2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Design and Model-Free Reinforcement Learning Based Control of a Modular Self-Balancing Robotic System

Mishek Musa and Uche Wejinya

I. Motivation

THE concept of utilizing a modular version of the two-wheeled self-balancing robot is motivated by the need for modern robotic systems to be scalable, making it possible to build robots of various sizes and capabilities by adding or removing modules, redundant in the case of system failure, and reconfigurable and customizable for industries with unique requirements, such as manufacturing, healthcare, and agriculture [1]. With this change comes the added complexity of dealing with a high degree of parametric uncertainty due to variations in parameters like mass, friction, and center of gravity. This necessitates the design of a control system capable of overcoming these issues and so model-free reinforcement learning (RL) is investigated for this purpose [2].

II. Methods

Robot Hardware Design: The robot design can be broken down into two main sections: the base module and the modular components, see Fig. 1(a). The base module of the robot is responsible for the primary functions of the robotic system and consists of the wheels, motors, drive-train, and control electronics. The main structural elements of the robot are made up of CNC and 3D printed components, and the drivetrain is driven by two brushless DC motors, see see Fig. 1(b). Motor commands are sent via a UART communication line from an Arduino Mega microcontroller to the motor controller. The orientation of the robot is monitored using an MPU6050 inertial measurement unit (IMU) whereby angular velocity and acceleration information is processed through a Kalman filter to report the pitch angle of the robot, see Fig. 1(d). The modular concept of the robot will consist of independent modules that will interface with the subsequent module using self-aligning pins to allow for ease of attachment. Once connected, spring-loaded electrical contacts will connect the module to the rest of the system for two purposes, power and data transmission. As the connected module receives power, a servo-locking mechanism will fix the module onto the rest of the system so that it becomes rigidly attached. Module functionality will include storage, sensing, & actuation.

Model-Free Reinforcement Learning Control: A simplified version of the robot was modeled using the URDF format and deployed using the pyBullet physics engine environment [3]. To simulate the modular self-balancing robotic system, the mass, friction, and inertia properties were randomized within the estimated working range of the robot. Three RL algorithms were then used to learn an optimal control system to maintain the robot in an upright position, namely Deep Q Network (DQN), Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO) [4].

M. Musa and U. Wejinya are with the Department of Mechanial Engineering, University of Arkansas, Fayetteville, AR, 72703 USA e-mail: mjmusa@uark.edu

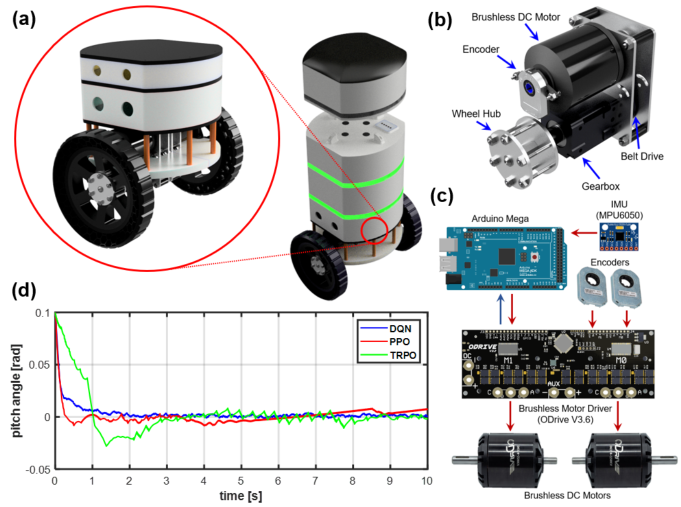

Fig. 1. (a) Modular self-balancing robotic system, (b) robot drive train, (c) robot control electronics, (d) RL balancing result

III. Results

The training of the models were done in a discrete manner due to the computational constraints. In each case, the robot was given an initial pitch angle of 0.1 rad. The learned control policies are able to achieve stability of the robot, see Fig. 1(c). It can be seen that the DQN model gives the best performance with a quick response time and no overshoot compared to PPO and TRPO that overshoot the upright position and have significant oscillations.

IV. Discussion & Conclusion

A modular self-balancing robot can be very useful in a variety of mobile robot applications. In this work, the design of the system is presented and initial simulation studies are performed to investigate the use of RL control algorithms to adapt to changes in the system plant. Future work will focus on deploying these algorithms in the real world environment.

References

[1] Jungwon, S. et al. "Modular reconfigurable robotics." Annual Review of Control, Robotics, and Autonomous Systems 2 (2019): 63-88.
[2] Sutton, R. S., and A. G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
[3] Coumans, E., and B. Yunfei. "Pybullet, a python module for physics simulation for games, robotics and machine learning." (2016).
[4] Raffin, A., et al. "Stable-baselines3: Reliable reinforcement learning implementations." Journal of Machine Learning Research 22.268 (2021)