

Predicting Mid-Air Interaction Movements and Fatigue Using Deep Reinforcement Learning

Noshaba Cheema
Max-Planck Institute for
Informatics & DFKI
Saarbrücken, Germany
ncheema@mpi-inf.mpg.de

Laura A. Frey-Law
Carver College of Medicine,
University of Iowa Health Care
Iowa City, USA
laura-freylaw@uiowa.edu

Kourosh Naderi
Games Research Group,
Aalto University
Helsinki, Finland
kourosh.naderi@aalto.fi

Jaakko Lehtinen
Aalto University &
NVIDIA Research
Helsinki, Finland
jaakko.lehtinen@aalto.fi

Philipp Slusallek
Computer Graphics Lab,
Saarland University & DFKI
Saarbrücken, Germany
philipp.slusallek@dfki.de

Perttu Hämäläinen
Games Research Group,
Aalto University
Helsinki, Finland
perttu.hamalainen@aalto.fi

ABSTRACT

A common problem of mid-air interaction is excessive arm fatigue, known as the “Gorilla arm” effect. To predict and prevent such problems at a low cost, we investigate user testing of mid-air interaction without real users, utilizing biomechanically simulated AI agents trained using deep Reinforcement Learning (RL). We implement this in a pointing task and four experimental conditions, demonstrating that the simulated fatigue data matches human fatigue data. We also compare two effort models: 1) instantaneous joint torques commonly used in computer animation and robotics, and 2) the recent Three Compartment Controller (3CC-*r*) model from biomechanical literature. 3CC-*r* yields movements that are both more efficient and relaxed, whereas with instantaneous joint torques, the RL agent can easily generate movements that are quickly tiring or only reach the targets slowly and inaccurately. Our work demonstrates that deep RL combined with the 3CC-*r* provides a viable tool for predicting both interaction movements and user experience *in silico*, without users.

Author Keywords

Computational Interaction; User Modeling; Biomechanical Simulation; Reinforcement Learning.

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI)**; *User models*; User studies; •**Theory of computation** → *Reinforcement learning*;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI'20, April 25–30, 2020, Honolulu, HI, USA

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6708-0/20/04.

DOI: <https://doi.org/10.1145/3313831.3376701>

INTRODUCTION

Interactive devices, such as touch screens or Virtual Reality (VR) goggles are becoming increasingly important in areas like entertainment, medicine, or virtual design. These devices enable a more intuitive and natural user experience with the use of mid-air gestures as an interaction tool. As such, physical ergonomics is an important design factor for mid-air interaction. In particular, arm fatigue, also known as “Gorilla arm effect” [34, 38], is a common problem that negatively effects user experience.

Designing interactive experiences is fundamentally difficult because the design goals are typically defined in terms of subjective qualities such as user enjoyment in games or low perceived exertion or effort in gestural interaction. Thus, the effects of design decisions are nearly impossible to predict without testing with actual users or players. In practice, interaction design is often an iterative process of trial-and-error where design is gradually and expensively improved through observing users interacting with prototypes.

A rising trend in design and human-computer interaction is to utilize computational models of users to predict the user experience [6, 19, 27, 54, 55, 70]. If this can be done with sufficient accuracy, one can rapidly evaluate alternative solutions to design problems *in silico*, without users, or at least preselect the most likely solutions to be tested in real life. Furthermore, if a computational model can evaluate a solution, a designer can deploy optimization algorithms to automatically find and propose high-value solutions.

Computational user models have been successfully applied in, e.g., game playing [28, 70, 72] and typing [54]. However, many complex interactions are still challenging to model, in particular in the domain of embodied experiences such as Virtual Reality (VR), which require modeling the user’s body and biomechanics. Fortunately, new and powerful tools are emerging: Recent advances in deep Reinforcement Learning (RL) [63, 49, 29, 42] provide a generic approach to train

intelligent agents for any kind of simulated system such as a video game or biomechanical simulation, provided that one can define the agent’s goals or tasks as a reward function such as a game score. For user modeling, this means that one needs to make minimal assumptions about user behavior; instead, the agents will explore and discover the behaviors of maximal utility – i.e., cumulative rewards – following a computational rationality model of behavior [26, 45]. Such AI agents can also be extended with models of intrinsic motivation and emotion [61, 50], which can allow prediction of the user experience and behavior beyond simple task-driven behavior and associated metrics like task success rate [27].

Contribution: We contribute the first user modeling experiment that combines deep RL with a biomechanical arm simulation model that allows both synthesizing mid-air interaction movements and predicting the associated embodied user experience, with a focus on subjective fatigue. We test the approach in a mid-air pointing task and four experimental conditions, replicating the experiment design of [38] who tested the same conditions on real humans and analyzed fatigue using motion capture data. We demonstrate that our agent learns the pointing movements needed for the tasks, doing away with the need to capture motions from real humans, and our simulation-based fatigue data provides a good fit with the human data of [38]. Compared to the optimization of mid-air pointing movements of Montano Murillo et al. [51], we do not rely on predetermined effort estimates for different spatial locations. Instead, all our data simply emerges from the biomechanical simulation model and rewarding the agent for both accomplishing the task and avoiding fatigue/discomfort.

As an additional contribution, we compare two different fatigue/effort models incorporated as reward function components: 1) instantaneous joint torques common in computer animation, robotics [58], and standard deep RL movement control benchmark tasks (MuJoCo) [66], and 2) the recent Three Compartment Controller (3CC- r) model from biomechanical literature [47]. We demonstrate that the 3CC- r model yields movements that are both more efficient and relaxed, whereas with instantaneous joint torques, the RL agent can easily generate movements that are quickly tiring or only reach the targets slowly and inaccurately. As the 3CC- r model causes no significant increase in computational complexity, we advocate deep RL researchers also incorporating it to their benchmark tasks to increase both the realism of biomechanical effort modeling and relaxedness of emerging movements.¹

BACKGROUND AND RELATED WORK

Simulating User Behavior

The literature on user modeling features different kinds of models. The most simple ones like Fitts’ law [21] allow predicting a quantity like pointing target acquisition time as a function of design variables like target distance using simple mathematical expressions. However, in many cases such mathematical models are not available, and one must instead resort to simulations of how users perceive, things, and act

while completing tasks. This was first proposed by Card et al. [10, 54] as early as in 1983. Their GOMS model was later extended by ACT-R and other more sophisticated cognitive architectures [65, 54].

A limitation of early user models like GOMS is their complexity: Successful application of the model in a design task requires the designer to provide a detailed breakdown of the user’s goals and expected behavior. Early cognitive architecture development was also plagued by various cognitive processes modeled in isolation, with difficulties of integrating them to general solutions for, e.g., autonomous skill acquisition [65]. However, this was prior to the recent deep neural network revolution; deep Reinforcement Learning (RL) agents have now been demonstrated to learn a wide variety of skills ranging from video game play [49] to controlling the movement of biomechanically simulated human bodies [42]. Although RL methods can be complex, they are simple to apply², which makes them lucrative for user simulation purposes.

Reinforcement Learning is an approach to discovering the optimal actions for a Markov Decision Process (MDP) [63]. It is assumed that at time t , the agent is in state s_t , takes action a_t , and observes a reward r_t and a next state s_{t+1} . The agent is optimizing utility, i.e., expected cumulative future rewards, in line with the computational rationality view of human behavior [26]. Thus, for user modeling, one only needs to define the states, actions, and rewards. At least in some cases, the reward function and other parameters can also be inferred from human data [11, 41, 3].

Yang et al. [69] provide a comprehensive study of what Machine Learning (ML) can offer to Human Computer Interaction (HCI). Traditionally, RL user simulation in HCI has been limited into simple MDP:s with discrete, enumerable states and actions, e.g., dialogue systems, menus and simple keyboards [44, 12, 43]. The discrete states and actions make the MDP:s solvable with classic RL methods like Q-learning [63]. However, recent deep RL methods like Proximal Policy Optimization [62] and Soft Actor Critic [29] also work with the high-dimensional states and actions required for intelligent control of human biomechanical simulation [42]. In this paper, we demonstrate that this makes deep RL a viable approach for modeling embodied interaction and predicting user experience outcomes such as fatigue.

A recent result that further motivates our work is that biomechanically realistic movement can be synthesized efficiently through simplified skeletal simulation without muscle and tendon detail, as long as the actuation effort minimized by an RL agent is computed with a higher degree of biomechanical realism through a Machine Learning model that predicts muscle activations from joint actuation torques [39]. Extending this approach, we actuate with joint torques and increase biomechanical realism through a fatigue model incorporated as an extra reward function component.

¹Source code is available at:
<https://github.com/noshaba/ArmFatigue>

²At least in principle; in practice, RL methods have their quirks and applying them successfully can require quite a bit of experimentation. See, e.g., [33].

It should be noted that modern neural-network based RL also generalizes to Partially Observable Markov Decision Processes (POMDP:s) where the agent cannot access the full environment or simulation state [32]. This is often the case in user modeling that incorporates realistic perception models.

Quantifying Mid-Air Interaction Fatigue

Muscle fatigue is the failure to maintain the required or expected force [16]. Fatigue depends on a multitude of simultaneous physiological and neurological processes, making it difficult to pinpoint a single mechanism responsible for the loss of force [68, 22, 1, 4]. It is also task-related and can vary across muscles and joints [68, 36, 17, 38, 23, 25], which partially explains the challenging nature of representing muscle fatigue analytically [68].

A widely used empirical model to estimate the effect of fatigue on the task endurance time (ET) of static load conditions is the Rohmert’s curve [60, 36]. Hincapié et al. [34] developed Consumed Endurance (CE), a metric to quantify arm fatigue of mid-air interactions, based on the Rohmert’s curve. Although straightforward, the approach lacks the ability to generalize to dynamic load conditions or recovery during rest periods [68, 38] as it is based on the Rohmert’s curve, which is only valid for static load conditions. Furthermore, CE is assumed to be zero at exertion levels below 15%. This limits the use of the model for evaluating mid-air interaction with low exertion levels [38, 23].

Liu et al. [46] have proposed a motor unit (MU)-based fatigue model which uses three muscle activation states: resting (M_R), activated (M_A) and fatigued (M_F). The model is able to predict fatigue at static load conditions but fails at submaximal or dynamic conditions [68, 38]. Xia et al. [68] have proposed a Three-Compartment Controller (3CC) model which improves upon the model of Liu et al. for dynamic load conditions by introducing a feed-back controller term between the active (M_A) and rest (M_R) muscle states. Frey et al. [25] validated the 3CC model in estimating ET under static load conditions and have obtained joint-specific parameters. Later, Jang et al. [38] have optimized the 3CC model for mid-air interaction tasks [38]. They further show that the 3CC model can be used to estimate human perceived fatigue ratings based on the Borg CR10 scale [8], which is a 10-point categorical rating system to quantify perceived fatigue ratings of individuals. It uses verbal anchors and numbers to map the magnitude of exertion to a scalar invariance scale [8, 38] (Table 1). The scale has been used in a variety of contexts, such as sports [18], ergonomics [7], or medicine [53] for subjective evaluation of fatigue based on physiological and psychological changes. While Jang et al. [38] findings are based on kinematic data only, they still require motion data captured from real humans.

More recently, Looft et al. [47] have published an improved 3CC- r model for intermittent tasks by introducing an additional rest recovery multiplier r and validated their results based on perceived fatigue from participants for specific joints.

PRELIMINARIES: FATIGUE MODELING

We investigate two different fatigue models: 1) instantaneous joint torques as a measure of instantaneous effort, and 2) the

| | | Borg CR10 Scale |
|-------|------------------|------------------------------------|
| Score | Definition | Note |
| 0 | Nothing at all | No arm fatigue |
| 0.5 | Very, very weak | Just noticeable |
| 1 | Very weak | As taking a short walk |
| 2 | Weak | Light |
| 3 | Moderate | Somewhat but not hard to go on |
| 4 | Somewhat heavy | |
| 5 | Heavy | Tiring, not terribly hard to go on |
| 6 | | |
| 7 | Very strong | Strenuous |
| 8 | | |
| 9 | | |
| 10 | Extremely strong | Extremely strenuous |

Table 1. Borg CR10 scales with verbal commentary.

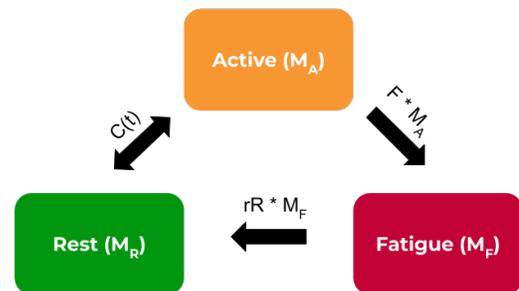


Figure 1. Three compartment controller model.

recently developed Three-Compartment Controller (3CC- r) Model by Looft et al. [47].

Instantaneous Joint Torque Effort

Instantaneous joint torque is a simple measure used in computer animation, robotics, and standard RL benchmark problems [9, 2, 67, 56] to measure and minimize the instantaneous effort of a given task a simulated agent is performing. When defining movement optimization objective functions, the torques are usually squared to make the optimization avoid using excessive strength.

Three-Compartment Controller (3CC- r) Model

An instantaneous effort model gives us a simple measure to determine the difficulty of a given task. However, it is not very biologically accurate as a simple task can become difficult when done long enough. A cumulative effort function, such as the 3CC- r model, thus gives a more accurate representation of perceived fatigue.

Akin to Liu et al. [46], the 3CC- r model [47] assumes motor units (MUs) to be in one of the three possible states:

- *active* - MUs contributing to the task
- *fatigued* - fatigued MUs without activation
- *resting* - inactive MUs not required for the task

Fig. 1 shows the relationship between these states. $M_A(\%)$ is the compartment of active MUs, $M_F(\%)$ the compartment of

fatigued, and $M_R(\%)$ the compartment of resting MUs. Each compartment is expressed as a percentage of the *maximum voluntary contraction (%MVC)* [68, 38].

In addition to that, the compartment theory is combined with control theory to define system behaviour which matches muscle physiology [68], i.e. active MUs' force production should begin to decay (fatigue) over time. This is expressed by the following equations:

$$\frac{\partial M_R}{\partial t} = -C(t) + rR \cdot M_F \quad (1)$$

$$\frac{\partial M_A}{\partial t} = C(t) - F \cdot M_A \quad (2)$$

$$\frac{\partial M_F}{\partial t} = F \cdot M_A - rR \cdot M_F \quad (3)$$

Where F and rR are the model parameters defining at which rate the motor units fatigue, and which rate they recover and enter the rest period, respectively. In contrast to the traditional 3CC model [68], the 3CC- r model introduces an additional rest recovery factor r , which enhances the recovery when the required force, i.e. target load (TL), is zero to better represent perceived fatigue estimates from user studies [47]:

$$rR = \begin{cases} r \cdot R & \text{if } M_A \geq TL \\ R & \text{else} \end{cases} \quad (4)$$

The 3CC- r [47] is equivalent to the 3CC [68] model when $r = 1$. Based on a sensitivity analysis r is set to 7.5. F is set to 0.0146, and R to 0.0022 based on Jang et al. [38] 3CC-model optimization for mid-air interactions.

$C(t)$ is the time-varying muscle activation-deactivation drive, which can produce the target load TL in percent by controlling the size of M_A and the availability of M_R . The following equations describe $C(t)$ mathematically:

$$C(t) = \begin{cases} L_D \cdot (TL - M_A) & \text{if } M_A < TL \text{ and } M_R > (TL - M_A) \\ L_D \cdot M_R & \text{if } M_A < TL \text{ and } M_R \leq (TL - M_A) \\ L_R \cdot (TL - M_A) & \text{if } M_A \geq TL \end{cases} \quad (5)$$

L_D is the muscle force development factor, and L_R is the relaxation factor. Based on the analysis by [68], these are set to 10.

SYSTEM

We implemented our system using the Unity game engine and their ML Agents Toolkit v0.8.2 [40] implementation of the Proximal Policy Optimization (PPO) [62] RL algorithm. The following details our effort model, the simulated pointing task, and the RL problem formulation and training settings.

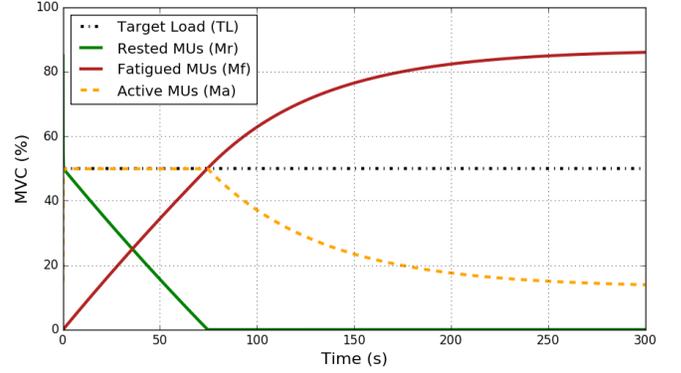


Figure 2. Behaviour of 3CC- r model at target load of 50% MVC (black dotted line). The load cannot be held after around 70 s (yellow dashed line).

Fatigue Model

Instantaneous Fatigue Model

In this paper, we compare instantaneous torques to the more advanced 3CC- r modeling. More specifically, we use instantaneous torques normalized with respect to maximum torque T_{max} :

$$Effort_I(\vec{T}) = \left(\frac{\|\vec{T}\|}{T_{max}} \right) \quad (6)$$

Cumulative Fatigue Model

Since the 3CC- r model is a relative unit-less system [68], TL can be described in a variety of options, e.g. the percentage of maximum voluntary torques (MVT) or forces (MVF).

Previous studies [38, 25, 47] set TL as the ratio of the magnitude of torque \vec{T} and the maximum voluntary torque T_{max} at a joint: $\frac{\|\vec{T}\|}{T_{max}} \cdot 100\%$. While this is a valid approach to measure the load at a given joint, we found it to be more accurate to model two 3CC- r models per degree-of-freedom (DOF) (one for the "positive", and one for the "negative" direction). Each DOF roughly corresponds to a muscle group. We avoid modeling at the level of individual muscles to reduce simulation and RL training time.

The target load can then be expressed as a vector of torque ratios for each direction:

$$\vec{TL}(\vec{T}) = \left[\frac{T_1^+}{T_{max}}, \frac{T_1^-}{-T_{max}}, \frac{T_2^+}{T_{max}}, \frac{T_2^-}{-T_{max}}, \frac{T_3^+}{T_{max}}, \frac{T_3^-}{-T_{max}} \right]^T \quad (7)$$

Where $\frac{T_i^+}{T_{max}}$ is the ratio of the torque at axis i in the "positive" direction, and $\frac{T_i^-}{-T_{max}}$ in the "negative" direction, respectively.

When $\frac{T_i^+}{T_{max}} \geq 0$, then $\frac{T_i^-}{-T_{max}} = 0$, and vice versa. Each value in \vec{TL} is used as input for a separate 3CC- r model.

For simulated agents we describe the cumulative effort $Effort_C$ as the difference between the actual target load

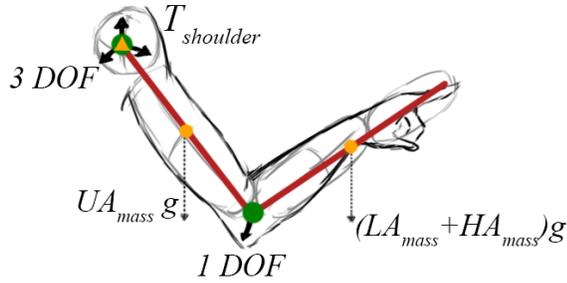


Figure 3. Forces acting on our biomechanical upper limb model. The limbs are modeled as rigid bodies (red) connected with joints (green). The degrees of freedom (DOF) of each joint are denoted by the arrows at the respective joint.

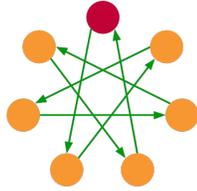


Figure 4. ISO 9241-9 reciprocal pointing task with 7 targets. Agents point to highlighted (red) target. Targets highlight in the pattern indicated by the arrows.

$\vec{T}L(\vec{T})$ at the joint and the desired muscle activation $\vec{M}_A = [M_{A_1}^+, M_{A_1}^-, M_{A_2}^+, M_{A_2}^-, M_{A_3}^+, M_{A_3}^-]^\top$ given by the 3CC- r model:

$$Effort_C(\vec{T}) = \left\| \frac{\vec{M}_A}{100} - \vec{T}L(\vec{T}) \right\| \quad (8)$$

The advantage of this over using M_F directly is that the cumulative fatigue given by the 3CC model stagnates after some time (Fig. 2 red line). Furthermore, it is not clear from M_F alone when a target load could not be held. Using the difference between the actual active motor units and the desired load tells us when the load is okay to be held, and when it becomes a burden. This is shown in Fig. 2, where the active motor units (yellow) start to decline after 70 s because the target load (black) was not sustainable.

Simulated Upper Limb Model

Similar to [38], we assume that arm fatigue is mostly attributed to shoulder-joint fatigue, due to the shoulder fatiguing faster than the elbow or wrist during arm movement [23]. To make our results comparable to theirs, we also only use the shoulder joint torques for our effort model. We simulate our arm using a 4 DOF serial chain - 3 for the shoulder and 1 for the elbow (Fig. 3), where the limbs are modeled as rigid bodies connected by joints [57]. To estimate the shoulder joint torques we use the method in [34].

Mid-Air Pointing Task

We model the mid-air pointing task after the ISO 9241-9 standard [37, 38, 64] based on Fitts' law [21, 48]. The standard is extensively used for evaluating 2D pointing devices, such as mice, pens, and touch screens [64]. The task has participants

point at a circle of targets, with a given width and distance to each other, in a given order. Akin to [38], we use seven targets with a width of 10 cm and a distance of 30 cm to each other, corresponding to an index of difficulty ID [21, 48] of $ID = \log_2 \left(\frac{30}{10} + 1 \right) = 2$. Fig. 4 shows the target sequence.

Previous studies [38, 34, 35] have shown that the height and distance from the arm's resting position affect the perceived fatigue ratings of participants, making them fatigue more rapidly the higher and further away the arm is. Additionally, rest periods are also a decisive factor of how fatigued we are [38]. We investigate these two factors in our experiments and compare them to human perceived fatigue ratings from a prior study [38].

Similar to Jang et al. [38], we switch between pointing and resting periods. Like them, we use the following four different rest periods: [5s, 10s, 15s, 20s].

RL Problem Formulation

To be able to apply RL methods, we need to define the MDP, i.e. states, actions, and rewards. Additionally, the PPO method we use requires the definition of a policy network for sampling an action given the current state. The MDP transition model $s_{t+1} = f(s_t, a_t)$ is implemented by Unity's ML-Agents Toolkit: After sampling an action, we use it to actuate the simulation and query the next state.

State and Action Space

The state vector \vec{s} of the agent, comprises a concatenation of the following features:

- limb positions with respect to the shoulder [9 values]
- linear velocities of upper and lower arm limbs [6 values]
- angular velocities of upper and lower arm limbs [6 values]
- direction vector from finger tip to target (not normalized) [3 values]
- target switch time [1 value]
- rest period Boolean [1 value]

We define our RL actions as actuation torques applied through Unity's AddTorque() method at the center of mass of the upper and lower arm limbs. The action vector \vec{a} from the policy specifies how much of the maximum voluntary torques (MVT) to apply to the limbs. It is comprised of 4 values, denoting the three actuation torque values for the upper, and one for the lower arm. The torques that are applied cannot overshoot shoulder MVT and elbow MVT values, respectively. The shoulder MVT is furthermore used as T_{max} in the effort calculation in Eq. 6 and 7.

Network

A policy π is represented by a neural network which maps a given state s to a distribution over action $\pi(a|s)$. The action distribution is modeled as a Gaussian, where the state dependant mean $\mu(s)$ and the diagonal covariance matrix Σ are specified by the network output:

$$\pi(a|s) = \mathcal{N}(\mu(s), \Sigma) \quad (9)$$

The inputs s to the network is processed by two fully-connected layers with 128 hidden units, each, using the Swish [59] activation function. During training the network adapts the mean and covariance matrix such that the actions become less noisy as the agent gradually starts to exploit instead of explore.

Reward

The reward $\rho(t)$ at each step t consists of two terms that encourage the character to pointing towards the target when there is one, while using minimal effort:

$$\rho(t) = (\omega_P \rho(t)_P + \omega_F \rho(t)_F) \cdot 0.01 \quad (10)$$

$\rho(t)_P$ and $\rho(t)_F$ are defined as the pointing and fatigue objectives, respectively, with ω_P and ω_F being their respective weights. The pointing objective encourages the agent to point towards the current target, while the fatigue objective encourages it to make use of actions which require less effort than others. We found that setting $\omega_P = 100$ and $\omega_F = 0.01$ resulted in the desired behavior for various settings.

The pointing reward $\rho(t)_P$ depends on the distance between the target and the finger tip, and is defined with:

$$\rho(t)_P = \begin{cases} 1 & \text{target has been hit} \\ \exp\left(-\frac{\|p_{target}(t) - p_{finger}(t)\|^2}{\tau_P^2}\right) & \text{else} \end{cases} \quad (11)$$

where $p_{target}(t)$ is the target's position, and $p_{finger}(t)$ the position of the finger tip at step t , respectively. τ_P the tolerance distance in meters for when the reward becomes $\rho(t)_P \approx 0.3679$, when the finger does not hit the target. We set $\tau_P = 0.15$, i.e. it starts heavily penalizing deviations of more than 15cm from the target. A minimal reward of approx. 0.001 is obtained at around 40 cm from the target with this objective function.

The fatigue reward $\rho(t)_F$ is defined using the respective *Effort* function defined in Eq. 6 or 8:

$$\rho(t)_F = \exp\left(-\frac{Effort(\vec{T}(t))^2}{\tau_F^2}\right) \quad (12)$$

τ_F is the tolerance in percentage of how much the torque ratio is allowed to deviate from the desired torque ratio, while obtaining a reward of at least approx. 0.3679. In the case of the instantaneous effort function this means how much percent is the shoulder muscle allowed to deviate from zero torque. However, in the case of the cumulative effort function based on the 3CC- r model, this means how much is it allowed to deviate from the allowed motor unit activation \bar{M}_A given by the 3CC- r model. The best tolerance value τ_F for each effort model is determined in the results section.

Training Settings for Pointing Task

| Setting | Distribution | Lower Bound | Upper Bound |
|----------------------|--------------|-------------|-------------|
| Target Height | Uniform | -40 cm | +20 cm |
| Target Distance | Uniform | +10 cm | +70 cm |
| Switch Time | Uniform | 1 s | 2 s |
| Pointing Period | Uniform | 30 s | 90 s |
| Rest Period Index | Uniform | 1 | 4 |
| Initial Target Index | Uniform | 1 | 7 |

Table 2. Settings of the pointing and resting task during training. Target height and distance are in relation to the shoulder position.

Note that in some movement optimization cases, squared cost terms are used without the exponentiation [52, 2]. Our use of exponentiation follows Peng et al. [56]; it converts minimized costs to maximized rewards and also limits the reward to a predefined range, which makes it easier to train PPO's value function predictor network.

Training

The neural network training is done through RL, such that the the agent maximizes the cumulative episode rewards. No data was used for this; RL proceeds through exploring random actions and learning to repeat those that yield high rewards. The policy training is done using the Proximal Policy Optimization (PPO) algorithm [62]. Standard hyperparameters defined in [40] were used, with the following adjustments: batch size = 2024, buffer size = 20240, $\gamma = 0.995$, max steps = $1.0e^6$, normalization = True, number of epochs = 3, time horizon = 1000, summary of frequency = 3000.

Initial State Distribution

PPO training proceeds in episodes, where at the start of each episode the agents and the environment are reset to an initial state s_0 . Each episode is simulated to a fixed time horizon with actions sampled from the policy, after which the agents and the environment are reset again. In total, we use $1e^6$ time steps, or 1000 episodes.

Many RL benchmark problems such as the MuJoCo locomotion environments use a fixed initial state s_0 or add only small random perturbations to it [9]. However, as demonstrated by [56], a diverse enough initial state distribution can greatly improve movement learning. To implement this, we sample multiple settings of the pointing task randomly from a uniform distribution. Table 2 shows an overview of these settings. *Target Height* and *Target Distance* are in relation to the shoulder position and the center point of the target circle. *Pointing Period* describes the duration of the pointing period before the user is supposed to rest, while *Rest Period Index* defines which of the four rest periods [5s, 10s, 15s, 20s] to use. The initial target index of the seven targets sequence is chosen randomly.

During training we use five agents to parallelize the simulation over multiple CPU cores. Their attributes are also sampled randomly for each episode. Hence, each RL training run uses a large and diverse population of simulated users. However, instead of sampling from a uniform distribution we sample from a Gaussian distribution to more accurately represent male and female strength properties. Table 3 shows which features

| Setting | Distribution | Agent Settings | |
|------------------|--------------|----------------|---------------|
| | | LB / μ | UB / σ |
| Body Weight (M) | Gaussian | 80 kg | 1.0 |
| Body Weight (F) | Gaussian | 65 kg | 1.0 |
| MVT Shoulder (M) | Gaussian | 54.9 Nm [30] | 1.0 |
| MVT Shoulder (F) | Gaussian | 29.2 Nm [30] | 1.0 |
| MVT Elbow | Uniform | +0 Nm | +10 Nm |

Table 3. Weight and Maximum Voluntary Torque (MVT) settings for upper body limbs. The elbow MVT is in relation to the agent’s shoulder MVT.

| Limb | Length | Arm Length & Weight | |
|-----------|----------|---------------------|--------|
| | | Male | Female |
| Upper Arm | 29.06 cm | 2.71% | 2.55% |
| Lower Arm | 29.44 cm | 1.62% | 1.38% |
| Hand | 21 cm | 0.61% | 0.56% |

Table 4. Length of arm limbs in cm and their corresponding weights in percentage of the total body weight based on [14].

we sample from what kind of distribution. For the MVT of shoulder, we use average values found in biomechanical literature [30]. Based on [24, 30] we estimate the average elbow MVT between 1 Nm to 10 Nm higher to the person’s shoulder MVT. Hence, the elbow MVT is sampled from a uniform distribution, where the lower bound (LB) is the agent’s shoulder MVT and the upper bound (UB) is an additional +10 Nm. Once the body weight is sampled, the actual arm weight is set according to average body weight percentages for the upper limbs (Table 4) [14].

For the 3CC- r model we furthermore set a random initial fatigue by sampling uniformly from an initial shoulder load (between $-T_{max}$ and T_{max} for each of the three DOF), which is then applied for a randomly set time (between 0 s and 180 s) onto the joint. With this we gather experiences for more long-term fatigue effects.

RESULTS

We evaluate our system in two experiments, with details provided below. The results are visualized in Figures 7 and 8.

First, we compare the movement synthesis quality of the two effort models in terms of both accuracy of reaching the pointing targets and relaxedness of movement (Subsection “Relaxedness vs. Accuracy”). As we define relaxedness as the arm using minimal effort when possible, there is an obvious accuracy-relaxedness trade-off for the agent, which can be adjusted with the reward function parameter τ_F from Eq. 12. To quantify relaxedness we propose a relaxedness metric η based on hand-crafted kinematic motion features that characterize typical movement behaviour (Eq. 13 - 18).

Second, using the model parameters that yield the best accuracy-relaxedness trade-off, we compare our model’s fatigue estimates based on synthesized movements with the average Borg CR10 ratings reported in [38], as well as their 3CC fatigue estimates which are based on human movement data

obtained from a Kinect (Subsection “Comparison to Ground Truth Human Data”). In contrast to [38], our fatigue estimates are solely based on the synthesized movements of the simulated users trained using deep RL, without the use of human movement data.

Relaxedness vs. Accuracy

To determine the best fatigue tolerance value τ_F for each effort model, we do a grid search on the trained networks and plot the results in terms of accuracy and relaxedness of the motions.

We train models with τ_F values between $0.0 \leq \tau_F \leq 0.5$ in 0.02 steps, totalling 52 trained models (26 for each effort function) for 25 different random seeds. Each model’s synthesized movements are evaluated on efficiency and relaxedness. In this experiment we use 20 agents, which are akin to 20 people, with different settings for each trained model. The parameters of the agents are seeded to have the same settings for each model. To make the models comparable to ground truth human data reported in [38], we set the targets’ switch time to 1.3 s, and the pointing period to 60 s. Jang et al. [38] determined that if subjects performed four mid-air interaction periods in a series they had a higher chance of learning and pre-fatigue effects [20]. Hence, we designed our experiment similar to their experiment with the following rest periods in between in this order: [20s, 5s, 15s, 10s]. This setup is akin to group 1 in [38] (Fig. 6). In total the pointing task lasts roughly 5 min. We placed the targets at four different interaction zones with five agents sharing the same zone: one at shoulder height and having the arm bent, one at waist height and having the arm bent, one at shoulder height and the arm straight, and finally one at waist height and having the arm straight. The four interaction zones for this experiment are shown in Fig. 5.

For the accuracy of a model we consider two separate measures: the median of the average distance over time from a target, and the median average time it takes to reach a target. If the agent could not reach the target, the time is set to the switch time. The median is computed over the average value for each agent over the 25 randomly seeded training sessions.

To determine the relaxedness η of arm movements, we use the following equation:

$$\eta = 4 - \left(\widetilde{d_E^{point}} + \widetilde{\phi_E^{rest}} + \widetilde{\phi_A^{rest}} + \widetilde{v_A^{rest}} \right) \quad (13)$$

with $d_E^{point}, \phi_E^{rest}, \phi_A^{rest}, v_A^{rest} \in [0, 1]$. The higher the value is, more natural and relaxed the arm motion is supposed to be. “ $\widetilde{}$ ” denotes the median value over the 20 agents and the 25 training sessions.

$d_E^{pointing}$ is the average distance over T time steps of the elbow to the plane that is spanned between the shoulder and finger tip and the direction of gravity, when the agent is pointing:

$$d_E^{point} = \frac{1}{T} \sum_t \frac{|\langle \vec{n}, \overrightarrow{ShEl} \rangle|}{\|\overrightarrow{ShEl}\|} \quad (14)$$



Figure 5. Four interaction zones used for determining the best model. 1) Target is shoulder height and arm is bent. 2) Target is waist height and arm is bent. 3) Target is shoulder height and arm is straight. 4) Target is waist height and arm is bent.

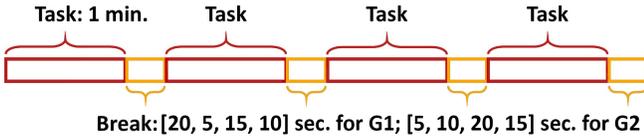


Figure 6. Experiment protocol of relaxedness vs. accuracy measure (G1), as well as during comparison against ground truth human data (G1, G2).

with \overrightarrow{ShEl} being the vector from the shoulder to the elbow position. The distance is divided by the length of this vector to obtain values between 0 and 1. \vec{n} is the plane normal of the shoulder-finger plane:

$$\vec{n} = \frac{\overrightarrow{ShFi}}{\|\overrightarrow{ShFi}\|} \times \frac{\vec{g}}{\|\vec{g}\|} \quad (15)$$

\overrightarrow{ShFi} is the vector from the shoulder to the finger tip. The higher d_E^{point} is, the further away the elbow is from this plane. The idea is that during pointing movements the agent should prefer to keep its elbow down since this position is perceived as less fatiguing than having the elbow point side-ways.

To measure the relaxedness of the arm during rest periods, we add ϕ_E^{rest} , ϕ_A^{rest} , and v_A^{rest} into the relaxedness equation.

ϕ_E^{rest} is the average elbow angle between the lower and upper arm. The idea is that during rest periods the agent is not supposed to flex its arm much. It is calculated the following way:

$$\phi_E^{rest} = \frac{1}{T} \sum_t \left(\left\langle \frac{\overrightarrow{ElSh}}{\|\overrightarrow{ElSh}\|}, \frac{\overrightarrow{ElHa}}{\|\overrightarrow{ElHa}\|} \right\rangle + 1 \right) \cdot 0.5 \quad (16)$$

\overrightarrow{ElSh} is the vector from elbow to shoulder and \overrightarrow{ElHa} from elbow to hand, respectively. When the arm is straight the dot product becomes -1 , when the lower arm is perpendicular to the upper arm the dot product is 0, and when the arm is flexing it is close to 1. To keep ϕ_E^{rest} between 0 and 1 (0 being straight and 1 flexing), we add 1 to the dot product and scale it with 0.5.

While this value lets us know when the arm is flexing during rest periods, with it alone the relaxedness measure would classify holding an arm in front of oneself as more relaxing than flexing it. Thus, we also calculate the average angle ϕ_A^{rest} between arm and the direction of gravity and incorporate it in our relaxedness equation:

$$\phi_A^{rest} = \frac{1}{T} \sum_t \left(\left\langle \frac{\overrightarrow{COMSh}}{\|\overrightarrow{COMSh}\|}, \frac{\vec{g}}{\|\vec{g}\|} \right\rangle + 1 \right) \cdot 0.5 \quad (17)$$

\overrightarrow{COMSh} is the vector from the center of mass of the arm to the shoulder.

While this gives us a good measure for policies where the arm learns to be static during rest periods, it still sometimes classifies moving arms to be more relaxed than flexing but resting arms during rest periods. This is because the average of all frames is taken and if the arm jerks around a lot, this average of that could still be an arm hanging down. To overcome this issue, we also add the average velocity v_A^{rest} of the arm during rest periods:

$$v_A^{rest} = \frac{1}{T} \sum_t \frac{v_{UA} + v_{LA}}{v_{Amax}^{rest}} \quad (18)$$

v_{UA} and v_{LA} are the upper arm and lower arm velocities obtained by the Unity engine. v_{Amax}^{rest} is the maximum velocity value of all parameter settings and agents.

Fig. 7 shows the results of the 3CC- r and the instantaneous effort model. Each point denotes a different τ_F value for a model. The models are ordered based on their τ_F value (lowest first). Models above a median time of 1.2 s learned to hang the arm down due to obtaining more reward from using as little effort as possible compared to the reward obtained from pointing. Models with relaxedness values below 3.4 resulted in unnatural movements and jerky arm behaviour during rest periods. Furthermore, the variance of the results in terms of accuracy increases as is suggested by Fig. 7. The sweet spot in which motions are relaxed, i.e. arm hangs down during rest periods and elbows are kept down during pointing periods, but

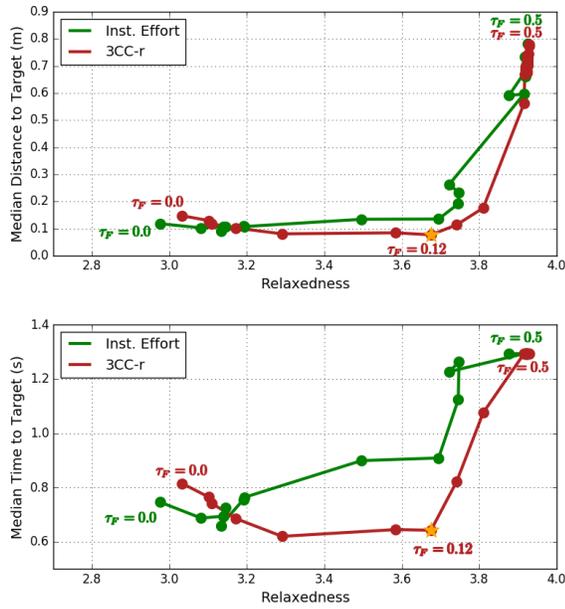


Figure 7. The evolution of the trade-off between pointing task performance and movement relaxedness when sweeping the τ_F parameter in the range $[0, 0.5]$ in steps of 0.02, plotted using both instantaneous torque effort (green) and the 3CC- r model effort (red). For each tested τ_F , the 20 agents are retrained and re-evaluated across 25 random seeds. The yellow stars indicate the best combinations of both relaxedness and pointing task performance. Overall, the 3CC- r yields better combinations of relaxedness and pointing task performance, across a range of τ_F .

still accurate lied usually within relaxedness values between 3.5 and 3.9 (Fig. 7). The plots in Fig. 7 show how the 3CC- r models consistently outperform the instantaneous fatigue model in terms of speed, accuracy and relaxedness within this region, i.e., points near the bottom-right corners of the plots. Based on the results in Fig. 7, we found that $\tau_F = 0.12$ for the 3CC- r model yields the best results in terms of efficiency and relaxedness. In the next section we will use this model to compare against ground truth perceived fatigue ratings from humans.

Comparison to Ground Truth Human Data

We compare our best model with the ground truth human data obtained from [38]: Jang et al. [38] use the 3CC model to predict fatigue ratings based on torque measures estimated from motion capture data from a Kinect [71] sensor. They compare their results with the subjects' perceived exertion rating using the Borg CR10 [8] scale. In this section we compare our 3CC- r fatigue estimates using the torque measures from synthesized movements to their 3CC fatigue estimates based on human movement data. We further compare our 3CC- r fatigue estimates to the subjects' average Borg CR10 ratings reported in [38].

To replicate the four conditions in [38], we use the first two interaction zones shown in Fig. 5, and two groups with different rest periods in between the four 60 s pointing periods: [20s, 5s, 15s, 10s] for group 1 and [5s, 10s, 20s, 15s] for group 2 (Fig. 6). In the following we refer to group 1 and 2 as G1

and G2, and the high and low interaction zones as H and L. Based on the findings of Jang et al. [38], the tasks based on the higher interaction zones should be more fatiguing than the ones based on the lower interaction zones. Furthermore, G1 should feel less fatigued compared to G2, due to a large rest period in the initial period of the task.

Jang et al. [38] use 24 participants in their study of which two were female. Since there was no ground truth data published of each participant's weight and their corresponding maximum torque estimate, we gauge their subjects in a virtual environment by using average torque and arm weight estimates found in literature. See Table 3 and 4 for details. We also use 22 male, and 2 female (virtual) subjects.

Similar to [38] we assume a linear relationship between the fatigued motor units M_F obtained from the 3CC- r model and the Borg CR10 scale with $\varphi(x) = 0.3 \cdot x$ denoting the linear mapping. To compute M_F for the Borg CR10 estimate, we use the ratio of magnitude of the torque $\frac{\|T\|}{T_{max}} \cdot 100\%$ as a target load for the model.

An overview of our results is shown in Fig. 8. Our 3CC- r estimates (black) on virtual data mostly follows the trend of the 3CC estimates (red) from [38] based on human data, as well as the ground truth average Borg CR 10 data (yellow). The average root mean squared error (RMSE) between the 3CC estimates from [38] and their average Borg CR10 ground truth data is 0.58, while ours to ground truth is 0.66. We find the largest deviation of our model in experiment G2-H. We believe this may be attributed to physiological and psychological factors that play into the role of an individual's perceived fatigue rating [38]. As the minimum and maximum values for G2H in Fig. 8 suggest, the variance of inter-individual Borg ratings is high for this case. However, despite using no ground truth human data for our calculations we achieve a similar accuracy to [38] on their data by just fitting a single scaling parameter φ to minimize the RMSE between our simulated fatigue and the average human Borg CR10 data of [38].

Parameter Values

A strength of our approach is that we only fit a single scalar parameter φ based on data. In summary, the following parameters were adjusted for our model: First, the ω parameters of the reward function (Eq. 10) were adjusted empirically until the simulations started to result in effective pointing behaviors. The τ_P tolerance for goal attainment (Eq. 11) was set to a reasonable value that starts heavily penalizing deviations of more than 15cm from the target. The τ_F was chosen to provide a good combination of naturalness and efficiency of movement (Fig. 7). The neural network weights were learned by maximizing the cumulative RL reward. Finally, the scale parameter $\varphi = 0.3$ was set to minimize the RMSE between our simulated fatigue and the average human Borg CR10 data of [38].

DISCUSSION

We make two main contributions to the HCI and ML community with our work: 1) a cumulative fatigue model for Reinforcement Learning of movement tasks 2) an *in silico* method for virtual user testing.

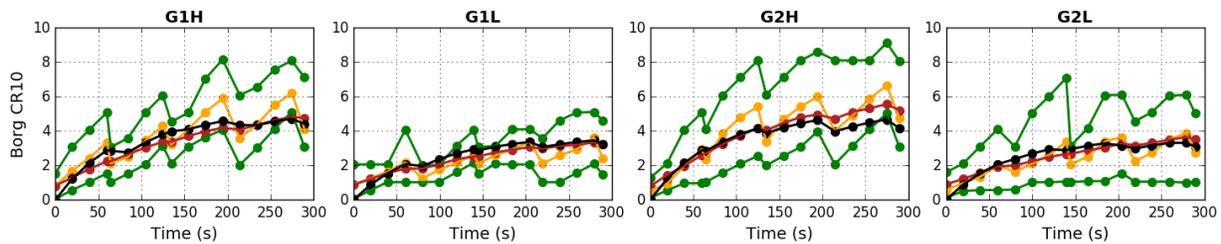


Figure 8. Results of predicting the Borg CR10 rating. Green: Upper/lower bound of ground truth. Yellow: Average of ground truth. Red: Average 3CC estimate of ground truth computed using motion capture data [38]. Black: Our simulation-based average 3CC- r estimate. Our simulation model yields similar modeling accuracy as [38], but does not require motion capture data.

Cumulative Fatigue for Reinforcement Learning

With our results we have shown that RL-agents trained on a cumulative fatigue model based on biomechanical literature, instead of instantaneous joint torques, learned more efficient and relaxed policies. This can be utilized for new optimization procedures in computer animation and robotics. To our knowledge, our model is the first one to use cumulative effort in such a way.

Reliable In Silico Subjective Fatigue Estimate

Our results confirmed that ground truth human movement data is not necessarily needed to obtain reliable fatigue estimates for our pointing task. To our knowledge this is the first method to achieve this. We believe that our model can be utilized for a multitude of HCI applications, where human data is not readily available or expensive to record, and open new pathways to virtual user testing. The advantage of such *in silico* methods is that many physical properties can be reliably modeled with standard games and physics engines [40, 66, 13, 15] making the prediction - in theory - more accurate than with non-invasive *in vivo* methods. With this new environments, e.g. effects of fatigue on the moon or under high pressure, could be easily explored.

Limitations and Future Work

While our model shows overall good results and performance, there are a number of limitations. Similar to [38], our proposed method is based on the assumption that perceived fatigue can be directly deduced from biomechanical information. In reality however, an individual's perceived fatigue can be attributed to a multitude of different factors [68, 38], e.g. through physiological and psychological changes. Previous studies [24, 38, 34, 8] have shown that individuals may experience fatigue and rest differently than others. However, this could be mitigated with extending the agents with models of intrinsic motivation and emotion [50] in future work.

With our work we demonstrate the capability to predict fatigue. However, predicting other variables may need more complex models. Nevertheless, the basic deep RL framework utilized in this paper should remain useful. For instance, modeling the speed-accuracy trade-off of movements through RL-controlled biomechanical simulation is an important topic for future work; presently, there is no research replicating classic Fitts' law experiments in silico using biomechanical forward models and neural network controllers trained without reference movement data. We believe this is more due to the

lack of attention to the topic rather than limitations of deep RL, and could possibly be tackled by incorporating aspects such as signal-dependent noise [31] and delayed feedback [5] to our model.

As an additional limitation, we only model shoulder fatigue. However, it is a central concern in mid-air interaction, and only modeling shoulder fatigue similar to [38] was essential to make our results comparable with [38] (Fig. 8). Nevertheless, for future work we find it would be interesting to model additional hand joints or even the full body for more accurate fatigue estimates.

Another limitation of our work is that for the trained agent to generalize, it must experience the full variance of environments and tasks during training. In our case, we varied the pointing targets, but more variation may be needed for other applications. Furthermore, the learning process itself is also time-consuming and laborious, and needs to be performed independently for each policy. While it takes around 2 hours on an i7 processor to learn pointing, training can take days for other, more complex tasks [56].

CONCLUSION

We presented a framework for evaluating subjective fatigue only using virtual embodied AI agents. The agents have been trained on a pointing task using Reinforcement Learning. For the training we compared two different effort models. First, using instantaneous joint torques; second, using a biomechanical cumulative fatigue model. We showed that the model trained with cumulative fatigue was able to learn more relaxed and efficient movements. We believe this is the first work to use cumulative fatigue in such a way. Finally, we used our trained model to estimate fatigue ratings under various conditions and compared the results with ground truth human data obtained from a previous study [38]. Overall, our model showed comparable results to ground truth Borg CR10 ratings and 3CC-estimates based on motion capture data, without using any human movement data. To the best of our knowledge, this is the first work to achieve this.

ACKNOWLEDGMENTS

This work was funded by the ITEA3 project MOSIM (grant no. 01IS18060C), the Academy of Finland (grant no. 299358), and an IMPRS-CS doctoral fellowship. The calculations presented in the paper were performed with the computational resources provided by the Aalto Science-IT project.

REFERENCES

- [1] Chris R Abbiss and Paul B Laursen. 2005. Models to explain fatigue during prolonged endurance cycling. *Sports medicine* 35, 10 (2005), 865–898.
- [2] Mazen Al Borno, Martin De Lasa, and Aaron Hertzmann. 2012. Trajectory optimization for full-body movements with complex contacts. *IEEE transactions on visualization and computer graphics* 19, 8 (2012), 1405–1414.
- [3] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K Dey. 2016. Modeling and understanding human routine behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 248–260.
- [4] Benjamin K Barry and Roger M Enoka. 2007. The neurobiology of muscle fatigue: 15 years later. *Integrative and comparative biology* 47, 4 (2007), 465–473.
- [5] Dan Beamish, I Scott MacKenzie, and Jianhong Wu. 2006. Speed-accuracy trade-off in planned arm movements with delayed feedback. *Neural networks* 19, 5 (2006), 582–599.
- [6] Pradipta Biswas, Peter Robinson, and Patrick Langdon. 2012. Designing inclusive interfaces through user modeling and simulation. *International Journal of Human-Computer Interaction* 28, 1 (2012), 1–33.
- [7] Gunnar Borg. 1990. Psychophysical scaling with applications in physical work and the perception of exertion. *Scand J Work Environ Health* 16, Suppl 1 (1990), 55–58.
- [8] Gunnar A Borg. 1982. Psychophysical bases of perceived exertion. *Med sci sports exerc* 14, 5 (1982), 377–381.
- [9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
- [10] Stuart K Card, Allen Newell, and Thomas P Moran. 1983. *The Psychology of Human-Computer Interaction*. (1983).
- [11] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. 2011. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Interspeech 2011*. 1025–1028.
- [12] Xiuli Chen, Gilles Bailly, Duncan P Brumby, Antti Oulasvirta, and Andrew Howes. 2015. The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 4217–4226.
- [13] Erwin Coumans. 2015. Bullet Physics Simulation. In *ACM SIGGRAPH 2015 Courses (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 7. DOI: <http://dx.doi.org/10.1145/2776880.2792704>
- [14] Paolo De Leva. 1996. Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of biomechanics* 29, 9 (1996), 1223–1230.
- [15] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* 54, 11 (2007), 1940–1950.
- [16] Richard HT Edwards. 1981. Human muscle function and fatigue. In *Ciba Found Symp*, Vol. 82. Wiley Online Library, 1–18.
- [17] Roger M Enoka and Jacques Duchateau. 2008. Muscle fatigue: what, why and how it influences muscle function. *The Journal of physiology* 586, 1 (2008), 11–23.
- [18] Roger Eston. 2012. Use of ratings of perceived exertion in sports. *International journal of sports physiology and performance* 7, 2 (2012), 175–182.
- [19] Gerhard Fischer. 2001. User modeling in human-computer interaction. *User modeling and user-adapted interaction* 11, 1-2 (2001), 65–86.
- [20] James Peter Fisher, Luke Carlson, James Steele, and Dave Smith. 2014. The effects of pre-exhaustion, exercise order, and rest intervals in a full-body resistance training intervention. *Applied Physiology, Nutrition, and Metabolism* 39, 11 (2014), 1265–1270.
- [21] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.
- [22] Robert H Fitts. 1994. Cellular mechanisms of muscle fatigue. *Physiological reviews* 74, 1 (1994), 49–94.
- [23] Laura A Frey Law and Keith G Avin. 2010. Endurance time is joint-specific: a modelling and meta-analysis investigation. *Ergonomics* 53, 1 (2010), 109–129.
- [24] Laura A Frey-Law, Andrea Laake, Keith G Avin, Jesse Heitsman, Tim Marler, and Karim Abdel-Malek. 2012a. Knee and elbow 3D strength surfaces: peak torque-angle-velocity relationships. *Journal of applied biomechanics* 28, 6 (2012), 726–737.
- [25] Laura A Frey-Law, John M Looft, and Jesse Heitsman. 2012b. A three-compartment muscle fatigue model accurately predicts joint-specific maximum endurance times for sustained isometric tasks. *Journal of biomechanics* 45, 10 (2012), 1803–1808.
- [26] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 6245 (2015), 273–278.
- [27] Christian Guckelsberger, Christoph Salge, Jeremy Gow, and Paul Cairns. 2017. Predicting player experience without the player.: An exploratory study. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. ACM, 305–315.

- [28] Stefan Freyr Gudmundsson, Philipp Eisen, Erik Poromaa, Alex Nodet, Sami Purmonen, Bartłomiej Kozakowski, Richard Meurling, and Lele Cao. 2018. Human-like playtesting with deep learning. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.
- [29] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and others. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).
- [30] Patricia A Hageman, Debra K Mason, Kelly W Rydlund, and Scott A Humpal. 1989. Effects of position and speed on eccentric and concentric isokinetic testing of the shoulder rotators. *Journal of Orthopaedic & Sports Physical Therapy* 11, 2 (1989), 64–69.
- [31] Christopher M Harris and Daniel M Wolpert. 1998. Signal-dependent noise determines motor planning. *Nature* 394, 6695 (1998), 780.
- [32] Matthew Hausknecht and Peter Stone. 2015. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*.
- [33] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [34] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1063–1072.
- [35] Marina Hofmann, R Brger, Ninja Frost, Julia Karremann, Jule Keller-Bacher, Stefanie Kraft, Gerd Bruder, and Frank Steinicke. 2013. Comparing 3d interaction performance in comfortable and uncomfortable regions. In *Proceedings of the GI-Workshop VR/AR*. 3–14.
- [36] Daniel Imbeau, Bruno Farbos, and others. 2006. Percentile values for determining maximum endurance times for static muscular work. *International Journal of Industrial Ergonomics* 36, 2 (2006), 99–108.
- [37] ISO ISO. 2000. 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9: Requirements for non-keyboard input devices (FDIS-Final Draft International Standard), 2000. *International Organization for Standardization* (2000).
- [38] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3328–3339.
- [39] Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. 2019. Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation. *arXiv preprint arXiv:1904.13041* (2019).
- [40] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. 2018. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* (2018).
- [41] Antti Kangasrääsio, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring cognitive models from data using approximate Bayesian computation. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. ACM, 1295–1306.
- [42] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 73.
- [43] Katri Leino, Antti Oulasvirta, Mikko Kurimo, and others. 2019. RL-KLM: automating keystroke-level modeling with reinforcement learning.. In *IUI*. 476–480.
- [44] Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1998. Using Markov decision process for learning dialogue strategies. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, Vol. 1. IEEE, 201–204.
- [45] Richard L Lewis, Andrew Howes, and Satinder Singh. 2014. Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science* 6, 2 (2014), 279–311.
- [46] Jing Z Liu, Robert W Brown, and Guang H Yue. 2002. A dynamical model of muscle activation, fatigue, and recovery. *Biophysical journal* 82, 5 (2002), 2344–2359.
- [47] John M Looft, Nicole Herkert, and Laura Frey-Law. 2018. Modification of a three-compartment muscle fatigue model to predict peak torque decline during intermittent tasks. *Journal of biomechanics* 77 (2018), 16–25.
- [48] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.
- [49] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.
- [50] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. 2018. Emotion in reinforcement learning agents and robots: a survey. *Machine Learning* 107, 2 (2018), 443–480.

- [51] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic Optimization of Immersive Virtual Environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 759–771. DOI: <http://dx.doi.org/10.1145/3126594.3126605>
- [52] Kouros Naderi, Joose Rajamäki, and Perttu Hämäläinen. 2017. Discovering and synthesizing humanoid climbing movements. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 43.
- [53] Bruce J Noble. 1982. Clinical applications of perceived exertion. *Medicine and science in sports and exercise* 14, 5 (1982), 406–411.
- [54] Antti Oulasvirta. 2017. User interface design with combinatorial optimization. *Computer* 50, 1 (2017), 40–47.
- [55] Antti Oulasvirta, Xiaojun Bi, and Andrew Howes. 2018. *Computational interaction*. Oxford University Press.
- [56] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 143.
- [57] Rositsa Raikova. 1992. A general approach for modelling and mathematical investigation of the human upper limb. *Journal of biomechanics* 25, 8 (1992), 857–867.
- [58] Joose Rajamäki and Perttu Hämäläinen. 2017. Augmenting sampling based controllers with machine learning. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 11.
- [59] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).
- [60] Walter Rohmert. 1960. Ermittlung von Erholungspausen für statische Arbeit des Menschen. *European Journal of Applied Physiology and Occupational Physiology* 18, 2 (1960), 123–164.
- [61] Shaghayegh Roohi, Jari Takatalo, Christian Guckelsberger, and Perttu Hämäläinen. 2018. Review of intrinsic motivation in simulation-based game testing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 347.
- [62] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [63] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [64] Robert J Teather and Wolfgang Stuerzlinger. 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 87–94.
- [65] Kristinn Thórisson and Helgi Helgasson. 2012. Cognitive architectures and autonomy: A comparative review. *Journal of Artificial General Intelligence* 3, 2 (2012), 1–30.
- [66] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033.
- [67] Jack M Wang, Samuel R Hamner, Scott L Delp, and Vladlen Koltun. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM transactions on graphics* 31, 4 (2012).
- [68] Ting Xia and Laura A Frey Law. 2008. A theoretical approach for modeling peripheral muscle fatigue and recovery. *Journal of biomechanics* 41, 14 (2008), 3046–3052.
- [69] Qian Yang, Nikola Banovic, and John Zimmerman. 2018. Mapping machine learning advances from hci research to reveal starting places for design innovation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 130.
- [70] Georgios N Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. 2013. Player modeling. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [71] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (2012), 4–10.
- [72] Alexander Zook, Brent Harrison, and Mark O Riedl. 2015. Monte-carlo tree search for simulation-based strategy analysis. In *Proceedings of the 10th Conference on the Foundations of Digital Games*.