Shared Autonomy Locomotion Synthesis With a Virtual Powered Prosthetic Ankle

Balint K. Hodossy and Dario Farina, *Fellow, IEEE*

***Abstract*— Virtual environments provide a safe and accessible way to test innovative technologies for controlling wearable robotic devices. However, to simulate devices that support walking, such as powered prosthetic legs, it is not enough to model the hardware without its user. Predictive locomotion synthesizers can generate the movements of a virtual user, with whom the simulated device can be trained or evaluated. We implemented a Deep Reinforcement Learning based motion controller in the MuJoCo physics engine, where autonomy over the humanoid model was shared between the simulated user and the control policy of an active prosthesis. Despite not optimising the controller to match experimental dynamics, realistic torque profiles and ground reaction force curves were produced by the agent. A data-driven and continuous representation of user intent was used to simulate a Human Machine Interface that controlled a transtibial prosthesis in a non-steady state walking setting. The continuous intent representation was shown to mitigate the need for compensatory gait patterns from their virtual users and halve the rate of tripping. Co-adaptation was identified as a potential challenge for training human-in-the-loop prosthesis control policies. The proposed framework outlines a way to explore the complex design space of robot-assisted gait, promoting the transfer of the next generation of intent driven controllers from the lab to real-life scenarios.**

***Index Terms*— AI and machine learning, human–robot interaction, locomotion synthesis, simulation, rehabilitation robotics.**

I. Introduction

COMMERCIALLY available powered lower limb Prosthetics and Orthotics (P&O) have yet to realise their potential impact [1], [2]. Price, weight and user perception are all key factors in determining whether these devices are a worthwhile intervention for their users [3]. Moreover, appropriate strategies are needed to synchronise the upcoming generated motion with the user's motor intent. Many wearable robotic devices are limited by unintuitive control interfaces, low number of degrees of freedom (DoFs) control and reliance on signals acquired as a result of a movement already in progress [2], [5]. This is especially true for non-steady-state locomotion (e.g. turning, changing walking speed or starting stair ascension) and for partial assistance systems [2]. The design, development and evaluation of novel control strategies are hindered by limited access to hardware and participants, as well as by the risks inherent to testing incomplete device controllers. The physical test-beds and frameworks, which can be used for validated, reproducible and safe tests require complex and unique equipment [6]. Time-efficient iterations on hardware and controller designs can be facilitated by device emulation hardware [7]. However, these may introduce restrictions on the range of test environments and locomotion tasks due to the mobility constraints of the emulation platform.

In the context of upper limb device design, virtual environments are a well established approach to alleviate the aforementioned issues [8]. However, there are additional challenges when this approach is applied to locomotion tasks. To provide the kinematic and kinetic context that is necessary for the operation of a simulated lower limb device, the user's movements need to be synthesised as well. It is insufficient to solely use inverse dynamics for this purpose. Indeed, motion trajectories reconstructed from experiments quickly become inaccurate with the occurrence of forces from a virtual device, which leads to instability if deviations from the prerecorded states are permitted. Instead, predictive forward simulations can generate stable walking policies, modelling key aspects of the simulated user's motor control [9], for example, the ability to:

- React and recover from disturbances and to take advantage of the assistance from the wearable device.
- Generate movement conditioned on a modelled gait pathology, and produce or learn compensatory movements.
- Perform long and short-term motion planning during non-steady-state locomotion tasks.

There are various methods for constructing gait policies with these characteristics, such as heuristic reflex-rule-based systems [10], trajectory optimisation [11], [12], evolutionary strategies [13], supervised learning [14], [15] and reinforcement learning [16], [17], [18]. Summaries on related work are available from the neuromechanical [9], [19] and computer graphics perspectives [20], [21]. Deep Reinforcement Learning (DRL) in particular has led to solutions that generalise well to multiple locomotion tasks simultaneously with realistic motion [16], while also reproducing key biomechanical aspects of

Manuscript received 11 June 2023; revised 16 October 2023 and 16 November 2023; accepted 20 November 2023. Date of publication 28 November 2023; date of current version 7 December 2023. This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/S02249X/1 for the Centre for Doctoral Training in Prosthetics and Orthotics and the Natural BionicS Initiative under Grant 810346. *(Corresponding author: Dario Farina.)*

The authors are with the Department of Bioengineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: d.farina@imperial.ac.uk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2023.3336713, provided by the authors.
Digital Object Identifier 10.1109/TNSRE.2023.3336713

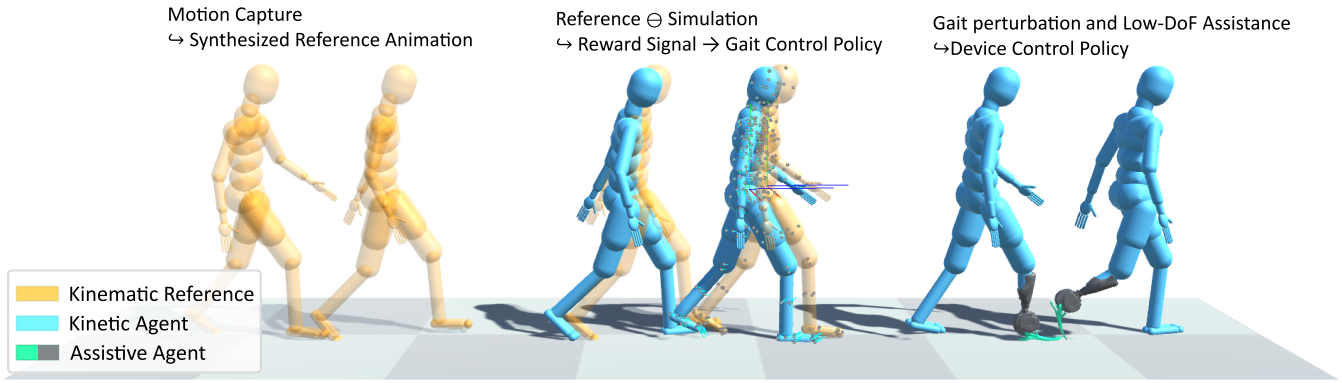

Fig. 1. Developing a virtual lower limb device testbed. A predictive walking policy was learnt, incorporating information from experimental data-driven references. A gait pathology was then simulated. Addressing the induced biomechanical deficit was the goal of the device's agent. Additional conditions and tasks can be introduced to assess the versatility of the controllers. 3D model from the Open Source Limb project used to represent the prosthesis [4].

gait [18]. In contrast to model predictive control methods, it requires an additional learning phase. However, once trained, DRL can perform inference using less resources than collocation methods. Furthermore, there have been promising examples of sim2real capability using DRL [22], [23], [24], an essential property when simulating robotics [25]. If a kinematic reference animation is available, learning time can be significantly reduced by using a motion tracking approach [16], [17], [18]. Musculoskeletal actuation may be essential when investigating orthotic devices [26]. However, direct torque actuation is multiple times faster to train in comparison [18], and may be appropriate for simulating users of prosthetic systems, which do not directly control the same DoFs as the device.

Simulating human locomotion along with the behaviour of active assistive devices is a promising way to introduce an inner design loop to the development of controllers. Signal modalities, model hyperparameters and calibration algorithms can be first investigated this way, before applying the gained insights to the real-life robot-human system. One benefit is the potential to use access to hardware, end-user subjects and testing equipment more efficiently. Alternatively, parameter ranges determined in simulations could be used as starting points when fitting models to real users through adaptive [27] or manual methods [28]. Finally, the device calibration and parameter tuning process, which is one of the most challenging aspects of current powered P&O [1], can be improved with insights from simulation.

Simulated gait policies have been previously proposed as suitable test environments for lower limb assistive devices [9], [29], and have been applied to orthotic [30], [31] and prosthetic [11], [18], [32] systems. However, existing examples primarily use passive devices during mostly steady-state locomotion tasks. Here, we explored a model of a Proportional Derivative (PD)-controlled unilateral powered ankle prosthesis, while its simulated user walked on a level surface with frequent turns and stops. Motion tracking DRL gait policies were learned based on reference kinematic animations generated with motion-matching [17], [33], which provided a data-driven representation not only of the user's desired movements, but also of their high-level abstract intent. Following this phase, agency over the user's below knee control signals on one side was assigned to a second control policy to mimic a transtibial prosthesis. The device control policy was also trained with DRL. We compared a prosthesis controller that only had inputs from implicit sensors [2], [34] with one that additionally received a representation of the simulated user's locomotion intent. This low-dimensional abstract intent serves as a surrogate signal of a neural interface in a late-fusion setting. In addition to reinforcement learning-based controllers tuned automatically using Proximal Policy Optimisation (PPO), we also reimplemented a Finite-State Machine (FSM) device controller [28]. The gait phase estimated by this rule-based system progressed naturally, indicating that the kinematic and kinetic context of the virtual device is plausible. The parameters of the FSM were tuned manually through a slider interface. An overview of the main stages of constructing the device controller testbed are illustrated in Figure 1.

Main Contributions: In summary, the following are the main contributions of this paper:

- Modelling of prosthesis use in a non-steady-state locomotion scenario involving stops and turns, with a virtual user that can react to perturbations or intent changes.
- Implementation of a motion tracking gait controller from the field of character animation, and demonstration that it generates a plausible dynamic context for a transtibial prosthesis.
- Modelling the co-adaptation setting where both the user policy and device controller could be learned through DRL simultaneously, but as separate agents.
- Demonstration that the desired horizontal walking velocity is a suitable control signal for a prosthetic ankle and reduces the need for compensatory gait patterns, compared to an ankle without inputs controlled by the user.

II. Materials and Methods

The following section details the key subsystems used to simulate the dynamic human-prosthesis model. First, the motion-matching approach for generating kinematic reference

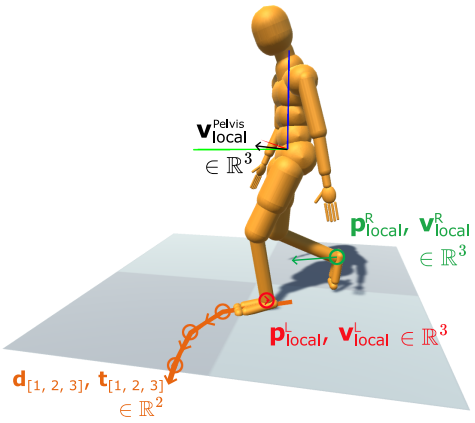

Fig. 2. Illustration of the features used in the nearest neighbour search for the motion-matching system, and the local orientation frame they were quantified in. The orange trajectory consists of the position and normalised forward direction of the pelvis reference frame, projected on the horizontal plane at 1/3, 2/3 and 1s in the future. This curve represents the upcoming walking path in the motion capture data set.

motions is explained. Second, the DRL environment for learning gait without a prosthesis is presented, followed by the changes introduced to model prosthesis use.

A. Locomotion Intent Synthesis

The high-level intent driving the motion synthesis was characterised as the desired horizontal two-dimensional velocity vector of the pelvis, ranging in amplitude from 0 to a moderate speed of 1.4 $\frac{\text{m}}{\text{s}}$ [35]. The process for generating the desired walking velocity during gait policy learning will be detailed in Section II-B.

To convert this abstract locomotive goal to specific mid-level motion trajectories, the high-level intent was used to extrapolate a 1 s long, critically damped walking path, which was used to drive a motion-matching system [33]. Motion-matching allows efficient usage of limited motion capture data sets by generating transitions between disparate parts of them. These transitions are key in augmenting the training data to include a diverse range of non-steady-state walking including turns and stops. The motion capture recordings of [36] were used as the data set in this study. They consist of unsegmented clips of diverse locomotion. A "meta-data" feature vector was calculated for each frame of these recordings, as described in [17]. This vector consists of the following elements (illustrated in Figure 2):

- Velocity of the pelvis $\left(\mathbf{v}^{Pelvis}_{local} \in \mathbb{R}^3\right)$.
- Position and velocity of the feet $\left(\mathbf{p}^{[L,R]}_{local}, \mathbf{v}^{[L,R]}_{local} \in \mathbb{R}^3\right)$.
- Position and normalised forward direction of the pelvis reference frame projected on the horizontal plane at 1/3, 2/3 and 1 second in the future $\left(\mathbf{t}_{[1,2,3]}, \mathbf{d}_{[1,2,3]} \in \mathbb{R}^2\right)$. These are referred to as the trajectory and direction features respectively.

This gives a total of 27 dimensions for motion-matching, which were all normalised to zero mean and unit standard deviation. These features were described from a semi-local frame of reference, which was located at the pelvis, with one axis aligned with the global vertical and another with pelvis forward directions and was assigned zero global velocity relative to the ground. Therefore, only the directions of velocities described in this local frame were influenced by pelvis kinematics, but not their magnitudes. Motion-matching velocity features were estimated using a first order Savitzky-Golay filter. To adjust the relative importance of the elements in this feature vector they can be scaled by a set of weights. These weights determine the trade-off between responsiveness to matching the desired trajectory and the smoothness of motion. A factor of 6 was used for velocity, 3 for position, 4 for trajectory and 2 for direction features, set through manual tuning.

During every 10^{th} frame of motion synthesis, the meta-data vector of the current motion frame was collected. Then, its walking trajectory and direction features were replaced by the critically damped walking path, as determined by the artificial high-level intent. This modified vector was then compared with all other meta-data vectors in the data set, and the kinematic motion continued from the frame that corresponded to the nearest neighbour of the current feature vector. Due to the inclusion of a single locomotion style and task (level ground walking with turns and stops), the data set was small enough that parallelisation or KD-tree based methods yielded no performance gains over a linear search for the nearest neighbour match [33]. "Inertialization" was applied on the kinematic animation targets to smooth discontinuities [37], a blending technique inspired by zero-jerk trajectory control principles [38]. The walking generated by motion-matching is kinematic in the sense that it cannot describe interactions between the humanoid, the ground and other elements in the scene (such as a virtual prosthesis). Indeed, tracking the output of motion-matching with a pure PD controller will almost immediately lead to tripping and falling. However, this movement provides a plausible first guess for target poses of a dynamically simulated humanoid controlled by a PD-DRL hybrid system, described below and illustrated in Figure 3. This combination of motion-matching with DRL was first proposed in [17].

B. Gait Policy

The "CMU humanoid" model described in [39] was simulated in the MuJoCo physics engine [40] at 125 Hz. The torso, lower back, neck, shoulder and hip joints were replaced in the model with spherical joints instead of serial hinge joints. A 2 DoFs ankle model was used, and all joints distal from the elbow were removed. This resulted in a total of 43 DoFs. The clavicle body segments, modelled as 2 DoF joints were fully passive, leading to 39 DoFs to be controlled by the gait policy. The agent received no external forces or torques aiding its balance. Scene construction, visualisation and task logic was performed in the Unity engine, using the ML-Agents package for communication with a learning framework implemented in PyTorch [41].

1) Reinforcement Learning: The gait control problem can be represented as a Markov decision process. Then, the probability of transitioning to a specific physics simulation state at the next time step is wholly determined by the current state ($\mathbf{s}_t$), the system dynamics and the actions ($\mathbf{a}_t$) taken by

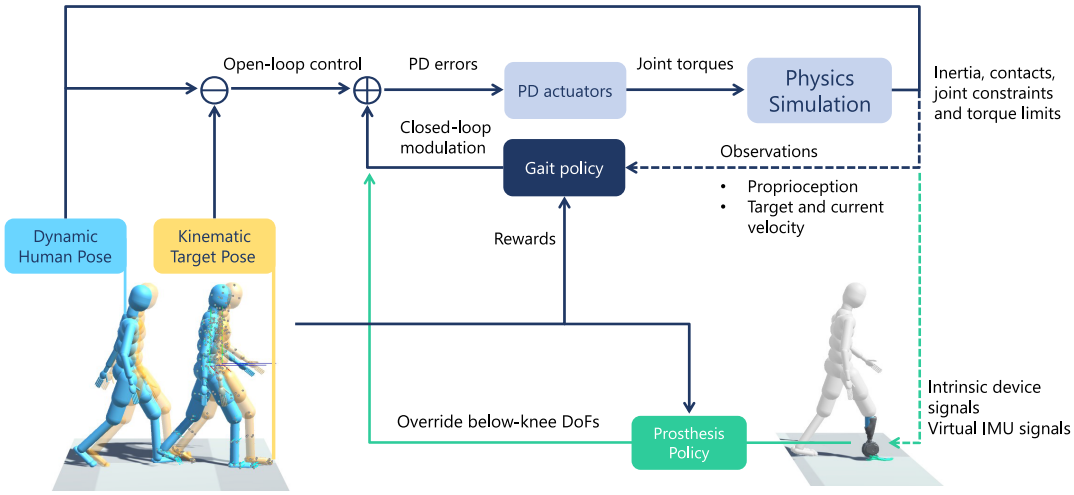

Fig. 3. The control problem solved using DRL. The target kinematic state from motion-matching is modulated by the gait policy, which receives observations from the simulation state. These targets are then converted into joint torques through stable PD. The simulation and reference states are then updated, and the process repeats. In prosthesis use conditions, below knee DoFs are controlled by a separate prosthesis policy, which receives its observations. Both gait and prosthesis policies are trained using the same reward signal.

the decision-making agents in the system. If we don't assume *a priori* what the ideal actions are, we can instead quantify desirable properties for the controlled system with a reward function $r_t(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{a}_t)$. DRL is a family of optimisation algorithms that use past experiences of states, actions and rewards to maximise the weighted sum of rewards gained within an episode of learning. The "deep" part of DRL refers to the use of deep learning function estimators in the agent's behaviour. In PPO [42], this is achieved through the use of an actor and critic system. The critic is iteratively updated to predict the future weighted sum of rewards that will be collected after taking a given action at a given state. The actor maps states to actions through a stochastic policy. The policy ($\pi(\mathbf{a}_t|\mathbf{o}_t)$) takes a subset of the state vector (called observations $\mathbf{o}_t$) as inputs, and outputs the probability distribution for each possible action to take. The parameters of the policy determine how likely it is to select an action conditioned on the state. When an action results in more/less reward than anticipated by the critic for a given observation, these parameters are adjusted by gradient ascent to make that action more/less likely in the future. PPO was chosen as the learning algorithm due to its robustness with respect to hyperparameter choice, its suitability for continuous action spaces and previous successes in applying it for locomotion [17], [18], [43].

The policy trained by PPO was a feedforward neural network with 3 hidden layers, 512 units each with swish activation [44]. The same architecture was used for the critic network. A visual graph of the gait policy's role in the simulation is shown in Figure 3. Training was performed on 6 parallel environments running on the same system. Further hyperparameters of the learning environment are detailed in the configuration file included in the supplementary materials.

2) Observations, Actions, Rewards: A vector of observations was sampled from the simulation environment at each control step queried at 60 Hz, based on the feature set used in [17]. Inside this observation vector, 6-dimensional kinematic information (position and linear velocity) was collected of the following body segments from the pelvis' coordinate frame:

- Left and right feet (12 dimensions)
- Left and right forearms (12 dimensions)
- Upper back (6 dimensions)
- Head (6 dimensions)

The difference between the desired and actual kinematics of these body segments was also provided as an observation (36 dimensions), as it was previously found to speed up the learning process [17]. This was concatenated with the desired and actual centre of mass velocity, as well as their difference (3×3 dimensions). The high-level walking intent in the form of the desired horizontal walking velocity used in the motion matching process was provided, along with its difference from the actual horizontal centre of mass velocity (2×2 dimensions). Lastly, the agent's previous actions (described below) were also provided as observations (39 dimensions). In total, the observation vector was 124 dimensional. This feature set [17] is different from many other implementations due to a lack of reliance on a phase variable [29], [43], [45], which is not straightforward to define during continuous but non-steady-state motions. The observation vector was concatenated with the previous decision step's observations before using them as inputs for the policy network.

Given this input, the policy outputs an action vector at each control step. Exponential smoothing was applied to this output with a smoothing factor of 0.9 [17], and was assumed to be constant between control steps. The actions modulated the open-loop poses of the motion-matched animation. They were not interpreted as velocity targets [17] or position targets [43] from which PD errors were then calculated. Instead, the action vector was added directly to the PD error vector calculated as the difference between the open-loop reference and the current pose. While this is equivalent to pose modulation in the case of hinge joints, it reduces the amount of computation necessary for spherical joints. This is because their error signal has a smaller dimensionality (3D) than their quaternion-based (4D) positional descriptions. Torques generated by Stable PD [46] actuators were limited under 190 N m, constraining the output to reasonable values for gait [47] and improving simulation

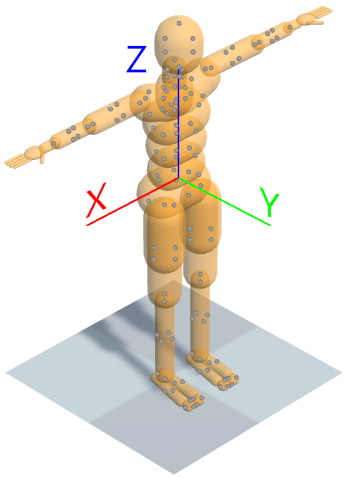

Fig. 4. The local reference frame and the bounding box surface points used in the reward calculation. Segments distal to the forearm were not included in the reward calculation.

stability. A position gain of 510 $\frac{\text{N·m}}{\text{rad}}$ and a velocity gain of 5.2 $\frac{\text{N·m·s}}{\text{rad}}$ was used, roughly approximating the human upper limit of stiffness in the knee and hip [48], [49]. By modulating the error signal the agent can mimic other stiffness and damping parameters. However, particularly in early stages of learning, a high stiffness starting point was found to be helpful in learning to avoid collapsing. A copy of the simulated humanoid followed the motion-matching animation, enforced through equality constraints on position and orientation. In this way, joint-space state and higher-order kinematics could then be queried from this "puppeteered" character for reference observations and actions.

The reward collected by the agent was calculated based on the kinematic differences between the reference and simulated body, consisting of four terms summed together (Equation 1).

$$r_t = e_{\text{fall}} \left(r_p + r_v + r_{local} + r_{v_{CM}} \right) \tag{1}$$

The first two were calculated using the "surface point" method introduced in [17].

The *position reward* was defined as the distance between the body segments of the dynamic humanoid and its kinematic reference (Equation 2).

$$r_p = \exp\left(\frac{-7.3}{N_{\text{segments}}} \sum_{i=1}^{6} \sum_{j=1}^{N_{\text{segments}}} \left\| \hat{\mathbf{p}}_{ij} - \mathbf{p}_{ij} \right\|_2^2 \right) \tag{2}$$

where N_{segments} is the total number of tracked segments. $\hat{\mathbf{p}}_{ij}$ denotes the i^{th} centre point among the j^{th} reference segment's bounding box faces. The square distance is then calculated as the positional difference from the corresponding point on the dynamic humanoid ($\mathbf{p}_{ij}$).

A squared distance measure rather than the L2 norm was used in the position, velocity and local pose rewards, based on the implementation of [50], which led to faster learning during preliminary results. The surface points' positions were resolved in the pelvis' reference frame, illustrated on Figure 4. Like [17], the dynamic humanoid's local reference frame was also considered to originate from its pelvis, but its orientation matches the reference frame's. This implicitly penalises facing the wrong direction with the dynamic humanoid.

The second term, the *velocity reward*, was used to match the linear velocities of these points (Equation 3).

$$r_v = \exp\left(\frac{-1}{N_{\text{segments}}} \sum_{i=1}^{6} \sum_{j=1}^{N_{\text{segments}}} \left\| \hat{\mathbf{v}}_{ij} - \mathbf{v}_{ij} \right\|_2^2 \right) \tag{3}$$

The linear velocities of these points were also affected by the angular velocity of their parent segments. While the direction of the velocity vectors was also resolved in the local reference frame, the pelvis's global velocity was not subtracted from their values. This strengthens the requirement to match the overall velocity of the locomotion.

The local rotation of each segment with respect to its parent segment determines the third term in the reward (Equation 4).

$$r_{local} = \exp\left(\frac{-6.5}{N_{\text{segments}}} \sum_{j=1}^{N_{\text{segments}}} \left\| \hat{\mathbf{a}}_j \ominus \mathbf{a}_j \right\|_q^2 \right) \tag{4}$$

where $\left\| \hat{\mathbf{a}}_j \ominus \mathbf{a}_j \right\|_q$ is the angle magnitude of the angle-axis decomposition of the quaternion difference of the two rotations.

The last reward term was provided based on matching the velocities of the two centres of mass (Equation 5).

$$r_{v_{CM}} = \exp\left(- \left\| \hat{\mathbf{v}}_{CM} - \mathbf{v}_{CM} \right\|_2^2 \right) \tag{5}$$

Finally, these reward signals were combined with a scaling factor based on the difference of the vertical location of the two head segments [17] (Equation 6). This prioritises learning how to avoid falling and tripping first, before reward from motion tracking can be gained.

$$e_{\text{fall}} = \max\left(\min\left(1.3 - 1.4 \left\| \hat{\mathbf{h}}_{CM_{head}} - \mathbf{h}_{CM_{head}} \right\|_2, 1 \right), 0 \right) \tag{6}$$

This reward was collected at every time step t when the agent takes an action. If an episode lasts more than 15 s, or the agent's rewards in a step fell below a near-zero threshold, the episode was terminated, and the simulated agent was reinitialised to the reference's state. As the reference animation was not restarted, episodes started at different states of walking. This approach can be thought of as a simplified version of Exploring Starts [51], or Reference State Initialisation [43].

3) Locomotion Task: The learning process took place in an $8 \times 8\,\text{m}^2$ area. The motion-matching animation was generated independently of the dynamic simulation (i.e. the state and actions of the agent did not influence the animation). This kinematic animation moved between target locations, smoothly transitioning between turning and straight walking (see Figure 5).

When the character was closer than 0.8 m to a target location, a new one was generated that was at least 2.4 m away, sampled from a uniform distribution within the learning area. A wait period was introduced between target locations with a 30% probability, sampled uniformly from the range [6-9] s. During a wait period, the animation comes to a stop,

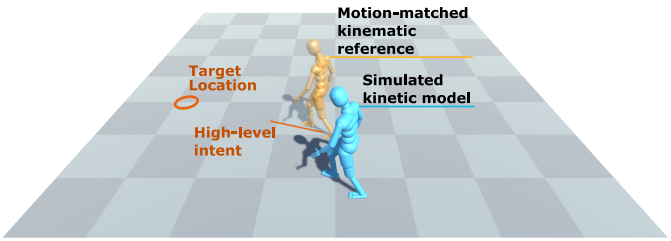

Fig. 5. Snapshot of the non-steady-state locomotion learning environment. The vector pointing from the reference to the target location is the high-level intent provided to the gait synthesis agent. Once a target location was reached, a new one was generated which may necessitate turning. Please refer to the video in the supplementary materials for demonstration of the environment in action.

and stands still with a desired horizontal velocity of 0 $\frac{\text{m}}{\text{s}}$, before receiving the next target location. Once the agent learnt to handle turns and the policy's performance converged to a static level ($\sim$10 million decision steps), additional perturbations were introduced every 1.2 s, in a uniformly sampled direction, applied for 0.24 s at a uniformly sampled body segment, with a force sampled uniformly from the range of [50-150] N. This type of perturbation can force the agent to explore recovery strategies in situations it would not encounter in its regular environment, but will once a change is introduced (such as a gait pathology) [22].

C. Virtual Device

The use of a transtibial prosthesis was modelled by reassigning control over below knee DoFs unilaterally to a separate agent [32], [52]. Dividing autonomy over the humanoid character, the locomotion and assistive agent then share the collaborative goal of restoring the character's original gait. This approach does not reproduce the behaviour or limitations of any one specific prosthetic device and assumes perfect weight matching and mechanical interface conditions, simplifying an already complex design space. Both flexion and adduction were controlled by the assistive agent, in contrast to more common flexion-only designs.

Similarly to the gait policy, PPO was used to train the prosthesis policy with the same network structure. However, only 128 units were used per hidden layer, and the assistive agent was queried only at 30 Hz.

1) Observations, Actions, Rewards: There is an important distinction to make in the types of observations used as inputs to the device policy when compared to the gait policy. The gait policy may use all privileged information available from the simulation. In contrast, the kinematic poses from the motion-matching process should not be used by the assistive agent. Instead, inputs to the device's control should be signals available through plausible instrumentation of real hardware. The simulated powered ankle device uses virtual accelerometer ($\in \mathbb{R}^3$) and gyroscope ($\in \mathbb{R}^3$) measurements from virtual Inertial Measurement Units (IMUs) placed on both the shank and foot, as well as virtual encoder signals for the ankle joint angle ($\in \mathbb{R}$) and angular velocity ($\in \mathbb{R}$) for a 14 dimensional observation vector in total.

One of the main benefits of simulated locomotion environments is the ability to test novel control schemes. Conditioning behaviour on high-level intent is one such improvement proposed for improving performance in non-steady-state and irregular environments [2]. Since biosignals commonly used in making this type of control schemes anticipatory and intuitive are not available in a straightforward way in simulation, an intermediate representation of the intent is necessary. In a real-life system, this representation would need to be estimated with human-machine interfaces (e.g., with electromyography). The intermediate signal could then be provided to the worn device as a compressed form of its user's intent. If the intermediate representation is chosen so that it is available during simulation, then it can be used as a surrogate input for controllers conditioned on high-level intent. As the desired walking velocity was used to drive the locomotion synthesis based on the experimentally determined relationship between movement and this abstract intent (enforced through motion matching), it is a natural representation to use for this purpose.

The actions of the device agent were interpreted as PD parameters of stiffness, damping and joint angles. No reference motion was used to influence this target for the device agent. Similarly to the gait policy, a high stiffness and damping starting point was used for the controller (35 $\frac{\text{N·m}}{\text{rad}}$ and 10 $\frac{\text{N·m·s}}{\text{rad}}$).

This horizontal velocity ($\in \mathbb{R}^2$) is a continuous high-level control signal and is conceptually between activity recognition and direct volitional control methods [2]. The prosthesis policy was conditioned on this intent by manipulating the policy network parameters through a hypernetwork [53].

D. Simulation Conditions

In early tests with simulated characters using a passive prosthesis, it was confirmed that robust locomotion policies can be learnt with passive devices. However, this was at the cost of compensatory movements [18]. Therefore, conditions were included where the gait policy was frozen after training without a prosthesis, then the device was introduced and only the prosthesis policy was trained. Once a gait policy was learned without a prosthesis (stopping training after $\sim$20 million decision steps), it was used as a starting point for these conditions after it. In these cases, compensatory changes to the gait were prevented, and the assistive agent had the responsibility to restore stable locomotion. When the gait policy was no longer trained, the action with the maximum likelihood was selected for it deterministically (the prosthesis policy remained stochastic). All conditions involved walking with turns and stops:

1) No prosthesis, perturbations added.
2) Active, non-intent-driven prosthesis, gait policy pre-trained.
3) Active, intent-driven prosthesis, gait policy pre-trained.
4) Active, non-intent-driven prosthesis, gait policy pre-trained and frozen.
5) Active, intent-driven prosthesis, gait policy pre-trained and frozen.

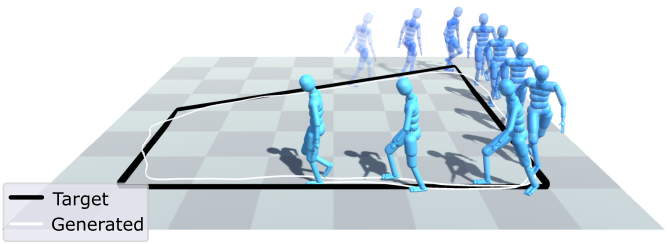

Fig. 6. Path tracking ability of the gait policy. "Stroboscopic" visual trail of the agent's poses made by following the learned gait policy recorded at 1 Hz. The agent was capable of performing continuous circuits without falling or deviating from the path.

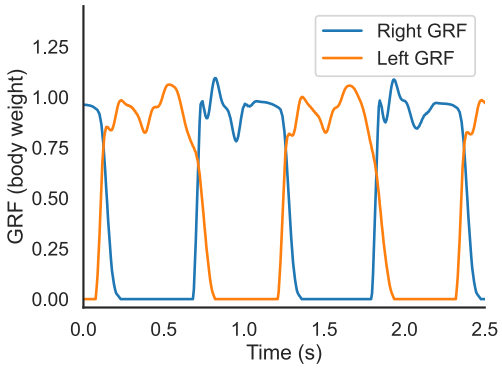

Fig. 7. Ground reaction force during walking as measured by virtual sensors placed on both feet of the character. Recorded at 1.3 $\frac{m}{s}$ straight walking.

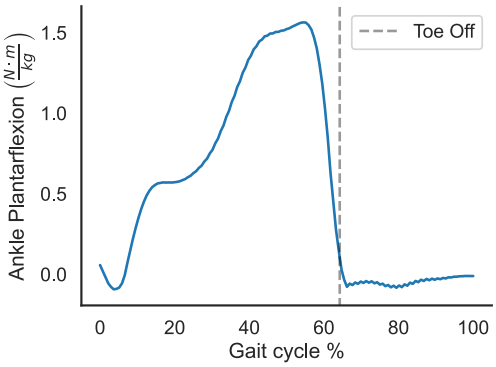

Fig. 8. Plantar flexion torque in one of the ankles during simulated gait. Early stance dorsiflexive and late stance plantar-flexive torque can be observed [47].

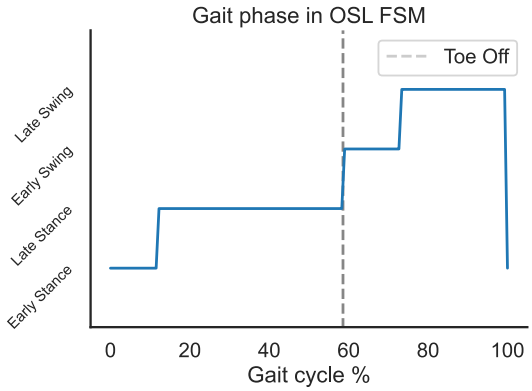

Fig. 9. Phase profile of the rule-based classifier used in a traditional Finite State Machine controller reproduced from [28], applied to the virtual user using an active prosthesis.

III. Results

A. Gait Policy

The learned policy generates robust walking between arbitrary landmarks, while being underactuated (no external forces applied to center of mass to help balancing). It was able to synthesise turning at various degrees and recover from losing balance (Figure 6). A video demonstration of the motion-matched reference and the synthesised dynamic gait is available in the supplementary materials.

While the DRL reward implemented only constrained kinematic properties of the motion synthesis, important characteristics of gait dynamics were reproduced by the final policy. This includes bimodal peaks of the ground reaction force (Figure 7) [47], and ankle dynamics (Figure 8).

The peaks of the ground reaction force were between 1 and 1.2 times the body weight, following normative data [54]. Moreover, there were contact forces (above 5% body weight) with both feet in the simulated walking during 21% of the gait cycle. This matched the expected ratio for double and single support within the gait cycle [55]. The peak ankle moment was 1.56 $\frac{N\ m}{Kg}$, which was also in accordance with experimental results [56].

To verify the suitability of the learned locomotion synthesizer to test prosthesis controllers, the Finite State Machine based controller of the Open-Source Leg [4], [28] was reproduced and tuned manually for the virtual user. This rule-based controller transitions between stance and swing states, as determined by the load on the prosthesis, and changes its behaviour based on knee velocity and ankle angle. Gait phase detection progressed naturally through early stance, late stance, early swing and late swing states (Figure 9), and the virtual device was able to restore walking. Transition to swing phases happened at 60% of the gait cycle (tracked between heel strikes), a value matching the expected timing in gait [55]. This FSM controller was primarily used as a validation of the locomotion synthesizer. Its natural progression through estimated gait phases indicated that the synthesized locomotion provided a reasonable kinematic and kinetic context to the virtual prosthesis. In further results the FSM controller was no longer used, instead a DRL policy controlled the prosthetic ankle.

Interestingly, in the condition where the gait policy was learned in parallel with the prosthesis policy, the additional observations of user intent provided no advantage to the joint performance of the human-prosthesis system (Figure 10).

In contrast, in the case where a pre-trained and frozen gait policy was used (hence no compensatory behaviour was adopted), conditioning the policy on user intent led to significant improvements in stability (Figure 11). In particular, the benefit was most apparent when the character transitioned between standing and walking. Without intent available, and with no compensatory movements, the assistive agent was unable to find an appropriate policy. This resulted in more frequent stumbling or tripping. When the intent was available, the prosthesis policy adjusted the control parameters

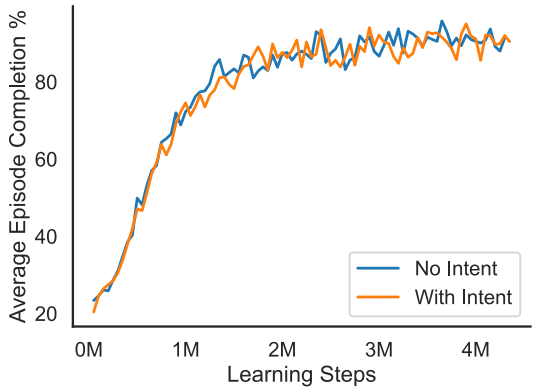

Fig. 10. Collaborative performance evolution of learning the (pretrained) gait policy and prosthesis policy simultaneously (conditions 2 and 3). Providing the intent as additional observation yields no improvement in these conditions.

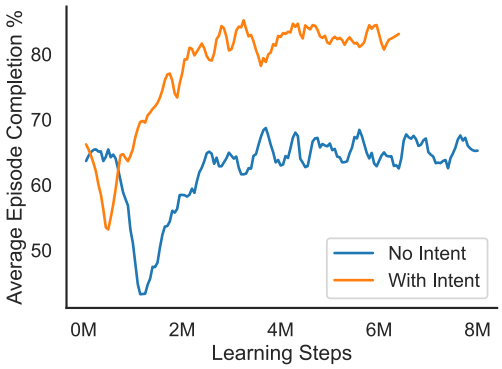

Fig. 11. Collaborative performance of user-prosthesis system, when the gait policy was pretrained without a prosthesis and could no longer learn. Only the prosthesis policy was adapted (Conditions 4 and 5). Note that the gait policy is deterministic here, as it no longer needs to explore by injecting noise in its actions. This yields the higher performance at the start compared to Figure 10.

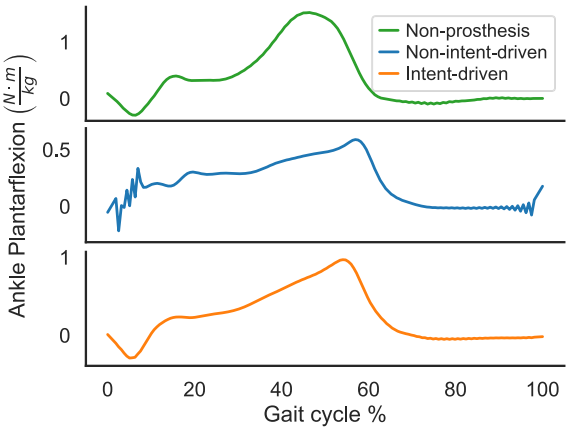

Fig. 12. Plantar flexion torque output of the ankle of the non-prosthesis-user (condition 1), the prosthesis with no intent input (condition 4), and intent-driven prosthesis (condition 5).

TABLE I
PERFORMANCE OF NON-PROSTHESIS (CONDITION 1), NON-INTENT-DRIVEN PROSTHESIS (CONDITION 4) AND INTENT DRIVEN PROSTHESIS (CONDITION 5) DURING STRAIGHT WALKING

Measure	No Prosthesis	No intent	Intent
Walking speed $[\frac{\mathrm{m}}{\mathrm{s}}]$	1.30	0.98	1.05
Peak ankle torque $[\frac{\mathrm{N\,m}}{\mathrm{kg}}]$	1.52	0.59	0.97
Pose error $[\mathrm{m}]$	0.026	0.045	0.043

immediately and continuously as soon as there was a change in intent, allowing the human-prosthesis team to transfer between locomotion modes (Figure 13). Without adaptation from the user, the mean time before the first trip event was 34 s and 12 s for the intent-driven and the non-intent-driven prosthesis respectively. If learning compensatory movements was allowed, the prosthesis user could walk for more than 3 minutes without tripping.

These two prosthesis conditions (4 and 5) were compared with the non-prosthesis-user (condition 1) through their average walking speeds, produced ankle torques and pose tracking error during steady-state straight walking (Table I). Both prosthesis conditions led to slower walking when their gait policy was prevented from learning compensatory motion. Furthermore, the intent-driven prosthesis exhibited torque profiles closer in magnitude to those of the original gait policy (Figure 12). This potentially arises from the non-intent-driven prosthesis policy being unsure whether it should prepare for a transition to standing or keep walking.

Pose tracking error was evaluated using by quantifying the average total Euclidean distance of the joint positions in Cartesian space and the position of the corresponding joint in the reference animation (Equation 7) [16]. The positions were relative to the location of the corresponding root joints (x^{root} and $\hat{x}^{\text{root}}$).

$$e_t = \frac{1}{N_{\text{joints}}} \sum_{j \in \text{joints}} \|(x_t^j - x_t^{\text{root}}) - (\hat{x}_t^j - \hat{x}_t^{\text{root}})\|_2 \quad (7)$$

IV. DISCUSSION

We have presented a human-prosthesis system model that provides key insights for its real-life equivalent. A high-level, abstract but continuous locomotive intent representation was shown to be helpful in non-cyclic gait scenarios, usually handled by classification processes [2]. By observation, most of the stability gained from relying on intent in conditions 4 and 5 was present when the agent was coming to a stop, or starting to walk from standing. The intent-driven policy was able to change its behaviour when the virtual user's goal changed, whereas the policy relying only on intrinsic sensing could react to movement already underway, leading to stumbling and trips (Figure 13). Sharing information through a limited channel is a known way of stabilising and improving performance in multi-agent learning scenarios [57], and intent estimation is an experimentally plausible approach to represent this.

However, it is important to note that adaptive controllers, such as those learned through DRL, have challenging learning dynamics when collaborating with a second non-static agent (conditions 2 and 3). Since the human has significantly more agency over managing the gait, exploration in assistive policies

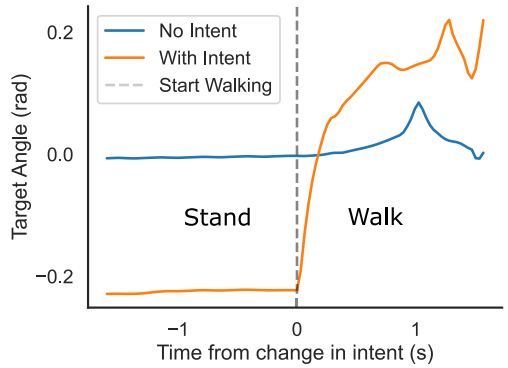

Fig. 13. Trajectory of the equilibrium ankle angle in the PD control parameters, set by the intent-driven (condition 5) and non-intent-driven (condition 4) systems when there's a step change between the virtual user's intent (going from standing to walking).

can hinder the gait policy's ability to exploit its competence, leading to lower the overall performance. This is apparent, for example, in the initial dip in the reward shown in Figure 11, where the gait policy was frozen. In a co-adaptation setting, this can encourage the prosthesis policy to act "lazily" [58], and let the human to learn safer, compensatory strategies. We believe this leads to an under-utilisation of the information available to the device policy, which explains the lack of difference when providing intent to the device (Figure 10). Indeed, the peak plantar flexion torque from the prosthesis in condition 3 is 0.5 $\frac{\text{N m}}{\text{kg}}$, half as much as in condition 5 where no compensation from the user is available. This type of situation may arise as well when applying reinforcement learning methods with real users who cannot risk falling over, who will also adapt their movements, potentially leading sub-optimal prosthesis policies. Through robust simulation environments, exploration can be enforced that discourages "lazy" policies (e.g., by disabling or slowing virtual user adaptation). Control parameters identified this way may provide starting points during fine tuning with real life users, which may avoid control strategies that do not fully take part in the shared autonomy of the movement they are supposed to assist. Although DRL was used to learn the device policy as well, the locomotion synthesizer could be combined with other, non-deep-learning based controllers [59].

During the development of the environment, sources of instability, errors in the implementation and suitable parameter ranges were identified for the prosthesis controller. Solving these issues first in the context of the synthesised locomotion partially mitigates the risk discovering these experimentally, and could contribute to using laboratory, equipment, and most importantly participant time more efficiently. In addition to the DRL device policy, reimplementing an FSM controller served as a proof-of-concept for testing rule-based policies in simulation, and practising or evaluating their tuning process.

The construction of the learning environments was considerably accelerated through the use of Unity, a development platform designed for efficient editing of virtual scenes. Its potential use case to model biomechanics and robotics has been recognised before, with the main criticism being towards the built-in physics engine's prioritisation of performance over accuracy [25]. Thanks to the recently released Unity plugin of the MuJoCo physics engine, it is possible to benefit from streamlined design and visualisation tools of Unity and still simulate biomechanically validated interactions.

The presented actuation scheme, managed by the DRL policy originated from the computer graphics field [17]. Despite this, there are rough parallels with current neuromechanical theories of human motor control. The controller consists of an open-loop control signal assembled from motion templates, which was then modulated based on signals of abstracted proprioception and high-level goal observations. This is reminiscent of the concept of Central Pattern Generator-based movement [60]. The motion-tracking and torque actuation-based method allowed for fast training of policies on limited hardware, while still producing plausible dynamics. For the purposes of simulating prosthesis use, torque actuation may be sufficient to provide the necessary context for the device. However, in partially over-actuated systems, like exoskeletons, realistic models of joint mechanics and actuation are more important, which is likely to involve musculoskeletal actuation [30]. Lastly, an important consideration is the simulation of the gait pathology. Prosthesis use can be characterised by various patterns, such as gait asymmetry or specific compensations such as circumduction. These could be closely replicated by tracking patient motion instead of non-prosthesis user data. However, this could be counter-productive, as then the virtual user would actively resist assistance from the prosthesis that would mitigate these gait patterns, just to better imitate its reference. It is more appropriate to motivate the agent to match non-compensatory movements, and constrain it mechanically (e.g., by limiting joint power, adding extra weight) or "physiologically" (e.g., by penalising joint load to simulate pain, or introducing fatigue mechanisms) in a way that the expected gait patterns emerge on their own. The scope of this study did not include a detailed comparison with experimental prosthesis use; however, this is a key aspect to be investigated in future work.

MuJoCo in particular is an attractive physics engine to use for future work due its efficient computation, which extends to muscle modelling [61], [62]. Furthermore, due to its flexible collision constraint configuration [40], surfaces of different materials and compliance can be easily simulated. This can be advantageous when trialling cushioned heels, or different types of challenging terrains.

Other key additions to this work would include other environments mimicking key activities of daily living, such as climbing stairs, tackling rough terrain or navigating crowds. The model of the prosthesis could also be extended. Key properties of the real socket-limb interface, such as pistoning, could be modelled to better capture the dynamics of gait during prosthesis use. This could reveal stress/strain relationships, which the optimisation process could take into account. The impact of aspects such as the number of DoFs, their range of motion, stiffness and power could also be further investigated to inform the design of new devices. The observations used as inputs to the policy are related to biological signals associated with human balancing (e.g., otholiths sensing linear acceleration of the head [63]). Further biologically inspired

input signals (e.g., the Golgi-tendon-like observations used in [29]) could not only improve artificial locomotion policies, but have implications on their role in human motion learning. Similarly, more nuanced multi-agent learning schemes should be explored to robustly model short- and long-term coadaptation. Lastly, augmenting the reference motion synthesizer with a more diverse motion capture data set could have positive effects on performance. Motion-matching is prone to reuse only segments of its database, therefore not all reference gait cycles will be equally represented, which may bias the policy. More complex reference motion synthesizer, such as neural state machines [64] could induce a more diverse training set. Alternatively, other DRL controllers that do not rely on synchronised reference motion in the first place could be applied [16].

There are various benefits to using DRL methods for finding the gait policy. While there is a computational overhead associated with training a DRL policy, once trained they are cheap to evaluate. A single policy may be trained to be robust for a range of virtual users and devices [29], [45], [65]. Locomotion agents with differences in weight, height and other parameters could be generated to evaluate devices and their controllers on a diverse virtual user population. Furthermore, DRL strategies can mimic various walking styles with a given humanoid model [43]. The function approximators commonly used in these methods can also establish connections between different representations of intent, sensory observations and the control policy. This is possible even if they are not directly accounted for in the cost or reward function, unlike trajectory optimisation methods.

The locomotion synthesis framework created and used in this study to train the gait policy was refactored, documented and released as an open-source project [66]. Additional features, such as the prosthesis environments updated for the latest package version along with more diverse walking environments (e.g., rough terrain and stairs) are planned additions for the future.

V. Conclusion

A gait policy learning environment was built through the combination of an accurate physics engine, a kinematic motion synthesizer and an accessible DRL framework. Autonomy over the simulated human's movement was shared between the gait policy and a second controller that operated a model of a unilateral transtibial prosthesis, forming a representative virtual test platform for wearable robotic devices. Controllers were trained and evaluated in a non-steady-state locomotion scenario involving walking, standing and turning. A continuous high-level intent representation was shown to be a useful control input, provided that compensatory gait patterns from the locomotion agent do not prevent the assistive device to capitalise on the additional information.

Human locomotion is capable of tackling various situations such as rough terrain or navigating crowds. By decoding the motor intent, assistive devices could adapt to their user's movements and their diverse environments. Using simulated gait and assistive devices, this complex design space can be explored in a low-cost and accessible way, promoting the transfer of the next generation of intent driven controllers from the lab to real-life scenarios.

References

[1] R. Gehlhar, M. Tucker, A. J. Young, and A. D. Ames, "A review of current state-of-the-art control methods for lower-limb powered prostheses," *Annu. Rev. Control*, vol. 55, pp. 142–164, Feb. 2023.

[2] M. R. Tucker et al., "Control strategies for active lower extremity prosthetics and orthotics: A review," *J. NeuroEng. Rehabil.*, vol. 12, no. 1, p. 1, 2015.

[3] D. Hill, C. S. Holloway, D. Z. M. Ramirez, P. Smitham, and Y. Pappas, "What are user perspectives of exoskeleton technology? A literature review," *Int. J. Technol. Assessment Health Care*, vol. 33, no. 2, pp. 160–167, 2017.

[4] A. F. Azocar, L. M. Mooney, J.-F. Duval, A. M. Simon, L. J. Hargrove, and E. J. Rouse, "Design and clinical implementation of an open-source bionic leg," *Nature Biomed. Eng.*, vol. 4, no. 10, pp. 941–953, Oct. 2020.

[5] R. Baud, A. R. Manzoori, A. Ijspeert, and M. Bouri, "Review of control strategies for lower-limb exoskeletons to assist gait," *J. NeuroEng. Rehabil.*, vol. 18, no. 1, pp. 1–34, Dec. 2021.

[6] D. Torricelli and J. L. Pons, "EuroBench: Preparing robots for the real world," in *Proc. Int. Symp. Wearable Robot.* Cham, Switzerland: Springer, 2018, pp. 375–378.

[7] J. M. Caputo and S. H. Collins, "A universal ankle–foot prosthesis emulator for human locomotion experiments," *J. Biomech. Eng.*, vol. 136, no. 3, pp. 1–10, Mar. 2014.

[8] R. S. Armiger et al., "A real-time virtual integration environment for neuroprosthetics and rehabilitation," *Johns Hopkins APL Tech. Dig.*, vol. 30, no. 3, pp. 198–206, 2011.

[9] F. De Groote and A. Falisse, "Perspective on musculoskeletal modelling and predictive simulations of human movement to assess the neuromechanics of gait," *Proc. Roy. Soc. B, Biol. Sci.*, vol. 288, no. 1946, Mar. 2021, Art. no. 20202432.

[10] A. R. Wu et al., "An adaptive neuromuscular controller for assistive lower-limb exoskeletons: A preliminary study on subjects with spinal cord injury," *Frontiers Neurorobot.*, vol. 11, p. 30, Jun. 2017.

[11] A. Falisse, G. Serrancolí, C. L. Dembia, J. Gillis, I. Jonkers, and F. De Groote, "Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies," *J. Roy. Soc. Interface*, vol. 16, no. 157, Aug. 2019, Art. no. 20190402.

[12] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4906–4913.

[13] E. Conti, V. Madhavan, F. Petroski Such, J. Lehman, K. Stanley, and J. Clune, "Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–15.

[14] L. Fussell, K. Bergamin, and D. Holden, "SuperTrack: Motion tracking for physically simulated characters using supervised learning," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–13, Dec. 2021.

[15] J. Merel et al., "Neural probabilistic motor primitives for humanoid control," 2018, *arXiv:1811.11711*.

[16] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "AMP: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–20, Aug. 2021.

[17] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes, "DReCon: Data-driven responsive control of physics-based characters," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–11, Dec. 2019.

[18] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–13, Aug. 2019.

[19] S. Song et al., "Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation," *J. NeuroEng. Rehabil.*, vol. 18, no. 1, pp. 1–17, Aug. 2021.

[20] T. Geijtenbeek and N. Pronost, "Interactive character animation using simulated physics: A state-of-the-art review," *Comput. Graph. Forum*, vol. 31, no. 8, pp. 2492–2515, Dec. 2012.

[21] L. Mourot, L. Hoyet, F. Le Clerc, F. Schnitzler, and P. Hellier, "A survey on deep learning for Skeleton-based human animation," *Comput. Graph. Forum*, vol. 41, no. 1, pp. 122–157, Feb. 2022.

[22] OpenAI et al., "Solving Rubik's cube with a robot hand," 2019, *arXiv:1910.07113*.

[23] S. Höfer et al., "Sim2Real in robotics and automation: Applications and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 398–400, Apr. 2021.

[24] X. Bin Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," 2020, *arXiv:2004.00784*.

[25] C. K. Liu and D. Negrut, "The role of physics-based simulators in robotics," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 4, no. 1, pp. 35–58, May 2021.

[26] D. J. Farris, J. L. Hicks, S. L. Delp, and G. S. Sawicki, "Musculoskeletal modelling deconstructs the paradoxical effects of elastic ankle exoskeletons on plantar-flexor mechanics & energetics during hopping," *J. Experim. Biol.*, vol. 217, no. 22, pp. 4018–4028, Jan. 2014.

[27] J. Zhang et al., "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science*, vol. 356, no. 6344, pp. 1280–1284, Jun. 2017.

[28] A. M. Simon et al., "Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes," *PLoS ONE*, vol. 9, no. 6, Jun. 2014, Art. no. e99387.

[29] J. Park, S. Min, P. S. Chang, J. Lee, M. S. Park, and J. Lee, "Generative GaitNet," in *Proc. Special Interest Group Comput. Graph. Interact. Techn. Conf.*, Aug. 2022, pp. 1–9.

[30] B. Lim, S. Hyoung, J. Lee, K. Seo, J. Jang, and Y. Shim, "Simulating gait assistance of a hip exoskeleton: Case studies for ankle pathologies," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1022–1027.

[31] J. I. Han, J.-H. Lee, H. S. Choi, J.-H. Kim, and J. Choi, "Policy design for an ankle-foot orthosis using simulated physical human–robot interaction via deep reinforcement learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2186–2197, 2022.

[32] L. De Vree and R. Carloni, "Deep reinforcement learning for physics-based musculoskeletal simulations of healthy subjects and transfemoral prostheses' users during normal walking," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 607–618, 2021.

[33] M. Büttner and S. Clavet, "Motion matching-the road to next gen animation," in *Proc. Nucl. AI*, 2015, vol. 1, no. 2015, p. 2.

[34] M. Tschiedel, M. F. Russold, and E. Kaniusas, "Relying on more sense for enhancing lower limb prostheses control: A review," *J. NeuroEng. Rehabil.*, vol. 17, no. 1, pp. 1–13, Dec. 2020.

[35] E. M. Murtagh, J. L. Mair, E. Aguiar, C. Tudor-Locke, and M. H. Murphy, "Outdoor walking speeds of apparently healthy adults: A systematic review and meta-analysis," *Sports Med.*, vol. 51, no. 1, pp. 125–141, Jan. 2021.

[36] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 1–60, Aug. 2020.

[37] D. Bollo, "High performance animation in Gears of War 4," in *Proc. ACM SIGGRAPH Talks*, 2017, pp. 1–2.

[38] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, Jul. 1985.

[39] S. Tunyasuvunakool et al., "dm_control: Software and tasks for continuous control," *Softw. Impacts*, vol. 6, Nov. 2020, Art. no. 100022.

[40] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.

[41] A. Juliani et al., "Unity: A general platform for intelligent agents," 2018, *arXiv:1809.02627*.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[43] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Aug. 2018.

[44] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[45] S. Lee, S. Lee, Y. Lee, and J. Lee, "Learning a family of motor skills from a single motion clip," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–13, Aug. 2021.

[46] J. Tan, K. Liu, and G. Turk, "Stable proportional-derivative controllers," *IEEE Comput. Graph. Appl.*, vol. 31, no. 4, pp. 34–44, Jul. 2011.

[47] D. A. Winter, "Kinematic and kinetic patterns in human gait: Variability and compensating effects," *Hum. Movement Sci.*, vol. 3, nos. 1–2, pp. 51–76, Mar. 1984.

[48] S. Pfeifer, R. Riener, and H. Vallery, "Knee stiffness estimation in physiological gait," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 1607–1610.

[49] K. Shamaei, G. S. Sawicki, and A. M. Dollar, "Estimation of quasi-stiffness of the human hip in the stance phase of walking," *PLoS ONE*, vol. 8, no. 12, Dec. 2013, Art. no. e81841.

[50] J. Booth and V. Ivanov, "Realistic physics based character controller," 2020, *arXiv:2006.07508*.

[51] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[52] Ł. Kidziński et al., "Artificial intelligence for prosthetics: Challenge solutions," in *Proc. NeurIPS*. Cham, Switzerland: Springer, 2020, pp. 69–128.

[53] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," 2016, *arXiv:1609.09106*.

[54] E. Y. Chao, R. K. Laughman, E. Schneider, and R. N. Stauffer, "Normative data of knee joint motion and ground reaction forces in adult level walking," *J. Biomech.*, vol. 16, no. 3, pp. 219–233, Jan. 1983.

[55] A. Kharb, V. Saini, Y. K. Jain, and S. Dhiman, "A review of gait cycle and its parameters," *IJCEM Int. J. Comput. Eng. Manage.*, vol. 13, pp. 78–83, Jul. 2011.

[56] K. C. Moisio, D. R. Sumner, S. Shott, and D. E. Hurwitz, "Normalization of joint moments during gait: A comparison of two techniques," *J. Biomech.*, vol. 36, no. 4, pp. 599–603, Apr. 2003.

[57] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, Eds. Cham, Switzerland: Springer, 2021, pp. 321–384, doi: 10.1007/978-3-030-60990-0_12.

[58] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv:1706.05296*.

[59] G. M. Bryan et al., "Optimized hip-knee-ankle exoskeleton assistance reduces the metabolic cost of walking with Worn loads," *J. NeuroEng. Rehabil.*, vol. 18, no. 1, pp. 1–13, Dec. 2021.

[60] J. Duysens and H. W. A. A. Van de Crommert, "Neural control of locomotion; Part 1: The central pattern generator from cats to humans," *Gait Posture*, vol. 7, no. 2, pp. 131–141, Mar. 1998.

[61] H. Wang, V. Caggiano, G. Durandau, M. Sartori, and V. Kumar, "MyoSim: Fast and physiologically realistic MuJoCo models for musculoskeletal and exoskeletal studies," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8104–8111.

[62] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "MyoSuite—A contact-rich simulation suite for musculoskeletal motor control," 2022, *arXiv:2205.13600*.

[63] P. A. Forbes, A. Chen, and J.-S. Blouin, "Chapter 4—Sensorimotor control of standing balance," in *Balance, Gait, and Falls* (Handbook of Clinical Neurology), vol. 159, B. L. Day and S. R. Lord, Eds. Amsterdam, The Netherlands: Elsevier, 2018, pp. 61–83. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444463916500045, doi: 10.1016/B978-0-444-63916-5.00004-5.

[64] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–14, Dec. 2019.

[65] G. Feng et al., "GenLoco: Generalized locomotion controllers for quadrupedal robots," 2022, *arXiv:2209.05309*.

[66] B. Hodossy and J. Llobera. (Oct. 2023). *Modular Agents: Extensions to the ML-Agents Toolkit, Focusing on Humanoid Control in Unity*. [Online]. Available: https://github.com/Balint-H/modular-agents