# Understanding Human Behaviour in a Robotic Guide Dog Using Parallel Deep Reinforcement Learning

Riccardo Aliotta*
*PRISMA Lab - DAS*
*University of Naples Federico II*
riccardo.aliotta@unina.it

Federico Esposito
*PRISMA Lab - CeSMA*
*University of Naples Federico II*
federico.esposito3@unina.it

Alessia Palumbo*, Fabio Ruggiero
*PRISMA Lab - DIETI*
*University of Naples Federico II*
alessia.palumbo@studenti.unina.it, fabio.ruggiero@unina.it

***Abstract*—This paper employs a massively parallel deep reinforcement learning approach to model human behavior in a quadruped robot system designed to function as a guide dog for visually impaired individuals. The supervisory network operates alongside a control scheme that ensures the safe interaction between the human user and the robot. It distinguishes between several simplified human behaviors, including stopping, turning, spinning, and walking straight. The trained network builds upon a low-level control network that effectively manages the quadruped's locomotion across challenging terrains. The effectiveness of the proposed approach is validated in a physics-engine-based simulation environment.**

## I. Introduction

The ability to walk and navigate one's environment is crucial for personal autonomy and quality of life. For blind and visually impaired people (BVIP), this ability is significantly hindered, creating substantial challenges in daily life and social integration. Guide dogs play a critical role in mitigating these challenges, as they are specially trained to assist BVIP by guiding them around obstacles, stopping at curbs and stairs, and ensuring safe navigation through various environments. Robotic solutions can also help in assisting walking-impaired individuals [1]. Examples include exoskeletons [2], [3], robot wheelchairs [4], [5], robotic guide canes [6], [7], robotic shopping trolleys [8], [9], and tethered robots [10]–[12].

This paper seeks to build upon the concept of using a quadruped robot in the role of a guide dog. The authors do not intend to replace guide dogs, which offer emotional support and companionship to BVIP, but rather to provide a robotic alternative when the challenges of selecting, training, and caring for a guide dog become prohibitive [13]. Additionally, robotic systems offer intrinsic benefits, such as easy customization and the potential to integrate with the user's biological data, enabling emergency calls if necessary. For a comparison of usability and efficiency between robot guide dogs and other assistive devices for BVIP, see [14].

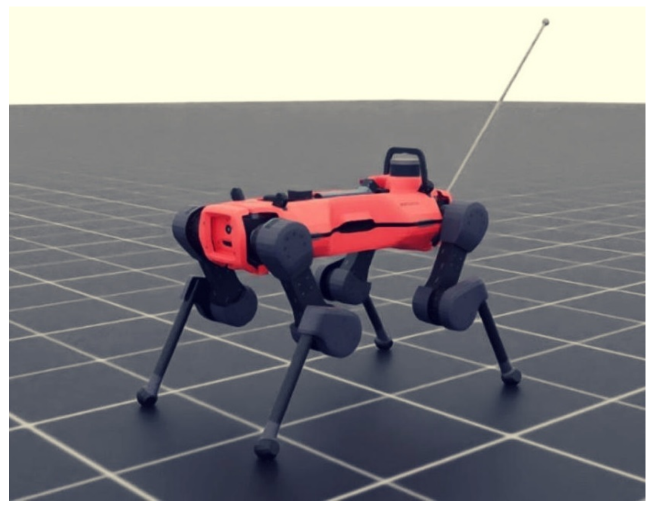

Fig. 1. The quadruped robot plays the role of a guide dog for BVIP.

This paper presents a preliminary study aimed at generalizing the interpretation of BVIP behaviors using deep learning. Instead of using real human data, we utilize a simplified set of behaviors to validate the approach. The quadruped robot is first trained to navigate robustly over challenging terrain, as demonstrated in [15]. Subsequently, a separate neural network is trained to interpret human intentions and relay velocity commands to the pre-trained locomotion controller, effectively generalizing the supervisory system. In this work, we omit the consideration of human-robot safe interaction, which has already been achieved through an admittance filter in [12], and focus instead on generating velocity commands based on human behaviors. We employ a rigid harness model, as recommended in [16], represented as a rigid bar connecting the robot and the human hand, modeled as a mass at the opposite end of the bar (see Fig. 1). Interaction forces are assumed to be measurable either through sensors or by employing the observer method outlined in [12]. By randomizing the mass of the quadruped robots and leveraging massively parallel

The research leading to these results has been supported by the COWBOT project, in the frame of the PRIN 2020 research program, grant n. 2020NH7EAZ_002, the ARIEL project, in the frame of the PRIN 2022 research program, grant n. 2022WS29WP, funded by the European Union Next-Generation EU, and the DARC project, in the frame of the PRIN 2022 PNRR research program, grant n. P2022MHR5C, funded by the European Union Next-Generation EU. The authors are solely responsible for its content.

*Riccardo Aliotta and Alessia Palumbo are both first authors.

deep reinforcement learning (PDRL), the system is able to train a network capable of distinguishing between different commands, such as continuing to walk, stopping, turning, and spinning in place. The validity of this approach is tested within a physics-engine-based simulation environment, such as NVIDIA's Isaac Lab [17].

## II. State of the Art and Contributions

The study of devices to aid BVIP dates back to the 1980s [18], [19], with numerous applications developed since then [6], [8], [9], [20]. Initial prototypes for tethered guide robots emerged in the early 2000s [7], [21]. Various robots were designed to assist BVIP [11], [21]–[23] or serve as assistants [24]–[27]. This work focuses on guide robot dogs, similar to the one described in [10], where the authors deployed a hybrid motion controller for human-robot interaction on a Mini Cheetah quadruped robot. A comprehensive control system for a quadruped robot was introduced in [12], where the commanded force is computed using a momentum-based estimator. Human-robot interaction is regulated at the force level through an admittance filter and at the planning level via a deterministic supervisor, which selects the appropriate action based on force input. The lower-level control is executed using a whole-body controller to manage the robot's movements.

In [13], the groundwork for a guide dog-like system is established by implementing a semantic-aware navigation framework. This system takes into account the human's presence, adjusting for factors critical to BVIP, like movement speed and directional control.

In [28], the control strategy emphasizes user comfort by modeling the leash as an elastic connection and developing a human force-motion model for a tethered environment. The robot's motion planner is driven by a human motion planner, which prioritizes user comfort by accounting for factors like the distance between the robot and the user, the exchanged forces, and minimizing frequent transitions between slack and taut leash conditions.

The concept of force-based interaction with a guide robot has been explored in [29], where pulling on the leash signals the robot's planning algorithm to initiate turns. Additionally, in [14], tugging is employed to modify the robot's speed, mimicking established human-guide dog communication protocols instead of using a joystick. Based on the paper's description in the previous section and the current state of the art, two key contributions can be identified: $(i)$ generalise the supervisor devised in [12] via a PDRL approach, and $(ii)$ use a trained network for understanding human behaviour over an already trained network for quadruped locomotion on rough terrains. This modular approach allows the possibility to change the supervisor policy without jeopardising the low-level quadruped controller, and vice versa.

## III. Methodology

The objective of this work is to train a neural network using PDRL, specifically employing Proximal Policy Optimization (PPO) [30], to interpret commands from a visually impaired user directed at a robotic guide dog via a harness. These commands, conveyed through pulling on the harness, signal the need for the robot to stop or rotate. The forces transmitted through the harness are translated into velocity commands for a low-level neural network responsible for managing the quadruped's locomotion, ensuring robustness against variations in physical parameters and terrain conditions. A spherical element is integrated at the gripping point in the human's hand, simulating the application of forces and the corresponding human actions.

The architecture of the robot controller is divided into two hierarchically related parts, as can be seen in Fig. 2. Both neural networks are designed as fully connected neural networks in Torch [31] and are subsequently trained via PDRL, leveraging the parallelisation capabilities of the PPO algorithm to reduce training time by simulating several environments in parallel[1]. This approach also facilitates easier and faster domain randomisation, as each environment can have different physical and dynamic parameters.

The low-level controller is designed to track velocity commands in the horizontal plane ($xy$ plane) of the robot's body frame, as well as the angular velocity around the vertical axis ($z$ axis). In the training environment, the network receives several inputs: joint positions and velocities; the floating base's linear and angular velocities; the commanded velocity (a three-dimensional vector comprising the aforementioned velocity commands); the previous actions generated by the network; the projected gravity vector; and a point cloud representing the height scan of a rectangular patch of terrain beneath the robot, totaling 235 elements. The network outputs position commands for the robot's 12 rotational joints. This fully connected neural network architecture consists of three hidden layers, containing $512, 256$ and $128$ neurons, respectively.

Training was conducted with $4096$ parallel environments (see Fig. 3), each with randomized values for mass and friction that remained constant throughout the simulation, while the initial body and joint positions were randomized at the start of every new episode. Random external disturbances were regularly applied to the body, with force disturbances chosen within the interval $(-100, 100)$ N to ensure robustness against strong human pulls. Terrain training followed a specified curriculum, enabling the controller to learn to traverse various challenging terrains with differing types and levels of difficulty, such as varying slope inclinations.

After training the first network, a second training environment was developed to train the supervisor network. This network is tasked with interpreting force commands applied to the robot through the harness and converting them into velocity commands for the low-level controller, thereby complying with the human's instructions. The pre-trained network is converted from Torch format to Open Neural Network Exchange (ONNX) format using the *onnx2torch* library [32]. The supervisor network receives the same inputs as the low-level

[1] Notice that the batch size does not vary, as it is the product between the number of environments in parallel and the number of simulation steps needed from every environment (see [15] for further details).

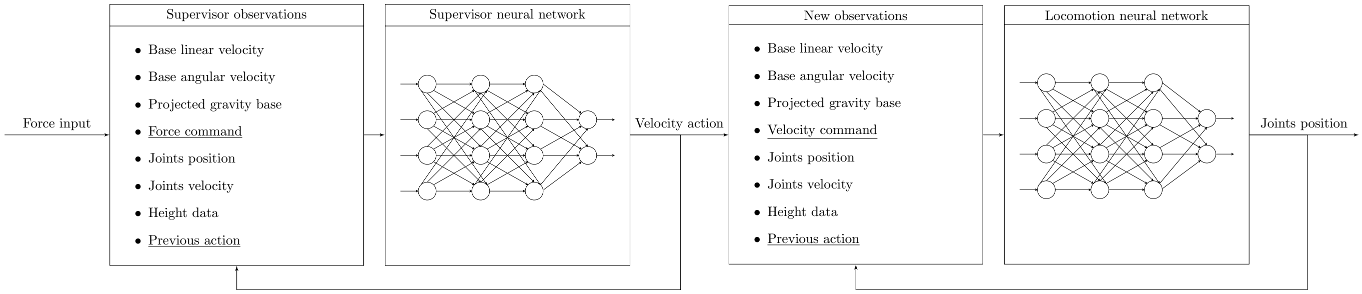


Fig. 2. Proposed controller architecture.

Fig. 3. 4096 robots learning to walk in parallel environments on rough terrains.

controller, with the only difference being that the dimension of the previous action is now based on commanded velocities, while the command interpretation focuses on the force inputs. Similar to the low-level controller, the supervisor network features three hidden layers, each with 128 neurons.

## IV. Supervisor policy

This section introduces the implemented supervisor policies to train the neural network (e.g., rewards and penalties description) to understand the proposed simplified human behaviour.

The supervisor policy is meant to analyse the pulling force exerted by the human, decomposing it into the $x$ (heading) and $y$ (lateral) directions. The $z$ component, although simulated and generated, is not considered relevant for the supervisor in this context. Two thresholds, $\sigma_x = 40$ N and $\sigma_y = 20$ N, were assigned to both $x$ and $y$ components, respectively; $\sigma_x$ was chosen accordingly to [13], while $\sigma_y$ was heuristically chosen as $\sigma_x/2$. Exceeding the thresholds dictates the desired behaviour of the supervisor as follows.

- Low force on both axes: the dog should keep walking straight at a given velocity.
- High force on the $x$ axis and low on the $y$ axis: the dog should stop.
- Low force on the $x$ axis and high on the $y$ axis: the dog should keep moving forward while curving towards the direction of the force.
- High force on both axes: the dog should stop and then rotate in place in the direction of the applied force.

TABLE I
Reward Functions

| Task | Reward Function | Weights |
|---|---|---|
| Behavioural[a] | $\exp(-10((v_x^* - v_x)^2 + (v_y^* - v_y)^2 + (\omega_z^* - \omega_z)^2)$ | 1 |
| Action Rate | $(y(k) - y(k-1))^2$ | $-10^{-4}$ |

[a] The target velociti es change based on the force input values. See Table II for more details.

TABLE II
Behavioural Rewards

| Task | Target Velocity | Condition |
|---|---|---|
| Walk | $v_x^* = 0.8\,\mathrm{m/s}, v_y^* = 0, \omega_z^* = 0$ | $f_x \geq -\sigma_x, \lvert f_y \rvert \leq \sigma_y$ |
| Stop | $v_x^* = 0, v_y^* = 0, \omega_z^* = 0$ | $f_x < -\sigma_x, \lvert f_y \rvert \leq \sigma_y$ |
| Turn | $v_x^* = 0.8\,\mathrm{m/s}, v_y^* = 0, \omega_z^* = 0.02 f_y$ | $f_x \geq -\sigma_x, \lvert f_y \rvert > \sigma_y$ |
| Spin | $v_x^* = 0, v_y^* = 0, \omega_z^* = 0.02 f_y$ | $f_x < -\sigma_x, \lvert f_y \rvert > \sigma_y$ |

To enable the network to learn the aforementioned policy, a structured set of rewards and a penalty were devised for training the second network outlined in Section III, while the forces exerted were in the ranges ($-80$, 0) N and ($-40$,40) N, respectively. It is crucial to highlight that the threshold for interpreting user commands was known solely to the reward function and not to the network itself. In total, one penalty aimed at reducing the action rate and four distinct rewards to foster the learning of the desired policy were defined. Notably, these rewards are variations of a single function, each adapted to the specific intended command. The details of these rewards and their corresponding parameters are summarized in Table I and Table II. In these tables, $v_x, v_y, \omega_z$ represent respectively the linear velocities along the $x$ and $y$ axes and the angular velocity around the $z$ axis of the robot, respectively. The asterisk is used to mark a desired quantity, while $y(k)$ represents the network output at the $k$-th iteration. For simplicity, the variable notation omits the index associated with the specific environment number.

Designed rewards and penalties follow.

- Stopping reward: when the associated condition is met, the robot should stop and stand still.
- Walking reward: when the associated condition is met, the robot should keep moving forward.

- Turning reward: when the associated condition is met, the robot should align itself with the harness's desired direction, while still moving forward, which coincides with a turning motion. The velocity is proportional to the exerted lateral force $f_y$.
- Spinning reward: when the associated condition is met, the robot should stand in place and spin on itself. Again, the angular velocity is proportional to $f_y$.
- Action rate penalty: the network output's derivative is penalised to avoid fast varying commands.

# V. Validation

## A. Hardware and set-up descriptions

The dynamic simulations and the networks' training were done on a workstation equipped with an Intel i9-13900KS processor, 128 Gb of DDR5-5600 RAM, and an NVidia RTX A6000 GPU with 48 Gb of VRAM. All the code was run on the Ubuntu release 22.04 LTS, according to the instructions from [17] to run NVidia's Isaac Lab. This choice was made mainly for two reasons: $(i)$ the Isaac Lab suite provides a direct interface for creating reinforcement learning environments and training agents, which reduces the implementation overhead of neural network interfaces, and $(ii)$ Isaac Lab leverages the capabilities of NVidia GPUs to perform end-to-end training on graphical computational resources. This setup allows for the simulation and data collection from hundreds or thousands of robots in parallel, enabling the use of massive PDRL. ANYmal D from ANYbotics has been used in simulation, given its similarities in dimensions to a medium build guide dog. The robot's URDF file has been taken from the official GitHub repository[2], and the needed modifications have been made. Specifically, the inspection payload has been removed as it is not needed in a robotic guide robot, while a leash-like element has been added to the back of the robot as described in Section III.

## B. Case studies description

The trained supervisor policy was evaluated under various force inputs to assess its performance and robustness. Initially, it was validated using force values that the policy had encountered during training to confirm that it exhibited the desired behavior. Subsequently, tests were conducted using the exact threshold values to examine its performance in limit conditions. Finally, the policy was challenged with inputs outside the training range, simulating scenarios where the human exerted greater forces or pulled in unexpected directions, resulting in a push. A video accompanying this paper shows the representative case studies[3].

## C. Discussion of the results

The results of the supervisor training have been evaluated both via the behavioural reward and by validating in simulation the network's performance. As can be seen in Fig. 4, the total reward tends to 1, which is its theoretical maximum value, reaching on average 0.61 after 3000 iterations: this means that the desired velocity is not being exactly tracked, but gives us little explicit information about the behaviours. By further looking at the single rewards, it is possible to notice that the Stop average reward (AR) surpasses 0.22, where 0.25 would be optimal, the Spin AR surpasses 0.16, while the Walk AR settles around 0.13 and the Turn AR barely reaches 0.08.

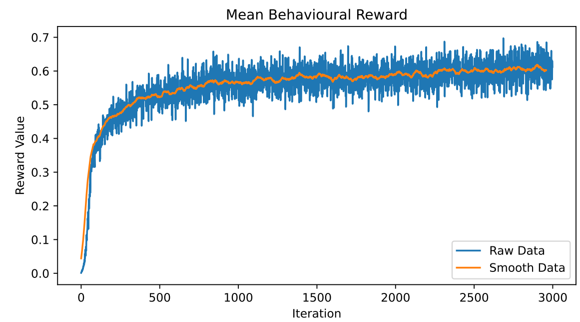


Fig. 4. The averaged sum of the behavioural rewards. In blue, raw data is shown, while in orange the data smoothed via a 50-sample wide moving average.

Moving on to simulation validation, a success rate has been evaluated for every scenario by checking if the correct behaviour is achieved. We tested 17 different force combinations numerous times, and the results can be seen in Fig. 5. We can see that the Spin has a $100\%$ success rate, while the Turn shows issues when subjected to inputs not previously explored: when subject to $0/50$ N, the robot starts to rotate around a point, but the movement is sideways and not forwards, deeming it as a failure, while when subjected to $20/50$ N it starts turning, but the high and unexpected force values make it fall. The Walk behaviour also has a high success rate, but sometimes a slight sidestep is present, often negligible; in the case of the push, however, all robots start turning to the left, which is a failure. Finally, the Stop behaviour often causes the robot to assume a relevant yaw position, in which case it is deemed as a failure. This could be solved by adding a penalty on the yaw value, as its slow spinning motion justifies the high value for the adopted reward.

In conclusion, we can see that in normal circumstances, as well as in some never seen before case, the behaviour is correctly interpreted, with some extreme case which are not managed correctly by the supervisor.

# VI. Conclusions and Future Work

This paper presented a framework for a robotic guide dog, featuring an architecture that uses a massively PDRL neural network to maintain balance on irregular terrains. Building upon this trained controller, a supervisory network was developed to differentiate between various human behaviors, including stopping, turning, and continuing to walk. The validity of this approach was demonstrated through simulations conducted in Isaac Lab. In the future, the policy will be

[2]https://github.com/ANYbotics/anymal_d_simple_description
[3]https://youtu.be/P2Dnm0Wtcsk

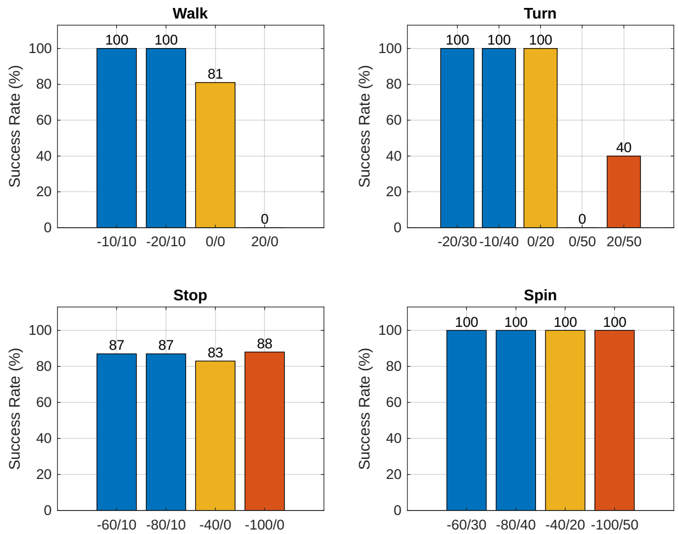


Fig. 5. The success rates for every force command tested. The forces, expressed in Netwons, are in the format $f_x/f_y$. Blue bars represent forces already applied in the training, yellow ones represents edge cases and red ones represents forces outside of the training ranges.

improved by considering BVIP's feedback, and the framework will be validated in real-life scenarios.

## References

[1] K. Kodur and M. Kyrarini, "Patient–robot co-navigation of crowded hospital environments," *Applied Sciences*, vol. 13, no. 7, p. 4576, 2023.

[2] Z. Shen, Y. Zhuang, J. Zhou, J. Gao, and R. Song, "Design and test of admittance control with inner adaptive robust position control for a lower limb rehabilitation robot," *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 134–142, 2020.

[3] Q. Wu, X. Wang, B. Chen, and H. Wu, "Development of a minimal-intervention-based admittance control strategy for upper extremity rehabilitation exoskeleton," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 1005–1016, 2017.

[4] L. Teng, M. A. Gull, and S. Bai, "Pd-based fuzzy sliding mode control of a wheelchair exoskeleton robot," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2546–2555, 2020.

[5] H. Ikeda, T. Toyama, D. Maki, K. Sato, and E. Nakano, "Cooperative step-climbing strategy using an autonomous wheelchair and a robot," *Robotics and Autonomous Systems*, vol. 135, p. 103670, 2021.

[6] J. Borenstein and I. Ulrich, "The guidecane-a computerized travel aid for the active guidance of blind pedestrians," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, 1997, pp. 1283–1288.

[7] A. A. Melvin, B. Prabu, R. Nagarajan, B. Illias, A. Allan Melvin, B. Prabu, R. Nagarajan, I. Bukhari, and A. Melvin, "Rovi: a robot for visually impaired for collision-free navigation," in *Proc. of the International Conference on Man-Machine Systems (ICoMMS 2009)*, 2009, pp. 3B5–1.

[8] C. P. Gharpure and V. A. Kulyukin, "Robot-assisted shopping for the blind: issues in spatial cognition and product selection," *Intelligent Service Robotics*, vol. 1, no. 3, pp. 237–251, 2008.

[9] V. Kulyukin and A. Kutiyanawala, "Accessible shopping systems for blind and visually impaired individuals: Design requirements and the state of the art," *The Open Rehabilitation Journal*, vol. 3, no. 1, 2010.

[10] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath, "Robotic guide dog: Leading a human with leash-guided hybrid physical interaction," in *2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 11 470–11 476.

[11] M. Tognon, R. Alami, and B. Siciliano, "Physical human-robot interaction with a tethered aerial vehicle: Application to a force-based human guiding problem," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 723–734, 2021.

[12] V. Morlando, V. Lippiello, and F. Ruggiero, "Tethering a human with a quadruped robot: A guide dog to help visual impaired people," in *31st Mediterranean Conference on Control and Automation*, 2023, pp. 547–553.

[13] H. Hwang, T. Xia, I. Keita, K. Suzuki, J. Biswas, S. I. Lee, and D. Kim, "System configuration and navigation of a guide dog robot: Toward animal guide dog-level guiding work," in *2023 IEEE International Conference on Robotics and Automation*, 2023, pp. 9778–9784.

[14] S. Cai, A. Ram, Z. Gou, M. A. W. Shaikh, Y.-A. Chen, Y. Wan, K. Hara, S. Zhao, and D. Hsu, "Navigating real-world challenges: A quadruped robot guiding system for visually impaired people in diverse environments," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI '24. Association for Computing Machinery, 2024.

[15] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*, 2022, pp. 91–100.

[16] H. Hwang, H.-T. Jung, N. A. Giudice, J. Biswas, S. I. Lee, and D. Kim, "Towards robotic companions: Understanding handler-guide dog interactions for informed guide dog robot design," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, ser. CHI '24. Association for Computing Machinery, 2024.

[17] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740—3747, 2023.

[18] S. Tachi, K. Tanie, K. Komoriya, Y. Hosoda, and M. Abe, "Guide dog robot—its basic plan and some experiments with meldog mark i," *Mechanism and Machine Theory*, vol. 16, no. 1, pp. 21–29, 1981.

[19] S. Tachi, R. W. Mann, and D. Rowell, "Quantitative comparison of alternative sensory displays for mobility aids for the blind," *IEEE Transactions on Biomedical Engineering*, no. 9, pp. 571–577, 1983.

[20] M. Bousbia-Salah and M. Fezari, "A navigation tool for blind people," in *Innovations and advanced techniques in computer and information sciences and engineering*, 2007, pp. 333–337.

[21] J. E. Young, Y. Kamiyama, J. Reichenbach, T. Igarashi, and E. Sharlin, "How to walk a robot: A dog-leash human-robot interface," in *2011 RO-MAN*, 2011, pp. 376–382.

[22] M. Avila, M. Funk, and N. Henze, "Dronenavigator: Using drones for navigating visually impaired persons," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, 2015, pp. 327–328.

[23] F. Mueller and M. Muirhead, "Jogging with a quadcopter," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 2023–2032.

[24] A. H. Basori, "Naocared: Intelligent and communally humanoid assistive robot for elderly care support," *Int. J. Comput. Sci. Netw.*, vol. 20, no. 4, pp. 113–120, 2020.

[25] P. T. A. Junior, B. d. F. V. Perez, R. Meneghetti, F. d. A. M. Pimentel, and G. N. Marostica, "Hera: Home environment robot assistant," in *II BRAHUR and III Brazilian Workshop on Service Robotics*, 2019.

[26] J. Bowen and C. Morosan, "Beware hospitality industry: the robots are coming," *Worldw. Hosp. Tour. Themes*, 2018.

[27] P. Balatti, I. Ozdamar, D. Sirintuna, L. Fortini, M. Leonori, J. M. Gandarias, and A. Ajoudani, "Robot-assisted navigation for visually impaired through adaptive impedance and path planning," in *2024 IEEE International Conference on Robotics and Automation*, 2024, pp. 2310–2316.

[28] Y. Chen, Z. Xu, Z. Jian, G. Tang, L. Yang, A. Xiao, X. Wang, and B. Liang, "Quadruped guidance robot for the visually impaired: A comfort-based approach," in *2023 IEEE International Conference on Robotics and Automation*, 2023, pp. 12 078–12 084.

[29] D. DeFazio, E. Hirota, and S. Zhang, "Seeing-eye quadruped navigation with force responsive locomotion control," in *Conference on Robot Learning*, 2023, pp. 2184–2194.

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

[31] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[32] E. developers, I. Kalgin, A. Yanchenko, P. Ivanov, and A. Goncharenko, "onnx2torch," https://enot.ai/, 2021.