Design, Development and Control of a Ball-on-Beam Control System Ising Industrial Equipment

N. Muškinja
University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia
nenad.muskinja@um.si

***Abstract* - This paper presents the design and implementation of a ball-on-beam system (BBS) using Mitsubishi industrial equipment. The system was implemented both mechanically and in simulations. An industrial servo motor with servo drive ensures the angle of the beam. An industrial PLC is used to control the system and an HMI display is used for the user interface. A dynamic mathematical model of the ball-on-beam system has been developed. The model was used in simulations to design the corresponding control algorithms and simulations of the designed algorithms were performed in the Matlab/Simulink environment. These algorithms were then implemented in the PLC, the hardware configured, and the on-screen HMI user interface created. We have measured the performance of the control on a real system and compared it with simulations. The device we have built allows us to control the position of the ball along the beam. Different control algorithms are available, and the controller parameters can be changed via the HMI interface. The developed system can be used as an alternative to commercial systems to learn control theory and at the same time to learn about industrial hardware and software.**

Keywords - ball on beam system, programmable logic controller, servo motor control, HMI

I. Introduction

A ball on beam system (BBS) is a type of control system that involves balancing a ball on a beam using control inputs. This system is often used as a teaching tool to demonstrate the principles of control theory and the use of feedback to stabilize a system. In BBS, the beam is typically mounted on a base that allows it to pivot or rotate. The ball is placed on top of the beam and is free to roll back and forth along its length. The goal of the control system is to maintain the ball in a stable position on the beam, despite external disturbances or perturbations that may cause the ball to move off balance. To achieve this, the control system typically includes sensors that measure the position of the ball on the beam, as well as actuators that can apply forces to the beam to move it and keep the ball balanced. These sensors and actuators are connected to a control algorithm, which processes the sensor data and generates control signals that are used to drive the actuators. The control algorithm in a BBS can be implemented in a variety of ways, such as using a programmable logic controller (PLC), a microcontroller, or a personal computer. The specific design of the control system will depend on the requirements of the application and the level of performance that is needed. Fig. 1 illustrates the ball on beam system.

INEA RBT d.o.o. and Mitsubishi Automation.

Recent research and applications of ball on beam systems present a comparison between different types of controllers, validated by simulation and experimental results [2-6].

The aim this work is to develop and manufacture such a system using industrial equipment and to implement algorithms for stabilizing and controlling the position of the ball along the length of the beam. Most of the equipment used was manufactured by Mitsubishi. A servomotor is used to incline the beam. The system is controlled using an industrial Human-Machine Interface (HMI) and the entire control program is implemented on a Mitsubishi Programmable Logic Controller (PLC). A three-dimensional (3D) model created in Autodesk Fusion 360 environment is used for the mechanical design. More complex mechanical components are 3D printed. From the mechanical part, the differential equation of the system and the geometrical connections are derived. A mathematical model of the system is built using the equations in Matlab and Simulink and simulations are performed. Different control algorithms are developed based on the model. Using Mitsubishi MELSOFT GXWorks3 software, a complete program is implemented for measuring the value of the sensors and controlling the servomotor. The designed control algorithms are implemented within this program. A touch screen HMI user interface is designed to display the operating status and to allow user control.

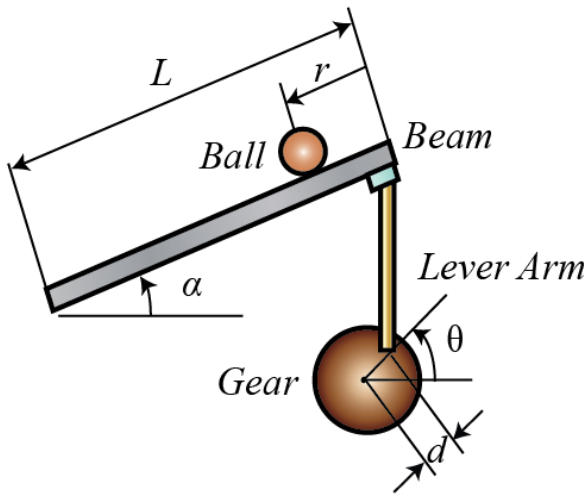

Figure 1. Ball on beam illustration [1].

II. Components of the Ball on Beam System

For our system, we used components used in industrial applications as well as a linear diaphragm potentiometer and self-build components.

A. Servo Motor, Servo Controller, PLC and HMI

The beam was inclined using a servomotor. The servomotor was chosen to be an HG-KN- 13J motor [7]. Fig. 2 and Table I shows the servomotor and its technical characteristics. The selected servomotor can be controlled using a servo amplifier. We have chosen Mitsubishi's MELSERVO MR-JE-10C [8]. The controller allows different motor control modes: speed, position and also torque mode. It supports both analogue and pulse-width modulated (PWM) input for the desired motor speed via the CN3 connector. It also supports the possibility of control via CC_Link communication, which is done via an Ethernet connector. The amplifier parameters can be set using the MR Configurator2 software package. It can be connected to a PC via USB or Ethernet. The amplifier is shown in Fig. 3. We chose a Mitsubishi MELSEC iQ-F series controller called FX5UC-32MT/DSS-TS [9] to control the whole system. The controller has 16 inputs and 16 outputs. It allows communication via Ethernet or RS-485 bus. It shall have a connection point for a standard SD card to which data can be logged during operation. An overview of the controller is shown in Fig. 4.

Figure 2. Servomotor HG-KNJ [7].

TABLE I. Servomotor Technical Data

Nominal power	100 W
Nominal torque	0,32 Nm
Maximum torque	0,95 Nm
Nominal speed	3000 min^{-1}
Maximum speed	5000 min^{-1}
Nominal current	0,8 A
Maximum current	2,4 A
Incremental encoder	17-bit, 131072 pulses per revolution

Figure 3. Digital AC-Servo Amplifier MR-JE [8].

Figure 4. PLC FX5UC-32MT/DSS-TS [9].

TABLE II. PLC Analogue Expansion Cards

FX5-4DA-ADP	FX5; Analog Output Module; 4 channels
FX5-4AD-ADP	FX5; Analog Input Module; 4 channels

Figure 5. HMI interface [10].

Figure 6. SoftPot Membrane Potentiometer [11].

Table II shows the two analogue expansion cards that were also attached to the PLC.

We used the GT2107-WTSD HMI [10] to interact with the system (Fig. 5). The interface has a seven-inch color touchscreen. It allows communication via Ethernet, RS-232 and RS-422/485 connectors and can also be connected to a PC via a USB port. Graphical elements can be created in the GT Designer3 software environment to allow monitoring of the system operation and modification of global variables in the PLK program.

B. Sensors

To measure the position of the ball on the beam, we chose the SoftPot SEN-08681 linear diaphragm potentiometer [11]. The sensor has a measuring range of 500 mm and a measured resistance of 12 kΩ. The sensor consists of a lower resistive layer, an air gap and an upper conductive membrane which contacts the lower layer when pressure is applied at the ball location. It works in much the same way as an ordinary potentiometer, except that the ball acts as a slide. The voltage across the upper membrane thus represents the position of the ball. The sensor is shown in Fig. 6.

C. Self-buid Conponents

A 3D model of the mechanical system was drawn in Autodesk Fusion 360. The basic element of the beam is an aluminum U-profile. We have designed plastic bearing supports at each end of the profile. Bearings were used in the pivots to minimize friction in the system.

Figure 7. 3D model of the ball and beam mechanical system.

We have designed a plastic arm for incline the beam. We have adjusted the length of this so that when the system is in the equilibrium position, the pivot of the lever with the beam is directly above the servomotor axis. This made it easier to derive the mathematical relationship between the angle of the beam at and the angle of the servomotor. We also designed a railing to ensure that the ball stays on the beam and moves only within the measuring range of the position sensor. Fig. 7 shows a drawing of a beam mechanical system.

III. Mathematical Modelling

The mathematical model was designed from a 3D model of the system, where we could easily measure all the necessary dimensions. A sketch of the system is shown in Fig. 8. The motor angle ϑ varies between -90° and 90°, the horizontal position of the motor arm is 0°. The movement of the beam angle α is physically limited by the length of the motor arm b and the length of the beam L. The ball is located at a distance from the pivot of the beam r. When the motor increases the angle ϑ, the angle α increases at the same time, which causes the ball to roll downhill due to the gravitational acceleration and its potential energy is converted into kinetic energy. In our example, we have neglected frictional forces.

We used the Euler-Lagrange method of mathematical model derivation [12], [13]. First, we determined the kinetic energy of the ball. This consists of the translation of the ball and the rotation of the ball. J represents the moment of inertia of the ball and ω represents the angular velocity of the ball rotation about its own axis. The Lagrangian equation is as follows:

$$\mathcal{L} = K - U, \tag{1}$$

where K is the kinetic energy and U is the potential energy of the BBS. The total kinetic energy consists of the kinetic energy of the ball K_1 and the kinetic energy of the beam K_2.

The kinetic energy of the ball K_1 consists of translational and rotational kinetic energy:

$$K_1 = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2, \tag{2}$$

where m is mass of the ball and v is the velocity of the center of the ball.

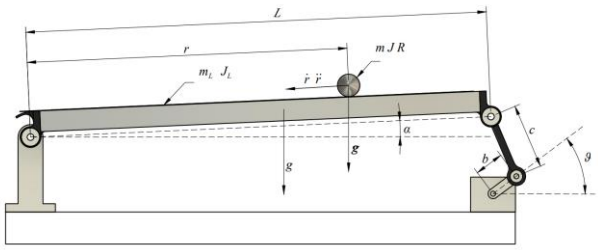

Figure 8. A sketch of the ball on beam system.

Assume that the ball is just rolling on the surface and not sliding. Thus, the angular velocity of the rotation of the ball can be written as:

$$\dot{r} = R\omega, \tag{3}$$

where R is radius of the ball.

Inserting (3) into (2), we obtain:

$$K_1 = \frac{1}{2}mv^2 + \frac{1}{2}J\left(\frac{\dot{r}}{R}\right)^2. \tag{4}$$

To calculate the translational kinetic energy, we need to calculate the velocity v of the ball. This is expressed in Cartesian coordinates:

$$v^2 = \dot{x}^2 + \dot{y}^2. \tag{5}$$

Converting them from radial to Cartesian coordinates gives:

$$\begin{aligned} x &= r \cdot \cos(\alpha) \\ y &= r \cdot \sin(\alpha) \end{aligned}. \tag{6}$$

After inserting (5) into (4) and further derivation, we get:

$$K_1 = \frac{1}{2}\left(\frac{J}{R^2} + m\right)\dot{r}^2 + \frac{1}{2}mr^2\dot{\alpha}^2. \tag{7}$$

The kinetic energy of the beam is:

$$K_2 = \frac{1}{2}J_L\dot{\alpha}^2, \tag{8}$$

where J_L represents the moment of inertia of the beam.

The potential energy of the ball is:

$$U_1 = mg\sin(\alpha)\,r, \tag{9}$$

and a potential energy of the beam is:

$$U_2 = \frac{1}{2}Lm_Lg\sin(\alpha), \tag{10}$$

where m_L is mass of the beam and g is a gravitational acceleration constant.

After summarizing all energies:

$$\begin{aligned} K &= K_1 + K_2 \\ U &= U_1 + U_2, \end{aligned} \tag{11}$$

and after solving Lagrange equation, we get final BBS mathematical model:

$$\left(m + \frac{J}{R^2}\right)\ddot{r} + mg\sin(\alpha) - mr\dot{\alpha}^2 = \tau, \tag{12}$$

where τ is the torque produced by the motor applied on the end of the beam.

Once we derived the differential equation of the system, we used this to design the mathematical model. The model was built in MATLAB's Simulink software package, which allows graphical system design. First, the relationships between the angles were determined using the physical constants of the system (Table III).

We have determined how the angle of the motor arm ϑ and the angle of the beam α depend on each other. Assume that the end of the beam moves only vertically, neglecting angular motion due to small angles. The value y_{off} represents the vertical distance between the beam pivot and the motor axis. In our system it is 0,078 m.

A sketch of the system (Fig. 9) shows the angle dependencies.

TABLE III. FIZICAL BBS CONSTANTS

Acceleration of gravity	g	$9{,}81\,\frac{\mathrm{m}}{\mathrm{s}^2}$
Length of the beam	L	0,555m
Mass of the beam	m_L	0,2 kg
Moment of inertia of the beam	J_L	0,037 kg m^2
Motor arm length	b	0,036 m
Length of baem handle	c	0,078 m
Radius of the ball	R	0,0135 m
Mass of the ball	m	0,082 kg
Moment of inertia of the ball	J	5,96 kg m^2

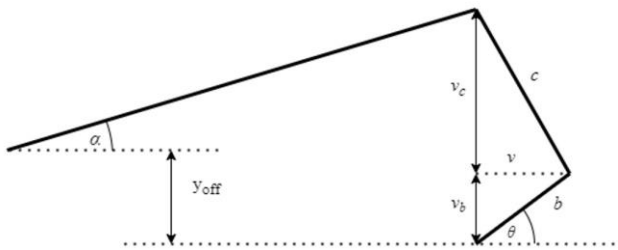

Figure 9. The BBS angle dependencies.

After some derivations and simplifications, we get:

$$\alpha = a\sin\left(\frac{h}{L}\right) = a\sin\left(\frac{b\sin(\vartheta) - y_{off} + \sqrt{c^2 - b^2\cos(\vartheta)^2}}{L}\right). \tag{13}$$

IV. DESIGN OF CONTROL ALGORITHMS

We used a cascaded proportional-integral-differential (PID) algorithm for control as proposed in [13]. PID is the most used control algorithm in industry because it is simple to implement, allows experimental parameter determination, and has good control characteristics. We used the PID-PD cascade structure as shown in Fig. 10. In the following, the name cascaded PID is used for this controller. For the determination of the controller parameters, we used the IMC-PID (Internal Model Control) procedure [14]. First, we need to measure the open-loop step response of the inner ball velocity loop to determine the process model, and then we can calculate the parameters of the inner PD ball velocity controller, as shown in Table IV. Fig. 11 shows the closed-loop response of the ball velocity control.

After the successful implementation of the internal PD velocity controller, an external PID position controller was added to the system. The input is the difference between the desired and actual ball position, and the output is the desired ball velocity. First, we need to measure the open-loop step response of the outer loop with the inner velocity loop closed. Then we can calculate the parameters of the external PID ball position controller as shown in Table V.

TABLE IV. VELOCITY PD CONTROLLER PARAMETERS

Process transfer function	K_p	T_D[sec]
$\frac{-0{,}48}{s}\cdot e^{-0{,}1\cdot s}$	4,9	0,04

Figure 10. Cascaded PID BBS.

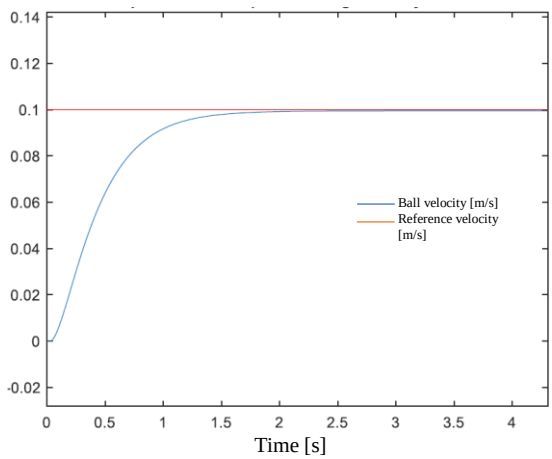

Figure 11. Closed-loop response of controlling the ball velocity.

TABLE V. POSITION PID CONTROLLER PARAMETERS

Process open loop transfer function	K_p	T_I[sec]	T_D[sec]
$\frac{1}{s} \cdot e^{-0,5 \cdot s}$	1,15	3,5	0,23

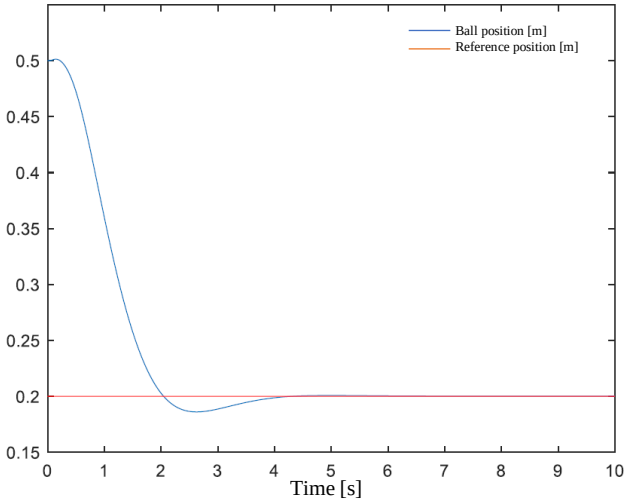

Figure 12. Closed-loop response of controlling the ball position.

Fig. 12 shows the closed-loop response of the ball position control.

In this section, we present the mathematical modelling of our BBS system and the design of the cascaded PID controller, as well as the simulation results of the ball position control. In the next section we will present the implementation of the algorithms in the PLK software and the real-time BBS experiments.

V. IMPLEMENTATION AND REAL-TIME EXPERIMENTS

A trolley made of extruded aluminum profiles was used to assemble all the elements. At the back of the trolley there are higher profiles where we mounted the electrical components. Most of the electrical components were mounted on DIN-rails, but the larger components were mounted directly on the mounting plate, for which we used a wooden furniture panel. Before starting to assemble the components, we created an approximate layout of the components in the Fusion 360 environment (Fig. 13).

When implementing the cascaded PID controller structure in the PLC, the same values of position and velocity PD controller parameters were used. Fig. 14 shows the response of the position control of the BBS ball.

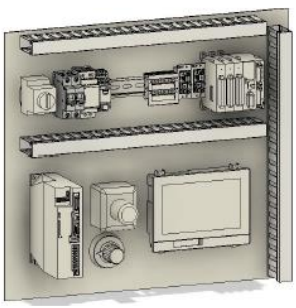

Figure 13. 3D model of the panel with electrical components.

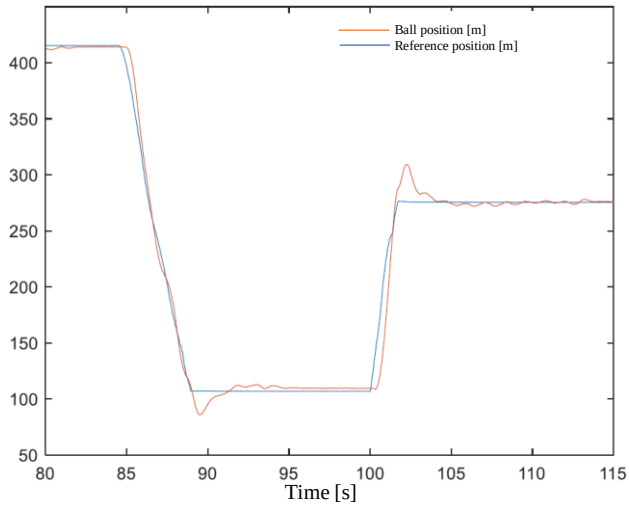

Figure 14. Closed-loop response of controlling the ball position on real BBC.

Figure 15. A real ball position control system.

VI. CONCLUSION

A ball-on-beam system was designed using industrial equipment (Fig. 15). The system was implemented both mechanically and programmatically. We designed the user interface for easy operation and at the same time added the possibility to change the control parameters. The developed system can be used as an alternative to commercial systems for learning control theory and at the same time for learning about industrial hardware and software.

The system successfully stabilizes the position of the ball using implemented algorithm. The position can be determined by using a potentiometer, by using a virtual slider and by using a changing reference that varies in the form of a PWM signal. The system is mounted on an aluminum trolley, which allows easy transport. It operates on its own local network, so there is no need for an external network router.

The Advanced Control System curriculum includes the study of complex control algorithms that are often abstract and difficult to understand, and many aspects of control of electronics. Therefore, in order to help students better understand them, it is necessary to implement laboratory exercises with real control systems and hardware implementation.

The practical implementation of a ball on a beam system involves a wide variety of integrated field such as:

- Control system components
  - Servo Motor and Servo drives
  - Potentiometer Sensor
  - Controller
- Feedback control systems
  - Reading Ball Position from Potentiometer Sensor
  - Derivative Filtering
  - System modeling
- System Identification Methods
  - Modeling of Actuator
  - Obtaining Transfer Function of Plant
- Real time measurements
  - Time Domain Characteristics
  - Steady State Response and Steady State Error
- Control system design
  - Time Response Characteristics
  - PID Controller and Fuzzy Logic Controller
  - Step Response and Steady-State Error
- Control system verification
  - Frequency Response Analysis
  - Experimental Bode Diagrams
  - Cut-Off Frequency Determination

Our ball on a beam system allows students to practice the theory of automatic control in a closed-loop experiment. Controlling the position of a ball on a beam is one of the classical problems of control theory. The ball and a beam system can be applied to a wide range of control system design implementations, from basic linear controllers to advance nonlinear methods. Students can understand approaches to system design using the Ball on a Beam method of learning by doing. They can use the curriculum to distinguish the effects of linearization's, assumptions, and modelling errors due to differences between simulations and real-world experiments.

REFERENCES

[1] "Control Tutorials for MATLAB and Simulink - Ball & Beam: System Modeling." https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SystemModeling (accessed Dec. 21, 2022).

[2] I. M. Mehedi, U. M. Al-Saggaf, R. Mansouri, and M. Bettayeb, "Two degrees of freedom fractional controller design: Application to the ball and beam system," *Measurement*, vol. 135, pp. 13–22, Mar. 2019, doi: 10.1016/j.measurement.2018.11.021.

[3] S. Zaare and M. R. Soltanpour, "The position control of the ball and beam system using state-disturbance observe-based adaptive fuzzy sliding mode control in presence of matched and mismatched uncertainties," *Mech. Syst. Signal Process.*, vol. 150, p. 107243, Mar. 2021, doi: 10.1016/j.ymssp.2020.107243.

[4] Z.-H. Pang, G. Zheng, and C.-X. Luo, "Augmented state estimation and LQR control for a ball and beam system," in *2011 6th IEEE Conference on Industrial Electronics and Applications*, Beijing, China, Jun. 2011, pp. 1328–1332. doi: 10.1109/ICIEA.2011.5975793.

[5] W. Yu, "Nonlinear PD Regulation for Ball and Beam System," *Int. J. Electr. Eng. Educ.*, vol. 46, no. 1, pp. 59–73, Jan. 2009, doi: 10.7227/IJEEE.46.1.5.

[6] S. Krafes, Z. Chalh, and A. Saka, "A Review on the Control of Second Order Underactuated Mechanical Systems," *Complexity*, vol. 2018, pp. 1–17, Dec. 2018, doi: 10.1155/2018/9573514.

[7] "HG-KN13J - Mitsubishi Electric Factory Automation." https://si.mitsubishielectric.com/fa/products/drv/servo/mr_je/rotary-servo-motor/hg-kn13j.html (accessed Dec. 21, 2022).

[8] "MR-JE-10C - Mitsubishi Electric Factory Automation." https://si.mitsubishielectric.com/fa/products/drv/servo/mr_je/servo-amplifier/mr-je-10c.html (accessed Dec. 21, 2022).

[9] "FX5UC-32MT/DSS-TS - Mitsubishi Electric Factory Automation." https://si.mitsubishielectric.com/fa/products/cnt/plc/plcf/cpu-module/fx5uc-32mt-dss-ts.html (accessed Dec. 21, 2022).

[10] "GT2107-WTSD | Human Machine Interface | HMI | Catalogue | Mitsubishi Electric Europe e-shop." https://mitsubishi-electric-eshop.mee.com/mee/FA_IA/en/EUR/Catalogue/HMI/Human-Machine-Interface/GT2107-WTSD/p/000000000000311489 (accessed Dec. 21, 2022).

[11] "SoftPot Membrane Potentiometer - 500mm - SEN-08681 - SparkFun Electronics." https://www.sparkfun.com/products/8681 (accessed Dec. 21, 2022).

[12] M. Keshmiri, A. F. Jahromi, A. Mohebbi, M. H. Amoozgar, and W.-F. Xie, "Modeling and Control of Ball and Beam System using Model Based and Non-Model Based Control Approaches," *Int. J. Smart Sens. Intell. Syst.*, vol. 5, no. 1, pp. 14–35, Jan. 2012, doi: 10.21307/ijssis-2017-468.

[13] P. V. M. Maalini, G. Prabhakar, and S. Selvaperumal, "Modelling and control of ball and beam system using PID controller," in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, India, May 2016, pp. 322–326. doi: 10.1109/ICACCCT.2016.7831655.

[14] B. W. Bequette, *Process control: modeling, design, and simulation*. Upper Saddle River, N.J: Prentice Hall PTR, 2003.