Project Report on the design and development of a Self-Balancing Stick

Samarth Katiyar

26/07/24

Introduction

The self-balancing stick, also known as an inverse flywheel pendulum, is a control systems project that demonstrates principles of dynamics, control theory, and robotics. The system consists of a vertical stick attached to a flywheel that can spin, allowing the stick to balance itself by adjusting the flywheel's angular momentum. The primary goal is to keep the stick upright by continually adjusting the torque applied to the flywheel based on feedback from the gyroscopic sensor.

The self-balancing stick is an example of an unstable system that can be stabilized through active control. It involves understanding and implementing feedback control loops, where the angular position and velocity of the stick are monitored, and appropriate control signals are sent to the motor driving the flywheel to maintain balance.

This report outlines the development and evaluation of three versions of the self-balancing stick, focusing on the motors and mechanisms used to achieve the desired balance.

Electronics used:

1. RS775
2. RMCS3016
3. DXL XL430-W250
4. Arduino Nano (For V1 and V2)
5. Arduino Mega (For V3)
6. Adafruit BNO055
7. Cytron MDD10A motor drive (For V1 and V2)

Version 1: RS775 Motor with 3D Printed Attachment and Flywheel

Description

The first version of the self-balancing stick utilized an RS775 motor. A 3D printed attachment connected the motor to a customized stand to alter pendulum length, the flywheel was equipped with screws for added torque, was used to provide the necessary angular momentum.

A grey metal object with holes

Description automatically generated A black wheel on an orange object

Description automatically generated

Fig2: RS775 sensor and pipe brace setup

Fig1: Version 1 setup



Fig3: RS775 flywheel

Pros

* Availability and Cost: RS775 motors are widely available and relatively inexpensive.
* Simple communication: The motor is controlled using a preexisting setSpeed function that changes the voltage provided to the motor, making the speed control very precise.

Cons

* Heavy weight: The RS775 motor weighs around 350g[1], contributing a big portion to the overall weight of the system. A concentrated weight around the center of the flywheel reduces the inertia making it extremely difficult for the flywheel torque to counteract the natural gravitational force from the falling pendulum.
* High RPM: RS775 can go up to very high speeds at peak voltage, resulting in lesser control over the acceleration/deceleration of the flywheel. Such high speeds also led to vibrations in the motor that eventually led to inaccuracies in gyroscope readings.
* Low torque: The peak torque of the RS774 is around 0.079Nm[1], compared to the overall weight of the system, it is insufficient to stabilize the system.

Result: Due to the motor torque not meeting the requirements, and the high RPM leading to less control over the system. This version was not viable.

Version 2: RMCS3016 Geared Motor with 3D printed brace and flywheel

Description

The second version employed an RMCS3016 geared motor, maintaining a similar flywheel and pipe connection design as the first version.

A circular object with holes

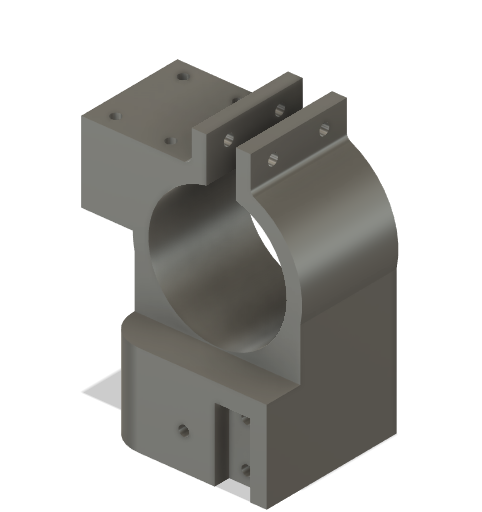
Description automatically generated

Fig5: RMCS3016 flywheel

Fig4: RMCS3016 sensor and pipe brace setup

Pros

Pros

* Increased Torque: The geared motor provided significantly more torque with a peak torque of 0.444Nm[2]. The motor was also lighter at 180g[2]. Giving it a higher weight:torque ratio, enabling us to use jerk motions to bring the system to equilibrium.
* Low RPM: A lower RPM compared to the RS775 eliminated vibrations making the gyroscope readings consistent and made the system easier to control.

Cons

* Axle position: The axle was not centered in the motor, so flywheel attachment and motor orientation were manually done, leading to slight inaccuracies in the alignment of the flywheel with the pivot.
* Insufficient torque: While the torque was higher compared to the RS775 motor, only the stall torque was sufficient to bring the motor to equilibrium but using a jerk-based motion to stabilize the system was impractical and puts a strain on the motor.
* Delicate: The gearbox of the motor was delicate as the fast switching of directions during test runs eventually eroded the gearbox leading to it being scrapped.

Result: While the torque provided by the motor was not enough to balance the system even after increasing the flywheel mass and radius. The motor also was unable to handle the rapid direction switches and the gearbox got damaged and the version was scrapped.

Version 3: DXL XL430-W250 Servo Motor

Description

A table with wires and a piece of paper on it

Description automatically generatedThe final version utilized a DXL XL430-W250 servo motor. It had a similar pipe connection as the previous two versions but a metal beam instead of a flywheel was used as the balancing weight. This version achieved the desired results and exhibited optimal performance.

Fig 6: Version 3 setup

Pros

* Lightweight: The servo motor weighs only 57.2g[3], which is significantly lighter compared to the other motors making majority of the weight coming from the flywheel itself. Giving it a much higher inertia.
* Torque: The motor had a substantially higher peak torque of 1.5Nm[3]. This made it possible to balance the system without excessive movements. Stabilizing the system in partial rotations.
* Reliability: The servo motor was robust and reliable, ensuring consistent performance over extended periods.

Cons

* Higher Cost: The DXL XL430-W250 servo motor was the most expensive option among the three versions.
* Flimsy weight connection: The servo motor was connected to the metal beam with two screws and additional weight to increase inertia were taped to the ends of the beams. This made the beam connection not as sturdy as desired and required occasional maintenance.

Result: The system was able to balance itself even with external disturbances of up to 2 degrees on either side.

|  |  |
| --- | --- |
| Challenges | Solutions |
| Finding the right shaft length: It was difficult to find the right length of the shaft to balance as the shorter the length the easier it was to control using the motor torque but the harder it was to control due to the increased natural frequency of the system. | Made a 3D printed attachment that could allow adjustment of the pipe length from the pivot to manually tune the system to find the right balance. |
| Insufficient inertia of the flywheels | Adjusted the 3D printed flywheels to allow screwing in bolts to increase the weight around the circumference, giving it a higher inertia, and taped weights to the metal beam. |
| Sensor reading inconsistencies: The gyroscope did not give consistent readings for the same position | Integrated a manual calibration before the main loop into the program to ensure accurate and consistent values. |
| Pipe overshot trying to stabilize itself: Instead of a dampening oscillation the flywheel often overshot and fell to the other side while trying to balance itself due to the stalling torque of the motor. | Added a conditional deceleration to prevent stalling torque from making the pipe fall back if the pipe is coming back to origin. |

Programming

Initially, a PID control was used for the first two version, this included using change in angle orientation to find the angular velocity of the system and include that in the calculations to set speed to the motor, however, the first two version were ones that weren’t physically capable of balancing the system and the final version did not require the use of PID control. The final control for version 3 used a P control with previous state logic and conditional deceleration. There was an absolute threshold value before which the motor did not move as the system remained stable within those bounds. Once crossed the p constant multiplied by the orientation angle was the power to the motor. Using previous state logic, If the system was going back to original, the p constant remained to ensure a gentler decrease in motor speed and there. During the process of going back to origin there was another ‘dead-zone’ during which there was no power supplied to the motor so as to make use of the initial momentum from the starting peak torque when the thresholds were crossed.

Conclusion

The development of the self-balancing stick progressed through three distinct versions, each with its own set of advantages and challenges. The RS775 motor provided a cost-effective starting point but lacked the necessary torque for effective balancing. The RMCS3016 geared motor improved the torque but was not sturdy enough and got the gearbox ruined due to the constant switches in direction. The final version, utilizing the DXL XL430-W250 servo motor, achieved the desired results with precise control and reliable performance, albeit at a higher cost. It allowed up to a ±2-degree deviation from origin. The system could be made better to stabilize bigger deviations with a stronger motor that provided significantly more torque as other factors like height and inertia could only be adjusted so much leaving the motor torque as the deciding factor in how much deviation could be stabilized in the system. This iterative development process highlights the importance of selecting appropriate components and the trade-offs involved in achieving optimal system performance.

Working Video link to a video of the working project:

<https://www.youtube.com/shorts/577FF8Fknhw>

Github repository link:

<https://github.com/SamarthK154/Inverse-Pendulum.git>

Acknowledgments

I would like to express my gratitude to Mr. Vyankatesh Ashtekara for his invaluable guidance and support throughout the project. His expertise and mentorship were instrumental in the successful completion of the inverse pendulum project. I am also deeply thankful to Dr. Ashish Dutta for granting permission and providing access to the necessary equipment at the center of mechatronics at Indian Institute of Technology Kanpur. Their assistance has been crucial to this project's development and implementation.

References

[1] - HandsOn Tech. "775 DC Motor: Product Specifications." Accessed July 26, 2024. <https://www.handsontec.com/dataspecs/motor_fan/775-Motor.pdf>.

[2] - Robokits. "RMCS-3016 V1: Robokits India Product Manual." Accessed July 26, 2024. <https://robokits.download/downloads/RMCS-3016_V1.pdf>.

[3] - Robotis. "XL430-W250: e-Manual." Accessed July 26, 2024. <https://emanual.robotis.com/docs/en/dxl/x/xl430-w250/>.