www.nature.com/scientificreports

scientific reports

OPEN

Reinforcement learning control of a biomechanical model of the upper extremity

Florian Fischer✉, Miroslav Bachinski, Markus Klar, Arthur Fleig & Jörg Müller

Among the infinite number of possible movements that can be produced, humans are commonly assumed to choose those that optimize criteria such as minimizing movement time, subject to certain movement constraints like signal-dependent and constant motor noise. While so far these assumptions have only been evaluated for simplified point-mass or planar models, we address the question of whether they can predict reaching movements in a full skeletal model of the human upper extremity. We learn a control policy using a motor babbling approach as implemented in reinforcement learning, using aimed movements of the tip of the right index finger towards randomly placed 3D targets of varying size. We use a state-of-the-art biomechanical model, which includes seven actuated degrees of freedom. To deal with the curse of dimensionality, we use a simplified second-order muscle model, acting at each degree of freedom instead of individual muscles. The results confirm that the assumptions of signal-dependent and constant motor noise, together with the objective of movement time minimization, are sufficient for a state-of-the-art skeletal model of the human upper extremity to reproduce complex phenomena of human movement, in particular Fitts' Law and the $\frac{2}{3}$ Power Law. This result supports the notion that control of the complex human biomechanical system can plausibly be determined by a set of simple assumptions and can easily be learned.

In the case of simple end-effector models, both Fitts' Law and the $\frac{2}{3}$ Power Law have been shown to constitute a direct consequence of minimizing movement time, under signal-dependent and constant motor noise[1,2]. Here, we aim to confirm that these simple assumptions are also sufficient for a full skeletal upper extremity model to reproduce these phenomena of human movement. As a biomechanical model of the human upper extremity, we use the skeletal structure of the *Upper Extremity Dynamic Model* by Saul et al.[3], including thorax, right clavicle, scapula, shoulder, arm, and hand. The model has seven actuated degrees of freedom (DOFs): shoulder rotation, elevation and elevation plane, elbow flexion, forearm rotation, and wrist flexion and deviation. While the thorax is fixed in space, the right upper extremity can move freely by actuating these DOFs. To deal with the curse of dimensionality and make the control problem tractable, following van Beers et al.[4], we use a simplified second-order muscle model acting at each DOF instead of individual muscles. These second-order dynamics map an action vector obtained from the learned policy to the resulting activations for each DOF. Following van Beers et al.[4], we assume both signal-dependent and constant motor noise in the control, with noise levels 0.103 and 0.185, respectively. Multiplying these activations with constant moment arm scaling factors, which represent the strength of the muscle groups at the respective DOFs, yields the torques that are applied at each DOF independently. Further details on the biomechanical model are provided in the *Methods* section below.

The Upper Extremity Dynamic Model is significantly more complex than standard point-mass or linked-segment models. In particular, there is no explicit formula for the non-linear and non-deterministic system dynamics. Together with the objective of movement time minimization, these properties make it difficult to use classical optimal control approaches. Instead, in this paper we learn a control policy using deep *reinforcement learning (RL)*. RL algorithms, just like the optimal control methods discussed below, aim to find a policy that maximizes a given reward function. Moreover, they do not require any explicit knowledge about the underlying model. Instead, the optimal value of a certain state is estimated from sampling different actions in the environment and observing the subsequent state and obtained reward[5].

In our approach, a control policy initially generates random movements, which are rewarded with the negative time to reach randomly placed 3D targets of varying size, with the right index finger (see Fig. 1). This reward signal implies movement time minimization for aimed movements. The policy is updated using the *soft-actor-critic* algorithm (SAC)[6]. The actor and critic networks both consist of two fully connected layers with 256 neurons each,

University of Bayreuth, Bayreuth, Germany. ✉email: florian.j.fischer@uni-bayreuth.de

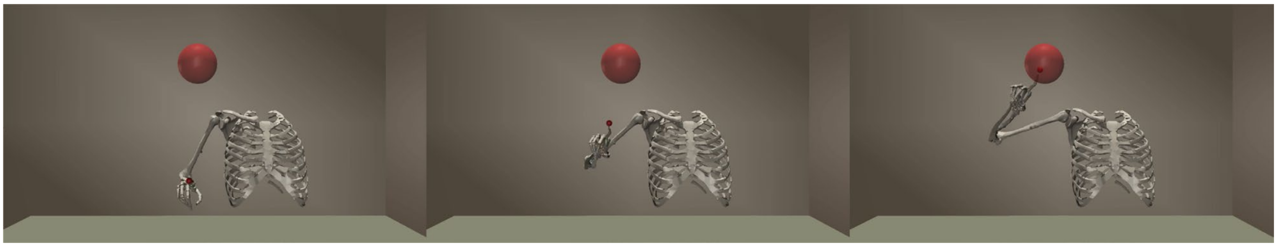

Figure 1. Synthesized reaching movement. A policy implemented as a neural network computes motor control signals of simplified muscles at the joints of a biomechanical upper extremity model from observations of the current state of the upper body. We use Deep Reinforcement Learning to learn a policy that reaches random targets in minimal time, given signal-dependent and constant motor noise.

followed by the output layer, which either returns the means and standard deviations of the action distributions (for the actor network) or the state-action value (for the critic network). Further information about the network architecture and a detailed description of all state components can be found in the *Methods* section below. To make reinforcement learning computationally feasible within a reasonable time period, a fast physics simulation is advantageous. Accordingly, we implemented the biomechanical model in MuJoCo[7].

It is important to note in this context that the assumption of minimizing total movement time does not provide any gradient information to the reinforcement learner. In particular, it is not possible to distinguish beneficial states and actions from inappropriate ones before the target has been reached, which terminates the episode and thus increases the total return. This, together with the fairly small subspace of appropriate actions relative to the number of possible control vectors, makes it very difficult to obtain a reasonable policy without additional aid. For this reason, we created an adaptive curriculum, which dynamically decreases the target diameter from 60 cm to less than 2 cm during training. This has proven to be both effective (targets with diameter around 2 cm are consistently reached by the final policy) and efficient (this minimum width was reached after 1.2M steps, while various predetermined curricula required more than 3M steps).

Related work

The question of how human arm movements are internally planned and controlled has received significant attention in the literature. Important phenomena that emerge from human arm movements include Fitts' Law and the $\frac{2}{3}$ Power Law. In this section we review related work in these areas.

Motor control.

Many models of human motor control assume that some objective function is optimized during the planning of the movement. A variety of objective functions have been proposed, including minimization of either jerk[8,9], peak acceleration[10], end-point variance[1], duration[2,11], or torque-change[12]. Moreover, combined objective functions have been used to model a trade-off between different objectives, e.g., between accuracy and effort[13,14], or jerk and movement time[15]. Extensions have been proposed that, e.g., focus on initial gating mechanisms[16] or motor synergies representing agonist and antagonist muscle groups[17].

While most of these models imply a separation between the planning and the execution stage, the *optimal feedback control* theory[18–22] assumes that sensory signals about the controlled quantity are fed back to the controller. These observations are then directly used to compute the remaining optimal control signals, resulting in a feedback loop. Extensions to infinite-horizon problems[23], which yield the optimal steady-state solution at the expense of neglecting transient behavior, and explicit non-linear time costs[24,25] have been proposed.

While many early works in motor control have modeled the biomechanics as point-mass models with linear dynamics[1,13] or linked-segment models[1], there is a growing interest in biomechanical models of increasing realism and fidelity. This is spurred by advances in biomechanical modeling[3,26,27] and simulation[28,29]. Biomechanical models allow control beyond the end-effector, for example on the level of joints[12,30–35], or muscles[36–39].

Joint-actuated models apply different optimality criteria for movement generation and coordination, minimizing, e.g., the angular accelerations with constraints[40], angular jerk[30], torque-change[12,31], mechanical energy expenditure[41], a combination of absolute work and angular acceleration[35], or some combination of accuracy and effort costs in the context of optimal feedback control[14]. The biomechanical plant in these works is usually represented as a linked-segment model, with simplified kinematic properties. In particular, the shoulder joint is commonly described as a rotation-only joint, ignoring the translatory part as well as complex movements related to the scapula and clavicle. Some of these models also include simplified muscles with simplified biomechanical attachment[14].

More recently, more complex, high-fidelity biomechanical musculoskeletal models have been introduced[36,37,42,43], where the control is muscle-based. In these models, the computation of the control values is commonly based on neural networks, particularly on deep learning and reinforcement learning. These methods have been applied successfully to predict coordinated muscle activations for multi-joint arm[42], lower body[43], and full body[36] movements. Moreover, a combination of 20 neural networks, each designed to imitate some specific part of the sensorimotor system, has recently been used to synthesize movements for such diverse sensorimotor tasks as reaching, writing, and drawing[37]. To make the control of muscle-based models feasible, these works apply multiple simplifications to the full biomechanical model, such as reducing or immobilising degrees of freedom[37,42] or even completely locking the movement to two dimensions[43], ignoring tendon's elasticity[36,37], limiting maximum passive forces[36], ignoring the muscle activation dynamics[36,37] or significantly reducing the number of independently controlled muscles[42,43]. Also, the control learning strategies differ from the pure reinforcement learning approach by applying imitation learning[36,37], or using artificial training data with simplified dynamics[37].

Up to now, these models have not been evaluated regarding the realism of the movements generated, in particular whether they exhibit features characteristic of human movements, such as *Fitts' Law* and the $\frac{2}{3}$ *Power Law*[34,36,37].

Fitts' law. Fitts' Law[44] describes the speed-accuracy trade-off of aimed movements towards a spatially defined target. Given the distance D between the initial position of the controlled end-effector (e.g., the hand or the finger) and the desired target position, and given the width W of the target, this law claims a logarithmic relationship between the distance-width ratio $\frac{D}{W}$ and the resulting movement time MT:

$$MT = a + b\log_2\left(\frac{D}{W} + 1\right), \tag{1}$$

where we used the *Shannon formulation*[45]. Most works that explored possible explanations for the emergence of Fitts' Law have postulated that it results from noise in motor control. Crossman and Goodeve[46] showed that Fitts' Law emerges from the assumptions of isochronal submovements towards the target and constant error-velocity ratio. Meyer et al.[47] demonstrated that a power form of Fitts' Law emerges from the optimization of the relative duration of two submovements in order to achieve minimal movement time, assuming that the standard deviation of submovement endpoints increases proportionally with movement velocity. Fitts' Law has also been derived within the infinite-horizon optimal control framework, assuming that the target is reached as soon as the positional end-effector variance relative to the target center falls below the desired target accuracy[23].

Harris and Wolpert[1] proposed that the central nervous system (CNS) aims to minimize movement end-point variance given a fixed movement time, under the constraint of signal-dependent noise. This signal-dependent noise is assumed to be the main factor determining the end-point accuracy: Faster movements can be achieved through applying larger control signals (in the extreme, this leads to the time-minimizing *Bang-bang* control), but only at the costs of larger deviations, which in turn induce a larger end-point variance and thus a greater risk of missing the target. This trade-off has a strong neuroscientific evidence[48] and is consistent with the speed-accuracy trade-off proposed by Fitts' Law[1,2]. Moreover, in the case of arm-reaching movements, it has been shown recently that the assumptions of feed-forward control and signal-dependent noise (using dynamics of a two-link planar arm model) directly imply Fitts' Law, with coefficients a and b related to the level of signal-dependent noise[49] . Both coefficients were also shown to depend on the dynamics and kinematics, e.g., on the viscosity, or the Jacobian matrix relating the joint space and the end-effector space.

$\frac{2}{3}$ Power law. Continuous, rhythmic movements such as drawing or hand-writing, exhibit a speed-curvature trade-off described by the $\frac{2}{3}$ Power Law[50]. This law proposes a non-linear relationship between the radius of curvature ρ_n and the corresponding tangential velocity v_n,

$$v_n = \quad k\rho_n^{1-\beta}, \tag{2}$$

$$\beta \approx \quad \frac{2}{3}, \tag{3}$$

where the parameter k determines the velocity gain. This particularly implies that higher curvature leads to lower velocity. It has also been demonstrated that the $\frac{2}{3}$ Power Law is equivalent to constant affine velocity[51].

The $\frac{2}{3}$ Power Law has been confirmed to hold for a variety of task conditions, including hand movement[52], eye movement[53], perceptuomotor tasks[54,55], and locomotion[56]. Moreover, it has been shown to apply under the assumption of signal-dependent noise[1]. Schaal and Sternad[57] observed that the perimeter of the ellipse has a large impact on the validity of this law, with β obtained from a non-linear regression showing deviations in the order of 30–40% for large ellipses (or, alternatively, with decreasing coefficient of determination R^2, i.e., decreasing reliability of the parameter fitting). Based on these observations, Schaal and Sternad argue that the $\frac{2}{3}$ Power Law cannot be an intrinsic part of the movement planning procedure, but rather occurs as a "by-product" from the generation of smooth trajectories in intrinsic joint space[57] Following this argumentation, the non-linearities arising from the transformation from joint space to end-effector space, i.e., from a non-trivial kinematic chain, may account for scale- and direction-dependent results. Other theories see the cause of the wide applicability of the $\frac{2}{3}$ Power Law either in trajectory planning processes such as motor imagery[58] or jerk minimization[59], or directly emerging from muscle properties[60] or population vectors in motor cortical areas in the CNS[61,62].

Results

Fitts' law. In order to evaluate the trajectories resulting from our final policy for different target conditions, we designed a discrete Fitts' Law type task. The task follows the ISO 9241-9 ergonomics standard and incorporates 13 equidistant targets arranged in a circle at 50 cm distance in front of the body and placed 10 cm to the right of the right shoulder (Fig. 2). The objective is for the end-effector to reach each target and to remain inside the target for 100 ms. In this case we deem the movement successful. Although not included in the training phase, remaining inside the target seemed to be unproblematic during evaluation. If either the movement was successful, or 1.5 s have passed, the next target is given to the learned policy.

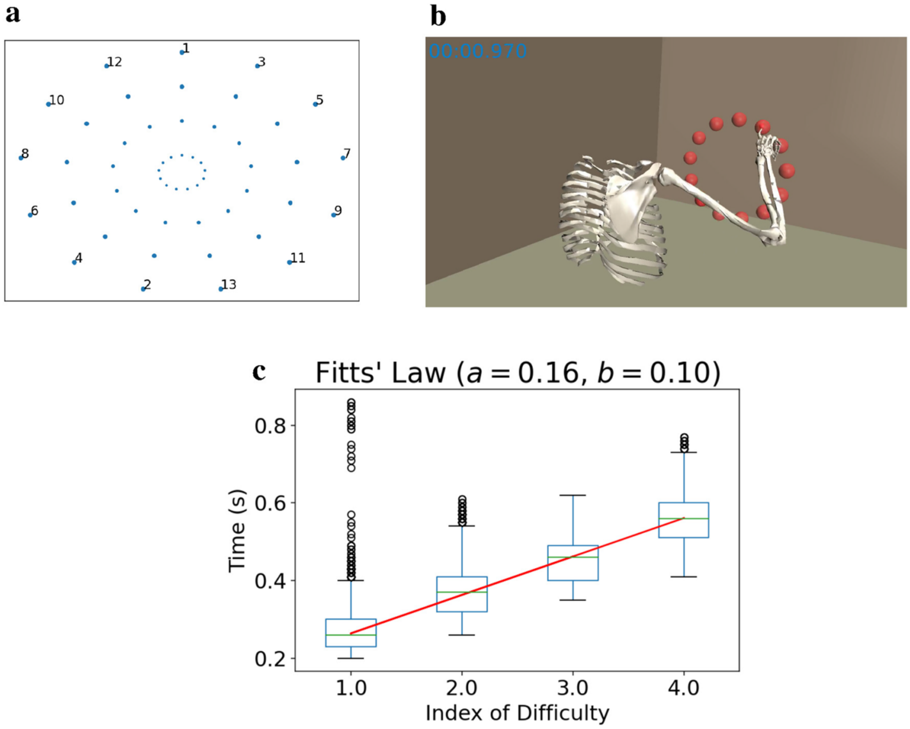

Figure 2. Fitts' Law type task. (**a**) The target setup in the discrete Fitts' Law type task follows the ISO 9241-9 ergonomics standard. Different circles correspond to different IDs and distances between targets. (**b**) Visualization of our biomechanical model performing aimed movements. Note that for each time step, only the *current* target (position and radius) is given to the learned policy. (**c**) The movements generated by our learned policy conform to Fitts' Law. Here, movement time is plotted against ID for all distances and IDs in the considered ISO task (6500 movements in total).

The Index of Difficulty (ID) of the tasks ranges from 1 to 4, where ID is computed as $\log_2(D/W + 1)$. D denotes the distance between the initial and target position, and W is the target size. We execute 50 movements for each task condition and each direction, i.e., 6500 movements in total—all were successful.

Using the trajectories from this discrete pointing task, we evaluate whether the synthesized movements follow Fitts' Law[44], i.e., whether there is a linear relationship between task difficulty (ID) and the required movement time. Figure 2c shows the total duration for each movement sorted by ID. The median movement times for each ID (green lines) are approximated by a linear function (red line, with coefficient of determination $R^2 = 0.9986$).

$\frac{2}{3}$ Power law. We evaluate whether our model exhibits the $\frac{2}{3}$ Power Law using an elliptic via-point task. To this end, we define an ellipse in 2D space (55 cm in front, 10 cm above, and 10 cm to the right of the shoulder) that lies completely within the area used for target sampling during training (ellipse radii are 7.5 cm (horizontal) and 3 cm (vertical)). Using the via-point method described in the *Methods* section below, our learned policy needs to trace the ellipse for 60 s as fast as possible . As shown in Fig. 3a, the simulation trajectories deviate from the desired ellipse, with the lower-right segment being flattened. Using these trajectories, we compute ρ_n and v_n for all time steps sampled at a rate of 100 Hz and then perform a log-log regression on the resulting values. This yields the optimal parameter values $\beta = 0.65$ and $k = 0.54$ (with correlation coefficient $R = 0.84$). Thus, the $\frac{2}{3}$ Power Law accounts for 71% of the variance observed in elliptic movements ($R^2 = 0.71$). Both the data points and the linear approximation in log-log space are shown in Fig. 3b.

Movement trajectories. In addition to Fitts' Law and the $\frac{2}{3}$ Power Law, we qualitatively analyze the movement trajectories generated by the model. Figures 4 and 5 show the position, velocity, and acceleration time series, as well as 3D movement path, Phasespace, and Hooke plots for multiple movements from the Fitts' Law type task for two representative task conditions (ID 4 respective ID 2, each with a 35 cm distance between targets) and one representative movement direction (between targets 7 and 8 shown in Fig. 2a). Apart from the 3D movement path, all plots show centroid projections of the respective trajectory onto the vector between the initial and target positions.

The movements exhibit typical features of human aimed movements, such as symmetric bell-shaped velocity profiles[63]. Movements are smooth, and gently accelerate and decelerate, as evident in the acceleration profiles and Hooke plots in Figs. 4 and 5. For high ID (Fig. 4), movements exhibit an initial rapid movement towards the target, followed by an extended phase of corrective movements. For low ID (Fig. 5), the phase of corrective movements is generally shorter.

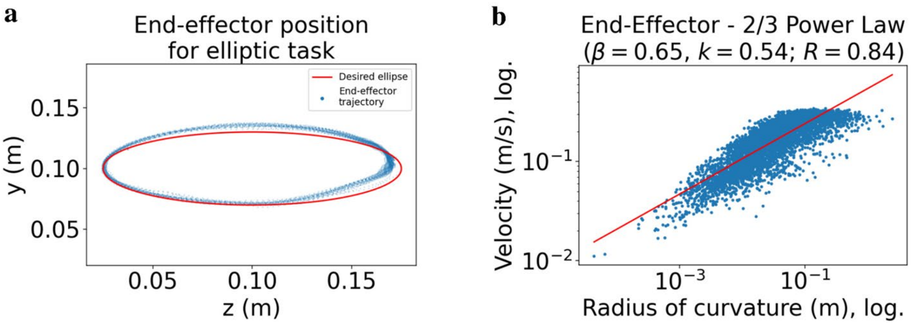

Figure 3. Elliptic via-point task. Elliptic movements generated by our learned policy conform to the $\frac{2}{3}$ Power Law. (**a**) End-effector positions projected onto the 2D space (blue dots), where targets were subsequently placed along an ellipse of 15 cm width and 6 cm height (red curve). (**b**) Log-log regression of velocity against radius of curvature for end-effector positions sampled with 100 Hz when tracing the ellipse for 60 s.

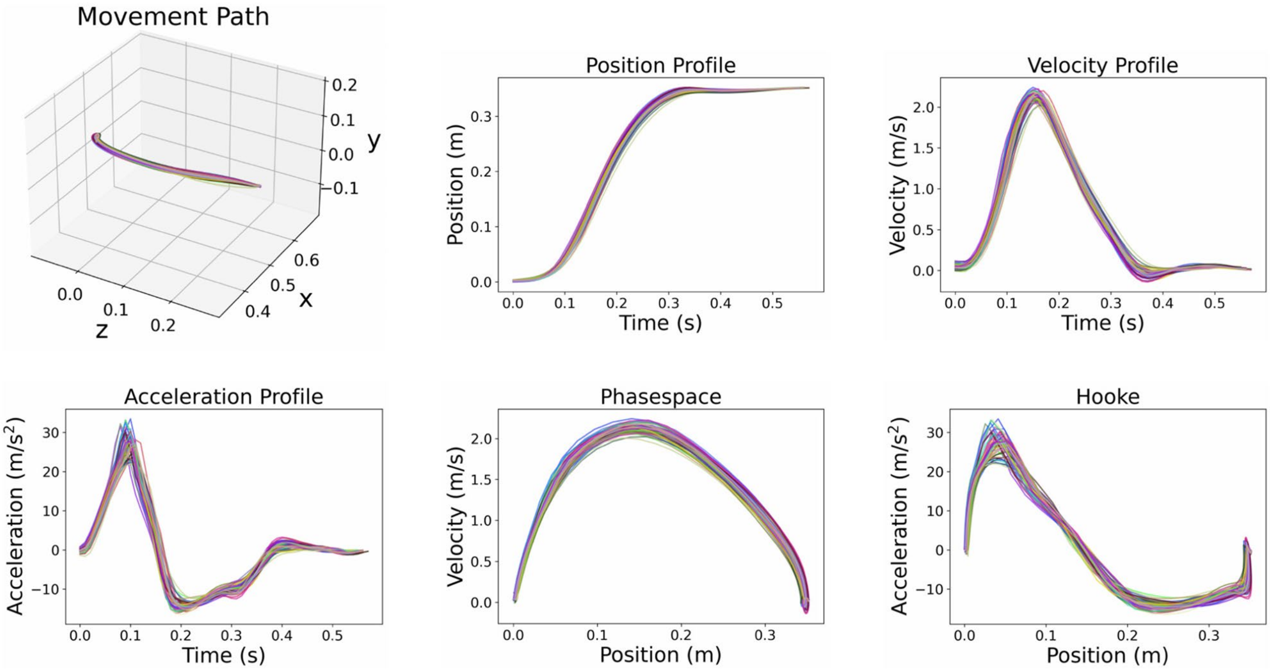

Figure 4. End-effector trajectories (ID 4). 3D path, projected position, velocity, acceleration, phasespace, and Hooke plots of 50 aimed movements (between targets 7 and 8 shown in Fig. 2a) with ID 4 and a target distance of 35 cm.

Movement trajectories towards the target are slightly curved and some of them exhibit pronounced correctional submovements at the end (see, e.g., Supplementary Fig. S1 and S2 online). The between-movement variability within one movement direction and task condition decreases with increasing ID. In particular, very simple ID 1 movements exhibit a large variability and are most prone to outliers (see, e.g., Supplementary Fig. S3 online).

For a few movement directions (mostly in ID 2 tasks), the corresponding plots seem to incorporate two different trajectory types (see, e.g., Supplementary Fig. S6 online): While some movements start with zero or even a negative acceleration and show a typical N-shaped acceleration profile, others exhibit a positive acceleration at the beginning, and their first peak is less pronounced. The reason for this behavior is corrective submovements at the end of the previous movement (see, e.g., Supplementary Fig. S4 and S5 online), leading to a different initial acceleration at the beginning of the subsequent movement. Apart from these notable features, almost all movements exhibit bell-shaped velocity and N-shaped acceleration profiles, as is typical for pointing tasks[63,64].

Discussion

Our results indicate that, under the assumption of movement time minimization given signal-dependent and constant motor noise, movement of the human upper extremity model produced by reinforcement learning follows both Fitts' Law and the $\frac{2}{3}$ Power Law. The movement times of aimed movements produced by the model depend linearly on the Index of Difficulty of the movement. For elliptic movements, the logarithm of the velocity

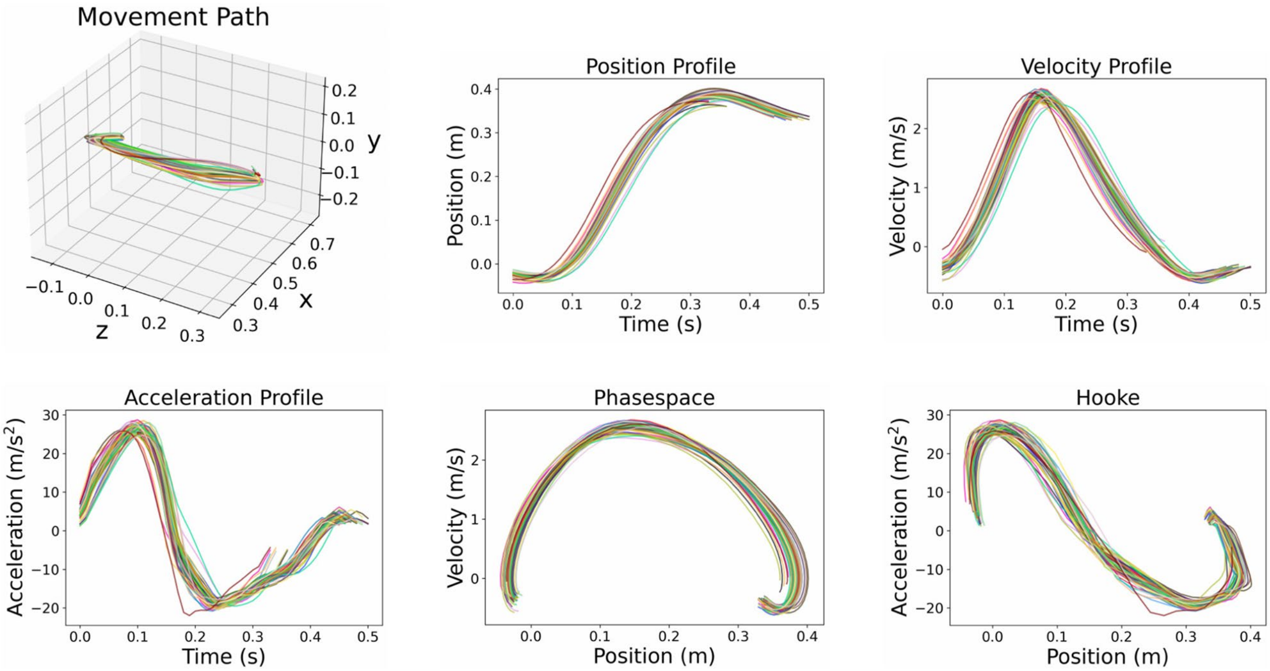

Figure 5. End-effector trajectories (ID 2). 3D path, projected position, velocity, acceleration, phasespace, and Hooke plots of 50 aimed movements (between targets 7 and 8 shown in Fig. 2a) with ID 2 and a target distance of 35 cm.

of the end-effector correlates with the logarithm of the radius of curvature[65] The optimal $\beta = 0.65$ obtained from log-log regression matches the proposed value of $\frac{2}{3}$, with a correlation coefficient of $R = 0.84$, which is consistent with previous observations, as the required ellipse has moderate size[57]. Finally, the generated trajectories exhibit features that are typical for human arm movements, such as bell-shaped velocity and N-shaped acceleration profiles.

The results confirm previous findings that demonstrated these phenomena in simpler models of the human biomechanics. In particular, the emergence of Fitts' Law and the $\frac{2}{3}$ Power Law from the assumption of signal-dependent noise has been demonstrated in the case of point-mass and linked-segment models of the human arm[1,2,49]. Our results support that insight by showing that Fitts' Law and the $\frac{2}{3}$ Power Law also emerge from those normative principles in a state-of-the-art biomechanical model of the human arm with simplified actuation.

In addition, we want to emphasize that the control signals that drive this model were obtained from RL methods. The fact that Fitts' Law and the $\frac{2}{3}$ Power Law hold for the generated trajectories provides evidence that behavior abiding these established laws of human motion can be generated using joint-actuated biomechanical models controlled by reinforcement learning algorithms. To the best of our knowledge, this has not yet been shown for state-of-the-art biomechanical models.

One limitation of our approach is the implicit assumption of *perfect observability*, as all state information (joint angles, end-effector position, etc.) are immediately available to the controller, without any disturbing noise. In the future, it will be interesting to combine state-of-the art models of sensory perception with the presented RL-based motor control approach. Promising approaches to address this problem include the usage of recurrent networks[66,67] and the internal formation of "beliefs", given the latest (imperfect) observations[68].

Another limitation is the usage of simplified muscle dynamics due to the curse of dimensionality. However, recent applications of deep learning methods, which approximate complex state-dependent torque limits[69] or muscle activation signals[37] using synthesized training data, raise hope for future approaches that efficiently combine RL or optimal control methods with state-of-the-art muscle models. It will be interesting to see whether well-established regularities such as Fitts' Law or the $\frac{2}{3}$ Power Law also emerge from such models.

Methods

Below, we first provide details on our biomechanical model. After discussing our general reinforcement learning approach, we focus on the individual components of our method, namely states, actions, scaling factors, rewards, and an adaptive target-selection mechanism. We also provide details on the implementation of our algorithm. Finally, we discuss the methods used for evaluation.

Biomechanical model of the human upper extremity.

Our biomechanical model of the human upper extremity is based on the *Upper Extremity Dynamic model*[3], which was originally implemented in *OpenSim*[28]. Kinematically, the model represents the human shoulder and arm, using seven physical bodies and five "phantom" bodies to model the complex movement of the shoulder. This corresponds to three joints (shoulder, elbow, and wrist) with seven DOFs and five additional joints with thirteen associated components coupled by thirteen constraints with the DOFs. Each DOF has constrained joint ranges (see Table 1), which limits the possible movements. In contrast to linked-segment models, the *Upper Extremity Dynamic model* represents both translational and rotatory components of the movement within shoulder, clavicle, and scapula, and also within the wrist. It also contains physiological joint axis orientations instead of the perpendicular orientations in linked-segment models. The dynamics components of the musculoskeletal model are represented by the weight and inertia matrix of each non-phantom body and the default negligible masses and inertia of all phantom bodies. The dynamics properties of the model were extracted from various previously published works on human and cadaveric studies. The active components of the *Upper Extremity Dynamic Model* consist of thirty-one Hill-type muscles as well as of fourteen coordinate limit forces softly generated by the ligaments when a DOF approaches the angle range limit. Further details of this model are given in Saul et al.[3]

In order to make reinforcement learning feasible, we manually implement the *Upper Extremity Dynamic Model* in the fast MuJoCo physics simulation[7]. With respect to kinematics, the MuJoCo implementation of the model is equivalent to the original OpenSim model and contains physiologically accurate degrees of freedom, as well as corresponding constraints. We assume the same physiological masses and inertial properties of individual segments as in the OpenSim model. We do not implement muscles in the MuJoCo model, as this would significantly slow down the simulation and make reinforcement learning computationally infeasible due to the exponential growth of decision variables in the (discretized) action space when increasing the number of DOFs – the *curse of dimensionality*. In particular, computing dynamic actuator lengths (which significantly affect the forces produced by muscle activation patterns) has proven challenging in MuJoCo[70]. Instead, we implement simplified actuators, representing aggregated muscle actions on each individual DOF, which are controlled using the second-order dynamics introduced by van der Helm et al.[71] with fixed excitation and activation time constants $t_e = 30$ ms and $t_a = 40$ ms, respectively. We discretize the continuous state space system using the *forward Euler method*, which yields the following dynamics:

$$\begin{bmatrix} \sigma_{n+1}^{(q)} \\ \dot{\sigma}_{n+1}^{(q)} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ \frac{-\Delta t}{(t_e t_a)} & 1 - \Delta t \frac{t_e + t_a}{t_e t_a} \end{bmatrix} \begin{bmatrix} \sigma_n^{(q)} \\ \dot{\sigma}_n^{(q)} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta t}{t_e t_a} \end{bmatrix} c_n^{(q)}, \tag{4}$$

where $c_n^{(q)}$ is the applied control and $\sigma_n^{(q)}$ the resulting activation for each DOF $q \in \mathcal{Q}$, and $\mathcal{Q}$ is the set that contains all DOFs. The controls are updated every Δt=10 ms, at time steps $n \in \{0, \ldots, N-1\}$. To get more accurate results, at each time step n, we compute five sub-steps (during which the control $c_n^{(q)}$ is constant) with a sampling time of 2 ms to arrive at time step $n+1$.

We assume both signal-dependent and constant noise in the control, i.e.,

$$c_n^{(q)} = (1 + \eta_n) a_n^{(q)} + \epsilon_n, \tag{5}$$

where $a_n = (a_n^{(q)})_{q \in \mathcal{Q}}$ denotes the action vector obtained from the learned policy, and η_n and ϵ_n are Gaussian random variables with zero mean and standard deviations of 0.103 and 0.185, respectively, as described by van Beers et al.[4] The torques, which are applied at each DOF independently, are obtained by multiplying the respective activation $\sigma_n^{(q)}$ with a constant scaling factor $g^{(q)}$, which represents the strength of the muscle groups at the this DOF, i.e.,

$$\tau_n^{(q)} = g^{(q)} \sigma_n^{(q)}. \tag{6}$$

We select the scaling factors, and respectively the maximum voluntary torques for the actuators given in Table 1, based on experimental data as described below. We currently do not model the soft joint ranges in MuJoCo, as the movements the model produces do not usually reach joint limits.

The used biomechanical model provides the following advantages over simple linked-segment models:

- Phantom bodies and joints allow for more realistic movements, including both translation and rotation components within an individual joint,
- Individual joint angle and torque limits are set for each and every DOF,
- Axes between joints are chosen specifically and not just perpendicular between two segments,
- The model includes physiological body segment masses, and yields better options for scaling individual body parts, e.g., based on particular individuals.

Reinforcement learning. We define the task of controlling the biomechanical model of the human upper extremity through motor control signals applied at the joints as a reinforcement learning problem, similar to recent work from Cheema et al.[34] In this formulation, a policy $\pi_\theta(a|s)$ models the conditional distribution over actions $a \in \mathcal{A}$ (motor control signals applied at the individual DOFs) given the state $s \in \mathcal{S}$ (the pose, velocities, distance to target, etc.). The subindex θ denotes the parameters of the neural networks introduced below. At each timestep $n \in \{0, \ldots, N\}$, we observe the current state s_n, and sample a new action a_n from the current policy π_θ. The physical effects of that action, i.e., the application of these motor control signals, constitute the new state s_{n+1}, which we obtain from our biomechanical simulation. In our model, given s_n and a_n, the next state s_{n+1} is not deterministic, since both signal-dependent and constant noise are included. Hence, we denote the probability of reaching some subsequent state s_{n+1} given s_n and a_n by $p(s_{n+1}|s_n, a_n)$, while $p(s_0)$ denotes the probability of starting in s_0. Given some policy π_θ and a trajectory $T = (s_0, a_0, \ldots, a_{N-1}, s_N)$,

$$p_\theta(T) = p(s_0)\prod_{n=0}^{N-1}\pi_\theta(a_n|s_n)p(s_{n+1}|s_n, a_n) \tag{7}$$

describes the probability of realizing that trajectory. Evaluating/Sampling equation (7) for all possible trajectories $T \in \mathcal{T}$ then yields the distribution over all possible trajectories, $\varrho_\theta^{\mathcal{T}}$, induced by a policy π_θ.

We compute a reward r_n at each time step n, which allows us to penalize the total time needed to reach a given target. The total return of a trajectory is given by the sum of the (discounted) rewards $\sum_{n=0}^{N}\gamma^n r_n$, where $\gamma \in]0, 1]$ is a discount factor that causes the learner to prefer earlier rewards to later ones. Incorporating the entropy term,

$$\mathcal{H}(\pi_\theta(\cdot \mid s)) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[-\log(\pi_\theta(a \mid s))], \tag{8}$$

yields the expected (soft) return

$$J(\theta) = \mathbb{E}_{T\sim\varrho_\theta^{\mathcal{T}}}\left[\left(\sum_{n=0}^{N-1}\gamma^n\big(r_n - \alpha\log(\pi_\theta(a_n \mid s_n))\big)\right) + \gamma^N r_N\right], \tag{9}$$

which we want to maximize with respect to the parameters θ, i.e., the goal is to identify the optimal parameters θ^* that maximize $J(\theta)$. Here, the *temperature parameter* $\alpha > 0$ determines the importance of assigning the same probability to all actions that yield the same return (enforced by maximizing the entropy $\mathcal{H}$), i.e., increasing the "stochasticity" of the policy π_θ, relative to maximizing the expected total return. It thus significantly affects the optimal policy, and finding an "appropriate" value is non-trivial and heavily depends on the magnitude of the rewards r_n. For this reason, we decided to adjust it automatically during training together with the parameters θ, using dual gradient descent as implemented in the soft-actor critic algorithm (see below)[6].

It is important to note that the soft return in Equation (9) is different from the objective function used in standard reinforcement learning. The MaxEnt RL formulation, which incorporates an additional entropy maximization term, provides several technical advantages. These include the natural state-space exploration[72,73], a smoother optimization landscape that eases convergence towards the global optimum[74–76], and increased robustness to changes in the reward function[77,78]. In practice, many RL algorithms have gained increased stability from the additional entropy maximization[79–81]. Conceptually, MaxEnt RL can be considered equivalent to *probabilistic matching*, which has been used to explain human decision making[82,83]. Existing evidence indicates that human adults tend to apply probabilistic matching methods rather than pure maximization strategies[82,84,85]. However, these observations still lack conclusive neuroscientific explanation[80].

In order to approximate the optimal parameters θ^*, we use a policy-gradient approach, which iteratively refines the parameters θ in the direction of increasing rewards. Reinforcement learning methods that are based on fully sampled trajectories usually suffer from updates with high variance. To reduce this variance and thus accelerate the learning process, we choose an approach that includes two approximators: an *actor network* and a *critic network*. These work as follows. Given some state s_0 as input, the actor network outputs the (standardized) mean and standard deviation of as many normal distributions as dimensions of the action space. The individual action components are then sampled from these distributions. To update the actor network weights, we must measure the "desirability" of some action a, given some state s, i.e., how much reward can be expected when starting in this state with this action and subsequently following the current policy. These values are approximated by the critic network.

The architecture of both networks is depicted in Fig. 6. For the sake of a simpler notation, the parameter vector θ contains the weights of both networks, however these weights are not shared between the two. These two networks are then coupled with the *soft actor-critic (SAC)* algorithm[6], which has been used successfully in physics-based character motion[86]: As a policy-gradient method, it can be easily used with a continuous action space such as continuous motor signals – something that is not directly possible with value function methods like *DQN*[5]. As an off-policy method that makes use of a replay buffer, it is quite sample-efficient. This is important, since running forward physics simulations in MuJoCo constitutes the major part of the training duration. Moreover, it has been shown that SAC outperforms other state-of-the-art algorithms such as PPO[87] or TD3[88]. Supporting the observations in Haarnoja et al.[6], we also found our training process to be faster and more robust when using SAC rather than PPO. Moreover, SAC incorporates an automatic adaption of the temperature α using dual gradient descent, which eliminates the need for manual, task-dependent fine-tuning. In order to obtain an unbiased estimate of the optimal value function, we use *Double Q-Learning*[89], using a separate target critic network. The neural network parameters are optimized with the Adam optimizer[90].

States, actions, and scaling factors. Using the MuJoCo implementation of the biomechanical model described above, the **states** $s \in \mathcal{S} \subseteq \mathbb{R}^{48}$ in our RL approach include the following information:

- Joint angle for each DOF $q \in \mathcal{Q}$ in radians (7 values),
- Joint velocity for each DOF $q \in \mathcal{Q}$ in radians/s (7 values),
- Activations $\sigma^{(q)}$ and excitations $\dot{\sigma}^{(q)}$ for each DOF $q \in \mathcal{Q}$ (2×7 values),
- Positions of the end-effector and target sphere (2×3 values),
- (positional) Velocities of the end-effector and target sphere (2×3 values),
- (positional) Acceleration of the end-effector (3 values),
- *Difference vector*: vector between the end-effector attached to the index finger and the target, pointing towards the target (3 values),
- Projection of the end-effector velocity towards the target (1 value),

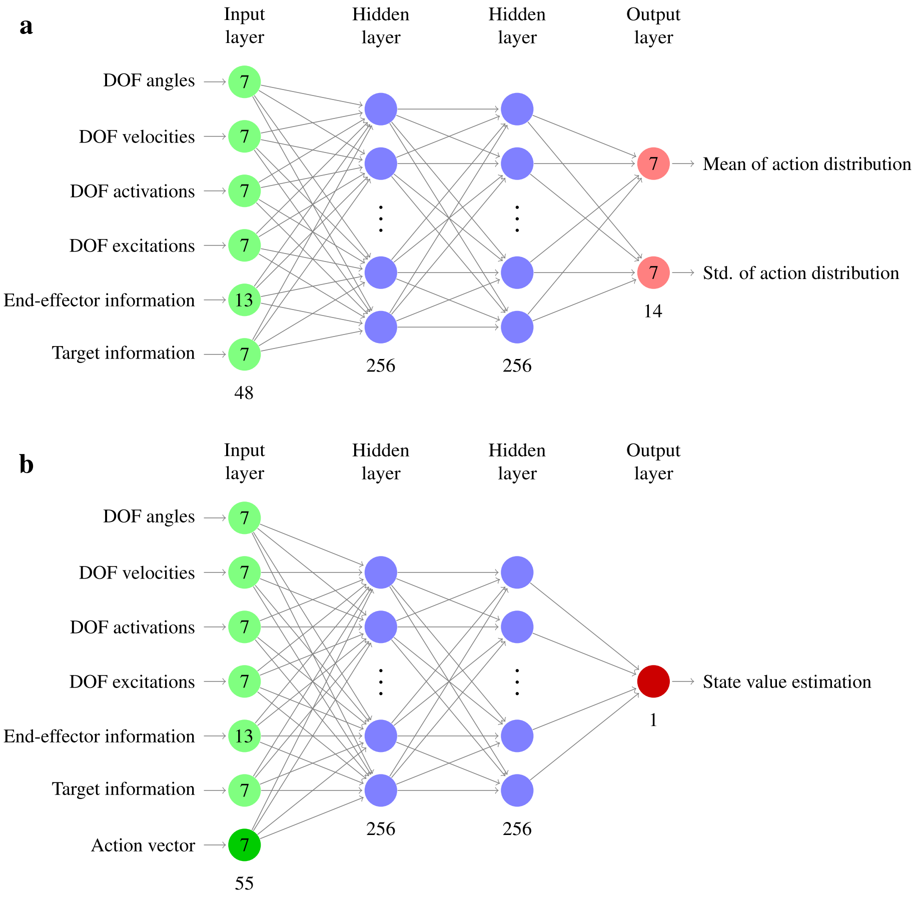

Figure 6. Neuronal network architectures. (**a**) The actor network takes a state s as input and returns the policy π_θ in terms of mean and standard deviation of the seven normal distributions, from which the components of the action vector are drawn. (**b**) The critic network takes both state s and action vector a as input and returns the estimated state-action value. Two critic networks are trained simultaneously to improve the speed and stability of learning (*Double Q-Learning*). Detailed information about the input state components are given in the *Methods* section.

- Radius of the target sphere (1 value).

We found that in our case, the target velocity (which always equals zero for the considered tasks), the end-effector acceleration, the difference vector, and the projection of the end-effector velocity can be omitted from state space without reducing the quality of the resulting policy. However, we decided to incorporate these observations, as they did not considerably slow down training and might be beneficial for more complex tasks such as target tracking or via-point tasks.

Each component $a^{(q)} \in [-1, 1]$ of the **action vector** $a = (a^{(q)})_{q\in\mathcal{Q}} \in \mathcal{A} \subseteq \mathbb{R}^7$ is used to actuate some DOF $q \in \mathcal{Q}$ by applying the torque $\tau^{(q)}$ resulting from Eqs. (4)–(6). Note that in addition to these actuated forces, additional active forces (e.g., torques applied to parent joints) and passive forces (e.g., gravitational and contact forces) act on the joints in each time step.

We determine experimentally the maximum torque a human would exert at each DOF in this task as follows. We implemented the Fitts' Law task described above in a VR environment displayed via the HTC Vive Pro VR headset. We recorded the movements of a single participant performing the task, using the Phasespace X2E motion capture system with a full-body suit provided with 14 optical markers. This study was granted ethical approval by the ethics committee of the University of Bayreuth and followed ethical standards according to the Helsinki Declaration. Written informed consent was received from the participant, which received an economic compensation for participating in the study. Using OpenSim, we scaled the *Upper Extremity Dynamic Model* to this particular person. We then used OpenSim to perform *Inverse Dynamics* to obtain the torque sequences that are most likely to produce the recorded marker trajectories. For each DOF $q \in \mathcal{Q}$, we set the corresponding **scaling factor** $g^{(q)}$ to the absolute maximum torque applied at this DOF during the experiment, omitting a small number of outliers from the set of torques, i.e., values with a distance to mean larger than 20 times the standard deviation. The resulting values are shown in Table 1.

Reward function and curriculum learning.

The behavior of the policy is determined largely by the reward r_n that appears in Eq. (9). We designed the reward following Harris and Wolpert[1], who argue that there is no rational explanation as to why the central nervous system (CNS) should explicitly try to minimize previously proposed metrics such as the change in torque applied at the joints[12], or the acceleration (or jerk) of the end-effector[8]. They argue that it is not even clear whether the CNS is able to compute, store, and integrate these quantities while executing motions.

Instead, they argue that the CNS aims to minimize movement end-point variance given a fixed movement time, under the constraint of signal-dependent noise. Following Harris and Wolpert[1], t his is equivalent to minimizing movement time when the permissible end-point variance is given by the size of the target. This objective is simple and intuitively plausible, since achieving accurate aimed movements in minimal time is critical for the success of many movement tasks. Moreover, it has already been applied to linear dynamics[2].

Therefore, the objective of our model is to *minimize movement time while reaching a target of given width*.

More precisely, our **reward function** consists only of a time reward, which penalizes every time step of an episode equally:

$$r_n = -100\Delta t. \tag{10}$$

This term provides incentives to terminate the episode (which can only be achieved by reaching the target) as early as possible. Since we apply each control a_n for 10 ms, Δt amounts to 0.01 in our case, i.e., $r_n = -1$ in each time step $n \in \{0, \ldots, N\}$.

According to our experience, it is possible to learn aimed movements despite the lack of gradient provided by the reward function, provided the following requirements are met. The initial posture needs to be sampled randomly, and the targets need to be large enough at the beginning of the training to ensure that the target is reached by exploration sufficiently often in early training steps to guide the reinforcement learner. However, creating a predetermined curriculum that gradually decreases the target width during training appropriately has proved very difficult. In most cases, the task difficulty either increased too fast, leading to unnatural movements that do not reach the target directly (and often not at all), or progress was slow, resulting in a time-consuming training phase.

For this reason, we decided to use an adaptive curriculum, which adjusts the target width dynamically, depending on the recent success rate. Specifically, we define a *curriculum state*, which is initialized with an initial target diameter of 60 cm. Every 10K update steps, the current policy is evaluated on 30 complete episodes, for which target diameters are chosen, depending on the current state of the curriculum. Based on the percentage of targets reached within the permitted 1.5 s (*success rate*), the curriculum state is updated. If the success rate falls below 70%, it is increased by 1 cm; if the success rate exceeds 90%, it is decreased by 1 cm. To avoid target sizes that are larger than the initial width or are too close to zero, we clipped the resulting value to the interval $[0.1\text{ cm},\ 60\text{ cm}]$.

At the beginning of each episode, the target diameter is set to the current curriculum state with probability $1-\varepsilon$, and sampled uniformly randomly between 0.1 cm and 60 cm with probability $\varepsilon = 0.1$, which has proven to be a reasonable choice. This ensures in particular that all required target sizes occur throughout the training phase, and thus prevents forgetting how to solve "simpler" tasks (in literature, often referred to as *catastrophic forgetting*; see, e.g., McCloskey et al.[91]).

Implementation of the reinforcement learning algorithm.

The actor and critic networks described in the *Reinforcement Learning* section consist of two fully connected layers with 256 neurons each, followed by the output layer, which either returns the means and standard deviations of the action distributions (for the actor network) or the state-action value (for the critic network). To improve the speed and stability of learning, we train two separate, but same-structuredidentically structured critic networks and use the minimum of both outputs as the teaching signal for all networks (*Double Q-Learning*)[6,89]. In all networks, ReLU[92] is used as nonlinearity for both hidden layers. The network architectures are depicted in Fig. 6.

The reinforcement learning methods of our implementation are based on the *TF-Agents* library[93]. The learning phase consists of two parts, which are repeated alternately: *trajectory sampling* and *policy updating*.

In the trajectory sampling part, the target position is sampled from the uniform distribution on a cuboid of 70 cm height, 40 cm width, and 30 cm depth, whose center is placed 50 cm in front of the human body, and 10 cm to the right of the shoulder. The width of the target is controlled by the adaptive curriculum described above. The biomechanical model is initialized with some random posture, for which the joint angles are uniformly sampled from the convex hull of static postures that enables keeping the end-effector in one of 12 targets placed along the vertices of the cuboid described above. The initial joint velocities are uniformly sampled from the interval $[-0.005\text{ radians/s},\ 0.005\text{ radians/s}]$.

In each step $n \in \{0, \ldots, N-1\}$, given the current state vector $s_n \in \mathcal{S}$ (see description above), an action is sampled from the current policy $\pi_\theta(\cdot \mid s_n)$. Next, the MuJoCo simulation uses this action to actuate the model joints. It also updates the body posture, and returns both the reward r_n and the subsequent state vector s_{n+1}. In our implementation, each episode in the learning process contains at most $N = 150$ of such steps, with each step corresponding to 10 ms (allowing movements to be longer than one and a half seconds did not improve the training procedure significantly). If the target is reached earlier, i.e., the distance between end-effector and target center is lower than the radius of the target sphere, and the end-effector remained inside the target for 100 ms, the current episode terminates and the next episode begins with a new target position and width. At the beginning of the training, 10K steps are taken and the corresponding transitions stored in a replay buffer, which has a capacity of 1M steps. During training, only one step is taken and stored per sampling phase.

In the policy updating part, 256 previously sampled transitions (s_n, a_n, r_n, s_{n+1}) are randomly chosen from the replay buffer to update both the actor network and the critic network weights. We use a discount factor of

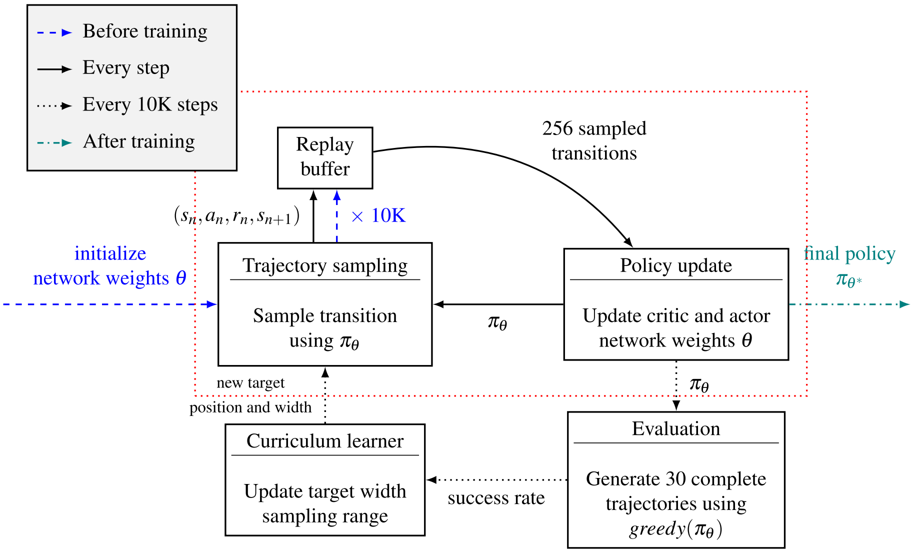
Before training
Every step
Every 10K steps
After training
Replay buffer
256 sampled transitions
(s_n, a_n, r_n, s_{n+1})
× 10K
initialize network weights θ
Trajectory sampling
Sample transition using π_θ
π_θ
Policy update
Update critic and actor network weights θ
final policy π_{θ^*}
new target position and width
π_θ
Curriculum learner
Update target width sampling range
success rate
Evaluation
Generate 30 complete trajectories using $greedy(\pi_\theta)$

Figure 7. Reinforcement learning procedure. Before training, the networks are initialized with random weights θ, and 10 K transitions are generated using the resulting initial policy. These are stored in the replay buffer (blue dashed arrows). During training (red dotted box), trajectory sampling and policy update steps are executed alternately in each step. The targets used in the trajectory sampling part are generated by the curriculum learner, which is updated every 10K steps, based on an evaluation of the most recent (greedy) policy. As soon as the target width suggested by the curriculum learner falls below 1 cm, the training phase is completed and the final policy π_{θ^*} is returned (teal dash-dotted arrow).

	Joint angle ranges (deg)		Joint torque ranges (Nm)	
Joint DOF	Minimum	Maximum	Minimum	Maximum
Elevation angle	– 90	130	– 36.01	36.01
Shoulder elevation	0	180	– 60.97	60.97
Shoulder rotation	– 90	20	– 19.37	19.37
Elbow flexion	0	130	– 12.57	12.57
Pronation/supination	– 90	90	– 1.03	1.03
Wrist deviation	– 10	25	– 2.14	2.14
Wrist flexion	– 70	70	– 1.53	1.53

Table 1. Joint ranges of individual DOFs. Angle and torque ranges of all joint DOFs, which are actuated via second-order muscle dynamics [Eq. (4)]. Moment arm scaling factors are defined as the magnitude of the torque range limits.

γ =0.99 in the critic loss function of SAC. All other parameters are set to the default values of the TF-Agents SAC implementation[93].

Both parts of our learning algorithm, the trajectory sampling and the policy update, are executed alternately until the curriculum state, i.e., the current suggested target diameter, falls below 1 cm. With our implementation, this was the case after 1.2M steps, corresponding to about four hours of training time. To evaluate a policy π_θ, we apply the action a_n^* with the highest probability under this policy for each time step (i.e., we use the corresponding *greedy* policy) and evaluate the resulting trajectory. Such an *evaluation* is done every 10K steps, for which 30 complete episodes are generated using this deterministic policy, and the resulting performance indicators are stored. After the training phase, θ^* is set to the latest parameter set θ, i.e., the final policy π_{θ^*} is chosen as the latest policy π_θ.

An overview of the complete training procedure is given in Fig. 7.

Evaluation.

For an evaluation of the trajectories resulting from the learned policy for different target conditions, we designed a discrete Fitts' Law type task. This task follows the ISO 9241-9 ergonomics standard and incorporates 13 equidistant targets arranged in a circle 50 cm in front of the body and placed 10 cm right of the right shoulder (Fig. 2). As soon as a target is reached and the end-effector remains inside for 100 ms, the next target is given to the learned policy. This also happens after 1.5 s, regardless of whether or not the episode was successful.

Based on the recommendations from Guiard et al.[94], we determine different task difficulty conditions by sampling "form and scale", i.e., the *Index of Difficulty (ID)* and the distance *D* between the target centers are sampled independently, instead of using a distance-width grid. We use the *Shannon Formulation*[45] of Fitts' Law [Eq. (1)] to compute the resulting distance between the initial and target point D, given the target width W and the ID:

$$\mathrm{ID} = \log_2\left(\frac{D}{W} + 1\right). \tag{11}$$

The used combinations of distance, width, and ID can be found as Supplementary Table S1 online, and the resulting target setup is shown in Fig. 2a.

The model executes 50 movements for each task condition and each direction, i.e., 6500 movements in total. All movements reached the target and remained inside for 100 ms within the given maximum movement time of 1.5 s. Plots for all task conditions and movement directions, together with their underlying data, can be found in a public repository[95].

In addition, an adaptive "moving target" mechanism is applied to generate elliptic movements from our learned policy. During training, the policy only learned to reach a given target as fast and accurate as possible—it was never asked to follow a specific path accurately. For this reason, we make use of the following method.

Initially, we place the first target on the ellipse such that 10% of the complete curve needs to be covered clockwise within the first movement, starting at a fixed initial position (leftmost point on the ellipse). In contrast to regular pointing tasks, the target already switches as soon as the movement (or rather the projection of the movement path onto the ellipse) covers more than half of this distance. The next target is then chosen so as to again create an incentive to cover the next 10% of the elliptic curve. Thus, roughly 20 via-points in total are subsequently placed on the ellipse. As shown in Fig. 3a, this indeed leads to fairly elliptic movements.

For our evaluation, we use an ellipse with horizontal and vertical diameters of 15 cm and 6 cm (similar to the ellipse used by Harris and Wolpert[1]), with its center placed 55 cm in front, 10 cm above, and 10 cm to the right of the shoulder. The task was performed for one minute, with end-effector position, velocity, and acceleration stored every 10 ms.

Comprehensive data for all of these movements can also be found in a public repository[95].

Data availability

The datasets generated during and/or analysed during the current study are available in a public repository, https://doi.org/10.5281/zenodo.4268230.

Received: 27 November 2020; Accepted: 29 June 2021
Published online: 14 July 2021

References

1. Harris, C. M. & Wolpert, D. M. Signal-dependent noise determines motor planning. *Nature* **394**, 780–784. https://doi.org/10.1038/29528 (1998).
2. Tanaka, H., Krakauer, J. W. & Qian, N. An optimization principle for determining movement duration. *J. Neurophysiol.* **95**, 3875–3886. https://doi.org/10.1152/jn.00751.2005 (2006).
3. Saul, K. R. *et al.* Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Comput. Methods Biomech. Biomed. Eng.* **5842**, 1–14. https://doi.org/10.1080/10255842.2014.916698 (2014).
4. van Beers, R. J., Haggard, P. & Wolpert, D. M. The role of execution noise in movement variability. *J. Neurophysiol.* **91**, 1050–1063. https://doi.org/10.1152/jn.00652.2003 (2004).
5. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (A Bradford Book, 2018).
6. Haarnoja, T. *et al.* Soft actor-critic algorithms and applications. arXiv:1801.01290 (2018).
7. Todorov, E., Erez, T. & Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033, https://doi.org/10.1109/IROS.2012.6386109 (2012).
8. Flash, T. & Hogan, N. The coordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.* **5**, 1688–1703 (1985).
9. Hoff, B. & Arbib, M. A. Models of trajectory formation and temporal interaction of reach and grasp. *J. Mot. Behav.* **25**, 175–192, https://doi.org/10.1080/00222895.1993.9942048 (1993).
10. Nelson, W. L. Physical principles for economies of skilled movements. *Biol. Cybern.* **46**, 135–147. https://doi.org/10.1007/BF00339982 (1983).
11. Artstein, Z. Discrete and continuous bang-bang and facial spaces or: Look for the extreme points. *SIAM Rev.* **22**, 172–185 (1980).
12. Uno, Y., Kawato, M. & Suzuki, R. Formation and control of optimal trajectory in human multijoint arm movement—Minimum torque-change model. *Biol. Cybern.* **61**, 89–101. https://doi.org/10.1007/BF00204593 (1989).
13. Todorov, E. *Studies of Goal-Directed Movements* (Massachusetts Institute of Technology, 1998).
14. Li, W. & Todorov, E. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics, (ICINCO 2004)*, vol. 1, 222–229 (2004).
15. Hoff, B. A model of duration in normal and perturbed reaching movement. *Biol. Cybern.* **71**, 481–488. https://doi.org/10.1007/BF00198466 (1994).
16. Bullock, D. & Grossberg, S. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychol. Rev.* **95**, 49 (1988).
17. Plamondon, R. A kinematic theory of rapid human movements: Part iii. Kinetic outcomes. *Biol. Cybern.* **78**, 133–145. https://doi.org/10.1007/s004220050420 (1998).
18. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* **5**, 1226–1235. https://doi.org/10.1038/nn963 (2002).
19. Scott, S. Optimal feedback control and the neural basis of volitional motor control. *Nat. Rev. Neurosci.* **5**, 532–46. https://doi.org/10.1038/nrn1427 (2004).
20. Todorov, E. Optimality principles in sensorimotor control. *Nat. Neurosci.* **7**, 907–915. https://doi.org/10.1038/nn1309 (2004).
21. Shadmehr, R. & Krakauer, J. A computational neuroanatomy for motor control. *Exp. Brain Res.* **185**, 359–381 (2008).
22. Diedrichsen, J., Shadmehr, R. & Ivry, R. B. The coordination of movement: Optimal feedback control and beyond. *Trends Cognit. Sci.* **14**, 31–39. https://doi.org/10.1016/j.tics.2009.11.004 (2010).
23. Qian, N., Jiang, Y., Jiang, Z.-P. & Mazzoni, P. Movement duration, fitts's law, and an infinite-horizon optimal feedback control model for biological motor systems. *Neural Comput.*https://doi.org/10.1162/NECO_a_00410 *(2012).*

24. Shadmehr, R., De Xivry, J. J. O., Xu-Wilson, M. & Shih, T.-Y. Temporal discounting of reward and the cost of time in motor control. *J. Neurosci.* **30**, 10507–10516 (2010).
25. Berret, B. & Jean, F. Why don't we move slower? The value of time in the neural control of action. *J. Neurosci.* **36**, 1056–1070. https://doi.org/10.1523/JNEUROSCI.1921-15.2016 (2016).
26. Holzbaur, K. R., Murray, W. M. & Delp, S. L. A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Ann. Biomed. Eng.* **33**, 829–840 (2005).
27. Seth, A., Dong, M., Matias, R. & Delp, S. Muscle contributions to upper-extremity movement and work from a musculoskeletal model of the human shoulder. *Front. Neurorobot.* **13**, 90. https://doi.org/10.3389/fnbot.2019.00090 (2019).
28. Delp, S. L. *et al.* Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**, 1940–1950 (2007).
29. Seth, A. *et al.* Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Comput. Biol.* **14**, 1–20. https://doi.org/10.1371/journal.pcbi.1006223 (2018).
30. Rosenbaum, D. A., Loukopoulos, L. D., Meulenbroek, R. G., Vaughan, J. & Engelbrecht, S. E. Planning reaches by evaluating stored postures. *Psychol. Rev.* **102**, 28–67. https://doi.org/10.1037/0033-295x.102.1.28 (1995).
31. Nakano, E. *et al.* Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. *J. Neurophysiol.* **81**, 2140–2155, https://doi.org/10.1152/jn.1999.81.5.2140 (1999).
32. Kawato, M. Optimization and learning in neural networks for formation and control of coordinated movement. *Attent. Perform.* 821–849 (1993).
33. Kawato, M. Trajectory formation in arm movements: Minimization principles and procedures. *Adv. Motor Learn. Control.* 225–259 (1996).
34. Cheema, N. *et al.* Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, 1–13, https://doi.org/10.1145/3313831.3376701 (Association for Computing Machinery, New York, NY, USA, 2020).
35. Berret, B., Chiovetto, E., Nori, F. & Pozzo, T. Evidence for composite cost functions in arm movement planning: An inverse optimal control approach. *PLoS Comput. Biol.* **7**, 1–18. https://doi.org/10.1371/journal.pcbi.1002183 (2011).
36. Lee, S., Park, M., Lee, K. & Lee, J. Scalable muscle-actuated human simulation and control. *ACM Trans. Graph.*https://doi.org/10.1145/3306346.3322972 *(2019).*
37. Nakada, M., Zhou, T., Chen, H., Weiss, T. & Terzopoulos, D. Deep learning of biomimetic sensorimotor control for biomechanical human animation. *ACM Trans. Graph.*https://doi.org/10.1145/3197517.3201305 *(2018).*
38. Si, W., Lee, S.-H., Sifakis, E. & Terzopoulos, D. Realistic biomechanical simulation and control of human swimming. *ACM Trans. Graph.*https://doi.org/10.1145/2626346 *(2015).*
39. Fan, J., Jin, J. & Wang, Q. Humanoid muscle-skeleton robot arm design and control based on reinforcement learning. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 541–546, https://doi.org/10.1109/ICIEA48937.2020.9248350 (2020).
40. Ben-Itzhak, S. & Karniel, A. Minimum acceleration criterion with constraints implies bang–bang control as an underlying principle for optimal trajectories of arm reaching movements. *Neural Comput.* **20**, 779–812. https://doi.org/10.1162/neco.2007.12-05-077 (2008).
41. Berret, B. *et al.* The inactivation principle: Mathematical solutions minimizing the absolute work and biological implications for the planning of arm movements. *PLoS Comput. Biol.* **4**, 1–25. https://doi.org/10.1371/journal.pcbi.1000194 (2008).
42. Tieck, J. C. V. *et al.* Learning continuous muscle control for a multi-joint arm by extending proximal policy optimization with a liquid state machine. In *International Conference on Artificial Neural Networks*, 211–221 (Springer, 2018).
43. Kidziński, Ł. *et al.* Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In Escalera, S. & Weimer, M. (eds.) *The NIPS '17 Competition: Building Intelligent Systems*, 121–153 (Springer International Publishing, Cham, 2018).
44. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psychol.* **47**, 381–391 (1954).
45. MacKenzie, I. S. A note on the information-theoretic basis for Fitts' law. *J. Mot. Behav.* **21**, 323–330. https://doi.org/10.1080/00222895.1989.10735486 (1989).
46. Crossman, E. R. F. W. & Goodeve, P. J. Feedback control of hand-movement and Fitts' law. *Q. J. Exp. Psychol.* **35**, 251–278 (1983).
47. Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E. & Keith Smith, J. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychol. Rev.* **95**, 340 (1988).
48. Matthews, P. Relationship of firing intervals of human motor units to the trajectory of post-spike after-hyperpolarization and synaptic noise. *J. Physiol.* **492**, 597–628 (1996).
49. Takeda, M. *et al.* Explanation of Fitts-law in reaching movement based on human arm dynamics. *Sci. Rep.* **9**, 19804. https://doi.org/10.1038/s41598-019-56016-7 (2019).
50. Lacquaniti, F., Terzuolo, C. & Viviani, P. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychol.* **54**, 115–130 (1983).
51. Pollick, F. E. & Sapiro, G. Constant affine velocity predicts the 13 power law of planar motion perception and generation. *Vis. Res.* **37**, 347–353. https://doi.org/10.1016/S0042-6989(96)00116-2 (1997).
52. Viviani, P. & Schneider, R. A developmental study of the relationship between geometry and kinematics in drawing movements. *J. Exp. Psychol. Hum. Percept. Perform.* **17**(1), 198–218 (1991).
53. de'Sperati, C. & Viviani, P. The relationship between curvature and velocity in two-dimensional smooth pursuit eye movements. *J. Neurosci.* **17**, 3932–3945 (1997).
54. Viviani, P. & Mounoud, P. Perceptuomotor compatibility in pursuit tracking of two-dimensional movements. *J. Mot. Behav.* **22**, 407–443. https://doi.org/10.1080/00222895.1990.10735521 (1990).
55. Viviani, P., Baud-Bovy, G. & Redolfi, M. Perceiving and tracking kinesthetic stimuli: Further evidence of motor-perceptual interactions. *J. Exp. Psychol. Hum. Percept. Perform.* **23**, 1232–1252. https://doi.org/10.1037//0096-1523.23.4.1232 (1997).
56. Hicheur, H., Vieilledent, S., Richardson, M., Flash, T. & Berthoz, A. Velocity and curvature in human locomotion along complex curved paths: A comparison with hand movements. *Exp. Brain Res.* **162**, 145–54. https://doi.org/10.1007/s00221-004-2122-8 (2005).
57. Schaal, S. & Sternad, D. Origins and violations of the 2/3 power law in rhythmic 3d movements. *Exp. Brain Res.* **136**, 60–72 (2001).
58. Karklinsky, M. & Flash, T. Timing of continuous motor imagery: The two-thirds power law originates in trajectory planning. *J. Neurophysiol.* **113**, 2490–2499. https://doi.org/10.1152/jn.00421.2014 (2015).
59. Todorov, E. & Jordan, M. I. Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *J. Neurophysiol.* **80**, 696–714 (1998).
60. Gribble, P. & Ostry, D. Origins of the power law relation between movement velocity and curvature: Modeling the effects of muscle mechanics and limb dynamics. *J. Neurophysiol.* **76**, 2853–2860. https://doi.org/10.1152/jn.1996.76.5.2853 (1996).
61. Schwartz, A. Direct cortical representation of drawing. *Science* **265**, 540–542. https://doi.org/10.1126/science.8036499 (1994).
62. Flash, T. & Handzel, A. Affine differential geometry analysis of human arm movements. *Biol. Cybern.* **96**, 577–601. https://doi.org/10.1007/s00422-007-0145-5 (2007).
63. Morasso, P. Spatial control of arm movements. *Exp. Brain Res.* **42**, 223–227 (1981).

64. Abend, W., Bizzi, E. & Morasso, P. Human arm trajectory formation. *Brain J. Neurol.* **105**, 331–348 (1982).
65. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences* (Academic Press, 2013).
66. Hausknecht, M. & Stone, P. Deep Recurrent q-learning for Partially Observable MDPS. arXiv:1507.06527 (2015).
67. Liu, J., Gu, X. & Liu, S. Reinforcement learning with world model. *Adapt. Learn. Optim.* **1908**, 11494 (2020).
68. Igl, M., Zintgraf, L., Le, T. A., Wood, F. & Whiteson, S. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, 2117–2126 (PMLR, 2018).
69. Jiang, Y., Van Wouwe, T., De Groote, F. & Liu, C. K. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Trans Graph (TOG)* **38**, 1–12 (2019).
70. Ikkala, A. & Hämäläinen, P. Converting biomechanical models from opensim to Mujoco. arXiv:2006.10618 (2020).
71. van der Helm, F. C. T. & Rozendaal, L. A. Musculoskeletal systems with intrinsic and proprioceptive feedback. in *Biomechanics and Neural Control of Posture and Movement* (eds Winters, J. M. & Crago, P. E.) 164–174 (Springer New York, NY, 2000).
72. Mnih, V. *et al.* Asynchronous methods for deep reinforcement learning. arXiv:1602.01783 (2016).
73. Eysenbach, B., Gupta, A., Ibarz, J. & Levine, S. Diversity is all you need: learning skills without a reward function. arXiv:1802.06070 (2018).
74. Ahmed, Z., Le Roux, N., Norouzi, M. & Schuurmans, D. Understanding the impact of entropy on policy optimization. In Chaudhuri, K. & Salakhutdinov, R. (eds.) *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, 151–160 (PMLR, 2019).
75. Fox, R., Pakman, A. & Tishby, N. Taming the noise in reinforcement learning via soft updates. arXiv:1512.08562 (2017).
76. Vieillard, N. *et al.* Leverage the average: an analysis of KL regularization in RL. arXiv:2003.14089 (2021).
77. Eysenbach, B. & Levine, S. If maxent RL is the answer, what is the question? arXiv:1910.01913 (2019).
78. Eysenbach, B. & Levine, S. Maximum entropy RL (provably) solves some robust RL problems. arXiv:2103.06257 (2021).
79. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. & Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, 1861–1870 (PMLR, 2018).
80. Abdolmaleki, A. *et al.* Maximum a posteriori policy optimisation. arXiv:1806.06920 (2018).
81. Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. arXiv:1805.00909 (2018).
82. Vulkan, N. An economist's perspective on probability matching. *J. Econ. Surv.* **14**, 101–118. https://doi.org/10.1111/1467-6419.00106 (2000).
83. Grünwald, P. D. & Dawid, A. P. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *Ann. Stat.* **32**, 1367–1433. https://doi.org/10.1214/009053604000000553 (2004).
84. Weir, M. W. Developmental changes in problem-solving strategies. *Psychol. Rev.* **71**, 473 (1964).
85. Gallistel, C. R. *The Organization of Learning* (The MIT Press, 1990).
86. Peng, X. B., Abbeel, P., Levine, S. & van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph* **37**, 1–14. https://doi.org/10.1145/3197517.3201311 (2018).
87. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. arXiv:1707.06347 (2017).
88. Fujimoto, S., van Hoof, H. & Meger, D. Addressing function approximation error in Actor-critic methods. arXiv:1802.09477 (2018).
89. Hasselt, H. V. Double q-learning. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S. & Culotta, A. (eds.) *Advances in Neural Information Processing Systems 23*, 2613–2621 (Curran Associates, Inc., 2010).
90. Kingma, D. P. & Ba, J. A. A method for stochastic optimization. arXiv:1412.6980 (2014).
91. McCloskey, M. & Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In Bower, G. H. (ed.) *Psychology of Learning and Motivation*, vol. 24, 109 – 165, https://doi.org/10.1016/S0079-7421(08)60536-8 (Academic Press, 1989).
92. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 807–814 (Omnipress, Madison, WI, USA, 2010).
93. Guadarrama, S. *et al.* TF-Agents: A library for reinforcement learning in tensorflow. https://github.com/tensorflow/agents (2018).
94. Guiard, Y. The problem of consistency in the design of Fitts' law experiments: Consider either target distance and width or movement form and scale. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, 1809–1818, https://doi.org/10.1145/1518701.1518980 (Association for Computing Machinery, New York, NY, USA, 2009).
95. Fischer, F., Bachinski, M., Klar, M., Fleig, A. & Müller, J. Reinforcement learning control of a biomechanical model of the upper extremity (dataset). *Zenodo*. https://doi.org/10.5281/zenodo.

Acknowledgements

We would like to thank Aldo A. Faisal (Imperial College London) for his very helpful advice and comments on the manuscript.

Author contributions

F.F. and J.M. wrote the main manuscript text; M.B. adjusted the biomechanical model; F.F., M.B., and M.K. worked on the MuJoCo implementation and collected simulation data; F.F., M.K., and A.F. analyzed and validated the results; F.F. created the plots; F.F. conducted and J.M. supervised the research process. All authors reviewed the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-93760-1.

Correspondence and requests for materials should be addressed to F.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.