



Muscles Reduce Neuronal Information Load: Quantification of Control Effort in Biological vs. Robotic Pointing and Walking

Daniel F. B. Haeufle^{1,2*}, Isabell Wochner^{3,4}, David Holzmüller^{5,6}, Danny Driess^{5,7}, Michael Günther³ and Syn Schmitt^{3,4}

¹ Hertie Institute for Clinical Brain Research, University of Tübingen, Tübingen, Germany, ² Center for Integrative Neuroscience, University of Tübingen, Tübingen, Germany, ³ Institute for Modelling and Simulation of Biomechanical Systems, University of Stuttgart, Stuttgart, Germany, ⁴ Stuttgart Center for Simulation Science, University of Stuttgart, Stuttgart, Germany, ⁵ Machine Learning and Robotics Lab, University of Stuttgart, Stuttgart, Germany, ⁶ Institute for Stochastics and Applications, University of Stuttgart, Stuttgart, Germany, ⁷ Max-Planck Institute for Intelligent Systems, Stuttgart, Germany

OPEN ACCESS

Edited by:

Thrishantha Nanayakkara,
Imperial College London,
United Kingdom

Reviewed by:

Leonardo Abdala Elias,
State University of Campinas, Brazil
Cosimo Della Santina,
Massachusetts Institute of
Technology, United States

*Correspondence:

Daniel F. B. Haeufle
daniel.haeufle@uni-tuebingen.de

Specialty section:

This article was submitted to
Soft Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 10 November 2019

Accepted: 07 May 2020

Published: 24 June 2020

Citation:

Haeufle DFB, Wochner I, Holzmüller D, Driess D, Günther M and Schmitt S (2020) Muscles Reduce Neuronal Information Load: Quantification of Control Effort in Biological vs. Robotic Pointing and Walking. *Front. Robot. AI* 7:77. doi: 10.3389/frobt.2020.00077

It is hypothesized that the nonlinear muscle characteristic of biomechanical systems simplify control in the sense that the information the nervous system has to process is reduced through off-loading computation to the morphological structure. It has been proposed to quantify the required information with an information-entropy based approach, which evaluates the minimally required information to control a desired movement, i.e., control effort. The key idea is to compare the same movement but generated by different actuators, e.g., muscles and torque actuators, and determine which of the two morphologies requires less information to generate the same movement. In this work, for the first time, we apply this measure to numerical simulations of more complex human movements: point-to-point arm movements and walking. These models consider up to 24 control signals rendering the brute force approach of the previous implementation to search for the minimally required information futile. We therefore propose a novel algorithm based on the pattern search approach specifically designed to solve this constraint optimization problem. We apply this algorithm to numerical models, which include Hill-type muscle-tendon actuation as well as ideal torque sources acting directly on the joints. The controller for the point-to-point movements was obtained by deep reinforcement learning for muscle and torque actuators. Walking was controlled by proprioceptive neural feedback in the muscular system and a PD controller in the torque model. Results show that the neuromuscular models consistently require less information to successfully generate the movement than the torque-driven counterparts. These findings were consistent for all investigated controllers in our experiments, implying that this is a system property, not a controller property. The proposed algorithm to determine the control effort is more efficient than other standard optimization techniques and provided as open source.

Keywords: muscle, control effort, morphological computation, reinforcement learning, reflexes during walking, information entropy, torque actuator

1. INTRODUCTION

To generate dynamic movements, biological and technical systems actively process information by sensing their state and deriving control signals. The part of the system that performs this active information processing is typically termed *controller*. A controller has to deal with the dynamics characteristics of the controlled system, e.g., the neuronal delays, and the muscular elasticities and nonlinearities in biological systems or the ideally linear torque characteristics in technical systems. While—from a classical engineering point of view—muscular elasticities and nonlinearities complicate the implementation of an adequate controller, several studies show that they are beneficial for the generation of movements in terms of robustness against perturbations (van Soest and Bobbert, 1993; Gerritsen et al., 1998; Wagner and Blickhan, 1999, 2003; Eriten and Dankowicz, 2009; van der Krogt et al., 2009; Haeufle et al., 2010, 2012; John et al., 2013). Examinations of the control of point-to-point movements in the human arm (Pinter et al., 2012; Kambara et al., 2013; Bayer et al., 2017; Stollenmaier et al., 2020; Wochner et al., 2020) as well as in a frog's leg (Giszter et al., 1993), suggest that the neuronal system explicitly relies on the visco-elastic characteristics of the muscles to stabilize a specific posture or to generate smooth dynamic trajectories from jerky control signals.

When discussing the potential contribution of morphology to control, researchers use conjectures like “reduce the control effort” (Blickhan et al., 2007) or “simplify control” (Full and Koditschek, 1999; Holmes et al., 2006) to suggest that less information has to be processed by the biological controller, i.e., the nervous system, during the movement due to the specific morphology. This part is then performed by the morphology, in the sense of “morphological computation” (Paul, 2006; Zahedi and Ay, 2013; Ghazi-Zahedi et al., 2016). A quantitative analysis of the information processing benefit that is gained by these characteristics of the biological system in direct comparison to (technical) systems with different characteristics is possible. For this purpose, we have proposed to measure the minimally required information to generate a movement, i.e., the *control effort* (Haeufle et al., 2014b). Applied to a simplified model of human hopping—with only one actuator and one mechanical degree of freedom—this approach showed that the muscle properties allow reducing the control effort almost by a factor of 20 in comparison to an ideal torque generator model-driven by a PD controller (Haeufle et al., 2014b). This and the existing evidence for muscular benefits in control suggests that relief of effort for the nervous control system may be engraved into muscle design, and may, in other words, have been one of several basic design criteria during the evolution of biological muscle. We, therefore, hypothesize that control effort is relevant in different and more complex movements.

To study this, we here extend the quantification of control effort to more complex movements as, e.g., human point-to-point arm movements and human walking, which is the first novelty of this paper. To determine control effort in complex musculo-skeletal or robotic models with many control signals, we propose a new algorithm (provided online), which is the second novelty of this paper. We applied this algorithm to

two existing musculo-skeletal models: one for arm movements (Driess et al., 2018; Stollenmaier et al., 2020), and one for planar walking (Geyer and Herr, 2010). For each model, a “robotic” version equipped with ideal torque generators was deployed (in analogy to “MOM” in van Soest and Bobbert, 1993). To obtain the controller for point-to-point arm movements, we considered deep reinforcement learning methods. Walking was controlled with proprioceptive neural feedback as well as a PD controller.

2. A SUMMARY OF THE APPROACH TO QUANTIFY CONTROL EFFORT

The measure of *control effort* previously introduced (Haeufle et al., 2014b) quantifies the minimal information required to generate a specific movement. The basis for this is the quantification of the information of the control signals—i.e., sensor signals and actuator command signals—based on SHANNON's information entropy (Shannon and Weaver, 1949). In a nutshell, the idea is to change the resolution of discretized control signals to reduce their information content. If the discretization is too coarse, the movement breaks down. The coarsest resolution where the movement still works represents the minimal information and is termed *control effort*. In the following, we will briefly summarize the concept.

We start with defining parameters for the discretization of the control signals: Each control signal $u_i(t)$ (with $i \in \{1, \dots, N_u\}$ and N_u the number of control signals) is discretized. Discretization limits the number of possible sensor measurement values to n_i (amplitude resolution) and the number of repeated measurements during the movement to m_i (time resolution). Both are positive natural numbers $n_i \in \mathbb{N}_1$ and $m_i \in \mathbb{N}_1$ (excluding 0). Each pair (n_i, m_i) represents the overall resolution of a specific signal u_i . The vector

$$\mathbf{r} = (n_1, m_1, n_2, m_2, \dots, n_i, m_i, \dots, n_{N_u}, m_{N_u}) \in \mathcal{R} \quad (1)$$

is the vector containing all amplitude and time resolution parameters. It has $2N_u$ elements. The set of possible parameter vectors is $\mathcal{R} := \mathbb{N}_1^{2N_u}$, the set of all possible vectors of length $2N_u$ with positive natural numbers.

The information of these control signals can then be calculated by (see also Appendix A):

$$I(\mathbf{r}) = \sum_{i=1}^{N_u} m_i \log_2 n_i \quad (2)$$

with $I: \mathbb{N}_1^{2N_u} \rightarrow \mathbb{R}$. This is a simple monotonic function, which depends on the resolution parameter vector. By reducing the values in \mathbf{r} , i.e., lowering the resolution, the information is reduced.

By reducing the information in the control signal, the movement performance will deteriorate and eventually break down. As an example: if the sensor resolution on the elbow joint position is reduced, the deviation from the target position will eventually increase. To quantify this, we define a performance function $P: \mathbb{N}_1^{2N_u} \rightarrow \mathbb{R}$. This performance function is movement specific and will be specified later (see sections 4.1.2, 4.2.2).

Finding the minimally required information, i.e., the *control effort*, is, thus, a constrained optimization problem of the form

$$\begin{aligned} \min_{\mathbf{r} \in \mathcal{R}} \quad & I(\mathbf{r}) \\ \text{subject to} \quad & P(\mathbf{r}) \leq 0 \end{aligned} \quad (3)$$

The cost function $I(\mathbf{r})$ is cheap to evaluate and straight-forward to optimize. Evaluating the constraint $P(\mathbf{r})$, however, requires the simulation of a movement which is computationally expensive.

3. NEW ALGORITHM TO QUANTIFY CONTROL EFFORT

In the publication, where we proposed the approach to quantify control effort, we searched for the minimal information by brute force (Haeufle et al., 2014b). This was possible due to the small number of control signals ($N_u \leq 3$). For more complex movements with many signals (as investigated here) such an approach needs to be replaced by a systematic optimization. As stated above, it is a constrained optimization problem with a very simple cost function, but a computationally expensive Boolean constraint (movement succeeded or failed), which is additionally stochastic in the presence of motor noise in the investigated arm movements. This prohibits the calculation of a derivative, even by numerical methods. Therefore, all constrained optimization techniques that rely on a gradient of the constraint are not applicable. As the cost function is computationally cheap, it should always be evaluated first and it should be avoided to calculate the constraint function for parameter sets for which it is already clear that the cost is larger than for the currently best parameter set. This makes it difficult to apply algorithms that rely on surrogate functions (e.g., SGHO), as the objective would become very unsteady, also due to the Boolean constraint (a failed movement would be interpreted as a very high cost). Direct search methods seem therefore appropriate (e.g., dual annealing, differential evolution, pattern search). Furthermore, we know two important aspects of our optimization problem: our cost function is monotonically decreasing and if the resolution becomes too coarse, the movement will break down. Thus, we expect a clear border above which the constraint is fulfilled and below it is not. With this knowledge, we can specifically tailor the optimization to minimize the costly calculation of the constraint function. We, therefore, developed a direct search approach that is specifically designed for our optimization problem (Equation 3). The algorithm is based on the pattern search concept, a class of derivative-free direct search algorithms (Todorov and Jordan, 2003; Lewis et al., 2000; Rios and Sahinidis, 2013). In the following, we will describe the concept of the algorithm. Its algorithmic details are given in the Appendix B and the algorithm can be found online at https://github.com/daniel-haeufle/Control_Effort_Optim_Algorithm.

3.1. Outline of the Algorithm

In every iteration of the algorithm, a new set of parameters $\mathbf{r} \in \mathcal{R}$ is selected (polled), evaluated, and the results are compared to the previous best solution (Algorithm 1, Appendix B.1). The key step

of the optimization algorithm is the selection (polling) of new parameter sets \mathbf{r} .

The initial guess of the parameter set \mathbf{r}_{init} has to be with high values for time and amplitude resolution n_i and m_i , almost resembling numerically continuous signals. With the high resolution parameters, \mathbf{r}_{init} fulfills the constraint function $P(\mathbf{r})$. Therefore, it becomes the currently best parameter set in the first iteration: $\mathbf{r} = \mathbf{r}_{\text{init}}$.

Starting from this initial guess, the pattern search algorithm searches for a better solution by exploratory moves (polling) in the parameter space by sampling the function in the vicinity of the currently best parameter set \mathbf{r} . Polling is performed by iteratively adding a specified set D of vectors $\mathbf{d}_l \in D$ (the *pattern*), multiplied by a current mesh size vector $\mathbf{m} \in \mathbb{R}^{2N_u}$ to the currently best solution as

$$\mathbf{r}_{\text{test}} = \mathbf{r} - \mathbf{m} \odot \mathbf{d}_l \quad (4)$$

(where \odot represents the element wise multiplication of the two vectors). The mesh size vector basically contains one “scaling factor” for each resolution parameter (entry in \mathbf{r}). In general, the mesh size vector is reduced (scaled by 0.5) if no better solution is found in the evaluated parameter space and increased (scaled by 2) if a better solution is found. This represents an adaptive search step width (mesh size). For a very helpful overview of the approach of pattern search algorithms, we refer the reader to Torczon (1997).

Our algorithm employs three different polling methods, which differ by the pattern vectors D .

3.1.1. Phase 1: Rapid Parallel Reduction of Resolution in All Signals

The first phase is an initial rough sweep where all signals are treated equally. Polling is done by a bisection search method working uniformly on all entries of \mathbf{r} , i.e., the pattern of this first phase D_1 contains only one vector

$$\bar{\mathbf{d}} = (1, 1, \dots, 1). \quad (5)$$

By adapting the mesh size as described above, this results in a global bisection algorithm acting in parallel on all entries of \mathbf{r} . The bisection search algorithm is shown in Algorithm 2, Appendix B.2. The benefit of phase 1 is that the performance function needs to be evaluated only a few times to identify a first rough performance limit, even for models with a high number of control signals.

3.1.2. Phase 2: Pattern Search

The more thorough sweep of the second phase is more closely inspired by pattern search algorithms. The pattern D_2 consists of the vectors $\mathbf{d}_l := \mathbf{e}_l$, where $\mathbf{e}_l = (0, \dots, 0, 1, 0, \dots, 0)$ is the l -th unit vector. It is motivated by the fact that the cost function is monotonically decreasing for each entry. Thus, with this set, the algorithm only polls in the direction of reduced cost $I(\mathbf{r})$ saving a lot of computational time to less specific search patterns. The vectors \mathbf{d}_l represent a linearly independent basis and only modify each variable individually. This is fine for most cases, but may cause the optimization to converge to an undesired

local minimum. To reduce this risk, we added a set of vectors $\{\bar{\mathbf{d}}_l\}$ in case the previous poll did not reveal any new and better solution. These vectors were constructed such that they had a positive value of 0.5 added to the entry of the previous successful polling direction: $\bar{\mathbf{d}}_l = \mathbf{d}_l + (0, \dots, 0.5, \dots, 0)$. Let us say the previous successful poll modified the second entry of \mathbf{r} . Then, the additional polling vectors would look like this:

$$\begin{aligned}\bar{\mathbf{d}}_1 &= (1, -0.5, 0, \dots, 0) \\ \bar{\mathbf{d}}_2 &= (0, 0.5, 0, \dots, 0) \\ \bar{\mathbf{d}}_l &= (0, -0.5, \dots, 1, \dots, 0) \\ \bar{\mathbf{d}}_{2N_u} &= (0, -0.5, 0, \dots, 1).\end{aligned}$$

These additional vectors represent linear combinations and allow the optimization algorithm to go “back” in one parameter to get out of a local minimum.

Please note: the mesh size vector \mathbf{m} is also adapted as in phase 1. The algorithm for phase 2 is shown in Algorithm 3 Appendix B.3.

3.1.3. Phase 3: Check Local Neighborhood and Calculate Error

The third and final phase is used to scan the local neighborhood of \mathbf{r}_{best} for better solutions and, at the same time, to calculate the error ΔI_{opt} . We allow for three simultaneously changed entries in \mathbf{r} as linear combinations of the vectors in the pattern D_2 to find potentially better solutions. For this systematic sweep, the mesh size vector is not adapted anymore. It is simply a vector of ones. This is shown in Algorithm 4, Appendix B.4. In principle, allowing more than three non-zero entries in \mathbf{d} may further improve the found vector \mathbf{r} . However, this would come with a high computational cost.

3.2. Optimal Result: Control Effort I_{min}

At the end of the third phase, $\mathbf{r}_{\text{opt}} = \mathbf{r}$ represents the best parameter vector found by the algorithm. With this, we calculate the control effort, which is the actual information content (Equation 15, Appendix A) of all signals

$$I_{\text{min}} = I^{\text{Sh}}(\mathbf{r}_{\text{opt}}) \quad (6)$$

$$= \sum_{i=1}^{N_u} \sum_{j=1}^{n_i} p_{ji}^{\text{opt}} \log_2 p_{ji}^{\text{opt}}. \quad (7)$$

This is the minimally required information content of all control signals to generate the desired movement, i.e., still fulfilling the performance constraint $P(\mathbf{r}) = 0$. We identify this minimal information as control effort and symbolize it with I_{min} .

Please note that during the optimization, we assumed equal distribution of the signal values u_{ji} in the range $u_i^{\text{min}} \leq u_{ji} \leq u_i^{\text{max}}$ with $j = 1, \dots, n_i$. Therefore, the probabilities were assumed to be $p_{ji} = 1/n_i$. The cost function of the optimization is based on this assumption and therefore requires no computationally expensive simulation to evaluate the cost function. However, the actual probability $p_{ji} = p(u_i(t) = u_{ji})$ that a signal $u_i(t)$ has the value u_{ji} at time t differs from the original assumption. Therefore,

the actual information at the optimal solution differs too. The real probabilities were estimated from the recorded control signals of the optimal walking simulation (p_{ji}^{opt}) using (Equation 7).

3.3. Error Estimation

We want to quantify the amount of error that we make by confining the components \mathbf{r} to integer values. To this end, our search algorithm calculates an error ΔI_{opt} specifying the maximum information reduction that can be achieved by reducing a single entry of \mathbf{r}_{opt} by one. Specifically, we define

$$\Delta I_{\text{opt}} := \max_{l \in \{1, \dots, 2N_u\}} I(\mathbf{r}_{\text{opt}}) - I(\mathbf{r}_{\text{opt}} - \mathbf{e}_l). \quad (8)$$

For small ΔI_{opt} , we thus expect the discretization of \mathbf{r} to only have little effect on the continuous information $I_{\text{min}} \in [0, \infty)$.

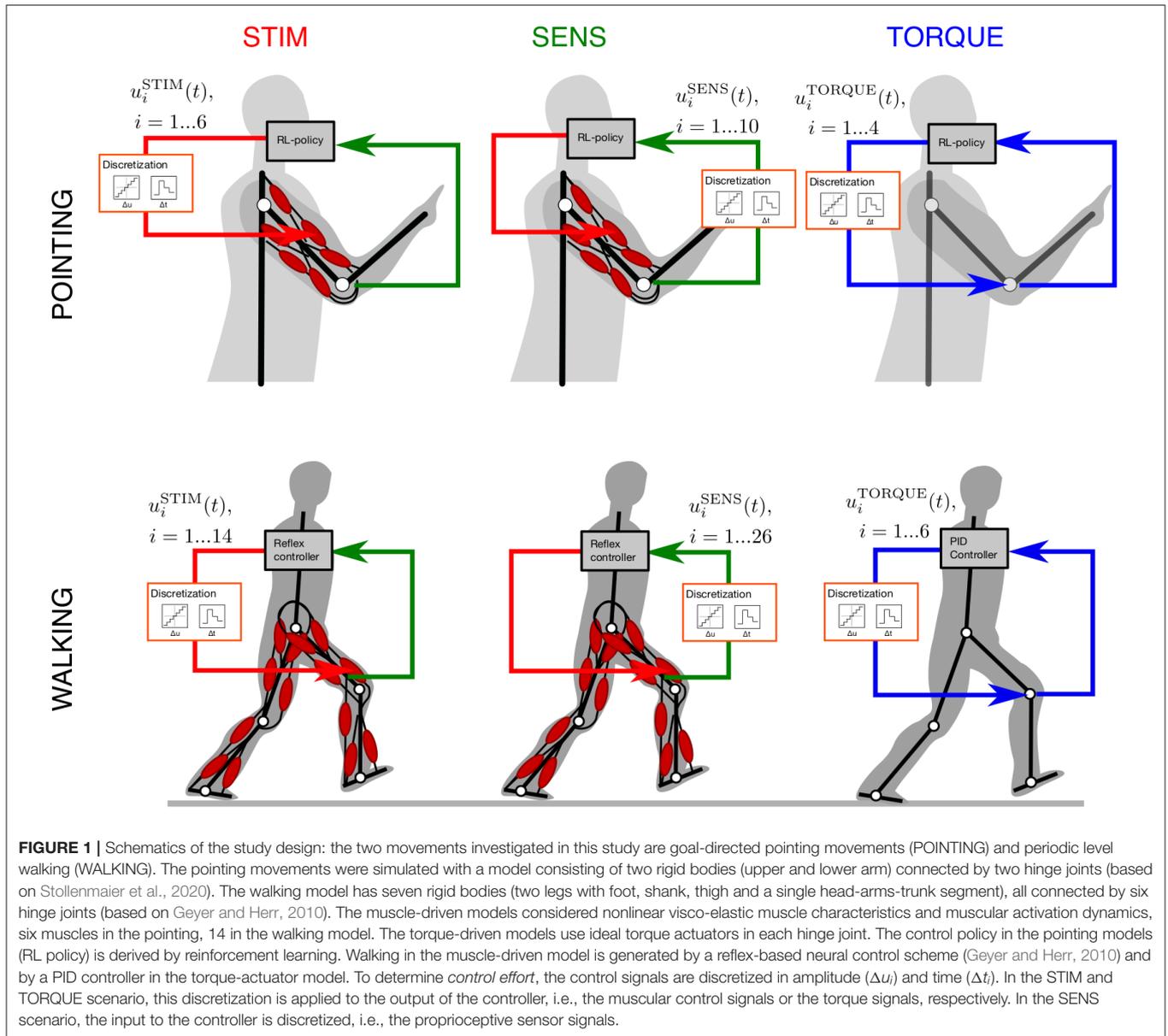
4. CONTROL EFFORT IN TYPICAL HUMAN MOVEMENT TASKS

This study hypothesized that reduced control effort for muscle models over torque actuators found in a simplified hopping model (Haeufle et al., 2014b) is also present in more realistic models of human movements. To test this, we applied the measure described above to biologically plausible models of human point-to-point arm movements and human walking. The models we employed for this study (or very similar ones) have been previously used to study motor control phenomena, where muscle characteristics play a role. Such arm models were used to investigate hypotheses on the control of fast arm movements (Kistemaker et al., 2006), motor learning to compensate for loads during arm movements (Gribble and Ostry, 2000), or the reaction to external forces (Stollenmaier et al., 2020). The walking model was originally used to demonstrate that level walking could be generated by simple reflex control schemes in the spinal cord and does not necessarily require central planning or pattern generators (Geyer and Herr, 2010). It reproduces human muscle activity patterns, joint torques, and kinematics quite well.

In both cases, muscle-driven models were the starting point. For comparison, we derived torque-driven models by stripping these models of all muscular dynamics and considering direct torque actuators in the joints. This resulted in a total of six different cases for which we quantified control effort: two different movements with three different scenarios to quantify control effort each. The two movements were goal-directed (pointing) arm movements and level walking. The three different scenarios to quantify control effort were the following (Figure 1)

- STIM: discretization of the motor signals stimulating the muscles (controller output) in the neuromuscular model.
- SENS: discretization of the sensor signals fed back to the controller (its input) in the neuromuscular model.
- TORQUE: discretization of the motor signals (controller output) fed to ideal torque generators.

STIM and SENS represent the biological, neuromuscular system. TORQUE represents the robotic, technical system. In the



following, we compare control effort for biological and technical systems in the same movement task.

4.1. Movement 1: Pointing

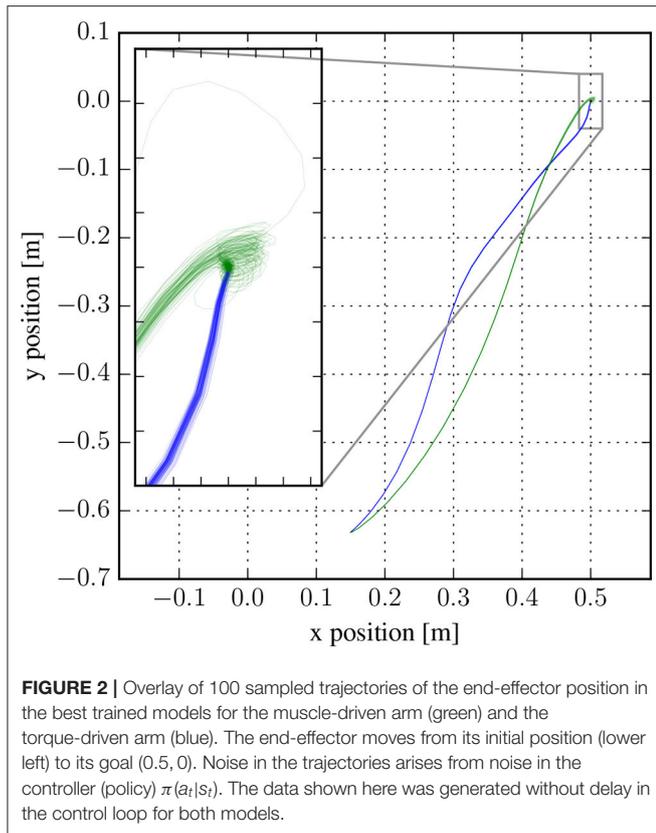
4.1.1. Models

The first movement investigated was a point-to-point arm motion simulated with a 2D arm model (Driess et al., 2018; Stollenmaier et al., 2020). The task is to reach a certain goal position, which also defines the performance criterion P and is described in section 4.1.2 below. The arm model consists of two segments representing the upper and lower arm, which are connected by the elbow and shoulder joint to the fixed shoulder (Driess et al., 2018).

We considered two different ways to generate the actuation torques at the joints: First, a muscle-driven arm model using

six Hill-type muscle-tendon units—four monoarticular and two biarticular muscles—that produce torques through nonlinear moment arms. The model of the muscle-tendon units considers the nonlinear force-length-velocity characteristics of the muscle fibers, the nonlinear elasticity of the tendon (Haeufle et al., 2014a), and the biochemical processes leading from neuronal stimulation to muscle force (Hatze, 1977). With this, it considers the visco-elastic and low-pass filter properties of muscles, which are considered to be important for stabilizing movements (Gerritsen et al., 1998; Haeufle et al., 2010; Pinter et al., 2012; John et al., 2013). The details and the parameters of the model can be found in **Supplementary Material (Data set 1)**.

As a second model to generate the actuation torques, we simply considered two ideal torque actuators that act directly on the joints of the arm.



In both cases, we used a deep reinforcement learning (RL) algorithm to obtain a controller for reaching a certain goal position. Details about the RL algorithms can be found in Appendix C. We use Deep RL since it bears parallels to biological learning (Neftci and Averbeck, 2019), and the task is simple enough so that we can find good controllers using such a very general learning scheme. The goal of the RL algorithm is to find a policy π which maps an observation (related to the state of the arm model) to an action (the control input), hence a closed loop controller, such that the expected sum of rewards $\mathbb{E} \left(\sum_{t=1}^T r_t \right)$ is maximized. A high reward here corresponds to a low deviation from the target position and low applied muscle activations resp. torques. See Appendix C for a proper mathematical definition of these terms.

In our case, the simulation is interrupted every 10 ms in order to get a new control input. We do this for a fixed number $T \in \mathbb{N}$ of iterations. In each of the T iterations, the simulation yields a state s_t . From the state, we compute an observation $o_t = f(s_t)$. A given policy π then yields a probability distribution $\pi(a_t|o_t)$ from which an action a_t is sampled. This action either corresponds to (normalized) muscle stimulations or to torques. Due to the sampling of the probability distribution, this action has small stochasticity included, similar to motor noise. A fixed reward function R is used to compute a reward $r_t = R(s_t, a_t)$. Using the action a_t as a control input, another 10 ms of movement is simulated and the next state $s_{t+1} = S(s_t, a_t)$ is obtained.

Note that the same RL algorithm was used to learn policies for both models (muscle- and torque-driven arm), but the

dimensions of the action and observation spaces differ among these models (cf. Appendix C). By construction of this RL algorithm, the distributions $\pi(a_t|o_t)$ are Gaussian, i.e., the output of the policy always contains additive Gaussian noise with non-zero variance. More specifically,

$$\pi(a_t|o_t) = \mathcal{N}(a_t | \text{NN}(o_t), \text{diag}(e^{2s_1}, \dots, e^{2s_{d_a}})),$$

i.e., a_t follows a Gaussian distribution with mean given by a learnable neural network applied to o_t and a diagonal covariance matrix with learnable parameters s_1, \dots, s_{d_a} , where d_a is the dimension of a_t . While the RL algorithm can adapt the variance of the noise during training, Faisal et al. (2008) suggest that humans can also manage to reduce noise in the nervous system by various complicated mechanisms that cannot easily be modeled or are not yet fully understood. **Figure 2** shows that the generated trajectories still contain remaining noise, especially toward the end of the muscle-arm trajectory.

Another difference between the muscle- and torque-driven arm is that we trained and tested the control policy for the muscle-driven model with a delay for sensor signals of 30 ms similar to the electromechanical delay (Mörl et al., 2012; Rockenfeller and Günther, 2016) in human muscles (De Vlugt et al., 2006), i.e., using $o_t = f(s_{t-3})$. We did not consider any control signal delays in our torque-driven model(s) as such delays can be neglected in real-world technical systems that employ torque drives.

4.1.2. Nonlinear Constraint: Movement Performance for Pointing

For pointing movements, we selected as performance criterion P the accuracy of pointing to a specific point in space. For each poll, five simulation runs were performed to ensure that the stochasticity of the controller does not affect the result. Accordingly, it was checked as a first part of the criterion P whether the arm model's "finger" trajectory ended up in a circle around the desired end goal x_{goal} with radius 2.5 cm:

$$(x_{\text{goal}} - x(t_{\text{end}}))^2 + (y_{\text{goal}} - y(t_{\text{end}}))^2 < (2.5 \text{ cm})^2. \quad (9)$$

Note, that the mean over the five simulation runs was taken for the end position of the trajectories $x(t_{\text{end}})$ and $y(t_{\text{end}})$ to account for the effect of movement variability. The second part of P is necessary to ensure that the "finger" not only passes through the target but actually holds this position. Therefore, it was checked whether both angle velocities \dot{q}_i (again averaged) were smaller than a certain threshold:

$$\dot{q}_i < 0.15 \text{ rad/s}, \quad \text{with } i = 1, 2. \quad (10)$$

Only if both criteria were fulfilled, the poll was considered successful, which gives a conditional expression for the performance criterion as follows:

$$P(\mathbf{r}) = \begin{cases} 0, & \text{if Equation (9) and Equation (10) are true} \\ 1, & \text{otherwise.} \end{cases}$$

4.2. Movement 2: Walking

4.2.1. Models

The second movement investigated in this study was human walking as “defined” by the performance criterion P described in section 4.2.2 below. For this, we resorted to an existing neuromuscular model (Geyer and Herr, 2010). It is a multi-body model with seven segments and hinge-joints in the ankle, knee, and hip. It is actuated by 14 Hill-type muscle-tendon complex models. The muscular control is based on neuronal reflex pathways processing in total 24 proprioceptive signals with biologically realistic neuronal delays. Such a control concept is inspired by the presence of mono-synaptic reflex pathways in the spinal cord, which could explain the low-level implementation of the rhythmic pattern generation of level walking (see Geyer and Herr, 2010, for more details). In forward-dynamic simulations, this model predicts robust walking patterns with strikingly realistic kinematics, ground reaction forces, and muscular activities.

Like for the arm model, we derived a technical model, which, in our case, is a torque-driven model without muscle-tendon characteristics and without neuronal control. This model had the same anthropometrics as the neuromuscular walking model. However, the joint torques for each of the six joints were generated based on PD controllers enforcing the joint kinematics φ_i^{ref} recorded from a reference simulation using the neuromuscular model (Geyer and Herr, 2010):

$$u_i^{\text{TORQUE}} = k_P(\varphi_i - \varphi_i^{\text{ref}}) + k_D(\dot{\varphi}_i - \dot{\varphi}_i^{\text{ref}}),$$

with $i = 1 \dots 6$ for the six joints (2x ankle, 2x knee, 2x hip). This represents a typical low-level control implementation in classical robotics. We here ignore all potential higher-level planning contributions and replace them with the recorded kinematics as the desired trajectory. This is, therefore, equivalent to the level of investigation in the reflex-driven neuromuscular model.

The joint torque was limited to 1.5 times the maximum active values generated by the muscles in the neuromuscular reference simulation. Two sets of feedback gain parameters k_P and k_D , one for stance and one for swing phase were determined in simulations with very fine discretization ($n = 10^{15}$ and $m = 10^{15}$) by a pattern search algorithm (Matlab (R) global optimization toolbox, with random initial conditions). We will show the results for the two best control parameter sets (CP2 and CP10).

The multi-body dynamics of both models are implemented in SimMechanics 1st generation within Matlab(R), SimulinkTM version 2016a. The differential equations are solved with a variable step solver (ode23s stiff/Mod. Rosenbrock) with relative and absolute tolerance of 10^{-3} and 10^{-4} , respectively. After an initial phase of approximately 5 s, the model’s walking pattern is fairly repetitive. Therefore, all evaluations were done on the interval $t \in [5 \text{ s}, 10 \text{ s}]$.

4.2.2. Nonlinear Constraint: Movement Performance for Walking

The nonlinear constraint for the walking model was a combination of a criterion for a desired walking speed and

a second one for “not falling”: From the continuous walking simulation, we can estimate the typical walking speed of the head-arms-trunk (HAT) segment with linear regression to $\dot{x}_{\text{HAT,cont.}} = 1.33 \text{ ms}^{-1}$. The first criterion for the performance limit is for the x-coordinate of the HAT segment x_{HAT} to stay within 6% to this walking speed:

$$\frac{|\dot{x}_{\text{HAT,cont.}} - \dot{x}_{\text{HAT}}(t)|}{\dot{x}_{\text{HAT,cont.}}(t)} < 0.06 \quad (11)$$

The second criterion—not falling—is simply described by the vertical position of the HAT segment y_{HAT} . Simulation tests showed that if the condition

$$y_{\text{HAT}}(t) > 1.24 \text{ m} \quad (12)$$

is violated, the model is falling and not walking anymore.

If at least one of these criteria is violated, the simulation stops (at t_{stop}) and the performance is the time difference to the desired simulation time

$$P(\mathbf{r}) = T - t_{\text{stop}}. \quad (13)$$

The optimization constraint

$$P(\mathbf{r}) = 0 \quad (14)$$

thus only allows for parameter sets which generate walking patterns not violating the above two conditions during the entire simulation time $T = 10 \text{ s}$.

4.3. Discretization of Signals to Determine Control Effort

In principle, it is not clear which signals need to be discretized to determine control effort. At first sight, all output signals of the controller which directly control the actuators—the muscle stimulation or the joint torque signals—seem the obvious choice. However, also the sensor signals provide important information to the system, so it could also be argued that all input signals to the controller need to be discretized (Haeufle et al., 2014b). Here, we tested three scenarios (Figure 1).

4.3.1. Discretize Muscle Stimulations (STIM)

In the first scenario, we discretized the muscle stimulations both in time and amplitude for all muscles with the algorithm given above. These discretized muscle stimulations u_i^{STIM} are then used as an input signal to each muscle. Because the muscles can be activated between 0 and 100%, we set $u_i^{\text{STIM,min}} = 0$ and $u_i^{\text{STIM,max}} = 1$ respectively, as well as the duration of the movement $T^{\text{POINTING}} = 1 \text{ s}$ for the pointing movements and $T^{\text{WALKING}} = 5 \text{ s}$. The time and amplitude resolution parameters m_i and n_i of each of the stimulation signals u_i^{STIM} were then varied with the algorithm described above.

4.3.2. Discretize Proprioceptive Sensor Signals (SENS)

In the second scenario, all proprioceptive sensor signals $u_i^{\text{SENS}}(t)$ are discretized in the neuromuscular models (Figure 1). Here

the minimum u_i^{\min} and maximum u_i^{\max} signal values were determined from a not discretized reference simulation. As above, the duration was $T^{\text{POINTING}} = 1$ s and $T^{\text{WALKING}} = 5$ s and the signal resolution parameters were optimized for minimal information with the algorithm above.

4.3.3. Discretize Torque Actuators (TORQUE)

In the third scenario, we discretized the control signals for the torque-driven system $u_i^{\text{TORQUE}}(t)$ (Figure 1). For the POINTING movement, we set $u_i^{\min} = -20$ Nm and $u_i^{\max} = 20$ Nm to approximately match the capacities of the muscles in the arm model. For the WALKING movement, the minimum u_i^{\min} and maximum u_i^{\max} signal values were determined from a not discretized reference simulation and the durations were again $T^{\text{POINTING}} = 1$ s and $T^{\text{WALKING}} = 5$ s. The signal resolution parameters were optimized for minimal information with the algorithm above.

5. RESULTS

The control effort, i.e., the minimally required information I_{\min} to generate pointing and walking movements, is lower in the neuromuscular models STIM and SENS as compared to the TORQUE model (Figures 3A,B).

In the pointing movements, the control effort is lowest for the STIM scenario ($I_{\min}^{\text{STIM}} = 67.5$ bit/s), where the information is reduced in the output of the controller. The control policy derived by reinforcement learning (RL), however, does not allow to reduce the information as much on the input side (SENS) as on the output side (STIM), resulting in almost three-fold higher control effort ($I_{\min}^{\text{SENS}} = 222.5$ bit/s). The torque model requires the most information, resulting in an almost four-fold higher control effort ($I_{\min}^{\text{TORQUE}} = 310.3$ bit/s).

In the walking model, the control effort for the STIM and SENS scenarios are quite similar and both about half of the best TORQUE model (Figure 3B and Table 1). The second best (CP2) PID controller parameters require double the amount of information than the best parameters (CP10).

The discretization—introduced to reduce the information content of the control signals—modifies the walking pattern. In the STIM and SENS scenarios, the parameters \mathbf{r}_{opt} (minimal information solution) result in slower walking patterns than in the reference solution (Figure 4). In the TORQUE scenario, the parameters \mathbf{r}_{opt} result in strong oscillations in the joint torques (Figure 5).

To demonstrate the reduction in information by changing the resolution parameters n_i and m_i to lower values (more coarse), we give the results for the different stages of the optimization algorithm (Table 1). The algorithm started with an initial guess of $n_i = 10^{15}$ and $m_i = 10^{15}$, resulting in high initial information content I_0 of the control signals which is about two orders of magnitude larger than the optimal result. This initial guess is highest for the SENS model as this model discretizes all 24 sensor signals—the highest number of signals investigated in this study. Therefore, I_0 is naturally high. Also, the number of required iterations is high, especially in the third stage, due to the high number of possible linear combinations checked in this stage. However, at the end of the third stage, the control effort I_{\min} of the SENS model is the lowest.

In this study (Figure 3A), the model scenarios STIM and SENS were trained and tested with a sensor delay ($\delta t = 30$ ms in pointing and $\delta t = 5 \dots 20$ ms in walking), representing the unavoidable neuronal delay (More et al., 2010) in biology, while the torque model had zero delay representing a modern technical solution. To investigate the influence of sensor delays on control effort in some more depth, we additionally trained

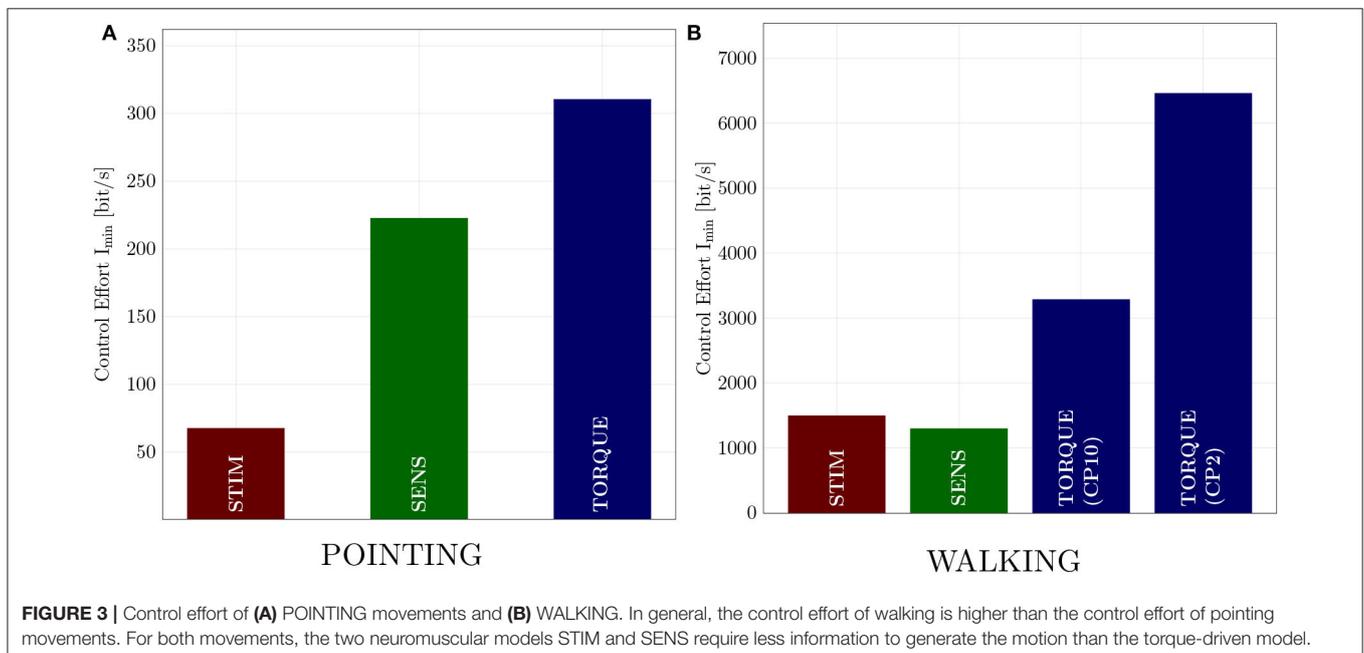
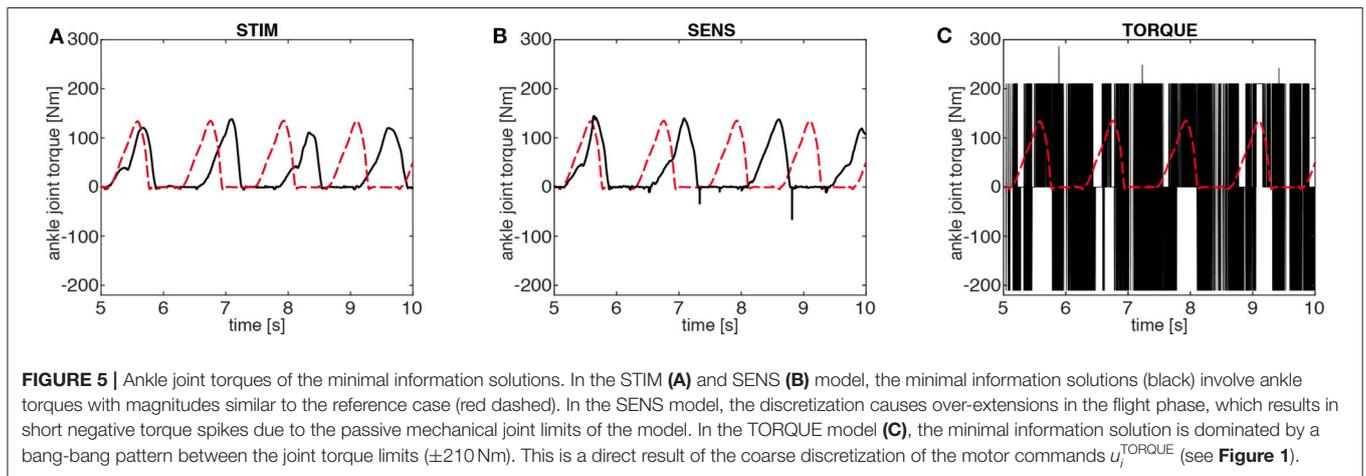
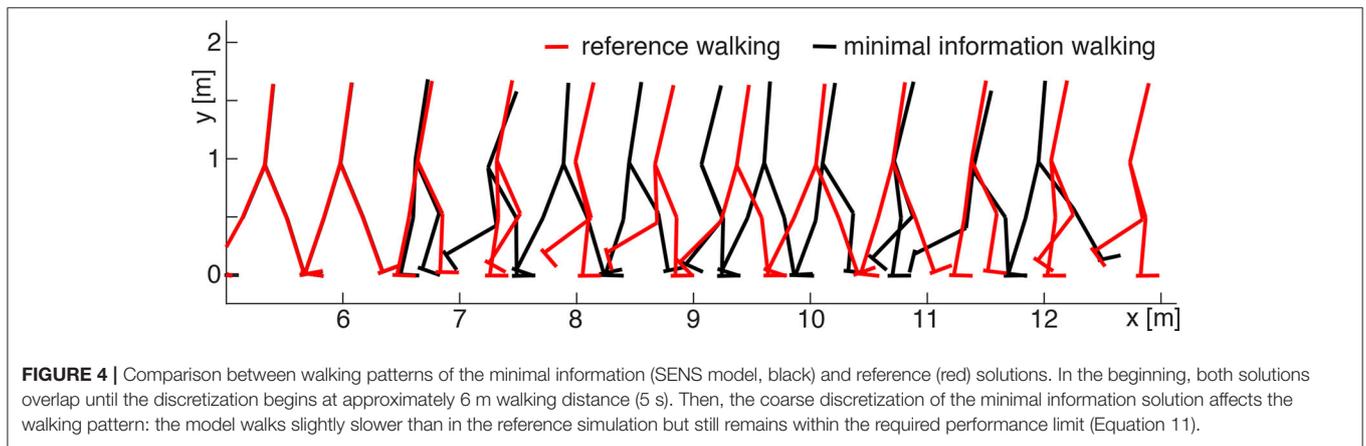


TABLE 1 | Control effort of walking as determined with the adapted pattern search algorithm at the different stages of the optimization.

Model	Number of control signals N_u	Initial I I_0 [kbit/s]	I stage 1 I_1 [kbit/s]	I stage 2 I_2 [kbit/s]	Control effort I_{min} [kbit/s]	Optim. error ΔI_{opt} [kbit/s]
STIM	14	688	3.48 #16	2.68 #611	1.49 #21,715	0.0016
SENS	24	1,229	7.74 #16	1.88 #3,275	1.29 #67,953	0.0063
TORQUE (CP10)	6	295	295 #16	3.54 #527	3.28 #209	0.0002
TORQUE (CP2)	6	295	220 #16	7.84 #590	6.46 #434	0.0060

Also given are the number of iterations # for each stage of the optimization. [kbit/s] means $[10^3 \text{ bit/s}]$.



and tested control policies for the neuromuscular POINTING movements without delay (unphysiological) in the muscle-driven and with delay (bad engineering) in the torque-driven model. The resulting optimized control effort is shown in Figure 6 in relation to the “correct” models. In the muscle-driven models, the control effort increases in the unphysiological zero-delay scenario, while the torque-driven model benefits from zero-delay.

The estimated error of the control effort in walking ΔI_{opt} is small with respect to the difference between the models. The certainty range is an order of magnitude higher than the optimization error. This means that in the last stage of the pattern search, the local neighborhood of the optimal solutions was checked intensively.

Finally, our algorithm is very efficient in finding the control effort. For comparison, we repeated the optimization of the

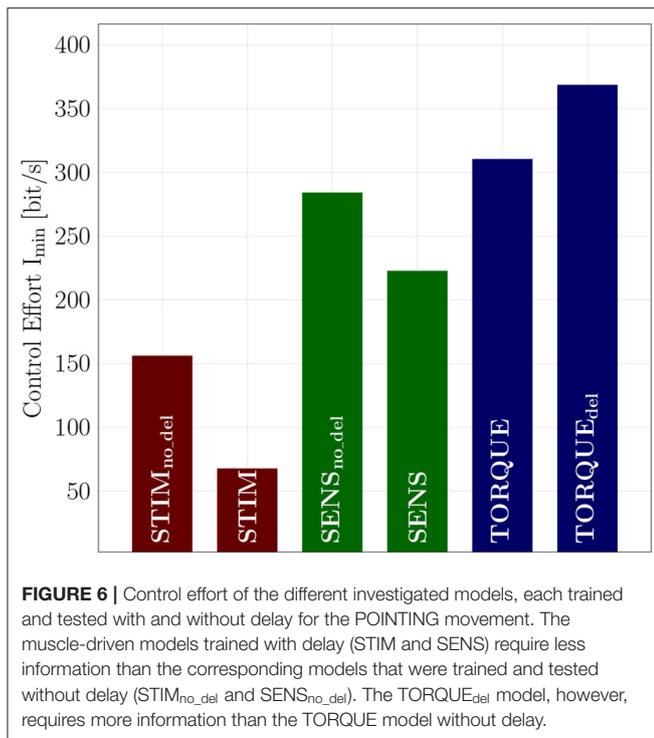


FIGURE 6 | Control effort of the different investigated models, each trained and tested with and without delay for the POINTING movement. The muscle-driven models trained with delay (STIM and SENS) require less information than the corresponding models that were trained and tested without delay (STIM_{no.del} and SENS_{no.del}). The TORQUE_{del} model, however, requires more information than the TORQUE model without delay.

torque model for the pointing movement with other standard optimization algorithms available in Python and found the following: Our algorithm converged in 42 iterations and found a value of 310 bit. In comparison, *dual annealing* stopped at 385 bit after 1000 iterations (the set limit) with 4,067 total function evaluations. *Differential evolution* stopped at 472 bit after 1,000 iterations (the set limit) with 30,033 total function evaluations. *SHGO* did not converge.

6. DISCUSSION

Control effort is reduced in muscle-driven systems compared to torque-driven systems for pointing and walking movements. This supports the general notion that muscle contraction (van Soest and Bobbert, 1993; Haeufle et al., 2010) and activation dynamics (Kistemaker et al., 2005; Rockenfeller and Günther, 2018, app. A) can serve as a low-level zero-delay feedback system (preflexes; Brown et al., 1995) supporting the generation and control of dynamic movements (Ekeberg et al., 2004; Proctor and Holmes, 2010). Here, we provide quantitative evidence for its contribution and the potential reduction in information load. From our point of view, this is interesting, because the two typical movements chosen differ greatly in their characteristics and by the number of muscles needed for their generation. Thus, control effort seems to be a general measure for the contribution of morphology to perform a specific task in biological and robotic motion.

Minimization of information processing may be a design principle for shaping bodies and structures during biological evolution (Niven and Laughlin, 2008) as it certainly comes along with the minimization in metabolic energy consumption

of the information processing structures themselves (Niven et al., 2007). However, it is *competing* with other movement criteria. A prominent example would be the performance, as demonstrated here, but probably other optimization criteria as well, such as maneuverability, jerk, stability, robustness, accuracy, or reproducibility. Pushing this surely incomplete list of potentially relevant movement criteria to extremes, minimization of control effort is definitely competing with the soundness of body tissue: damage and failure are even more costly than corrections and compensations in movement execution. However, we see that the minimization of information processing may be crucial in the evolution of morphology, and our approach allows us to quantify it.

Having this said, we would like to emphasize that we do not expect the actual system to process only this minimal amount of information. Especially in biology, there is an abundance of structures (Latash, 2012)—many muscles, sensors, and neurons—which are not considered here. Also, in robotics, one would never control a robot at this limit, as it is just on the verge of instability. However, by applying this minimal information approach systematically to the same movement but different morphologies, the contribution of the latter can be uncovered and quantified.

6.1. Influence of Delay on Control Effort

Delay in information processing seems per definition unavoidable (Nishikawa et al., 2007; Shadmehr et al., 2010). In robotic systems with their electric cables, the delay can be very small—usually smaller than the typical time resolution Δt . However, it is a universal characteristic of neuronal information processing in biological systems that the delay is, in general, much larger than the time resolution, and scales with the size of the animal (More et al., 2010). Despite these large delays, animals can perform quite well in uncertain environments. In fact, our approach shows that neglecting this delay in the neuromuscular model increases control effort (Figure 6). On the other hand, engineers who employ widespread electric motors do good in trying to minimize delay (Figure 6).

Neuronal systems have additional possibilities that allow them to compensate drawbacks of delays, which are not considered in our models. They may use open-loop control signals—potentially from an inverse model or a model template (Full and Koditschek, 1999; Holmes et al., 2006)—to drive a movement and only use feedback if a perturbation occurs (Todorov and Jordan, 2003). Furthermore, by predicting sensor states with a forward model (e.g., a template), they may deal with possible instabilities arising from delays (Shadmehr, 2010), at least as long as no external perturbation occurs (Kalveram and Seyfarth, 2009). Despite these neuronal capabilities, the control approach can still rely on the stabilizing response of the visco-elastic muscles to external perturbations (van Soest and Bobbert, 1993; Wagner and Blickhan, 2003; Haeufle et al., 2012; Stollenmaier et al., 2020). Brown et al. (1995) termed these responses “preflexes,” due to their zero time-delay response. There are strong indications that such strain-rate-dependent actuator properties, even more in combination with positive muscle force feedback (Geyer et al., 2003), as well as

position-plus-rate characteristics of proprioceptors (McMahon, 1984, p. 154–155) can also provide predictive information that is valuable for movement stabilization. Thus, a delay well-tuned to the controller/control-system interaction may even improve performance (Hedrick and Daniel, 2006; Shadmehr, 2010), and potentially allow to reduce the control effort, as our results indicate.

6.2. Information Processing in Walking Machines

The processed information in a digitally controlled walking machine can be estimated with Appendix A, Equation 17. Although the necessary parameters would be easy to determine for the construction engineer, they are usually not published. As one example, we estimated the parameters for a walking pattern reported for the robot MABEL from information given in Park et al. (2011) and Sreenath et al. (2011). MABEL seems to be interesting for comparison, as it is a 2D walking machine that considers elasticities in the drive. Based on the data provided by the papers, we estimate the total information processed in MABEL per second to be $I = 6.4 \cdot 10^4$ bit/s. The derivation of this is described in more detail in Appendix D.

This value is large in comparison to the minimal information I_{\min} predicted by our models but low in comparison to our initial guess I_0 . Obviously, the choice of encoder resolution is not made to generate walking with the least amount of information. This is very well not recommended in a technical system, as low resolutions entail the risk of significant oscillations, as seen in our optimized TORQUE results. However, with this comparison, we speculate that our results are in a reasonable range. To evaluate the contribution of morphology to the control and to verify our model calculations, it would yet be quite interesting to apply our algorithm to MABEL while further modifying the characteristics of this machine's actuators.

6.3. A Hypothetical Scenario Where an Ideal Torque Generator Would Be Advantageous

Above, we exclusively cited papers indicating and demonstrating the benefit of muscles, and our results fit in this picture. Therefore, it is important to point out that control effort cannot be expected to always be lower in muscle-driven systems. For this, we would like to perform a short though experiment. Imagine a hypothetical task: an arm with a single joint has to generate exactly the same tangential endeffector force at any given joint angle (at rest). An ideal torque generator, would require a single constant input signal. Such a signal, per definition of Equation (2), contains the minimally possible control effort as the resolution parameters could be reduced to $m = 1$ and $n = 1$, and, consequently, $I_{\min} = 0$. A muscle-actuated arm, on the other hand, would have to adapt the stimulation to each and every angle as the muscle force depends on its length and therefore also on the joint angle. This is only to highlight that muscles are very well suited for particular dynamic tasks, but not in general the best actuator for everything.

6.4. Other Optimal Control Approaches for Measuring Simplicity

In the present work, we utilize control effort, which has recently been proposed by us (Haeufle et al., 2014b), to quantify the minimal information required to generate a specific movement. This measure is based on the quantification of the information of the control signals, i.e., sensor signals and actuator command signals, based on SHANNON's information entropy (Shannon and Weaver, 1949, see section 2). By comparing control effort for different morphologies, it quantifies, to some extent, how "simple" it is to generate a specific movement depending on the morphology of the system.

Brockett (1997) argue to consider simplicity as a way to *synthesize* controllers, which they call Minimum Attention Control (MAC). In order to measure simplicity, they introduce the concept of attention, which quantifies the required rate of change of the control to achieve desired changes in the system state. This can be interpreted as the difficulty to implement a respective controller (Brockett, 1997). For example, a control system which can be controlled by just a constant input would require minimal (no) attention. Thus, the basic idea is to find controllers through an optimal control framework where the objective function trades-off system performance with attention, i.e., simplicity of the controller.

In Della Santina et al. (2017), MAC was found as a beneficial solution for controller design in soft robots. Biomechanical systems such as the arm model used in the present work have properties which enable to reach a desired system state with a constant control input. This has been exploited in Driess et al. (2018), Wochner et al. (2020) and Driess et al. (2019) to learn a controller for such systems efficiently. The controllers of Driess et al. (2018) and Driess et al. (2019), Wochner et al. (2020) are, by design, optimal with respect to attention with zero attention, since the controller produces constant controls for each desired system state.

The measure of attention from Brockett (1997) is, in a way, similar to control effort of the present work, as it is also driven by the idea that a specific design of a control system could be beneficial to achieve a certain system behavior without an overly complex controller. Thus, the process of evolving structures and functioning—simultaneous and codependent control system and controller design—can also be supported by MAC.

However, there is also an important difference between MAC and control effort as considered in the present work. MAC is a paradigm to *synthesize* controllers by integrating it directly into the cost function of an optimal control framework. In contrast, we use control effort here as a measure to analyze the contribution of the systems dynamics to the control of the movement. Therefore, it measures a system property. Indeed, the controllers that are either learned or hand-tuned in this work at no point have the objective to minimize control effort. Solving the optimal control problem with MAC as an objective is non-trivial, especially for nonlinear systems. In the future it could be investigated whether MAC

can be extended to nonlinear biomechanical models and to test whether it allows to find controllers that show a difference in attention between musculoskeletal and torque-driven actuators.

6.5. The Optimization Algorithm

Quantifying control effort requires to solve an optimization problem (Equation 3). The algorithm proposed here is novel and specifically designed to efficiently optimize the given problem. The three stages of the algorithm differ in their computational expense, with the first stage being computationally cheap (16 iterations for the walking model), while the other two require more iterations (**Table 1**). For the few control signals discretized in the TORQUE model, the final search in stage 3 is also computationally cheap. For more control signals, the linear combinations tested in the third stage are computationally expensive. It may be considered to exclude the final stage, as we did for the POINTING movements, since the difference in the results between stage two (I_2) and final result (I_{\min}) in the walking model are not very large, and the general trend can already be seen. In general, this algorithm can easily be applied to any other simulation of movements, and also to robotic systems (which would, however, require safety measures to avoid damage in the low-resolution trials). By providing it as open source, we hope to foster the quantitative evaluation of control effort and a more systematic study of the contribution of morphology to control in biological systems.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

REFERENCES

- Bayer, A., Schmitt, S., Günther, M., and Haeufle, D. F. B. (2017). The influence of biophysical muscle properties on simulating fast human arm movements. *Comput. Methods Biomech. Biomed. Eng.* 20, 803–821. doi: 10.1080/10255842.2017.1293663
- Blickhan, R., Seyfarth, A., Geyer, H., Grimmer, S., Wagner, H., and Günther, M. (2007). Intelligence by mechanics. *Philos. Trans. R. Soc. A* 365, 199–220. doi: 10.1098/rsta.2006.1911
- Brockett, W. (1997). “Minimum attention control,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 3 (San Francisco, CA), 2628–2632. doi: 10.1109/CDC.1997.657776
- Brown, I. E., Scott, S. H., and Loeb, G. E. (1995). Preflexes—programmable high-gain zero-delay intrinsic responses of perturbed musculoskeletal systems. *Soc. Neurosci.* 21:562.
- De Vlugt, E., Schouten, A. C., and Van Der Helm, F. C. (2006). Quantification of intrinsic and reflexive properties during multijoint arm posture. *J. Neurosci. Methods* 155, 328–349. doi: 10.1016/j.jneumeth.2006.01.022
- Della Santina, C., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M., et al. (2017). Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robot. Autom. Mag.* 24, 75–83. doi: 10.1109/MRA.2016.2636360
- Driess, D., Schmitt, S., and Toussaint, M. (2019). “Active inverse model learning with error and reachable set estimates,” in *Proc. of the Int. Conf. on*

AUTHOR CONTRIBUTIONS

DFBH, MG, and SS: project idea. DFBH, SS, and MG: control effort algorithm. DFBH, IW, DH, and DD: model. DFBH and IW: data. All authors: paper.

FUNDING

The research was supported by the Ministry of Science, Research and the Arts Baden-Württemberg (Az: 33-7533.-30-20/7/2). This work was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanys’ Excellence Strategy—EXC 2075-390740016. We further acknowledge support by the International Max-Planck Research School for Intelligent Systems and by the Open Access Publishing Fund of the University of Tübingen.

ACKNOWLEDGMENTS

We would like to thank Daniel Ossig, who spent many hours with a tedious search for specifics of the control frequency and sensor resolution of a humanoid walking robot. We would also like to thank Hartmut Geyer, who provided the walking model, and Dan Suissa, who developed the first version of the arm model. We would like to thank Marc Toussaint for his support in the project. Furthermore, DFBH would like to thank Keyan Ghazi-Zahedi for the fantastic discussions on everything, but also on control effort and morphological computation.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00077/full#supplementary-material>

Intelligent Robots and Systems (IROS) (Macau). doi: 10.1109/IROS40897.2019.8967858

- Driess, D., Zimmermann, H., Wolfen, S., Suissa, D., Haeufle, D., Hennes, D., et al. (2018). “Learning to control redundant musculoskeletal systems with neural networks and SQP: exploiting muscle properties,” in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)* (Brisbane, QLD). doi: 10.1109/ICRA.2018.8463160
- Ekeberg, Ö., Blümel, M., and Büschges, A. (2004). Dynamic simulation of insect walking. *Arthrop. Struct. Dev.* 33, 287–300. doi: 10.1016/j.asd.2004.05.002
- Eriten, M., and Dankowicz, H. (2009). A rigorous dynamical-systems-based analysis of the self-stabilizing influence of muscle. *J. Biomech. Eng.* 131, 011011-1-9. doi: 10.1115/1.3002758
- Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nat. Rev. Neurosci.* 9, 292–303. doi: 10.1038/nrn2258
- Full, R., and Koditschek, D. (1999). Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *J. Exp. Biol.* 202(Pt 23), 3325–3332.
- Gerritsen, K. G., van den Bogert, A. J., Hulliger, M., and Zernicke, R. F. (1998). Intrinsic muscle properties facilitate locomotor control—a computer simulation study. *Motor Control* 2, 206–220. doi: 10.1123/mcj.2.3.206
- Geyer, H., and Herr, H. (2010). A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and

- muscle activities. *IEEE Trans. Neural Syst. Rehabil. Eng.* 18, 263–73. doi: 10.1109/TNSRE.2010.2047592
- Geyer, H., Seyfarth, A., and Blickhan, R. (2003). Positive force feedback in bouncing gaits? *Proc. R. Soc. Lond. B* 270, 2173–2183. doi: 10.1098/rspb.2003.2454
- Ghazi-Zahedi, K., Haeufle, D. F. B., Montúfar, G., Schmitt, S., and Ay, N. (2016). Evaluating morphological computation in muscle and dc-motor driven models of human hopping. *Front. Robot. AI* 3:42. doi: 10.3389/frobt.2016.00042
- Giszter, S., Mussa-Ivaldi, F., and Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *J. Neurosci.* 13, 467–491. doi: 10.1523/JNEUROSCI.13-02-00467.1993
- Gribble, P. L., and Ostry, D. J. (2000). Compensation for loads during arm movements using equilibrium-point control. *Exp. Brain Res.* 135, 474–482. doi: 10.1007/s002210000547
- Haeufle, D. F., Grimmer, S., and Seyfarth, A. (2010). The role of intrinsic muscle properties for stable hopping - stability is achieved by the force-velocity relation. *Bioinspir. Biomimet.* 5:016004. doi: 10.1088/1748-3182/5/1/016004
- Haeufle, D. F. B., Grimmer, S., Kalveram, K.-T., and Seyfarth, A. (2012). Integration of intrinsic muscle properties, feed-forward and feedback signals for generating and stabilizing hopping. *J. R. Soc. Interface* 9, 1458–1469. doi: 10.1098/rsif.2011.0694
- Haeufle, D. F. B., Günther, M., Bayer, A., and Schmitt, S. (2014a). Hill-type muscle model with serial damping and eccentric force-velocity relation. *J. Biomech.* 47, 1531–1536. doi: 10.1016/j.jbiomech.2014.02.009
- Haeufle, D. F. B., Günther, M., Wunner, G., and Schmitt, S. (2014b). Quantifying control effort of biological and technical movements: an information-entropy-based approach. *Phys. Rev. E* 89:012716. doi: 10.1103/PhysRevE.89.012716
- Hatze, H. (1977). A myocybernetic control model of skeletal muscle. *Biol. Cybern.* 25, 103–119. doi: 10.1007/BF00337268
- Hedrick, T. L., and Daniel, T. (2006). Flight control in the hawkmoth *manduca sexta*: the inverse problem of hovering. *J. Exp. Biol.* 209, 3114–3130. doi: 10.1242/jeb.02363
- Holmes, P., Full, R., Koditschek, D., and Guckenheimer, J. (2006). The dynamics of legged locomotion: models, analyses, and challenges. *SIAM Rev.* 48, 207–304. doi: 10.1137/S0036144504445133
- John, C. T., Anderson, F. C., Higginson, J. S., and Delp, S. L. (2013). Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation. *Comput. Methods Biomech. Biomed. Eng.* 16, 451–462. doi: 10.1080/10255842.2011.627560
- Kalveram, K. T., and Seyfarth, A. (2009). Inverse biomimetics: how robots can help to verify concepts concerning sensorimotor control of human arm and leg movements. *J. Physiol.* 103, 232–243. doi: 10.1016/j.jphysparis.2009.08.006
- Kambara, H., Shin, D., and Koike, Y. (2013). A computational model for optimal muscle activity considering muscle viscoelasticity in wrist movements. *J. Neurophysiol.* 109, 2145–2160. doi: 10.1152/jn.00542.2011
- Kistemaker, D., van Soest, A., and Bobbert, M. (2005). Length-dependent [Ca²⁺] sensitivity adds stiffness to muscle. *J. Biomech.* 38, 1816–1821. doi: 10.1016/j.jbiomech.2004.08.025
- Kistemaker, D. a., Van Soest, A. J., and Bobbert, M. F. (2006). Is equilibrium point control feasible for fast goal-directed single-joint movements? *J. Neurophysiol.* 95, 2898–2912. doi: 10.1152/jn.00983.2005
- Latash, M. L. (2012). The bliss (not the problem) of motor abundance (not redundancy). *Exp. Brain Res.* 217, 1–5. doi: 10.1007/s00221-012-3000-4
- Lewis, R. M., Torczon, V., and Trosset, M. W. (2000). Direct search methods: then and now. *J. Comput. Appl. Mathematics* 124, 191–207. doi: 10.1016/S0377-0427(00)00423-4
- McMahon, T. (1984). *Muscles, Reflexes, and Locomotion*. Princeton, NJ: Princeton University Press.
- More, H. L., Hutchinson, J. R., Collins, D. F., Weber, D. J., Aung, S. K., and Donelan, J. M. (2010). Scaling of sensorimotor control in terrestrial mammals. *Proc. R. Soc. B* 277, 3563–3568. doi: 10.1098/rspb.2010.0898
- Mörl, F., Siebert, T., Schmitt, S., Blickhan, R., and Guenther, M. (2012). Electro-mechanical delay in hill-type muscle models. *J. Mech. Med. Biol.* 12:1250085. doi: 10.1142/S0219519412500856
- Neftci, E. O., and Averbeck, B. B. (2019). Reinforcement learning in artificial and biological systems. *Nat. Mach. Intell.* 1, 133–143. doi: 10.1038/s42256-019-0025-4
- Nishikawa, K., Biewener, A. A., Aerts, P., Ahn, A. N., Chiel, H. J., Daley, M. A., et al. (2007). Neuromechanics: an integrative approach for understanding motor control. *Integr. Compar. Biol.* 47, 16–54. doi: 10.1093/icb/icm024
- Niven, J., Anderson, J., and Laughlin, S. (2007). Fly photoreceptors demonstrate energy-information trade-offs in neural coding. *PLoS Biol.* 5:e116. doi: 10.1371/journal.pbio.0050116
- Niven, J., and Laughlin, S. (2008). Energy limitation as a selective pressure on the evolution of sensory systems. *J. Exp. Biol.* 211(Pt 11), 1792–1804. doi: 10.1242/jeb.017574
- Park, H. W., Sreenath, K., Hurst, J. W., and Grizzle, J. W. (2011). Identification of a bipedal robot with a compliant drivetrain: parameter estimation for control design. *IEEE Control Syst. Mag.* 31, 63–88. doi: 10.1109/MCS.2010.939963
- Paul, C. (2006). Morphological computation: a basis for the analysis of morphology and control requirements. *Robot. Auton. Syst.* 54, 619–630. doi: 10.1016/j.robot.2006.03.003
- Pinter, I. J., Van Soest, A. J., Bobbert, M. F., and Smeets, J. B. J. (2012). Conclusions on motor control depend on the type of model used to represent the periphery. *Biol. Cybern.* 106, 441–451. doi: 10.1007/s00422-012-0505-7
- Proctor, J., and Holmes, P. (2010). Reflexes and preflexes: on the role of sensory feedback on rhythmic patterns in insect locomotion. *Biol. Cybern.* 102, 513–531. doi: 10.1007/s00422-010-0383-9
- Rios, L. M., and Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Global Optim.* 56, 1247–1293. doi: 10.1007/s10898-012-9951-y
- Rockenfeller, R., and Günther, M. (2016). Extracting low-velocity concentric and eccentric dynamic muscle properties from isometric contraction experiments. *Math. Biosci.* 278, 77–93. doi: 10.1016/j.mbs.2016.06.005
- Rockenfeller, R., and Günther, M. (2018). Inter-filament spacing mediates calcium binding to troponin: a simple geometric-mechanistic model explains the shift of force-length maxima with muscle activation. *J. Theor. Biol.* 454, 240–252.
- Shadmehr, R. (2010). Control of movements and temporal discounting of reward. *Curr. Opin. Neurobiol.* 20, 726–730. doi: 10.1016/j.conb.2010.08.017
- Shadmehr, R., Smith, M. A., and Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annu. Rev. Neurosci.* 33, 89–108. doi: 10.1146/annurev-neuro-060909-153135
- Shannon, C., and Weaver, W. (1949). *The Mathematical Theory of Communication*. Urbana: University of Illinois Press.
- Sreenath, K., Park, H.-W., Poulakakis, I., and Grizzle, J. W. (2011). A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *Int. J. Robot. Res.* 30, 1170–1193. doi: 10.1177/0278364910379882
- Stollenmaier, K., Ilg, W., and Haeufle, D. (2020). Predicting Perturbed human arm movements in a neuro-musculoskeletal model to investigate the muscular force response. *Front. Bioeng. Biotechnol.* 8:308. doi: 10.3389/fbioe.2020.00308
- Todorov, E., and Jordan, M. I. (2003). “A minimal intervention principle for coordinated movement,” in *Advances in Neural Information Processing Systems*, 27–34.
- Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM J. Optim.* 7, 1–25. doi: 10.1137/S1052623493250780
- van der Krogt, M. M., de Graaf, W. W., Farley, C. T., Moritz, C. T., Richard Casius, L. J., and Bobbert, M. F. (2009). Robust passive dynamics of the musculoskeletal system compensate for unexpected surface changes during human hopping. *J. Appl. Physiol.* 107, 801–808. doi: 10.1152/jappphysiol.91189.2008
- van Soest, A., and Bobbert, M. (1993). The contribution of muscle properties in the control of explosive movements. *Biol. Cybern.* 69, 195–204. doi: 10.1007/BF00198959

- Wagner, H., and Blickhan, R. (1999). Stabilizing function of skeletal muscles: an analytical investigation. *J. Theor. Biol.* 199, 163–179. doi: 10.1006/jtbi.1999.0949
- Wagner, H., and Blickhan, R. (2003). Stabilizing function of antagonistic neuromusculoskeletal systems: an analytical investigation. *Biol. Cybern.* 89, 71–79. doi: 10.1007/s00422-003-0403-0
- Wochner, I., Driess, D., Zimmermann, H., Haeufle, D. F. B., Toussaint, M., Schmitt, S., (2020). Optimality principles in human point-to-manifold reaching accounting for muscle dynamics. *Front. Comput. Neurosci.* 14. doi: 10.3389/fncom.2020.00038
- Zahedi, K., and Ay, N. (2013). Quantifying morphological computation. *Entropy* 15, 1887–1915. doi: 10.3390/e15051887

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Haeufle, Wochner, Holzmüller, Driess, Günther and Schmitt. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.