learning to walk with MS-HUMAN-700 via Two-Stage Hierarchical Training (TSHT).

that can be used to perform neuro-musculo-skeletal control simulation of human full-body movements.

We also develop a hierarchical deep reinforcement learning (RL) algorithm with a low-dimensional representation, Two-Stage Hierarchical Training (TSHT), which is capable of controlling the model to generate biologically plausible movements. We make both the musculoskeletal model and our control algorithm available to the research community with the goal of fostering the development of more accurate models, the design of better assistive robots, and a better understanding of human neuromuscular motion control for embodied intelligence.

Our Contributions. We build a dynamic model of full-body human musculoskeletal system with physiologically feasible body segments, joints, and multi-muscle-tendon units. We embed the whole-body model in simulation environments that support contact interaction simulations, providing a scalable, adaptable, and computationally efficient simulation framework for data-driven musculoskeletal system control. We propose a deep reinforcement learning algorithm, two-stage hierarchical training, for the control of the high-dimensional and highly nonlinear full-body musculoskeletal model. Our control algorithm can generate desired motions in full-body musculoskeletal model.

II. RELATED WORK

Human Musculoskeleton Modeling. Many previous studies focused on the musculoskeletal models actuated by Hill-type muscles [3] for specific human body parts. An upper extremity model has been built and validated [7]–[9]. Lower extremity models have been continuously improved and used in the field [10]–[14]. Some researchers built the model of the trunk [15] and extended it with simple muscles-actuated limbs [16]. Other researchers created models with muscles of the torso and limbs, but the models do not cover a large range of important skeletal muscles and joints of the torso [17], [18]. These models were mainly built on physics-based musculoskeletal simulation platforms. Representative research efforts include OpenSim [6], Anybody [19], etc. However, running dynamic simulations on them is computationally expensive. Some recently developed physics engines [5], [20]–[22] are becoming popular for robotic simulation, and they are capable for efficient dynamic simulations that support contact interactions. MuJoCo’s modeling approach is able to provide a simplified and fully supported interface for muscle modeling among them. Efforts have been made

to convert the musculoskeletal models from OpenSim to MuJoCo, resulting in acceleration of simulation speed on the model by two orders of magnitude [23]–[25].

Motion Control of the Musculoskeletal System. Researchers have explored the use of biological actuators for controlling human musculoskeletal systems. Evolutionary algorithms were used to optimize control parameters for a 2D lower limb model comprising 16 muscle-tendon units [26], and previous work [27] developed a more complex musculoskeletal model. However, these methods fell short when handling complex models. Recently, a hierarchical RL algorithm, coupled with imitation learning, was applied to a musculoskeletal model featuring 346 muscles [17]. Xu et al. [28] introduced a high-performance differentiable simulator and an algorithm that effectively enhanced control over the human musculoskeletal model by leveraging gradient information. A review was conducted on the application of RL in neuromechanical simulations [29]. Beyond the human musculoskeletal model, a 3D musculoskeletal model of an ostrich was constructed using the MuJoCo engine [30]. This model successfully performed various locomotion tasks with deep RL algorithm TD4. Schumacher et al. [31] proposed the DEP-RL algorithm to attain superior control over the aforementioned ostrich model. Furthermore, the concept of reward shaping to achieve natural walking in musculoskeletal models of human lower limbs was introduced [32]. Multi-task learning [33] and synergistic action representation [34], [35] have demonstrated their effectiveness in the motion control of specific musculoskeletal body parts.

In the open-source musculoskeletal modeling community, the current model with the largest number of muscles contains 552 individual muscle units [16], which lacks the coverage of limb muscles to complete the simulation of full-body movements. Control algorithms that can drive the high-dimensional musculoskeletal models to produce realistic motions are still lacking.

III. MUSCULOSKELETAL SYSTEM

As demonstrated in Fig.1, our full-body musculoskeletal model includes 90 rigid body segments, 206 joints, and 700 muscle-tendon units. The model can simulate the joint movements of the torso and limbs driven by skeletal muscles. We use musculoskeletal models [8], [9], [12], [14], [15] on OpenSim platform as the references to set anatomical parameters of the following models of body segments, joints and muscle-tendon units, following the pipeline of [23]. We

also embed the MARCH exoskeleton model in simulation to demonstrate human-machine interactive walking.

A. Skeletal Modeling

We choose the pelvis as the root of the kinematic tree. We set the position, mass and inertia properties of each body segment to enable dynamic simulation of the model. The shape of each body segment is defined with mesh files, reserving their convex hulls for collision calculation during simulation. The body segments are connected mainly by hinge joints. There are also slide joints to represent translational characteristic between segments. The axis and range of motion of each joint are defined in the model. The coupling relationship between joints is modeled by polynomial equations [12], [36].

The dynamics of the human musculoskeletal system can be formulated with the Euler-Lagrangian equations as

$$M(q)\ddot{q} + c(q, \dot{q}) = J_m^T f_m(act) + J_c^T f_c + \tau_{ext}, \quad (1)$$

where q represents the generalized coordinates of joints, $M(q)$ is the mass distribution matrix, $c(q, \dot{q})$ is Coriolis and gravitational forces. $f_m(act)$ is the vector representing muscle forces generated by all muscle-tendon units, and is determined by muscle activations (act). f_c is the constraint force, and τ_{ext} is the external torque from the interaction with environments. J_m and J_c are Jacobian matrices that map forces to the space of generalized coordinates.

B. Muscle Modeling

Our full-body musculoskeletal model has 700 muscle-tendon units. They exert tensile forces on bones to move joints.

Muscle-Tendon Pathway. These muscle-tendon units typically originate and are inserted at pathway points attached to different body segments. For wide muscle attachments, it will be modeled as multiple independent muscle-tendon units. Thus, the total number of units, 700, in our model is larger than the common biomedical estimations for the number of skeletal muscles, which is around 640 [37]. Wrapping geometry surfaces are added to the pathways to model physical constraints imposed by bones and soft tissue when needed. Our multi-muscle-tendon and wrapping geometry models are demonstrated in Fig. 3a and Fig. 3b.

Neuro-Muscle Dynamics. The activation-contraction dynamics of muscles exhibit non-linearity and temporal delay, thereby posing challenges to neuromuscular control. With the employment of the Hill-type muscle model [3], the maximum isometric force, the tendon length, and the optimal fiber length of each unit is defined in the model. The force produced by a single muscle-tendon unit can be formulated as

$$f_m(act) = f_{max} \cdot [F_l(l_m) \cdot F_v(v_m) \cdot act + F_p(l_m)], \quad (2)$$

where f_{max} stands for the maximum isometric muscle force and act, l_m, v_m respectively stand for the activation, normalized length and normalized velocity of the muscle. F_l and F_v represent force-length and force-velocity functions fitted

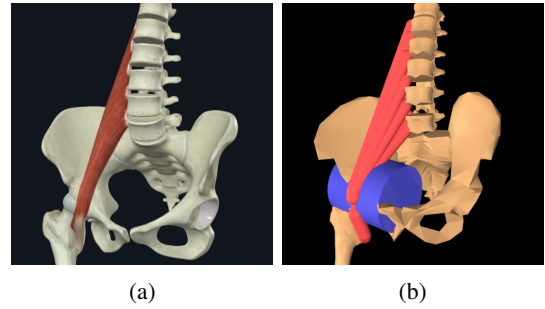


Fig. 3: (a) The anatomy of Psoas Major muscle from the medical atlas, image courtesy of Complete Anatomy. (b) Its multi-muscle-tendon units and the wrapping geometry constructed in our model.

using data from biomechanical experiments [38]. The muscle activation act is calculated by a first-order nonlinear filter in simulation as Eq. (3) where u is the input control signal of the musculoskeletal model. $(\tau_{act}, \tau_{deact})$ is a time constant for activate and deactivate latency with defaults (10ms, 40ms).

$$\frac{\partial act}{\partial t} = \frac{u - act}{\tau(u, act)}, \tau(u, act) = \begin{cases} \tau_{act}(0.5 + 1.5act) & u > act \\ \frac{\tau_{deact}}{0.5 + 1.5act} & u \leq act \end{cases} \quad (3)$$

IV. CONTROL

To tackle this high-dimensional musculoskeletal model control problem, we propose a training algorithm based on hierarchical action space representation, which is capable of generating natural motions through two stages of training.

A. Problem formulation for Musculoskeletal Model Control

We formulate the control problem of the musculoskeletal model as a Markov Decision Processes, denoted by $\langle \mathcal{S}, \mathcal{A}, r, p, \rho_0, \gamma \rangle$, where $\mathcal{S} \subseteq \mathbb{R}^n$ represents the continuous space of all valid states, $\mathcal{A} \subseteq \mathbb{R}^m$ represents the continuous space of all valid actions. $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. p stands for the state transfer probability function, where $p(s'|s, a)$ denotes the probability of an agent taking an action a to transfer from the current state s to the state s' , ρ_0 is the probability distribution of the initial state with $\sum_{s \in \mathcal{S}} \rho_0(s) = 1$ and $\gamma \in [0, 1)$ is the discount factor. We adopt an off-policy RL algorithm, Soft Actor-Critic (SAC) [39], to optimize our policy. This algorithm aims to obtain the optimal policy by maximizing the accumulative discounted return, $\sum_t \gamma^t R_t$ and the policy entropy, $\mathcal{H}(\pi)$, where

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{(o_t, a_t)} \left[\sum_{t=0}^T \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot | s_t))) \right]. \quad (4)$$

State space \mathcal{S} consists of $\{t, q, \dot{q}, f_m, l_m, \dot{l}_m, act, \mathcal{E}\}$, represents simulation time, joint position, joint velocity, muscle force, muscle length, muscle velocity, muscle activation, and other task-specific observations respectively. Task-specific observations are introduced in section V-A. Action space \mathcal{A} consists of muscle excitation u . We can also call muscle excitation as motor neuron signals.

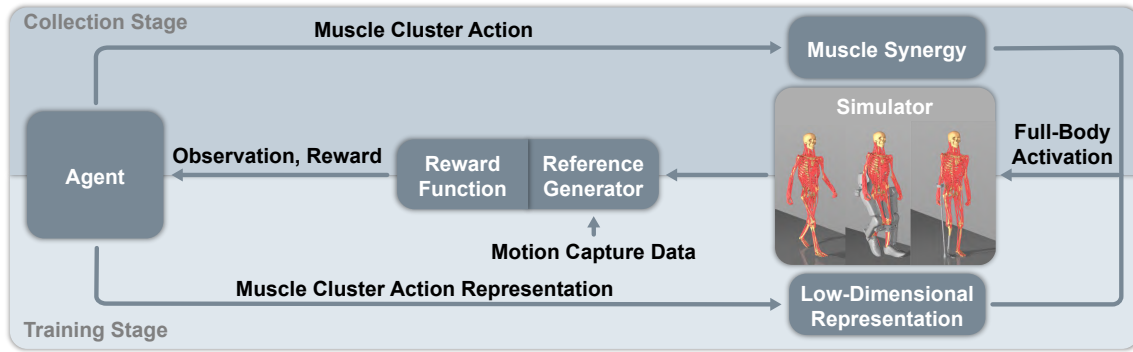


Fig. 4: Two-Stage Hierarchical Training (TSHT). In the collection stage, the agent determines muscle cluster actions. Muscle Synergy function then maps these muscle cluster actions into full-body muscle activation, leveraging prior physiological knowledge. Meanwhile, the reference generator generates the desired joint states based on motion capture data and computes corresponding rewards. Once a sufficient number of sub-optimal trajectories are collected, low-dimensional representations are extracted by encoder-decoder architecture and utilized for the training stage.

B. Motion Imitation

When the reward function is defined in a straightforward manner using forward velocity, the agent is still able to learn a policy to get a high reward, but the gait turns out to be very unnatural. Given the complex nature of elaborate human musculoskeletal models, researchers commonly regulate the search space by reward shaping. Using tracking motion capture clips is one of the very effective reward setting schemes. A recent work [30] integrated motion capture data into the reward function to generate natural movements.

The reference generator shown in Fig. 4 can generate reference joint position q^r and velocity \dot{q}^r taking motion capture data from different tasks. The optical motion capture system calculated the position of markers, based on which we perform an inverse kinematic to reconstruct motions. We then obtain a time series of joint angles. Taking the distance between the actual joint state $\{q, \dot{q}\}$ and the desired joint state $\{q^r, \dot{q}^r\}$ as input, the policy is able to accomplish all considered tasks efficiently.

C. Hierarchical Action Space Representation

Due to the high-dimensional nature of the action space, previous algorithms [30], [31] struggled to maintain control over our full-body musculoskeletal model during locomotion generation. Consequently, the inadequately controlled models frequently collapsed during simulation. Given the vastness of the action space, resolving this issue hinges on the representation of the action space. We employ physiological muscle synergy and a low-dimensional representation of muscle activation to expedite the policy learning process.

Physiological Muscle Synergy. We optimize our policy according to the muscle synergy effect [40]–[42]. Components in human musculoskeletal system are highly redundant. However, each of the muscles is not activated individually. Muscles work synergistically or antagonistically during human movement, where many of them often form clusters that are being activated or deactivated simultaneously for motor control. Muscle synergy could contribute to dimension

reduction. In our experiments, we divide the muscles into anatomical clusters. Muscle Synergy (MS) function assign the excitation to each muscle-tendon unit in the muscle clusters. The control signal, muscle excitation u , is determined by the muscle cluster action u_c , where $u = f_c(u_c)$. f_c is the function mapping the cluster action space into the muscle excitation space. Due to the high degree of freedom of muscles in the torso, we introduced biological muscle synergy. For limb muscles, the control of each muscle-tendon unit is independent.

Low-Dimensional Representation of Muscle Activation.

From a physiological point of view, the majority of high-dimensional human motion tasks are controlled by low-dimensional signals from the brain. Control by relying solely on physiological muscle synergy often fails to perform very well and accurately. Therefore, how to find potential representations in the temporal signals of a large number of muscle activations is the key to solving the problem. Manifold learning [43] aims to find a low-dimensional representation in high-dimensional data as the essential features. AutoEncoder is a powerful neural network architecture that efficiently learns low-dimensional representations from the data. Works such as VAE [44] utilize a large amount of data to find the distribution that fulfills the image generation task. In our work, we use an encoder $E_\phi : \mathcal{X} \in \mathbb{R}^n \rightarrow \mathcal{Z} \in \mathbb{R}^d$ that embed the high-dimensional data into a latent space, where the decoder $D_\phi : \mathcal{Z} \in \mathbb{R}^d \rightarrow \mathcal{X} \in \mathbb{R}^n$ can be used to recover the original data. We use certain sub-optimal muscle-activating signals to formalize the encoder and the decoder. The agent is then trained in the latent space.

D. Two Stages Hierarchical Policy Training

Little effective information can be obtained by adopting End-to-End (E2E) strategies when controlling the motion of musculoskeletal models with imitation learning. Therefore, we divide the entire training process into two stages, namely the collection stage and the training stage. The entire training algorithm is shown in Fig. 4.

Collection Stage. Consider muscle synergy to improve the E2E methods to a certain extent, and then extract low-dimensional representations of the trajectories generated by this policy. At this stage, the policy controls the action of each muscle cluster, and the muscle synergy maps it back to the high-dimensional space of all muscle activations.

Training Stage. Before attaining a detailed understanding of human motor control, people remain restricted by the inherent limitations of physiological muscle synergy. We therefore employ an encoder-decoder architecture to extract more precise low-dimensional representations from the generated trajectories. During the collection stage, we optimize the encoder-decoder architecture to minimize reconstruction errors. The encoder-decoder architecture encompasses various dimensionality reduction and expansion methods. In the training stage, we use the extracted low-dimensional representation for policy learning. The agent outputs the actions to be implemented in the low-dimensional space, and the control of the full-body model is completed through recovery realized by the decoder. By further fine-tuning the rewards, the trained agent can produce more natural actions.

V. EXPERIMENTS

We compare the TSHT algorithm on MS-HUMAN-700 model with representative baselines, including SAR [35] and E2E RL. We design a series of full-body motion tasks and interaction scenarios to demonstrate the superiority of our model and algorithm.

A. Experimental Setup

We designed three tasks: Walking forward, Human-Exoskeleton (Exo) interaction, and Prosthetic walking, to demonstrate full-body motion control, human-machine interaction, and further personalization of pathological conditions. In all environments, the joints on the torso are constrained to achieve better control, with the exception of certain spinal joints. We use optical motion capture to collect motion data for the motion imitation, as shown in Fig. 6a.

Walking forward. We expect the musculoskeletal model to mimic a realistic walking trajectory going forward. The trajectory is generated through the Reference Generator mentioned in section IV-B.

$$r = 10e^{-w_q(q-q^r)^2 - w_{\dot{q}}(\dot{q}-\dot{q}^r) - w_c(q_c - q_c^r)}$$

The reward consists of three parts, where w_q , $w_{\dot{q}}$ and w_c are the weights of the reward for position tracking, velocity tracking and concerned joint position tracking respectively. $w_q = 1, w_{\dot{q}} = 5e^{-3}, w_c = 1$. The musculoskeletal model may be initialized at any point throughout a trajectory cycle. The task does not have task-specific observation.

Human-Exo interaction. We also design a task on human interaction with the MARCH exoskeleton. In this environment, we expected the model to walk in the exoskeleton with minimal contact force in legs. Similarly, we use trajectory tracking for its upper limbs. The joint angles of the exoskeleton are tracking the reference trajectory by a PID

controller. We use spring elements with high stiffness to model the bandage interaction between legs and exoskeleton.

$$r = w_m e^{-(q_u - q_u^r)^2} - w_c \|f_{\text{contact}}\|_2$$

The reward comprises a penalty for contact force and a reward for tracking the upper limb trajectory, where w_m and w_c are the weights of the corresponding rewards. $w_m = 10, w_c = 5e^{-2}$. The task-specific observations for this task include the distance between the ends of the elastic elements on the leg and the exoskeleton.

Prosthetic walking. We replace the right calf and foot of the full-body model with a prosthetic and add a crutch to its right hand. This is done to demonstrate its resistance to external disturbances and imitate gait under pathological conditions. The reward and observations are identical to those used in the walking forward task.

B. Baselines

We compared our algorithm with current state-of-the-art methods: E2E, SAR, and SAR-ours. Our algorithm begins with the collection stage over a time step of length M with muscle synergy, followed by optimization over a training stage of length N . When selecting of encoder-decoder architecture, inspired by SAR, we employed PCA [45] and ICA [46], where $u_c = \text{ICA}(\text{PCA}(u))$. In contrast, E2E is directly trained over a time step of $M + N$. SAR collects low-dimensional representations during a time step of M and is trained over a time step of N . SAR-ours, on the other hand, extracts low-dimensional representations using muscle synergy during the time step of M and then trains over a time step of length N .

C. Experimental Results

In our experiments, we used the SAC implementation from the open-source RL library stable-baselines3 [47]. For the Walking forward and Human-Exo interaction tasks, both the policy network and the value function network took dimensions of (512, 300). For Prosthetic walking, the dimensions were (400, 300), while all other parameters remained at their default settings. All observation values were normalized to a maximum value of 10. For the Walking forward task, the collection stage has a length (M) of 15M, while the training stage (N) is 20M. In the Human-Exo interaction and Prosthetic walking tasks, both M and N are set to 15M. The latent dimension d is 50.

TABLE I: Comparing the maximum returns of algorithms on the three tasks.

	Ours	E2E	SAR	SAR-Ours
Walking forward	371.07	14.72	109.58	23.78
Human-Exo interaction	986.91	-1111.48	-275.09	-1022.01
Prosthetic walking	274.67	101.24	54.83	30.28

By comparing our algorithm with baseline methods across all tasks, we achieved SOTA performance as shown in Table.I and Fig. 5. Neither E2E nor SAR performs well in such high-dimensional tasks.

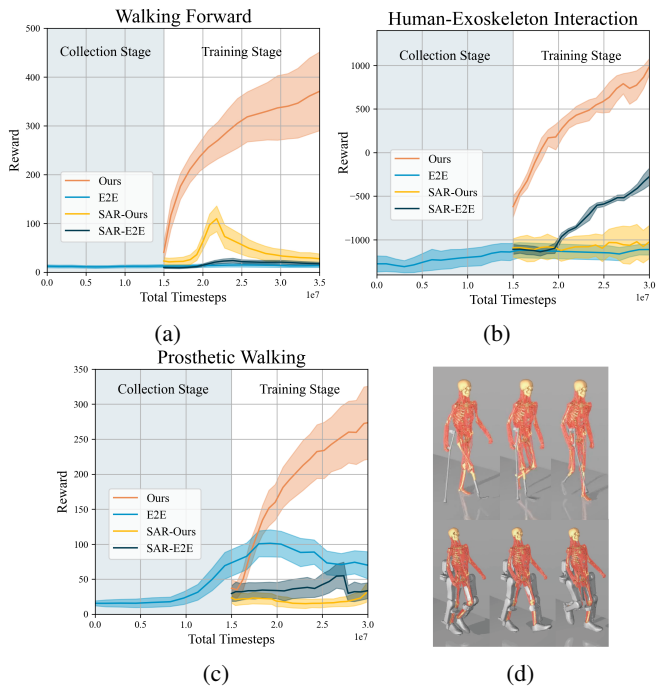


Fig. 5: (a)-(c) Our algorithm outperforms baseline algorithms on different tasks. (d) Learnt locomotion with a prosthetic leg and a crutch (top) and an exoskeleton (bottom).

D. Analysis and Discussion

We demonstrate the visualization of our model’s motion performance in Fig. 2 and Fig. 5. We also provide a supplementary video. As shown in Fig. 5, in the human-exoskeleton interaction task, the algorithm is able to learn a policy that not only maintains upper body balance but also minimizes interaction forces between human and the exoskeleton.

Fig. 6c and Fig. 6d shows the representative joints’ angular position and muscle activity of the lower limbs in simulation. A walking simulation was presented along with the motion capture data and electromyography (EMG) recordings [12], which we consider as our reference. The periodicity and stability of the curves substantiate the robustness of our algorithm’s control efficacy. The discrepancy between the simulated and reference angular positions of the joints can be attributed to suboptimal reference trajectories that fail to ensure balance in the dynamics simulation of the full-body model. Consequently, an agent may opt for a more stable policy, such as increasing hip flexion and ankle dorsiflexion to elevate toe height and prevent tripping or falling, which is indicated in Fig. 6c. The relaxation training phase after imitation [48] may be used to improve the dynamic motion performance. The temporal pattern of simulated muscle activations corresponds with key features of the EMG signals recorded during walking, despite exhibiting discrepancies in timing and magnitude for certain muscle activations when compared to EMG data. Achieving simulated activations that accurately replicate the measured EMG activity across all muscles remains a crucial challenge in studies utilizing muscle-driven simulations [12], [17].

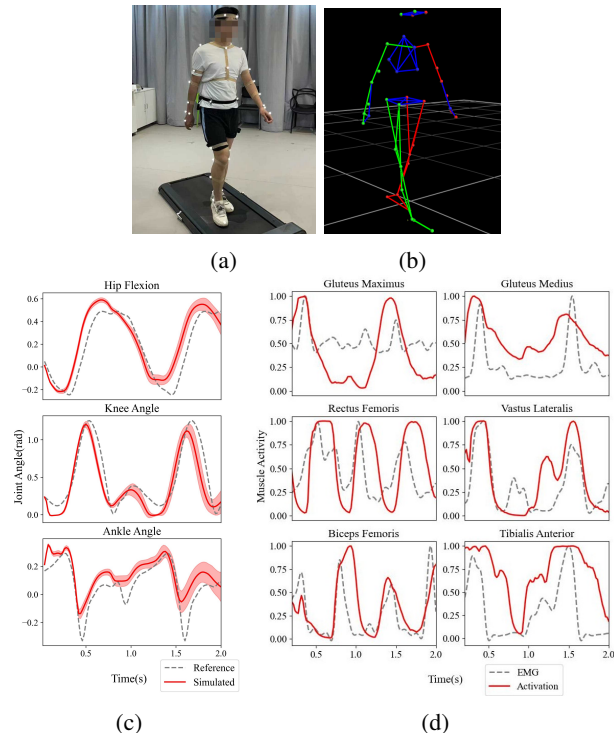


Fig. 6: (a) Our motion capture experiment of human walking. (b) The reconstruction of a full-body walking trajectory in Vicon Nexus. (c) The joint angles in the sagittal(X-Z) plane of the lower limbs, the positive joint directions indicates hip flexion, knee flexion and ankle dorsiflexion respectively. (d) The comparison of simulated muscle activations and the experimental EMG data in 6 representative muscle groups.

VI. CONCLUSION

This work presents a musculoskeletal model (MS-HUMAN-700) with 90 rigid body segments, 206 joints, and 700 muscle-tendon units that encompasses the torso and limbs of human body. The model is adaptable to efficient open-source simulation engines, offering a simulation framework with a versatile interface for neuro-musculo-skeletal control simulation of human full-body movements. We introduce a novel control algorithm designed for high-dimensional and nonlinear musculoskeletal models to generate biologically plausible human motion. Our algorithm employs a hierarchical low-dimensional representation approach, integrating deep reinforcement learning and imitation techniques. Our proposed method exhibits superior performance in full-body musculoskeletal model control tasks, surpassing baseline algorithms significantly. Furthermore, the derived control policy of the agents provides valuable insights into the quantitative analysis of human motion control. Modeling and controlling musculoskeletal systems hold the potential to deepen our understanding of human motor intelligence and human factors in human-machine interactions. As a self-model of human for embodied intelligence, it could serve as a testing ground for the design of interactive robots and offer insights into humanoid behavior.

REFERENCES

- [1] L. H. Ting, S. A. Chvatal, S. A. Safavynia, and J. Lucas McKay, "Review and perspective: neuromechanical considerations for predicting muscle activation patterns for movement," *International journal for numerical methods in biomedical engineering*, vol. 28, no. 10, pp. 1003–1014, 2012.
- [2] N. Bernstein, "The co-ordination and regulation of movements," *The co-ordination and regulation of movements*, 1966.
- [3] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control," *Critical reviews in biomedical engineering*, vol. 17, no. 4, pp. 359–411, 1989.
- [4] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature neuroscience*, vol. 3, no. 11, pp. 1212–1217, 2000.
- [5] E. Todorov, T. Erez, and Y. 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.
- [6] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: open-source software to create and analyze dynamic simulations of movement," *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.
- [7] S.-H. Lee, E. Sifakis, and D. Terzopoulos, "Comprehensive biomechanical modeling and simulation of the upper body," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 4, pp. 1–17, 2009.
- [8] K. R. Saul, X. Hu, C. M. Goehler, M. E. Vidt, M. Daly, A. Velisar, and W. M. Murray, "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model," *Computer methods in biomechanics and biomedical engineering*, vol. 18, no. 13, pp. 1445–1458, 2015.
- [9] D. C. McFarland, E. M. McCain, M. N. Poppo, and K. R. Saul, "Spatial dependency of glenohumeral joint stability during dynamic unimanual and bimanual pushing and pulling," *Journal of Biomechanical Engineering*, vol. 141, no. 5, 2019.
- [10] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, and J. M. Rosen, "An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 8, pp. 757–767, 1990.
- [11] E. M. Arnold, S. R. Ward, R. L. Lieber, and S. L. Delp, "A model of the lower limb for analysis of human movement," *Annals of biomedical engineering*, vol. 38, no. 2, pp. 269–279, 2010.
- [12] 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," *IEEE transactions on biomedical engineering*, vol. 63, no. 10, pp. 2068–2079, 2016.
- [13] Y. Sui, K. ho Kim, and J. W. Burdick, "Quantifying performance of bipedal standing with multi-channel emg," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3891–3896.
- [14] A. K. Lai, A. S. Arnold, and J. M. Wakeling, "Why are antagonist muscles co-activated in my simulation? a musculoskeletal model for analysing human locomotor tasks," *Annals of biomedical engineering*, vol. 45, pp. 2762–2774, 2017.
- [15] A. G. Bruno, M. L. Boussein, and D. E. Anderson, "Development and validation of a musculoskeletal model of the fully articulated thoracolumbar spine and rib cage," *Journal of biomechanical engineering*, vol. 137, no. 8, p. 081003, 2015.
- [16] S. Schmid, K. A. Burkhart, B. T. Allaire, D. Grindle, and D. E. Anderson, "Musculoskeletal full-body models including a detailed thoracolumbar spine for children and adolescents aged 6–18 years," *Journal of biomechanics*, vol. 102, p. 109305, 2020.
- [17] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Transactions On Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.
- [18] J. Chang, D. Chablat, F. Bennis, and L. Ma, "A full-chain opensim model and its application on posture analysis of an overhead drilling task," in *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Human Body and Motion: 10th International Conference, DHM 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part 1*. Springer, 2019, pp. 33–44.
- [19] M. Damsgaard, J. Rasmussen, S. T. Christensen, E. Surma, and M. De Zee, "Analysis of musculoskeletal systems in the anybody modeling system," *Simulation Modelling Practice and Theory*, vol. 14, no. 8, pp. 1100–1111, 2006.
- [20] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.
- [21] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. Karen Liu, "Dart: Dynamic animation and robotics toolkit," *The Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018.
- [22] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.
- [23] A. Ikkala and P. Hämmäläinen, "Converting biomechanical models from opensim to mujoco," in *International Conference on NeuroRehabilitation*. Springer, 2020, pp. 277–281.
- [24] H. Wang, V. Caggiano, G. Durandau, M. Sartori, and V. Kumar, "Myosim: Fast and physiologically realistic mujoco models for musculoskeletal and exoskeletal studies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8104–8111.
- [25] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "Myosuite – a contact-rich simulation suite for musculoskeletal motor control," <https://github.com/MyoHub/myosuite>, 2022. [Online]. Available: <https://sites.google.com/view/myosuite>
- [26] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.
- [27] Y. Lee, M. S. Park, T. Kwon, and J. Lee, "Locomotion control for many-muscle humanoids," *ACM Trans. Graph.*, vol. 33, no. 6, nov 2014. [Online]. Available: <https://doi.org/10.1145/2661229.2661233>
- [28] J. Xu, V. Makovychuk, Y. Narang, F. Ramos, W. Matusik, A. Garg, and M. Macklin, "Accelerated policy learning with parallel differentiable simulation," 2022. [Online]. Available: <https://arxiv.org/abs/2204.07137>
- [29] S. Song, Ł. Kidziński, X. B. Peng, C. Ong, J. Hicks, S. Levine, C. G. Atkeson, and S. L. Delp, "Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation," *Journal of neuroengineering and rehabilitation*, vol. 18, pp. 1–17, 2021.
- [30] V. La Barbera, F. Pardo, Y. Tassa, M. Daley, C. Richards, P. Kormushev, and J. Hutchinson, "Ostrichrl: A musculoskeletal ostrich simulation to study bio-mechanical locomotion," 2021. [Online]. Available: <https://arxiv.org/abs/2112.06061>
- [31] P. Schumacher, D. Häufle, D. Büchler, S. Schmitt, and G. Martius, "Dep-rl: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems," 2022. [Online]. Available: <https://arxiv.org/abs/2206.00484>
- [32] J. Weng, E. Hashemi, and A. Arami, "Natural Walking With Musculoskeletal Models Using Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4156–4162, Apr. 2021, conference Name: IEEE Robotics and Automation Letters.
- [33] V. Caggiano, S. Dasari, and V. Kumar, "Myodex: A generalizable prior for dexterous manipulation," 2023.
- [34] R. Cheng, Y. Sui, D. Sayenko, and J. W. Burdick, "Motor control after human sci through activation of muscle synergies under spinal cord stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 6, pp. 1331–1340, 2019.
- [35] C. Berg, V. Caggiano, and V. Kumar, "Sar: Generalization of physiological agility and dexterity via synergistic action representation," *arXiv preprint arXiv:2307.03716*, 2023.
- [36] P. Walker, E. Rovick, and D. Robertson, "The effects of knee brace hinge design and placement on joint mechanics," *Journal of biomechanics*, vol. 21, no. 11, pp. 965–974, 1988.
- [37] J. Gray, *Gray's anatomy: Selected writings*. Penguin UK, 2013.
- [38] M. Millard, T. Uchida, A. Seth, and S. L. Delp, "Flexing computational muscle: modeling and simulation of musculotendon dynamics," *Journal of biomechanical engineering*, vol. 135, no. 2, p. 021005, 2013.
- [39] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel et al., "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.
- [40] S. A. Chvatal and L. H. Ting, "Common muscle synergies for balance and walking," *Frontiers in computational neuroscience*, vol. 7, p. 48, 2013.
- [41] L. H. Ting and J. L. McKay, "Neuromechanics of muscle synergies for posture and movement," *Current opinion in neurobiology*, vol. 17, no. 6, pp. 622–628, 2007.
- [42] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle

- synergies in the construction of a natural motor behavior,” *Nature neuroscience*, vol. 6, no. 3, pp. 300–308, 2003.
- [43] A. J. Izenman, “Introduction to manifold learning,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 5, pp. 439–446, 2012.
- [44] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [45] M. Ringnér, “What is principal component analysis?” *Nature biotechnology*, vol. 26, no. 3, pp. 303–304, 2008.
- [46] J. V. Stone, “Independent component analysis: a tutorial introduction,” 2004.
- [47] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, “Stable baselines3,” 2019.
- [48] Y. Jin, X. Liu, Y. Shao, H. Wang, and W. Yang, “High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning,” *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1198–1208, Dec. 2022, number: 12 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s42256-022-00576-3>