Jurnal Polimesin
Department of Mechanical Engineering
Lhokseumawe State Polytechnic
http://e-jurnal.pnl.ac.id/polimesin
e-ISSN : 2549-1999 No : 2 Month : August
p-ISSN : 1693-5462 Volume : 20 Year : 2022

Received : 2022-01-13 Accepted : 2022-04-22
Revised : 2022-04-20 Available online : 2022-08-31

Synchronization of Treadmill Speed to Bicycle Model Speed Based on PID Controller

Baskoro Latu Anurogo
Mechanical Design Engineering, University of Sanata Dharma
Sleman, 55282, Indonesia
*Corresponding Author: baskorolatu@usd.ac.id

Abstract

Bad weather and crowded vehicle lanes are often obstacles for cyclists during practice. Bicycle treadmills are the right solution to answer these demands. A person can cycle on a treadmill without having to leave the room. The rotation speed of the treadmill will be determined by the rotational speed of the bicycle wheels, so the bicycle will not fall off the treadmill. For this reason, this study aims to design a PID-based controller that will control the rotational speed of the treadmill so that it is in sync with the rotational speed of the bicycle wheel. The rotation of the bicycle wheel is represented by the wheel of a bicycle model that rotates on a treadmill, while the treadmill is represented by a model in the form of a conveyor. The results showed that in the first second, the conveyor speed was not the same as the speed of the bicycle model. Even so, after 1.5 seconds, the conveyor can catch up to the speed of the bike model. Despite some glitches, the conveyor can keep up with the speed of the bike model.

Keywords:
Treadmill, bicycle, microcontroller (Arduino), motor DC, PID

1 Introduction

Sport is an important need for humans to maintain physical and spiritual health. One of the most popular sports is cycling. As quoted by Muhammad Rizal and friends in the Journal of Public Health in 1986, it was stated that cycling activities would reduce the risk of death by 40% [1].

Cycling has become a hobby for many people, one of which is using a road bike. This can be proven by the increasing number of cycling activities in Indonesia, with the emergence of several bicycle communities and the emergence of new athletes. The obstacle that athletes often encounter when exercising on road bikes is unfavorable weather conditions, for example, when it rains. When it rains, cycling outside can threaten the safety of cyclists, but also threaten the health of athletes. To solve some of these problems, we need a tool for cycling training needs. A bicycle treadmill is one solution that can be used to answer these problems. This research became the basis for designing a bicycle treadmill.

This research begins with the results of research by Zuly Budiarso and his friends about calculating the speed of the motor to move the robot's wheels, where the movements are in the form of forwarding, backward, circular motion, turning, and winding. The movement will be obtained with the difference in RPM of the two driving motors (DC motors) of the right and left wheels. Setting the speed of the DC motor as the driving force of the robot car is done by adjusting the delay time in the program made with Arduino. So it can be said that the DC motor will rotate based on the frequency of the digital pulses generated by the Arduino, and the pulse frequency is set through the program [2]. In contrast to the research conducted by Dio Taufiq Arif and Aswandi, M.T., the control system still uses Arduino while the signal generator is Pulse Width Modulation (PWM). PWM can regulate the rotation speed of the DC motor. In addition to PWM and Arduino, there are many variations of the DC motor speed regulation system. Muhammad Irhas and his friends analyzed DC motor speed regulation with PID control. PID control is a standard control that can be used to correct errors between measurement values and deviations. By tuning or control parameters obtained a good response from a PID control [3]. Various variations in DC motor speed regulation can be carried out with controls that use conventional calculations or smart controls, such as research conducted by Safah Tasya A. and friends. In this research, we compare the PI control system and fuzzy control [4].

Reread from existing references, the design of a special treadmill for bicycles will begin by synchronizing the rotation of the bicycle wheel and DC motor on the conveyor.

The purpose of this study was to test the performance of the PID controller used to synchronize the speed of the treadmill to the actual speed of the bicycle model running on it. This synchronization is carried out based on the real speed of the treadmill and the real speed of the bicycle model so that a flat or sloping surface has no effect.

2 Research Method

This research was conducted through four steps, as shown in Fig. 1.

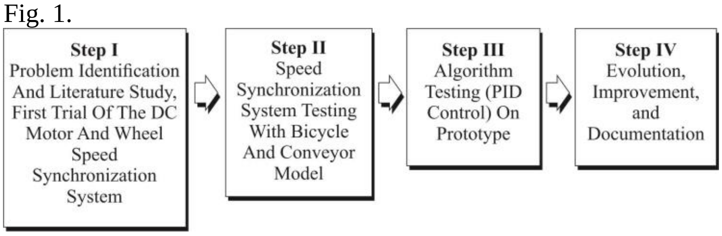

Fig. 1. The Steps of research methods

2.1 Step I

This step is in the form of a specific study to study the alignment (synchronization) of the RPM of a bicycle wheel with the RPM of the DC motor. Alignment experiments will be carried out with a tool in the form of modelling a bicycle wheel, where the rotational speed of the bicycle wheel will be captured by a speed sensor (optic), and then the data obtained will be processed on the Arduino UNO (first) into RPM speed data. The characteristics of sensors that only produce analogue quantities are a challenge in applying digital technology using sensors. The change from an analogue system to a digital system is one of the things that became the beginning of the development of a digital system. By changing the analogue system to a digital control system, the types of devices used also change [5].

Then the RPM data is sent to the Arduino UNO (second), which contains the PID program. In this PID program, the wheel speed will be the reference speed for the conveyor, which is a model of the treadmill. PID controller is the most popular controller in the industry [6][7]. This is because it has a simple structure and effectiveness [8]. The problem that often arises is the tuning [6]. Even so, many tuning methods have been offered [9]. The most widely used method is the Ziegler-Nichols method [10]. However, this method still produces a large overshoot and is not very strong against load variations [11]. A more robust method against variations in load is the Cohen-Coon method. However, this method still produces a large overshoot and an oscillating response [12]. PID controller block diagram shown in fig. 2.

The PID control system is a system that has a working system in a closed-loop method, consisting of three controls, proportional (P), Derivative (D), and the integral (I), where each has strengths and weaknesses in the response. Uniting The three controls aim to maximize the system in achieving the set points that have been given. In the application, the system requires to set the value of Kp, Ki, and Kd, so that the resulting output response signal corresponds to the desired [13].

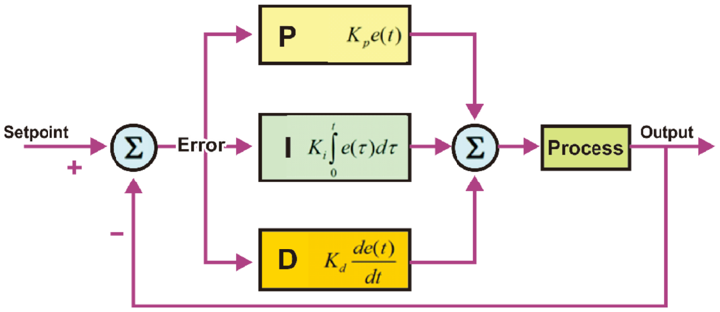

Fig. 2. PID controller block diagram

. The speed of the DC motor driving the treadmill model will be read by the rotary encoder so that the reading results become feedback for the Arduino UNO (second). Then the speed between the rotating wheel and the DC motor is not balanced, the Arduino UNO (second) will give an additional speed command for the DC motor so that the RPM between the bicycle wheel and the treadmill conveyor does not occur at an intersection.

2.2 Step II

After understanding the problem, the next step is to create a bicycle and conveyor model that will be used for testing and speed synchronization between the treadmill rotating motor and the bicycle wheel speed. The bicycle model will stand on the conveyor to be able to easily find out when there is a deviation from the conveyor belt speed with the wheel rotation speed of the treadmill model. So at this stage, the speed deviation will be seen from the test results, not based on mathematical formula calculations. Conveyor Prototype fig. 3.

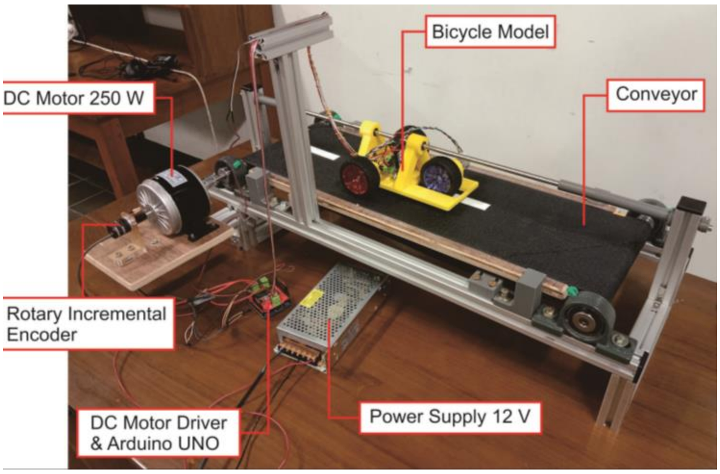

Fig. 3. Conveyor Prototype

2.3 Step III

Next is the testing of the PID controller algorithm on the prototype motor rotation speed synchronization test.

2.4 Step IV

The last stage is evaluation and improvement according to the evaluation and documentation of research activities.(Fig. 4)

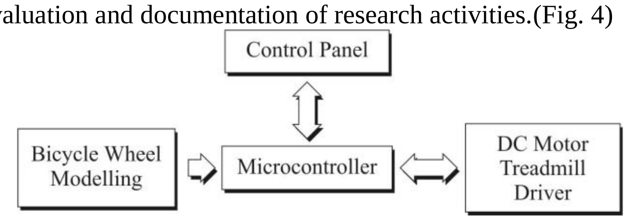

Fig. 4. Design System Carried Out

3 Result dan Discussion

3.1 Mechanical and Electronic Design

The first step is to attach a black sticker to the paddle wheel and place the wheel on a special tripod paddock for a peddle bike. The optical sensor is directed right in front of the bicycle rim. This mechanic will be equipped with Arduino UNO (first) (fig. 5).

Fig. 5. Paddlewheel mechanics, optical sensors, and Arduino UNO

The bicycle wheel will rotate so that it produces rotation, and the optical sensor will capture the light and dark pattern resulting from the color of the wheels (light) and the black sticker (dark). So when rotated, the patterns will be detected by the optical sensor, which forms a series of electronic pulses. Then the data will be sent to the Arduino UNO (first), which will convert a series of electronic pulses into speed data. Then the data is sent to the Arduino UNO (second), and the data will be converted into input which is processed into a command to drive a DC motor (the driving source on the conveyor).

When the DC motor moves the belt conveyor following the instructions from the Arduino UNO (second), the rotation of the motor will be read by the rotary encoder so that it produces data. The data generated by the rotary encoder will be compared with the input, so it is expected that the rotational speed of the conveyor will be the same as the rotational speed of the pedaling bicycle wheel. Electronic block diagram of a bicycle model and treadmill model and the communication between them.(Fig. 6).

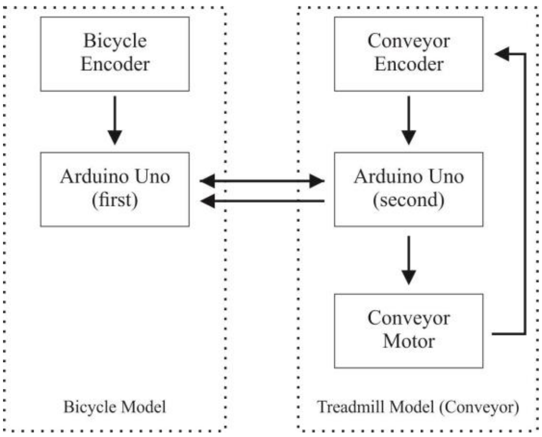

Fig. 6. Electronic block diagram of a bicycle model and treadmill model and the communication between them

The next research mechanical tool is to prepare conveyors and bicycle models. The bicycle model is equipped with Arduino UNO (first), rotary encoder, and DC motor as front-wheel drive (Fig. 7).

Meanwhile, the conveyor is equipped with a belt drive DC motor, Arduino UNO (second), a motor driver, and a rotary encoder. On the Arduino UNO (second), it is used as a control for the rotation speed of the conveyor motor and DC motor speed controlled by PID control (fig. 8).

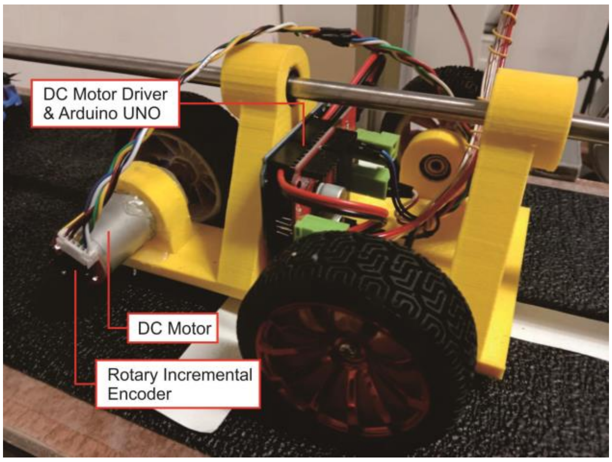

Fig. 7. Bicycle Model

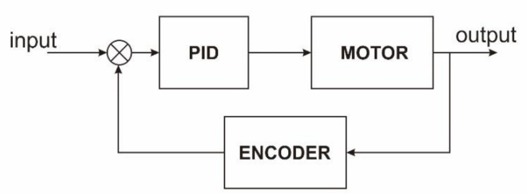

Fig. 8. PID Control Block

Based on Fig. 8, the input data is the real speed of the bicycle model obtained from the bicycle encoder data read by Arduino UNO (first). This data will then be compared with the rotary encoder conveyor data processing feedback system. The error value from the comparison of input and feedback will be processed by Arduino UNO (second), which contains a PID program to give a command signal to the DC motor driving the conveyor so that it moves according to the speed of the bicycle model. This control system architecture is expected to produce a DC conveyor motor rotation speed that matches the rotation speed of the bicycle model.

3.2 Algorithm testing (PID control) on prototype

Fig. 9 shows the process of synchronizing the speed of a bicycle model on a conveyor from a stop position.

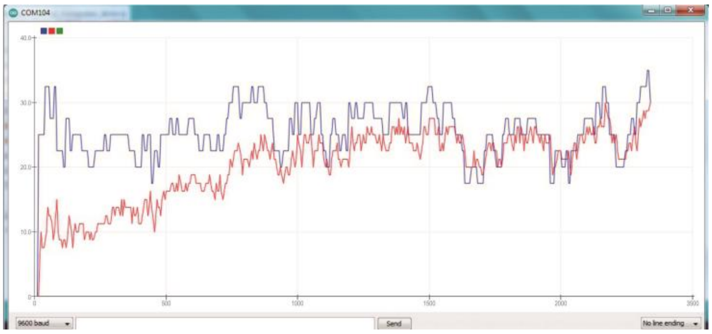

Fig. 9. Speed sync starting from zero speed

Fig. 9 shows the process of synchronizing the speed of a bicycle model on a conveyor from a stop position. The blue line is the speed reference of the bike model, and the red line is the conveyor speed. The bicycle model moves first, followed by the conveyor. In the early seconds, the speed of the conveyor is not the same as the speed of the bicycle model. Even so, after 1.5 seconds the conveyor can catch up to the speed of the bike model. After that, the conveyor can follow the speed of the bicycle model even though there are variations in the speed reference as shown in Fig. 10.

The 1.5-second delay is closely related to the response characteristics of the closed-loop control system, namely the rise time. Rise time can be minimized by increasing the Proportional Constant and reducing the Derivative Constant. But in this case, the 1.5-second delay is not too much of a problem, because the synchronization speed or rise time can be adjusted according to the rider's comfort. Rise time that is too fast will cause discomfort and even have the potential to trigger an accident while riding a bicycle on a treadmill. Fig. 10 shows the response of the conveyor after a while of running.

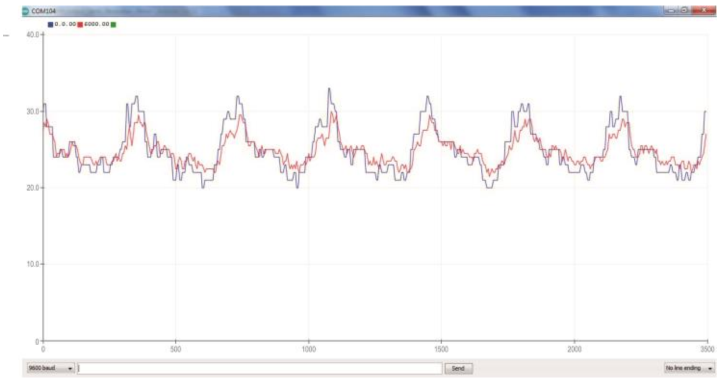

Fig. 10. Speed synchronization after 1.5 seconds the motor runs

Fig. 10 shows the response of the conveyor after a while of running. It appears in the picture that the conveyor can always follow the speed dynamics of the bicycle model. The speed profile that rises and falls is caused by disturbances in the slightly stiff and profiled conveyor belt.

4 Conclusion

Disturbance occurs due to seams at the belt connection. The conveyor belt is made of Oscar material with a slightly hard and stiff material and the seams on the belt connection are uneven. (Fig. 11).

Fig. 11. Conveyor belt materials and seams

Experiments show that the bicycle model can still walk on the conveyor and the bicycle model remains in the center position on the conveyor. The conveyor only takes a maximum of 1.5 seconds to synchronize its speed with the bike model. If the conveyor cannot synchronize its speed with the speed of the bicycle model, then the bicycle model will either advance or lag behind the conveyor. If the conveyor goes faster than the bicycle model, the bicycle model will lag and fall behind the conveyor. If the speed of the conveyor is slower, than the speed of the bicycle model, the bicycle model will accelerate and fall in front of the conveyor.

5 Acknowledgment

Thanks are given to God the Most Creative for His blessings and gifts, as well as to fellow lecturers at the Vocational Faculty, Sanata Dharma University Yogyakarta, especially Dr. Eng., Petrus Sutyasadi, and Mr. Martinus Bagus Wicaksnono, S.T., M.Eng. who has provided, direction, input so that the journal with the title "Synchronizing Treadmill Speed Against the Speed of a PID Controller-Based Bike Model" can run smoothly.

References

[1] M. Rizal, B. K. Prastiwi, and P. Kresnapati, "SURVEI MOTIVASI OLAHRAGA BERSEPEDA PADA MASSA PANDEMI COVID-19 DI MASYARAKAT KABUPATEN KOTAWARINGIN BARAT," *J. STAND Sport. Dev.*, vol. 2, no. 2, pp. 70–76, 2021.

[2] Z. Budiarso, E. N. Raharjo, and H. Listiyono, "SISTEM KENDALI KECEPATAN ROBOT MOBIL DENGAN DUA PENGGERAK MOTOR DC BERBASIS ARDUINO," *Din. Inform.*, vol. 12, no. 1, pp. 1–8, 2020.

[3] M. Irhas, I. Iftitah, and S. A. Azizah Ilham, "Penggunaan Kontrol Pid Dengan Berbagai Metode Untuk Analisis Pengaturan Kecepatan Motor Dc," *JFT J. Fis. dan Ter.*, vol. 7, no. 1, pp. 78–86, 2020, doi: 10.24252/jft.v7i1.13846.

[4] S. T. Aprilyani, I. Irianto, and E. Sunarno, "Desain dan Komparasi Kontrol Kecepatan Motor DC," *Ecotype*, vol. 7, no. 2, pp. 127–134, 202AD.

[5] Z. Budiarso, W. Hadikurniawati, and A. Prihandono, "Rekayasa Sistem Kendali Generator Sinyal XR-2206 Berbasis Arduino UNO R3," *J. eknologi Inf. Din.*, vol. 19, no. 2, pp. 101–111, 2014.

[6] C.-C. Yu, *Autotuning of PID Controllers A Relay Feedback Approach*. Springer, 2006.

[7] S. Skogestad, "Probably the best simple PID tuning rules in the world," *J. Process Control*, 2001.

[8] K. H. Ang, G. Chong, and Y. Li, "PID Control System Analysis, Design, and Technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, 2005.

[9] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," *IEE Proc.-Control Theory Appl.*, vol. 149, no. 1, pp. 46–53, 2002.

[10] J.G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Trans. ASME*, vol. 64, pp. 759–765, 1942.

[11] J. Han, P. Wang, and X. Yang, "Tuning of PID Controller Based on Fruit Fly Optimization Algorithm," in *IEEE International Conference on Mechatronic and Automation*, 2012, pp. 409–413.

[12] R. Gorez, "New design relations for 2-DOF PID-like control systems," *Automatica*, vol. 39, pp. 901–908, 2003.

[13] P. Sutyasadi and M. Bagus Wicaksono, "Joint control of a robotic arm using particle swarm optimization based H2/H∞ robust control on Arduino," *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 18, no. 2, pp. 1021–1029, 2020.