Stability Improvements for Physics-Simulated Mobility Aids

Stefan Teodor Maxim & Kiyn Chin

Received October 08, 2024
Accepted December 20, 2024
Electronic access December 31, 2024

Globally, mobility aid usage is estimated to increase significantly. Cane stability is crucial for individuals who rely on canes for mobility and balance. In this work, we explore how to address the cane balance problem. The study objective is to find a theoretical model for understanding the stability of a cane and propose a model that will lead to improved stability. We utilized a real-time physics simulator to model the environment. First, we created a balancing mechanism for a cane by simulating it as a 3D extension of the canonical Cart Pole problem. We established thresholds for deviation from vertical, displacement, and jerkiness. Second, we defined cane stability by the time the cane is vertical within a threshold and established a baseline. Finally, we conducted experiments, and the longest time the cane was upright, 74.184 seconds, was achieved by attributing equal weights to the weight function's angle, time, and jerkiness, and the reward function skewed towards the jerk. Experiments demonstrate that our model outperforms the benchmark by 5.94%. This research can be further extended by building a physical mechanism based on modeling results.

Introduction

Using mobility aids like canes has exploded in recent decades and is necessary for a sizable percentage of the global population[1]. During the last decade, the US alone has seen an estimated 26% increase[2] in cane usage for people aged 65+. Unfortunately, canes also lead to tripping or falling[2] more frequently than other mobility aids like walkers and wheelchairs. For visually impaired individuals, using a cane is necessary to walk independently and safely[3]. In 2023, approximately 304.1 million people have moderate to severe blindness while around 49.1 million are completely blind, a 42.8% increase from 34.4 million in 1990[4]. Currently, the only standardized balancing mechanism is one's senses and hand-eye coordination, which may work for some but not others. Given the aging population increase and the more significant increase in visually impaired people, we believe this problem is worth investigating. The study objective is to find a theoretical model for understanding the stability of a cane. We measured several variables and observed the impact on the cane balance. We hypothesize that jerk is one of the main factors influencing stability. The cane inclination, θ, ranges from -90 to +90 degrees (TH1). The cane is upright if θt (the deviation from vertical) satisfies the following threshold (TH2):

$$\text{(TH1)} \quad -90^\circ \leq \theta \leq 90^\circ$$
$$\text{(TH2)} \quad -12^\circ \leq \theta_t \leq 12^\circ$$

We hypothesize that the timesteps will be maximized when we use a reward function that optimizes only for timesteps rather than jerk and angle from upright. A person using a cane will apply a force at a certain angle on the cane. The cane should "respond" to changes in orientation, ultimately supporting the person using it. There are several parts to improve stability for real-time physics-simulated mobility aids: software, material science, and mechanics. By simulating the stability problem, we can understand what materials and mechanical parts are necessary to build stability into the cane. This paper focuses on the software part by employing an AI algorithm to determine the reaction of the cane to remain in a stable state. The testing environment is limited to simulation with synthetic data. Testing with "real" data is challenging to do for unsafe scenarios. Simulation is a good proxy. Simulation makes it faster to try many variable combinations in a short amount of time. We used OpenAI simulation and a classic Reinforcement Learning (RL)[5] loop. RL is a machine learning function where an agent takes action by learning how to maximize a predefined reward function. In Figure 1, the agent optimizes function R and moves from the state S_t to state S_{t+1}.

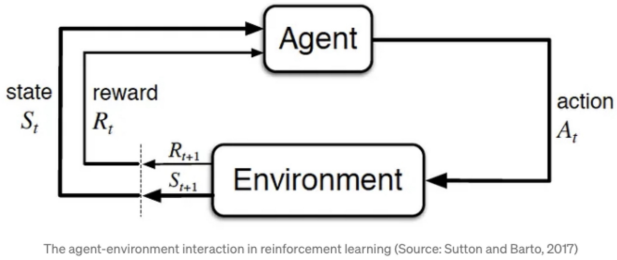

Fig. 1 The reward loop in an RL environment[5]

Our method consists of defining the metrics and using close-

loop experiments to collect the resulting actual value of the metrics. After comparing different experiment results, we will understand the key factors influencing cane stability.

Literature Review

Cane-related research is primarily limited to attaching cameras or other position-detecting tools and using them to map the surroundings. One such method is using deep-learning models like YOLOv8[6], a commonly used DNN in image deception software to detect the position of one's surroundings and alert the user when they approach obstacles[7,8] . Others opt to use traditional programming solutions by optimizing the route that someone with visual impairments would use to navigate by creating a wearable device to calculate the optimal path for a person to walk[9]. Another paper[10] proposes to combine a robotic adjustable stick with an object detection system to create a more efficient method for obstacle detection. However,[9] does not address the balance problem in case of a collision, for example. Pole balancing research uses several approaches to stabilize a pole using a control strategy:

- PID (Proportional-Integral-Derivative) control adjusts the applied force based on the pole angle and its derivatives[11]
- Model-based control builds a mathematical model to predict the pole's behavior[12].

Approaches proposed in[11,12] are very complex and might not be feasible for real-time applications. Moreover, the actual physical implementations proposed are mostly industrial-based and unsuitable for a cane.

Several types of walking canes are available on the market, including quad, folding, or build-in-seat canes[13]. Some canes help with gait symmetry, are portable, and help alleviate conditions like carpal tunnel syndrome and arthritis[14]. However, they do not fully address the stability issues, leading to tripping or falling[15].

We propose addressing those problems by designing a stabilizing algorithm for a hypothetical robotic mobility cane that can self-stabilize via a moving cart at the bottom. The study is conducted by simulating inertial feedback and the position of the cane relative to the ground and applying a counterbalancing force to keep the person standing upright.

Method

A cane is fully stable if the cane is in a perpendicular position to the ground as defined in (TH_1). The cane stability is the amount of time a cane is in a stable position. The goal is to maximize the time a cane maintains the perpendicular position.

Environment 3D CartPole

We have created a cane balancing mechanism (Figure 2) by simulating it as a 3D extension of the canonical Cart Pole problem[16,17].

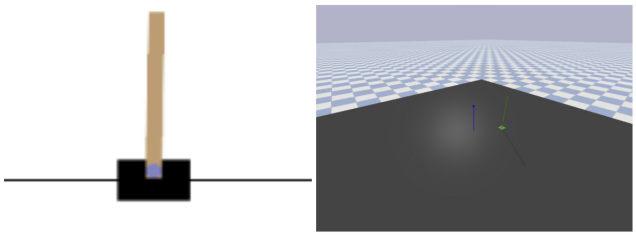

Fig. 2 Left -Original Cart Pole simulation from OpenAI[16] and Right -our simulation.

The cartpole is a suitable environment for evaluating cane stability because it is a simple and well-understood inverted pendulum system. In the case of a cane, the user will apply different forces that can lead to destabilization. This can be simulated with a cartpole sensitive to parameters such as the pendulum's length, jerk, mass, and friction. The cartpole physics is well-understood, allowing us to simulate the entire system's behavior.

The cart is simulated as a box that can move on a flat XY plane. We define the following constraints for the size of the box and pole (C1) and (C2):

- (C1) The box size is 1m radially from the origin
- (C2) The size pole is 2m x 0.1m x 0.1m (L x W x H)

Attached to that box is a pole that rotates along the two axes at its base. The pole is connected to the cart. However, the current version of the cartpole is 1D and can move left or right in the direction of +x or -x. We extrapolated this movement to 3 discrete forms of motion: +x, -x, +y, -y,-z, and +z (Figure 3). We aim to design a balancing controller/policy so that the pole stays vertical and supports the user. The system is actuated by moving the cart along the ground. Every time we move the pole, we apply force on the cart because the pole is attached to the cart, and torque results in the center mass of the pole.

We can make the pole stand close to vertically by using discrete cart movements. The cart will move continuously, trying to balance the pole. We have created a Gymnasium environment[17] to house our new version of the Cart Pole. The code has five components: step, get observations, reset, close, and reward. We coded in pybullet[18], a real-time physics simulator using 3D rendering for the cart and the pole.

Policy

We are using Proximal Policy Optimization (PPO)[19] since it is efficient and stable. PPO does not require complicated parameter

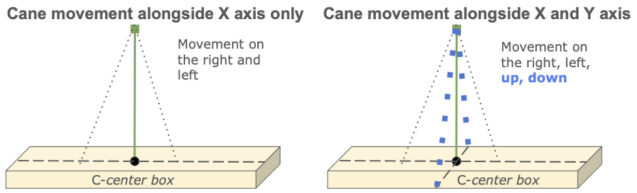

Fig. 3 (left) The current version of the cartpole can move along the X axis, and (right) our version can move in the X and Y axes.

optimization and can be applied to diverse tasks like our cane balancing one.

Termination condition

The experiment ends (with a failure) when the pole goes beyond a certain degree θ (Theta) from vertical or when the cart moves a certain distance from the center where it started. In both cases, the pole is unstable beyond repair and will destabilize the person using it. One possible reward method is to add a reward for every non-terminal step given by the (R_1) equation:

One possible reward method is to add a reward for every non-terminal step given by the (R_1) equation:

(R_1) $R_t = 1 + R_{t-1}$.

Another way to model the reward function is based on Θ, where the reward changes depending on how big ω is from vertical. We need to minimize Θ. The reward will be high for small values of Θ (R_2). The reward will be small or even negative for large values of Θ.

(R_2) $R_t = 100 - \omega^3 + R_{t-1}$, where ω is defined in (TH-1).

Another option is to normalize the value of ω in the $[0,1]$ interval (R_3).

(R_3) $R_t = \frac{\omega}{\omega_t} + R_{t-1}$.

The last reward policy evaluation was boned on the jerk. In robotics, jerk (the third derivative of the acceleration with respect to time, which is the change in acceleration per unit time) is one of the best ways to analyze the safety and stability of a movement. We want to keep the jerk as small as possible. A smaller jerk means smoother motion. To calculate the positions (Figure 4), we are using vector calculations given by

(V_1) $V_{bc} + V_p = V_{pc}$

(V_2) $\cos\omega = \frac{\mathbf{V_v}\cdot\mathbf{V_p}}{|\mathbf{V_v}|\cdot|\mathbf{V_p}|} \quad \Rightarrow \quad \omega = \cos^{-1}\left(\frac{\mathbf{V_v}\cdot\mathbf{V_p}}{|\mathbf{V_v}|\cdot|\mathbf{V_p}|}\right)$

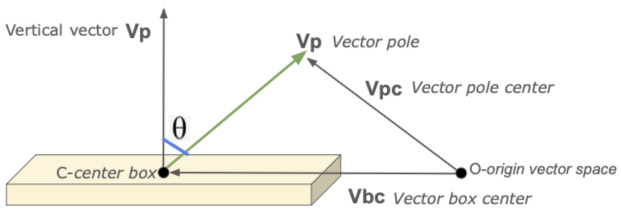

Fig. 4 Position vector calculations.

KPIs (Key Performance Indicators)

We chose three KPIs to evaluate the following performing reward function:

- Time standing = the amount of time in ms that the cane is standing.
- Angular velocity = rate of change of an object's position with respect to time as it rotates around a central axis.
- The measure of jerkiness = the amount of jerk.
- Reward value = the value of the reward function.

For each experiment, we computed the KPIs and the value of the Reward function.

Simulation description

A .urdf file in pybullet is used for 3D rendering and to define schematics, constraints, and shapes defined in (C_1) and (C_2). We have used the properties of vectors to measure the other positions:

A. The angle of the pole

B. Distance of the pole

C. Angular acceleration of the pole and the change in the angular acceleration, which is the jerk

D. Cartesian accelerator

E. Position of the pole

To find A-E above, we have used the getLinkState methods from the pybullet (Table 1):

Table 1: Different calculations for vectors of position.

Simulation calculations	
Position vector	Calculated with
(A), (B) and (E)	linkWorldPosition returns the Cartesian position of the center of mass
(C)	worldLinearVelocity returns Cartesian world velocity. Only returned if computeLinkVelocity is non-zero.
(D)	worldLinkAngularVelocity returns Cartesian world velocity. Only returned if computeLinkVelocity is non-zero.

Baseline

Our baseline is an RL model that only takes the angle into consideration. We will evaluate the time upright and compare it with time upright using our proposed method.

Results

Experiments were conducted by adjusting the following variables: learning rate, the number of epochs, weighting, reward, and number of steps. The weight function is defined by (FW_1):

(FW_1) $W(x_1,x_2,x_3) = w_1x_1 + w_2x_2 + w_3x_3$,
where $\sum_{i=1}^{3} w_i = 1$ and $x_1,x_2,x_3 \in \mathbb{R}$.

We define $x_1 = \text{angle}, x_2 = \text{time}, x_3 = \text{jerk}$. For example, $x_1 = 0$ means that angle variations are not considered, and $x_1 = 100$ means that the angle has 100 as the weighting average between the three numbers.

The reward function is defined by (FW_2):

(FR_2) $R(x_1,x_2,x_3) = b_1x_1 + b_2x_2 + b_3x_3$, where b_i is 0 or 1.

We define $x_1 = \text{angle}, x_2 = \text{time}, x_3 = \text{jerk}$. All are booleans with 0 meaning that the attribute does not contribute to the reward function, and 1 meaning that it does.

First, we established the error margin for the experiment values by running ten rounds with the same settings. We used the W(1,1,1) and R(0,0,1) to record the variability in the Average Timesteps, Average displacement, Average jerk, and Reward value, respectively. For each variable, we calculated the following values: minimum, 25 percentile, median, 75 percentile, and maximum across all ten measurements (Table 2).

Table 2: Evaluating experiment error margin

	Minimum value	25 percentile	Median value	75 percentile	Maximum value
Average timesteps	74.18397531	74.18397556	74.18398	74.18397638	74.18397626
Average displacement	0.704293718	0.704293997	0.704294	0.704294764	0.704294823
Average jerk	0.443876583	0.443876961	0.443877	0.443878162	0.443878282
Reward value	0.443876507	0.443876977	0.443878	0.443877686	0.443877569

Also, we plotted the BoxPlot to show the measure of experiment variability visually (Figure 5-8). The variation observed in the range of 1E10-6 in the experiment's outcomes is within the acceptable limits of <1E10-3 and thus does not change the conclusions being drawn.

The steps are the number of training steps. We ran several experiments (Table 3) with the following values for the W, R, and the steps:

We trained the RL model for each experiment, saved the raw data per iteration, made an aggregation per epoch, and then drew four graphs. In each graph, the X-axis is the number of epochs. The Y axis is different for each graph and shows the variation of time, distance from the vertical, degree of jerkiness, and the reward gain. For each experiment, we also calculated the average over all the epochs of all four metrics considered: timesteps, average distance from the vertical, average jerk, and average reward (Figure 9). The experiments were analyzed in

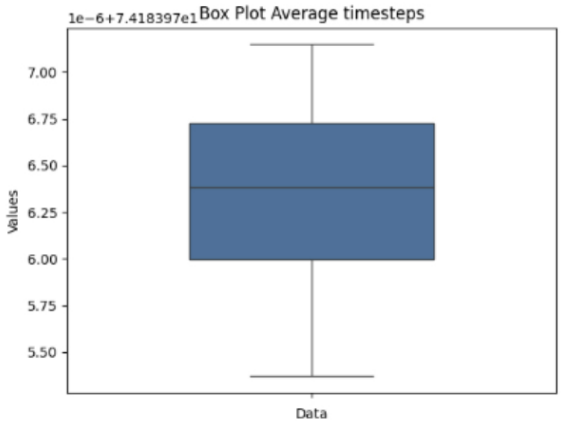

Fig. 5 Average timestamp experiment variability

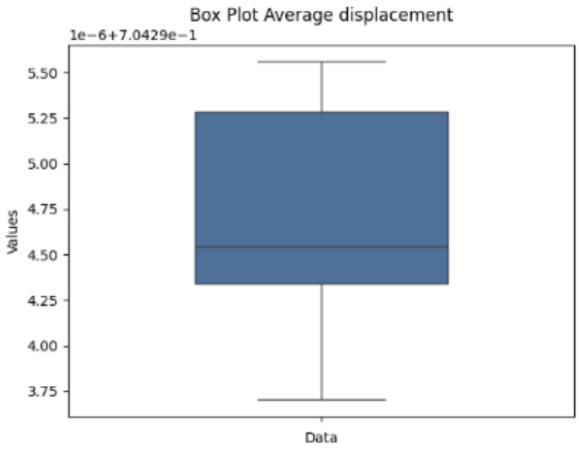

Fig. 6 Average displacement experiment variability

terms of the average time of standing. The best-performing experiment is the e1 with the average_total_timesteps of 74.18 sec (Figure 10). In Figure 10, we also plotted our benchmark, which has an uptime of 70.02 seconds. Experiment 12 (Figure 6) was performed using similar values to Experiment 1 except for increasing the training steps. The average_total_timesteps for this experiment was lower than experiment 1. The worst performing experiment was the e9 (average time of standing = 0.68 sec), where we increased the weight of the jerk to 10 from 1.

The second worst was e5, with an average standing time of 70.42 sec. We tried to increase the weight of the jerk, but the results were not satisfactory. By expanding the jerk to 10x or 100x like in experiments e8 and e10, we did not arrive at better for the average_total_timesteps. We also tried to double the number of training steps from 25000 to 50000 (Experiment 11), but the result was similar to that of Experiment 1 (Table 4).

For experiment 1, we evaluate the average timesteps per epoch (Figure 11).

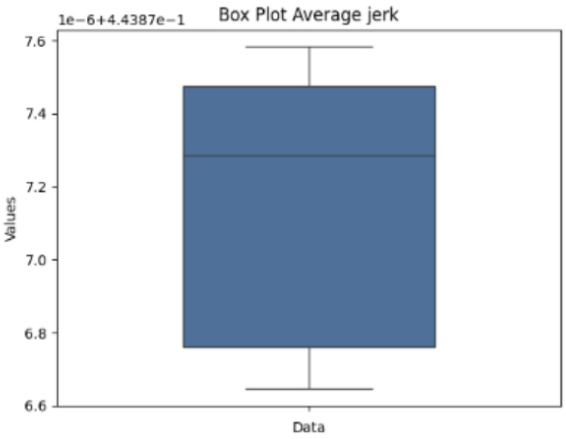

Fig. 7 Average jerk experiment variability

Fig. 8 Reward value experiment variability

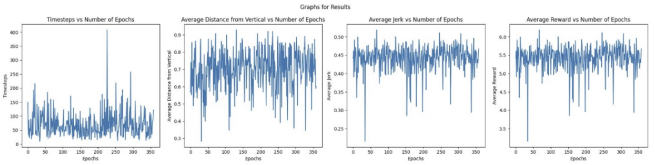

Fig. 9 Experiment data that shows the variation of time, distance from the vertical, degree of jerkiness, and the reward gain.

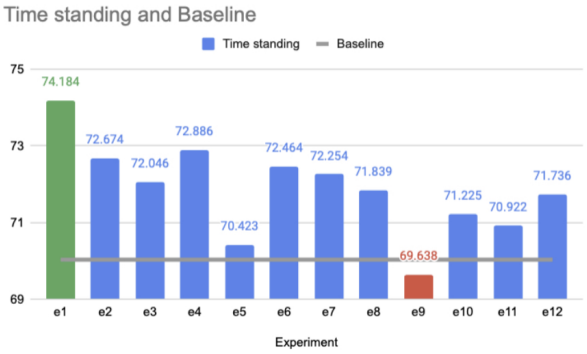

Fig. 10 Experiments ranked by timesteps showing the best in green (e1) and worst in red(e9). The horizontal grey bar represents the baseline as defined in the method section. The Y axis has been resized from 0-100 to 69-75 for clarity of visuals.

Discussion

The most crucial measure of cane stability is determined by the time that the cane stands. By this measure, the best number (in seconds) was achieved by a reward function skewed towards the jerk: R(0,0,1). We also found that training for over 25000 steps (Experiment 12) leads to overfitting. Figure 7 shows that the average time steps start to decrease after epoch 25000, ultimately reducing the time the cane is held upright. Most of our experiments perform better than the baseline, which means that our model better fits the problem we are trying to solve. The best-performing model is from experiment e1, which is 5.94% more performant than the benchmark.

We aimed to maximize the timesteps and the reward while minimizing the displacement and jerk. Experiment 1, our best performing experiment, also has the minimal number for jerk and close (one from the minimum) to the minimal number for the displacement. Thus, the initial hypothesis was refuted, and

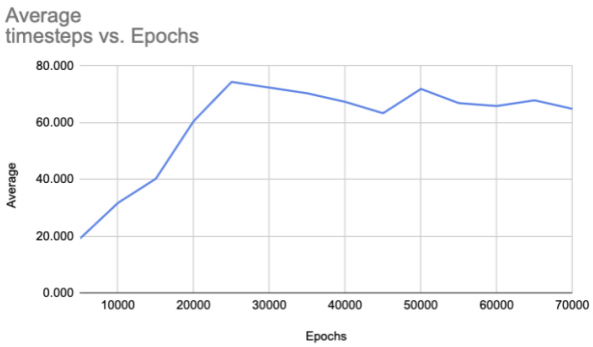

Fig. 11 Average timesteps per epoch.

Table 3: Experiments with the weight and reward function and number of steps.

Experiments with different weights, rewards, and steps			
Experiment ID	Weight function	Reward function	Number of steps
e1	W(1,1,1)	R(0,0,1)	25000
e2	W(1,1,1)	R(0,1,0)	25000
e3	W(1,1,1)	R(0,1,1)	25000
e4	W(1,1,1)	R(1,0,0)	25000
e5	W(1,1,1)	R(1,0,1)	25000
e6	W(1,1,1)	R(1,1,0)	25000
e7	W(1,1,1)	R(1,1,1)	25000
e8	W(1,1,10)	R(0,0,1)	25000
e9	W(1,1,10)	R(0,1,1)	25000
e10	W(1,1,100)	R(0,0,1)	25000
e11	W(1,1,1)	R(1,1,1)	50000
e12	W(1,1,1)	R(0,0,1)	50000

Table 4: KPI values for all experiments.

Experiments with different weights, rewards, and steps							
Experiment ID	Weight function	Reward function	Number of steps	Average timesteps	Average displacement	Average jerk	Reward function
e1	W(1,1,1)	R(0,0,1)	25000	74.184	0.704	0.444	0.444
e2	W(1,1,1)	R(0,1,0)	25000	72.674	0.704	0.446	1
e3	W(1,1,1)	R(0,1,1)	25000	72.046	0.709	0.446	1.446
e4	W(1,1,1)	R(1,0,0)	25000	72.886	0.707	0.447	0.707
e5	W(1,1,1)	R(1,0,1)	25000	70.423	0.709	0.442	1.152
e6	W(1,1,1)	R(1,1,0)	25000	72.464	0.711	0.444	1.711
e7	W(1,1,1)	R(1,1,1)	25000	72.254	0.707	0.446	2.153
e8	W(1,1,10)	R(0,0,1)	25000	71.839	0.704	0.445	4.449
e9	W(1,1,10)	R(0,1,1)	25000	69.638	0.704	0.442	5.424
e10	W(1,1,100)	R(0,0,1)	25000	71.225	0.684	0.445	44.476
e11	W(1,1,1)	R(1,1,1)	50000	70.922	0.708	0.444	2.152
e12	W(1,1,1)	R(0,0,1)	50000	71.736	0.698	0.446	0.446

we now contend that jerk is more valuable as a reward than the timesteps.

This result makes sense in practice as well. It will be very noticeable for a user if the cane is continuously out of balance. Also, jerkiness is the primary measure of comfort, and humans will notice it immediately. The angle is the least important factor since a human will not see a 1-2-degree difference in the inclination. We have met our objective of finding a method to balance the cane. The simulation results show that jerk is the main factor contributing to the balance. Jerk is intrinsically a better reward because, unlike timesteps, they provide feedback about how the current run is going. Timesteps only tell you the end of the run and nothing about how it went. Thus, jerk is a more informative reward, and kpi for how well the cane is doing would naturally lead to better rewards. Our results are confirmed by biological studies[20] on primates balancing mechanisms.

Conclusion

Current research in improving the functionality of a cane focuses on vision algorithms to create a more efficient method for obstacle detection. Still, the cane balance needs to be addressed, which is one of the essential aspects for a user. This paper addresses this problem by simulating a robotic cane with inertial feedback that constantly updates the position of the cane relative to the ground and applies a counterbalancing force to keep the person standing upright. Through simulation, using the Farama Gymnasium CartPole environment, we have validated the hypothesis that jerk contributes the most to the balance of a cane, which is defined as the time a cane stays vertically within a threshold. The results of this research can be further extended by building a physical mechanism that alleviates jerk.

This research targets three performance variables: timesteps, angle, and jerk, which we consider the main stability factors for a cane. However, many other factors like contact surfaces (wet asphalt, snow on the ground) or weather (rain, wind) can influence the stability of a cane. Moreover, more than our chosen parameter may be required to understand the problem space fully. The next step would be to conduct a grid search of hyperparameter space, the range of weights, and reward function combinations explored. This sensitivity analysis will further help to confirm our selection. Furthermore, we can consider extending by adding other parameters like weight, length, and friction and study the effect on the time upright. Additionally, our study is centered on modeling in simulation and does not tackle concerns of physical implementation. An extension of the study can consist of building a physical mechanism to ensure balancing based on the simulation results. For example, we can imagine a cartpole-like structure with a cane linked to a pivoting base. A control algorithm monitors the stability and maintains an upright position.

Generally, further actions are necessary to address the global society's challenges due to the aging population. We should focus research on such problems and build products that will make a difference in all our lives.

References

1 University of Vermont, *ScienceDaily*, 2024.

2 N. Gell, R. Wallace, A. LaCroix, T. Mroz and K. Patel, *Journal of the American Geriatrics Society*, 2015, **63**, 853–859.

3 National Federation of the Blind, *Free White Cane Program*.

4 S. Flaxman, P. Briant, M. Bottone, T. Vos, K. Naidoo, T. Braithwaite, M. Cicinelli, J. Jonas, H. Limburg, S. Resnikoff, A. Silvester, V. Nangia, H. R. Taylor, R. Bourne and J. Adelson, *Ophthalmology and Vision Science*, 2020, **61**, 2317.

5 R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction, Second Edition*, MIT Press, Cambridge, MA, 2018.

6 Ultralytics, *YOLO v8 model*.

7 B. N. Ilag and Y. Athave, *International Journal of Computer Applications*, 2019, **182**, year.

8 D. Kim and R. Emerson, *Journal of Visual Impairment Blindness*, 2018.

9 J. Bai, S. Lian, Z. Liu and K. Wang, *IEEE Transactions on Consumer Electronics*, 2018, **PP**, 1–1.

10 M. Tangjitjetsada, 2023 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Phanom, Thailand, 2023.

11 H. Li and M. Z. Ch, *ISA Transactions*, 2023, **134**, year.

12 M. Fu, *IFAC-PapersOnLine*, 2023, **56**, year.

13 *Assistive Device Training: Canes (Rehabilitation Therapy)*, 2024.

14 B. MK, S. M, S. D, T. J, K. MV, B. K, I. E and M. WE, *Physiotherapy Canada*, 2009.

15 Senior Supported, *Best Quad Canes – Everything You Need to Know About Them*.

16 A. G. Barto, R. S. Sutton and C. W. Anderson, *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, **SMC-13**, 834–846.

17 Farama Gymnasium, *Cartpole*, 2004.

18 Bullet Physics SDK, *Real-time Collision Detection and Multi-Physics Simulation for VR, Games, Visual Effects, Robotics, and Machine Learning*.

19 J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, 2017.

20 N. Hogan, *Journal of Neuroscience*, 1984, **4**, 2745–2754.