Using Reinforcement Learning to improve stability for real-time physics simulated mobility aids

By **Stefan Maxim, Kiyn Chin**

Abstract

Globally, mobility aid usage is estimated to increase significantly. Current research is focused on obstacle detection and classification but does not address the cane balance problem, which is one of the most important aspects for a user. The study objective is to find a theoretical model for understanding the stability of a cane. Utilizing a real-time physics simulator to model the environment and a Reinforcement Learning model, we conducted experiments to determine the parameters that most influence balance. We have 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. We defined cane stability by the time the cane is vertical within a threshold. By this measure, the longest time the cane was upright was achieved by attributing equal weights to the angle, time, and jerkiness in the weight function, and the reward function skewed towards the jerk. We also found that training for more epochs will lead to overfitting and thus reduce the timesteps. This research may be extended further by looking into building a physical mechanism that alleviates jerk. With the increasing percentage of the aging population, we believe that more research should be focused on having an impact on their safety.

Keywords: mobility aids, cane balance, physics simulator, reinforcement learning.

Introduction

Usage of mobility aids like canes has exploded in recent decades and is now necessary for a sizable percentage of the global population [1]. The US alone has seen an estimated 26% increase in cane usage for its population of people aged 65+ compared to just the last decade. Unfortunately, canes also lead to instances of tripping or falling[2] more frequently than other mobility aids like walkers and wheelchairs. Additionally, for visually impaired individuals, using a cane is necessary to walk independently and safely [3]. Currently, approximately 304.1 million people have moderate to severe blindness while around 49.1 million are completely blind, a stark 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 for others. Given the aging population increase and the even more steep increase in visually impaired people, we believe this problem is worth investigating.

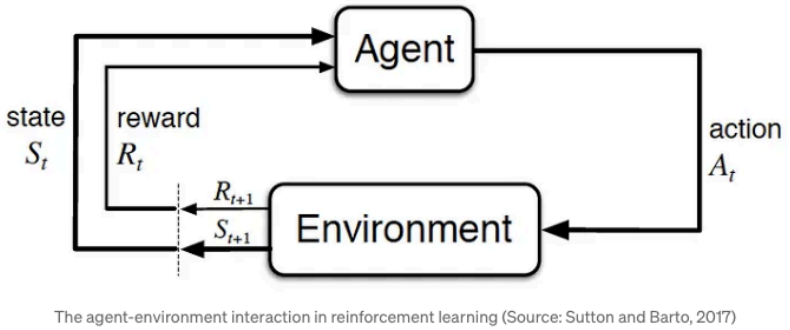
Current research is mostly limited to attaching cameras or other position-detecting tools and using them to map the surroundings. One such method is using deep-learning models like YOLOv8[5], common in image deception software to detect the position of one's surroundings and alert the user when they approach obstacles[6] [7]. 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 [8]. Lastly, another paper[9] proposes to combine a robotic adjustable stick with an object detection system to create a more efficient method for obstacle detection. However, this paper does not address the balance problem in case of a collision for example. We propose addressing this problem 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 dependent on that data to keep the person standing upright.

The results of this research, focusing on a control algorithm that aims to stabilize a cane, will enhance the field's understanding of the parameters important for cane stabilization. Following simulation results, new insights can be gained into materials needed to build tomorrow's canes to increase stability.

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. This paper intends to determine what type of reward function will maximize the time the cane will be vertical enough to support someone's weight. The cane is considered upright if θ (the deviation from vertical) satisfies the following threshold:

(TH_1) $-90^0 \leq \theta \leq 90^0$
(TH_2) $-12^0 \leq \theta_t \leq 12^0$

We hypothesize that the number of 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 its orientation trying to support the person using it. There are several parts to improve stability for real-time physics simulated mobility aids: software, material science, and mechanics. The software part is responsible for using physics formulas to simulate the physical dynamics of the cane. The material science will focus on the materials necessary to build such a cane. The mechanics part will focus on the mechanism to deploy the resulting decision of how to stabilize the cane. 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. Real data is difficult to gather for unsafe scenarios. Simulation makes it faster to try many combinations of variables in a short amount of time. We are proposing using OpenAI simulation and a classic Reinforcement Learning (RL) [10] loop where the agent is evaluating several reward functions. RL is a type of machine learning where an agent is taking action by learning how to maximize a predefined reward function. In Figure 1, the agent is trying to optimize function R and moves from the state S_t to state S_{t+1}.



The agent-environment interaction in reinforcement learning (Source: Sutton and Barto, 2017)

Figure 1: The reward loop in a RL environment

Our method consists of defining the metrics and using close-loop experiments to collect the resulting actual value of the metrics. By comparing different experiment results, we will understand the key factors influencing cane stability.

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 measured as the amount of time the 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 balancing mechanism (Figure 2) for a cane by simulating it as a 3D extension of the canonical Cart Pole problem[11] [12].

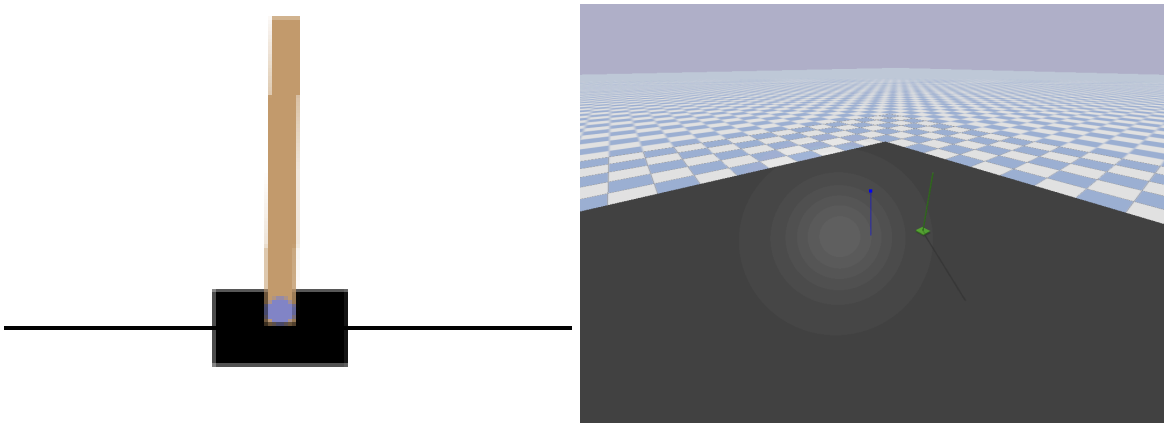


Figure 2: Left -Original Cart Pole simulation from OpenAI [11] and Right -our simulation.

The cart is simulated as a box that can move on a flat *XY* plane. We define the following constraints:

(C_1) The box size is 1m radially from the origin

(C_2) 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 about two axes at its base where it is attached to the cart. However, the current version of the cartpole that exists is 1D dimension and it can move left or right in the direction of +x or -x. We extrapolated this movement to 3 discrete forms of motion and now it can move +x, -x, +y, -y, and -z and +z (Figure 3 shows movement in X and Y). We aim to design a balancing controller/policy for the pole to stay vertical to support the user. The system is actuated by moving the cart along the ground. Every time we move the pole we apply a force on the cart because the pole is attached to the cart. torque results about the center mass of the pole. By using discrete movements of the cart we can make the pole stand close to vertically. The cart will move continuously trying to balance the pole.
We have created a Gymnasium environment [12] to house our new version of the Cart Pole. The code has 5 main components: step, get observations, reset, close, and reward. It was coded in a real-time physics simulator (pybullet) [13] using 3D rendering for the cart and the pole.

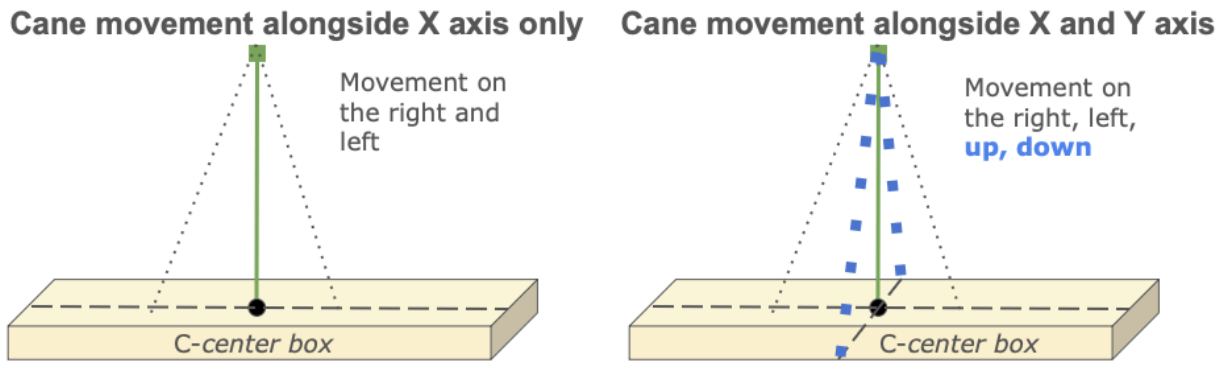


Figure 3: (left) Current version of cartpole that can move along the X axis and (right) our version can move in X and Y axis.

Policy

We are using Proximal Policy Optimization (PPO [14]) since it is efficient and stable. PPO does not require complicated parameter 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.

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

Another way to model the reward function is based on Theta, where the reward changes depending on how big the θ is from vertical. We need to minimize theta. For small values of Theta, the reward will be high. For large values of Theta, the reward will be small or even negative.

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

Another option is:

(R_3) $R_t = \theta/\theta_t + R_{t-1}$

which will normalize the value in the [0,1] interval. This will convert the Theta from radians.

The last reward policy evaluation was based 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.

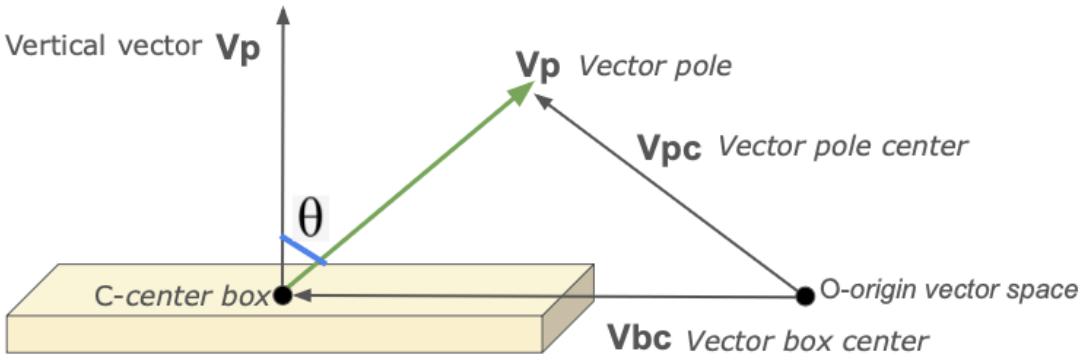


Figure 4: Position vector calculations.

To calculate the positions (Figure 4) we are using vector calculations:

$$(V_1)\ Vbc + Vp = Vpc$$

$$(V_2)\ \cos \boldsymbol{\theta} = \frac{Vv \,.\, Vp}{|Vv| \,.\, |Vp|} \Rightarrow \boldsymbol{\theta} = \cos^{-1} \frac{Vv \,.\, Vp}{|Vv| \,.\, |Vp|}$$

KPIs (Key Performance Indicators)

We chose three KPIs for evaluating the next 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 like:

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):

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

Table 1: Different calculations for vectors of position.

Results

Several experiments were done adjusting the AI model parameters, the learning rate, and the number of epochs. There are three main variables: weighting, reward, and number of steps. The weight function is defined by:

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

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

The reward function is defined by:

$$(FR_1)\ R(x_1, x_2, x_3) = b_1x_1 + b_2x_2 + b_3x_3 \text{ where } b_i \text{ is 0 or 1}$$

We define x_1=angle, x_2=time, x_3=jerk. are all booleans with 0 meaning that the attribute does not contribute to the reward function and 1 meaning that it does. The steps are the number of training steps. We ran several experiments (Table 2) with the following values for the W, R, and the 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 2: Experiments with the weight and reward function and number of steps.

For each experiment, we trained the RL model, 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. The experiments were analyzed in terms of the average time of standing. The best-performing experiment is the e1 with the average_total_timesteps of 74.18 sec (Figure 5).

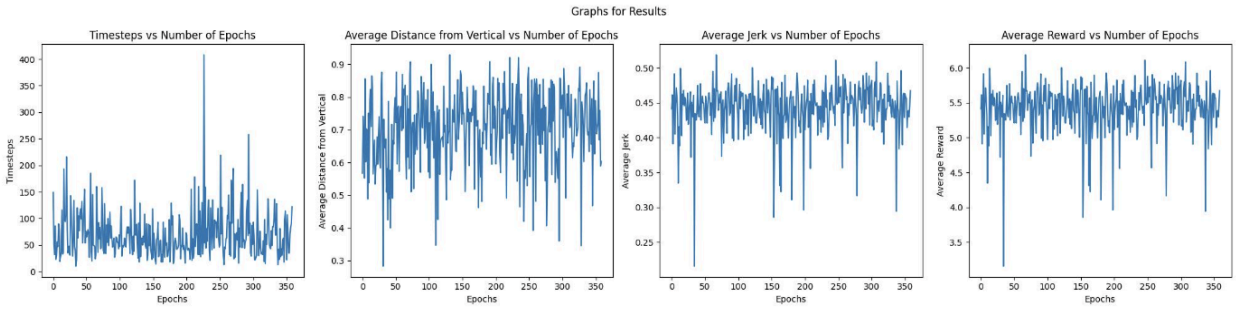


Figure 5: Experiment e1 with constant weights and focus on time to vertical.

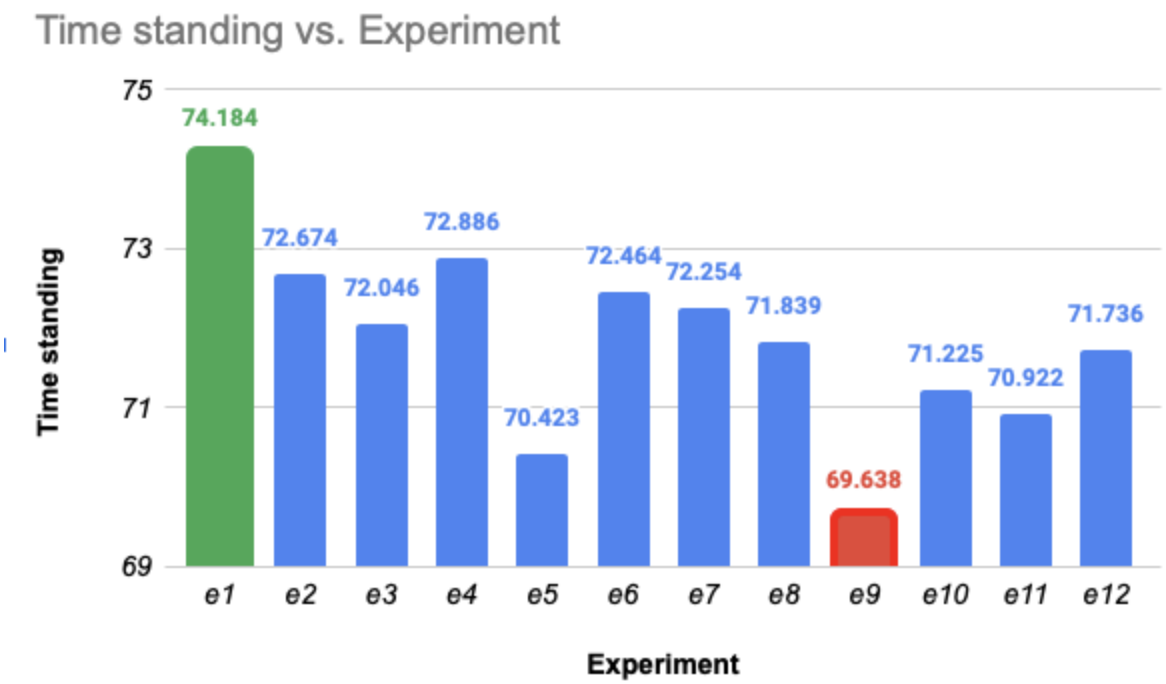


Figure 6: Experiments ranked by timesteps showing the best in green (e1) and worst in red(e9). The Y axis has been resized from 0-100 to 69-75 for clarity of visuals.

Experiment 12 (Figure 6) was performed using similar values as 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 time of standing of 70.42 sec. We tried to increase the weight of the jerk but the results were not satisfactory. By increasing the jerk to 10x or 100x like in experiments e8 and e10 respectively 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 not better than Experiment 1 (Table 3).

Experiments values for timesteps, distance, jerk and reward							
Exp ID	Weight function	Reward function	Number of steps	Average timesteps	Average displace-ment	Average jerk	Reward value
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	1.000
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

Table 3: KPI values for all experiments

Discussion

The most important 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 more than 25000 steps (Experiment 12) leads to overfitting and thus reduces the time the cane is held upright.

Our goal was to maximize the timesteps and the reward while minimizing the displacement and jerk. Experiment 1 also has the minimal number for jerk and close (one from the minimum) to the minimal number for the displacement. This enforces the idea that an equally weighted average is the best combination. Thus, the initial hypothesis was refuted, and we now contend that jerk is more useful as a reward than the timesteps.

This result makes sense in practice as well. For a user, it will be very noticeable if the cane is continuously out of balance. Also, jerkiness is the main measure of comfort and humans will notice jerkiness immediately. The angle is probably the least important factor since a human will not notice a difference of 1-2 degrees 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 the run itself went. Thus, jerk is a more informative reward and kpi for how well the cane is doing, so it would naturally lead to better rewards. Our results are confirmed by biological studies[15] on primates balancing mechanisms.

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. 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 mechanism to ensure balancing based on the simulation results.

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

Conclusion

Current research in improving the functionality of a cane focuses on vision algorithms to create a more efficient method for obstacle detection but does not address the cane balance which is one of the important aspects of 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 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. With an increasing percentage of the aging population, we believe that more research should be focused on having an impact on their safety.

References

1 University of Vermont. (2015, May 11). Friend or foe? Study examines seniors' increasing use of walking aids. ScienceDaily, 2024.

2 N.M. Gell, R.B. Wallace, A.Z. LaCroix, T.M. Mroz, K.V. Patel. Mobility device use in older adults and incidence of falls and worry about falling. 2011-2012 national health and aging trends study. J Am Geriatr Soc. 2015 May;63(5):853-9. doi: 10.1111/jgs.13393.

3 National Federation of the blind. Free White Cane Program, https://nfb.org/programs-services/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.A. Bourne, J. Adelson. Global prevalence of blindness and distance and near vision impairment in 2020. Ophthalmol. Vis. Sci., page 61(7):2317, 2020.

5 Ultralytics YOLO v8 model, https://docs.ultralytics.com/models/yolov8/

6 B. N. Ilag, Y. Athave. A Design review of Smart Stick for the Blind Equipped with Obstacle Detection and Identification using Artificial Intelligence, International Journal of Computer Applications (0975 – 8887) Volume 182 – No. 49, April 2019.

7 D.S. Kim,R.W. Emerson. Obstacle Detection with the Long Cane: Effect of Cane Tip Design and Technique Modification on Performance. J Vis Impair Blind. 2018.

8 J. Bai, S. Lian, Z. Liu, K. Wang. Virtual-Blind-Road Following Based Wearable Navigation Device for Blind People.
March 2018, IEEE Transactions on Consumer Electronics PP(99):1-1 .

9 M. Tangjitjetsada. Development and study of the impacts of automated walking aids on the elderly, 2023 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Phanom, Thailand, 2023.

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

11 A. G. Barto, R. S. Sutton and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 5, pp. 834-846, Sept.-Oct. 1983.

12 Farama Gymnasium - Cartpole (2004)
https://gymnasium.farama.org/environments/classic_control/cart_pole/

13 Bullet Physics SDK: real-time collision detection and multi-physics simulation for VR, games, visual effects, robotics, machine learning etc. https://github.com/bulletphysics/bullet3.
14 J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. Proximal Policy Optimization Algorithms, 2017.
15 N. Hogan. An organizing principle for a class of voluntary movements. J Neurosci. 1984 Nov;4(11):2745-54.