Tekhnê
January - June 2021, Vol. 18, No. 1, pp. 21 – 36

ISSN 1692-8407

Temperature control system for a hatchery

Sistema para el control de temperatura de una incubadora de huevos de gallina

Karen L. Cruz[1] and Fernando F. Vera[2]
[1]Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
klcruzr@correo.udistrital.edu.co
[2]Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
ffvera@correo.udistrital.edu.co

A closed-loop PID controller tuning method is presented, as well as a control mechanism that functions as an egg incubator controller to control the temperature. The method's performance is compared using digital simulation tests, and the PID controller gains are determined based on the system requirements. Finally, recommendations for using the tested method as well as arguments for why we reject other methods are provided.

Keywords: Control, incubator, PID, temperature, transfer function

Se presenta un método de sintonización de controladores PID de lazo cerrado, un mecanismo de control que opera como regulador de una incubadora de huevos, el cual controla la temperatura. Mediante pruebas de simulación digital se compara el desempeño del método y se determinará las ganancias del controlador PID según los requerimientos del sistema. Finalmente, se ofrecen recomendaciones sobre la utilización del método probado y los argumentos por los cuales descartamos otros métodos.

Palabras clave: Control, función de transferencia, incubadora, PID, temperatura

Article typology: Research

Received: August 14, 2020
Accepted: May 28, 2021

Research funded by: Universidad Distrital Francisco José de Caldas (Colombia).

How to cite: Cruz, K., Vera, F. (2021). *Temperature control system for a hatchery*. Tekhnê, 18(1), 21 -36.

ISSN 1692-8407

Introduction

Poultry production is growing in Colombia, as the consumption of chicken meat and eggs has been increasing for a long time (Páramo et al., 2020). They have become an important source of food for humans, providing higher quality proteins, offering a wide range of amino acids and essential minerals that benefit physical development (Blanco & Ramos, 2019; Zutta, 2005). The high demand for these products has led to the fact that natural reproduction is not enough, therefore, it has been necessary to use artificial incubators to increase production (Millette et al., 2020). Poultry breeding is a meticulous process, full of details and very strict parameters that must be controlled for the eggs to be successful (Marçal et al., 2015).

The control parameters to be taken into account in an artificial incubator are temperature, ventilation, turning, and humidity of the environment (Guzmán & Ramírez, 2017; Santoso et al., 2020). These parameters or physical variables will be controlled by a control design and verified by the simulation to satisfy the optimal breeding characteristics (Bello et al., 2013). Making an egg incubator that is efficient and inexpensive is a challenge because egg embryos are very fragile, and any sudden change in temperature or humidity can significantly affect chick growth and time to hatch ("Diseño e implementación de una incubadora de huevos con aire forzado basada en microcontrolador," n.d.).

Within the industry three types of temperature controls are generally known, ON-OFF, which work according to their name, it turns on and off continuously to maintain the exact desired temperature set point (Che et al., 2019; Khera & Kohli, 2018). When the temperature exceeds the set point it turns off and when the temperature is below the set point it turns on. Proportional controllers reduce the power of the heater when it detects that the temperature will exceed the setpoint, maintaining a stable temperature and fast response, but may exhibit oscillations in the controlled variable or present error in steady-state, and PID controllers offer proportional, integral, and derivative control, is feedback control to make the error between the output and input is zero, compensates for temperature changes thanks to its combination and can adjust each variable, making it more accurate (López, 2014).

Of the controller types, the most suitable for incubator temperature control is the PID controller which is the most susceptible to changes and can react quickly to compensate for them (Kapen et al., 2020). In addition, it can predict any change or effect on the output, thanks to its derivative function. Currently, they have already realized temperature control systems for hatcheries, for example, Microcontroller-based controls have been used that can control temperature, humidity, and reverse eggs automatically and through the Internet of Things (IoT) system help farmers to monitor the hatchery in real-time and remotely (Farfán et al., 2011; Sanjaya et al., 2018).

Microcontrollers have also been used that have been designed using fuzzy control, which is about logically determining what process to do to achieve the control objectives from a basis provided by the designer (Colmenares et al., 2003). This controls the egg position, temperature, and humidity of the incubator and ensures the best conditions for different types of eggs. Next is to build an IoT system to control the incubator remotely (Aldair et al., 2018). Finally, a study was conducted using Arduino IDE Software and Matlab 2015a to simulate and operate a PID system that would be used, as well as design the plant that would produce a temperature response in the hatching of the egg in the incubator (Shafiudin & Kholis, 2018).

Problem statement

The starting point for this project is an already implemented incubator from which 18 temperature data are obtained from a transient response test by feeding the resistance and measuring the behavior of the temperature concerning time. It is desired to design a control system to maintain the temperature of the artificial incubator because if it varies, it can alter the growth of the embryo.

In the data obtained in the test, an increase in temperature is observed after 1 minute, reaching 48.0 degrees Celsius, when the ideal temperature for incubation is 37.5 degrees Celsius, this behavior is because it is an open-loop control. If this temperature is maintained in the egg incubator, poultry production will be affected by the loss of eggs or faster development, altering the growth of the embryo and thus affecting the quality of the chick.

The incubator currently implemented has a heating element, a temperature sensor, and a fan to ensure the movement of hot air inside the incubator. The heating of the eggs is produced by the heat exchange between the air and the eggs, therefore the air temperature is a fundamental factor in the process so the main objective is to design an automaton device that controls in a closed-loop the temperature and feeds back the input with the previous output, ensuring the proper control of temperature to be maintained at 37.5 degrees Celsius inside the egg incubator and that despite external disturbances the control responds quickly and stabilizes the temperature. The system would have the behavior shown in Fig. 1.

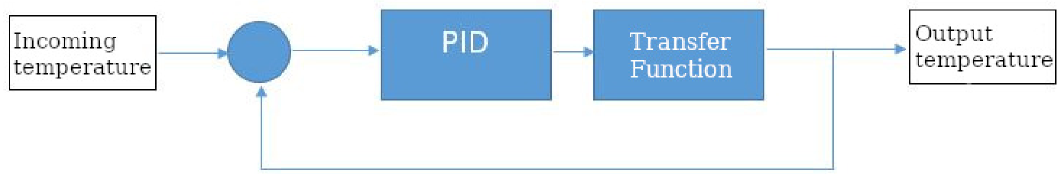
Methods

The design of the control system for the hen egg incubator will be based on the data obtained in the transient response test in which the temperature behavior concerning time was obtained as shown in Table 1 (Torres & Ramírez, 2012). These data were captured in the laboratory directly from the incubator understudy, and calibrated measuring equipment

ISSN 1692-8407

Figure 1

Block diagram of the control system.



was used. Although they correspond to a single reading, multiple tests have been carried out on this plant, and it has been found that the behavior does not show significant variations (Lata et al., 2020; Maasoum et al., 2020).

Table 1

Table of laboratory data

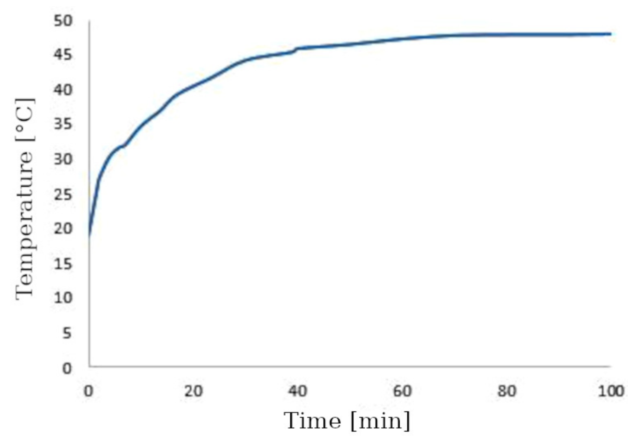
Time [minutes]	Temperature [Degrees]
0	19
1,6	25.7
2	27.1
4	30.3
6	31.7
7	32.0
10	34.7
13,5	36.8
14,16	37.3
17	39.3
23,5	41.7
30	44.2
39	45.4
40	45.9
50	46.5
60	47.3
70	47.8
80	47,9
90	47.9
100	48.0

From these data the following procedures will be done for the design of the closed-loop controller:

1. Obtain the real temperature behavior of the incubator by graphing the data obtained in the test.
2. Based on the obtained graph, find the mathematical function that describes the behavior of the incubator.

Figure 2

Graph of real behavior.



3. Validate the behavior of the model.
4. Find the transfer function.
5. Perform the stability analysis and find the interval of K in which the system is stable.
6. Evaluation of PID controller method. This requires a study of existing tuning strategies.
7. Perform the simulation of the design using the Root Locus function of MATLAB.
8. External disturbance test.
9. Analyze the results of the model.

Development procedure

Incubator temperature behavior

The first analysis of the data is carried out graphically by means of a representation of the laboratory test. Fig. 2 shows the actual temperature behavior of the egg incubator.

ISSN 1692-8407

Mathematical function

The behavior of the curve resembles a first-degree transfer function. The canonical equation has the following form:

$$y(t) = K - K * e^{\left(-\frac{t}{\tau}\right)} \tag{1}$$

$$y(t) = K + (K - C.initial) * e^{\left(-\frac{t}{\tau}\right)} \tag{2}$$

$$K = 48 \tag{3}$$

With the values in the table, we will find the equation that simulates the behavior of the incubator, making a linear regression to find the time value that corresponds to 63 percent of K. That is, we will obtain the time in minutes for a temperature of 37.27 degrees. We will perform the linear regression between the times of 13.5 and 17 minutes.

$$m = \frac{39.3 - 36.8}{17 - 13.5} \tag{4}$$

$$m = 0.714 \tag{5}$$

$$b = 36.8 - 0.714 * 13.5 \tag{6}$$

$$Temperature = m * Time + b \tag{7}$$

$$Temperature = 0.714 * Time + 27.16 \tag{8}$$

$$Time = \frac{Temperature - 27.16}{0.714} \tag{9}$$

We only take into account when the exponential form starts.

$$0.63 * (K - 19) = 18.27 \tag{10}$$

$$18.27 + C.initial = 37.3 \tag{11}$$

$$Time = \frac{37.3 - 27.16}{0.714} \tag{12}$$

$$Time = 14.2 = \tau \tag{13}$$

Finally, the resulting equation is as follows:

$$Temperature(t) = 48 - 29 * e^{\left(-\frac{t}{14.2}\right)} \tag{14}$$

$$Temperature(\infty) = 48 = K \tag{15}$$

$$Temperature(0) = 19 = C.initial \tag{16}$$

A canonical equation that simulates the behavior of the egg incubator is obtained by verifying the important points. The two graphs are compared to see if this is true. Fig. 3 shows the behavior of both curves, the blue one was obtained from the incubator data sample and the orange one was obtained using the canonical equation found, it is concluded that the canonical equation corresponds to the real behavior of the incubator since both graphs are similar. From the equation, the transfer function of the system is found.

Transfer function

The Laplace transform is applied to the canonical equation to obtain the closed-loop transfer function.

$$Laplace\{48 - 29 * e^{\left(-\frac{t}{14.2}\right)}\} \tag{17}$$

$$Output(s) = \frac{19(s + 0.1779)}{s(s + 0.0704)} \tag{18}$$

Finally, taking into account the unit step of the input the transfer function that simulates the behavior of the data obtained in the laboratory is:

$$H(s) = \frac{19(s + 0.1779)}{(s + 0.0704)} \tag{19}$$

Stability analysis

Fig. 4 shows the structure of the entire system including the PID controller.

Simplifying by block algebra.

$$G(s) = \frac{K19(S + 0.1779)}{S + 0.0704} \tag{20}$$

$$H(s) = 1 \tag{21}$$

$$\frac{G(s)}{1 - G(s) * H(s)} \tag{22}$$

$$\frac{G(s)}{1 + \frac{K19(S+0.1779)}{S+0.0704}} \tag{23}$$

$$\frac{\frac{K19(S+0.1779)}{S+0.0704}}{\frac{S+0.0704+K19(S+0.1779)}{S+0.0704}} \tag{24}$$

$$\frac{K19(S + 0.1779)}{S + 0.0704 + K19(S + 0.1779)} \tag{25}$$

$$\frac{C(s)}{\mu(s)} = \frac{K19(S + 0.1779)}{(K19 + 1)S + 0.1779(K19 + 0.395728)} \tag{26}$$

$$\frac{C(s)}{\mu(s)} = \frac{K19(S + 0.1779)}{(K19 + 1)S + 0.1779(K19 + 0.395728)} \tag{27}$$

ISSN 1692-8407

Figure 3

Real vs. simulated behavior comparison.

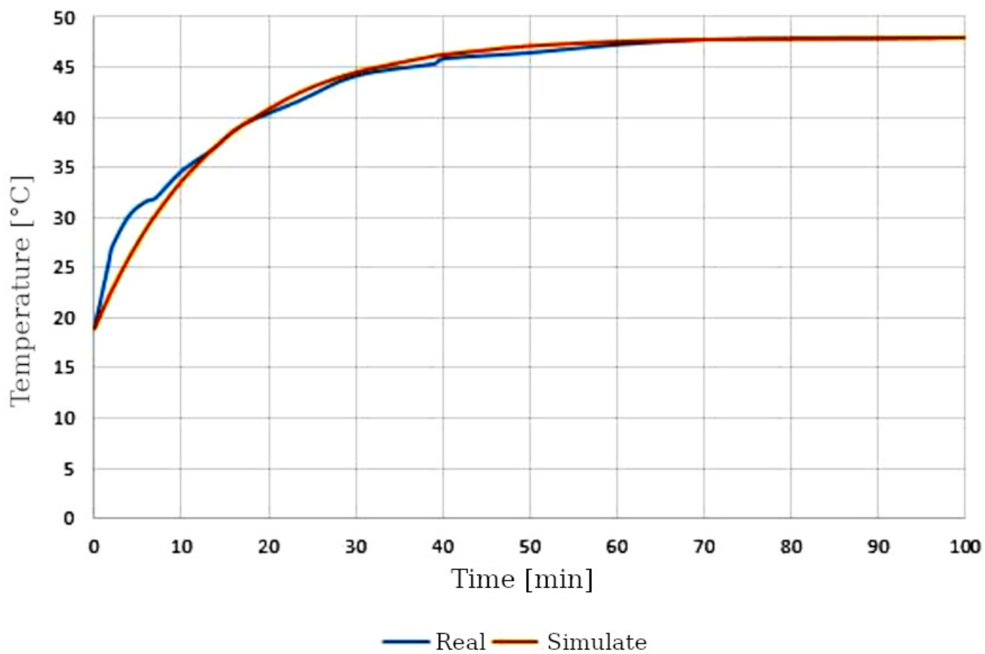
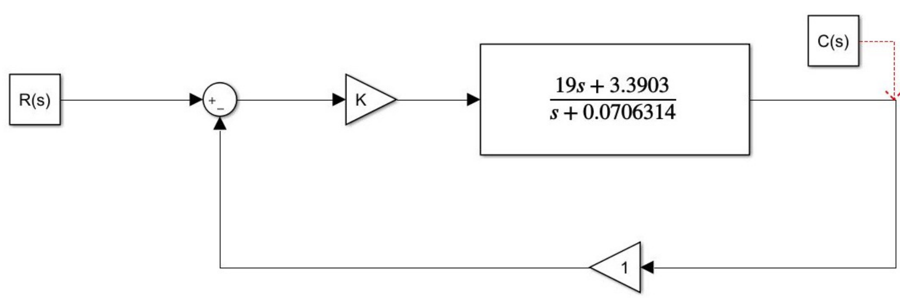


Figure 4

Closed-loop system model.



$$(19K + 1)S + 3.3801K + 0.0704 = 0 \tag{28}$$

$$\begin{aligned} S1 &= (19K + 1) \\ S0 &= 3.3801K + 0.0704 \end{aligned} \tag{29}$$

Case 1:

$$19K + 1 > 0 \tag{30}$$

$$19K > -1 \tag{31}$$

$$K > -\frac{1}{19} \tag{32}$$

$$\underbrace{K > -0.052} \tag{33}$$

Case 1:

$$3.3801K + 0.0704 > 0 \tag{34}$$

$$3.3801K > -0.0704 \tag{35}$$

ISSN 1692-8407

$$K > -\frac{0.0704}{3.3801} \tag{36}$$

$$\underbrace{K > -0.0208} \tag{37}$$

Therefore, for the system to be stable:

$$K > 0 \tag{38}$$

PID controller method evaluation

The control system to be implemented is as shown in Fig. 5.

Therefore:

$$\frac{y(s)}{r(s)} = \frac{(Kp + \frac{K1}{S} + SKD)\left(\frac{19(S+0.1779)}{S+0.0704}\right)}{1 + \left(Kp + \frac{K1}{S} + SKD\right)\left(\frac{19(S+0.1779)}{S+0.0704}\right)} \tag{39}$$

$$\frac{y(s)}{r(s)} = \frac{\frac{(KpS+K1+S^2KD)(19(S+0.1779))}{S^2+S0.0704}}{1 + \left(\frac{(KpS+K1+S^2KD)(19(S+0.1779))}{S^2+S0.0704}\right)} \tag{40}$$

$$a = \frac{(KpS + K1 + S^2KD)(19(S + 0.1779))}{S^2 + S0.0704} \tag{41}$$

$$\frac{y(s)}{r(s)} = \frac{a}{\frac{S^2+S0.07+(KpS+KI+S^2KD)(19(S+0.1779))}{S^2+S0.0704}} \tag{42}$$

$$b = (KpS + KI + S^2KD)(19(S + 0.1779)) \tag{43}$$

$$\frac{y(s)}{r(s)} = \frac{b}{(S^2 + S0.07 + b)} \tag{44}$$

The following will show some methods for tuning a PID controller. The objective of these methods is to calculate the gains of our PID controller or the individual cases, i.e. proportional only Proportional and integral gain (PI), proportional and derivative gain (PD), or proportional integral and derivative (PID).

Results

Ziegler-Nichols method

The Ziegler-Nichols method allows practically tuning a PID controller, without the need to know the equations of the plant or the system to be controlled. It is one of the most widely used methods and allows the proportional, integral and derivative gains to be defined from the system output in either open or closed-loop, but is best suited for open-loop systems.

The method consists of replacing the PID controller with a unitary stepper and having a sensor supplying the system output. With the output behavior, we can calculate the parameters of the PID controller. For the article, the Ziegler-Nichols method would not be the best procedure to calculate the PID controller because it is used when the plant equations are not known and it is based only on the output of the system through the reading of a sensor. In the case of the article, we already know the plant equations and we do not have a real-time sensor reading of the output.

The gains of the PID controller are calculated using another method that uses what has already been obtained as the system's transfer function, and it is calculated for a closed-loop system to ensure accuracy. There is an indeterminacy when applying the equations proposed by the Ziegler-Nichols method, for this reason, our system is not suitable to apply this solution method.

Ziegler-Nichols oscillation

In this method, it is not required to remove the PID controller from the closed-loop. This method proposes to take K_i and K_d to 0, and alternate K_p until the system oscillates constantly, at this point it is necessary to measure the proportional gain K_p called critical gain or K_c, and the period of oscillation T_c in minutes. Once these two values are measured, you can calculate the gains of the PID controller or the individual cases, i.e. only proportional and integral gain (PI), proportional and derivative gain (PD), or proportional, integral and derivative (PID).

For our system by taking our gains K_i and K_d to zero and varying K_p in different values we did not obtain any oscillation, therefore the system has no oscillatory response any fundamental parameter for obtaining the critical gain, the reason that our system has no oscillatory response alternating K_p may be because it is linear for any value of K_p.

Root Locus Simulation in MATLAB

Initially the plant transfer function is written and the SISOTOOL function is called which plots the Root Locus (Figs. 6 and 7).

H= tf ([19 3.3801], [1 0.0704]); sisotool (H)

Fig. 6 identifies the Root locus of the plant by locating the zero and the pole, both real on the left side of the complex plane. Being the pole -0.0704 and the zero of -0.1779. Fig. 7 shows the response to the passage of the plant with a proportional controller $K_p = 1$, which has no overshoot and its stabilization time is greater than 60 s, stabilizing at 0.98.

As the proportional control is displaced on the straight line between pole and zero, the step response changes, therefore, this will be the strategy to achieve the type of response expected under the following design parameters:

- Stabilization time < 10 s.
- Overshoot < 5 percent.

ISSN 1692-8407

Figure 5

Complete system block diagram.

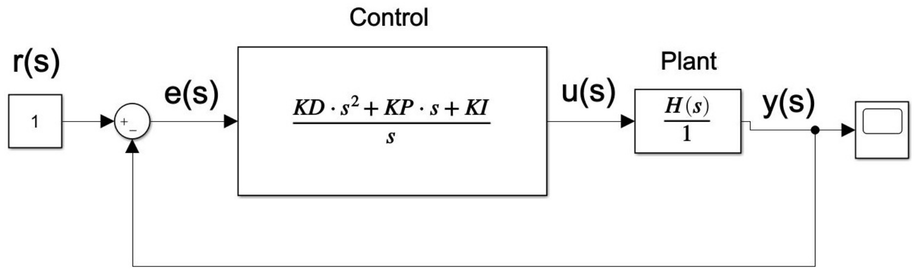
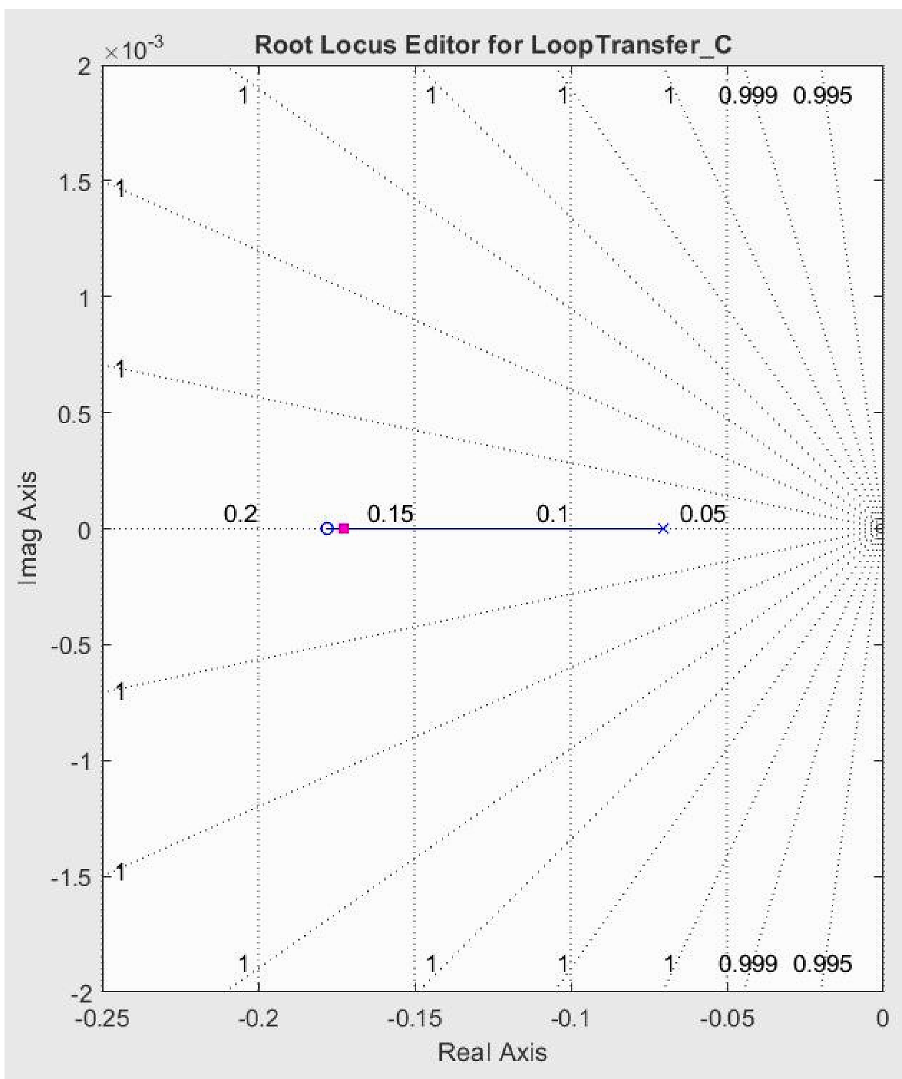


Figure 6

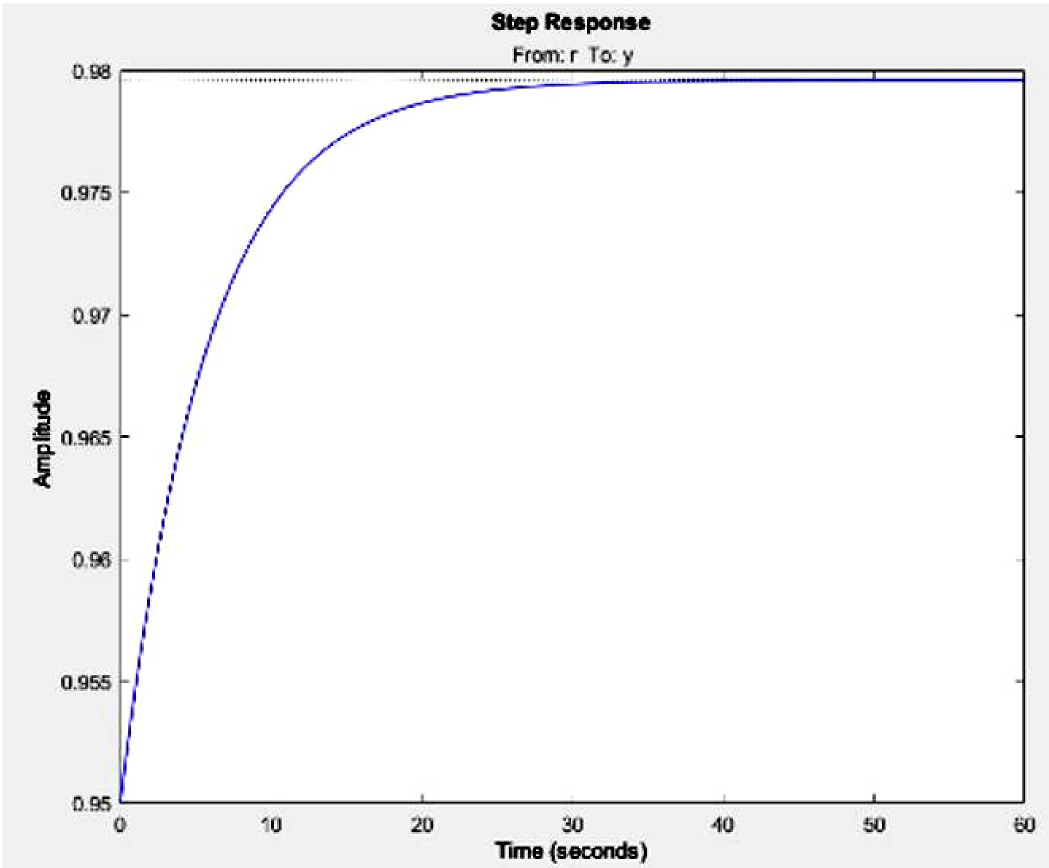
Root Locus plant - System Poles and Zeros.



ISSN 1692-8407

Figure 7

Step response of the system - $K_p = 1$.



- Steady state error $ess = 1$ percent.

In the Root locus, the stabilization and overshoot time parameters were configured to establish the area where the controller should be located and comply with the parameters. Fig. 8 shows the design area. In the step response, the stationary time error is set to 1 percent (Fig. 9).

Since the transfer function provides us with the position of the zeros and poles, it is not sufficient to meet the previously established design criteria since even if the gain of K_p is increased, it always stabilizes in 26 s. To improve the stabilization time, a pole will be included which, together with the proportional controller, generates a PI controller.

In MATLAB an integrator was added with a pole at the origin and a zero at -1, this causes an overshoot of 0.24 percent and a stabilization time of 14 s, a time that does not meet the design criteria. Fig. 10 shows the addition of a pole and a zero, and Fig. 11 shows the response of the controller over time.

To bring the stabilization time to a value that meets the criteria, the controller gain will be moved along the path obtained by Root Locus and if necessary change the poles or zeros over the design zone.

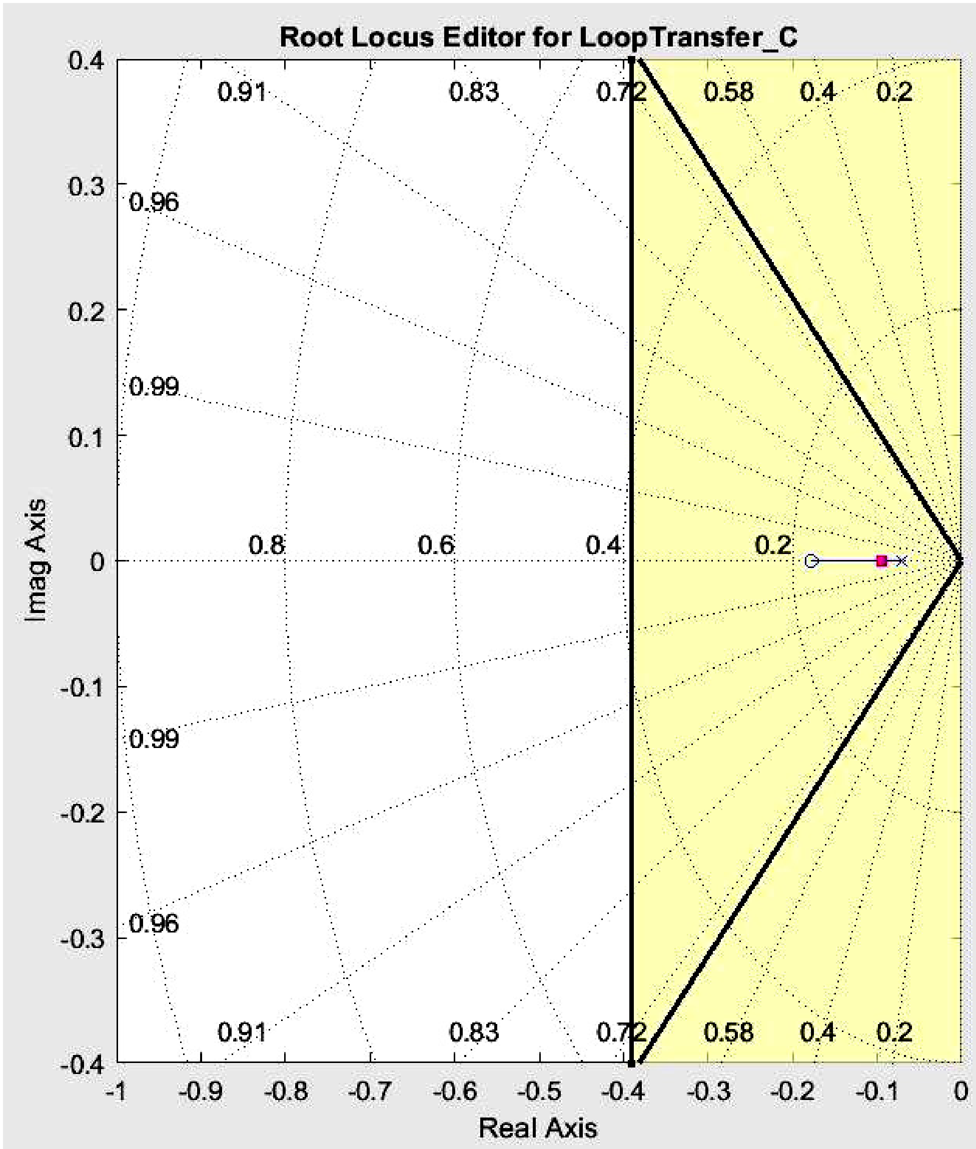
The response that best met the design parameters is stabilized at a time of 9.82 s, with an ess of 1 percent and an overshoot of 2.98 percent, as shown in Fig. 13. To reach this response, the real zero was shifted to -17.5 as shown in Fig. 12 and the gain of the controllers to 0.128, thus obtaining a PI controller with a value of $(0.128\ (\ 1{+}0.054s\ ))/s$, where $K_p = 0.0069$ and $K_i = 0.128$.

It was identified that only by using a controller with proportional and integral part does the system behaves stable, therefore it was not necessary to implement the differential controller. These data were chosen to perform the external disturbance test with the SIMULINK Simulator since it did not allow very large values that also made the system stable and satisfy the design criteria.

ISSN 1692-8407

Figure 8

Design area.



To verify the veracity of the results obtained with the MATLAB Simulator, the values of the constants were taken and simulated in Python (Figs. 14 and 15).

External disturbance test

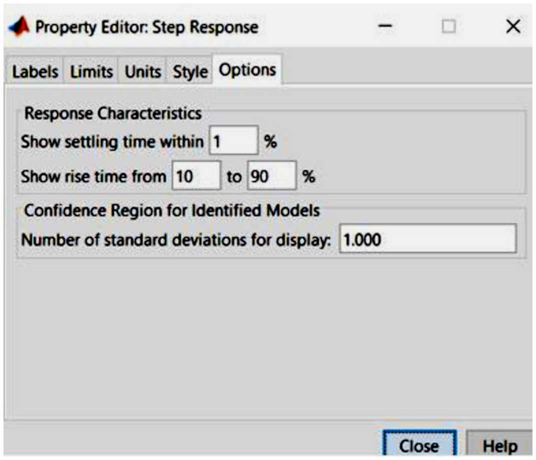
The system's external disturbance test was performed using a 0.5 magnitude pulse with a duration of 3 s and a period of 30 s (Kokieva et al., 2020; Roscoe et al., 2020). This signal type was used to simulate the opening of an incubator door. The circuit was assembled in SIMULINK as shown in Fig. 16. Fig. 17 shows the behavior of the PID controller when tuned with the circuit of Fig. 16.

Fig. 18 shows the response graph to the simulated disturbance, where there is an elevation of 1.039 during the first 3 seconds (pulse duration). After these three seconds the signal decreases to 0.089, begins to stabilize until it reaches the ideal temperature, and takes 6.7s for its complete stabilization. This shows that the designed PI controller corrects the disturbances quickly and stably (Fig. 19).

ISSN 1692-8407

Figure 9

Stationary error.



Acknowledgement

This work was supported by the Universidad Distrital Francisco José de Caldas, specifically by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by Universidad Distrital.

Conclusion

Through the development of this experiment, the different possible responses of the system to the different types of controllers are observed, observing the best response when using a PD or PID controller. These, however, must be correctly adjusted to obtain the desired response, since in case of generating an erroneous adjustment, adverse and unforeseen effects can be generated, or in more serious cases, cause the destabilization of the system. An automatic system was implemented to control the temperature of an incubator, managing to maintain a stable temperature, improving the quality of egg production in an artificial incubator. Since the transfer function is of the first order, no matter how much the value of Kp is increased and K_i and K_d are kept at zero, the response of the system will never oscillate, therefore it is not possible to calculate the value of P_c. Different behavior if it were a function of second degree that as K_p increases if it oscillates. The incubator behavior curve has no delay, so it is not possible to use the Ziegler-Nichols reaction curve and Cohen-Coon methods in which the delay time is necessary to find the constants.

For the control of the incubator, a design with a PI controller was implemented since it was identified that the differential controller is not necessary for the system to be stable and to have external disturbance to be able to stabilize the system quickly. Using the MATLAB software with the SISOTOOL tool facilitates the obtaining of the PID controller thanks to the Root locus in which the poles and zeros can be identified quickly with only knowing the plant of the system, in addition, it allows to visualize the graph of the response to the system step and thus to identify the key points of evaluation such as the overshoot, the stationary error, the stability time, the rise time, among others.

The usefulness of using software to obtain the Root Locus is important because it reduces the design time, calculation and prevents errors and attributes more precision and accuracy when obtaining the controller. Although MATLAB gives a possible answer in many cases it does not meet the design requirements, therefore, the designer must modify the data obtained by MATLAB, so that it fits the necessary criteria.

The PID controller is a robust tool, widely known and used at the industrial level, and really powerful to achieve optimal process operation. However, on many occasions the tuning requirements of its parameters are unknown or disregarded, restricting the process to a game of trial and error. These types of exercises, in addition to their industrial applications, provide very useful academic tools for young designers interested in the field, especially now that design tools such as MATLAB and Python are available. It is therefore expected that the results of this article, presented in a practical way, will help develop professionals in the field and help to extend design applications for this important component of control systems.

References

Aldair, A. A., Rashid, A. T., & Mokayef, M. (2018). Design and implementation of intelligent control system for egg incubator based on iot technology. *2018 4th International Conference on Electrical, Electronics and System Engineering (ICEESE)*, 49–54.

Bello, D., García, L., & Martínez, F. (2013). Planta robusta de levitación neumática para investigación y formación en control y visión artificial. *Tekhnê*, *10*(2), 33–41.

Blanco, L., & Ramos, M. (2019). Sustainable diets and meat consumption reduction in emerging economies: Evidence from colombia. *Sustainability*, *11*(23), 6595. https://doi.org/10.3390\/su11236595

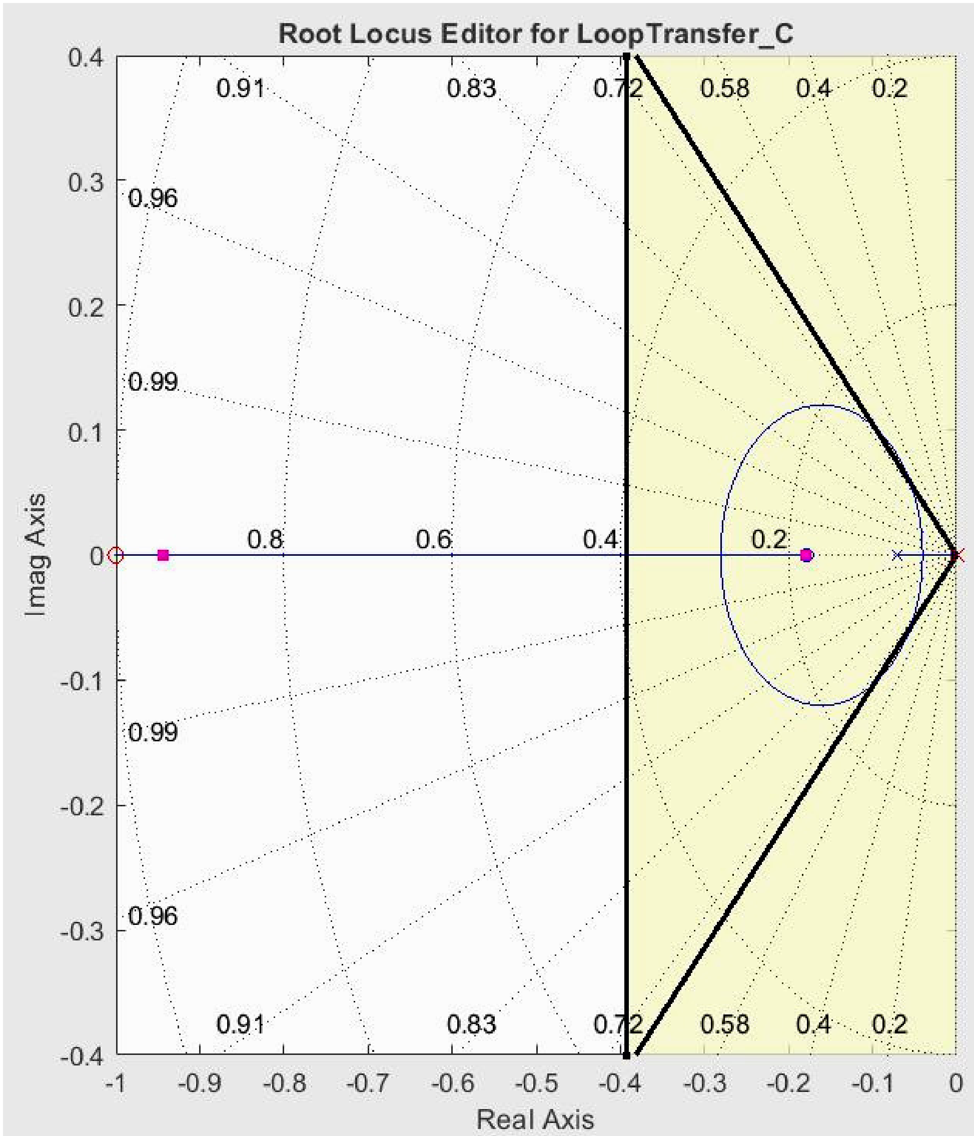
Che, A., Muzaffar, A., Shukri, A., Mohd, R., Rozali, S., Mohamad, S., & Hadinata, A. (2019). Analysis of light bulb temperature control for egg incubator design. *International Journal of Integrated Engineering*, *11*(4), 268–276. https://doi.org/10.30880\/ijie.2019.11.04.031

Colmenares, J., Ospino, C., & Vargas, O. (2003). Adecuación y automatización para un horno industrial a gas natural. *Tekhnê*, *1*(1), 24–29.

ISSN 1692-8407

Figure 10

Root Locus PI.



Diseño e implementación de una incubadora de huevos con aire forzado basada en microcontrolador. (n.d.). *2018 Conferencia Internacional sobre Avance en Ingeniería Eléctrica y Electrónica (ICAEEE).*

Farfán, D., Martínez, F., & Pirajan, R. (2011). Development system for atmel microcontroller oriented towards plc. *Tekhnê*, *8*(1), 47–54.

Guzmán, J., & Ramírez, J. (2017). How can concrete be improved through processes that are bio-friendly to the environment and that enhance its construction qualities? *Tekhnê*, *14*(2), 49–58.

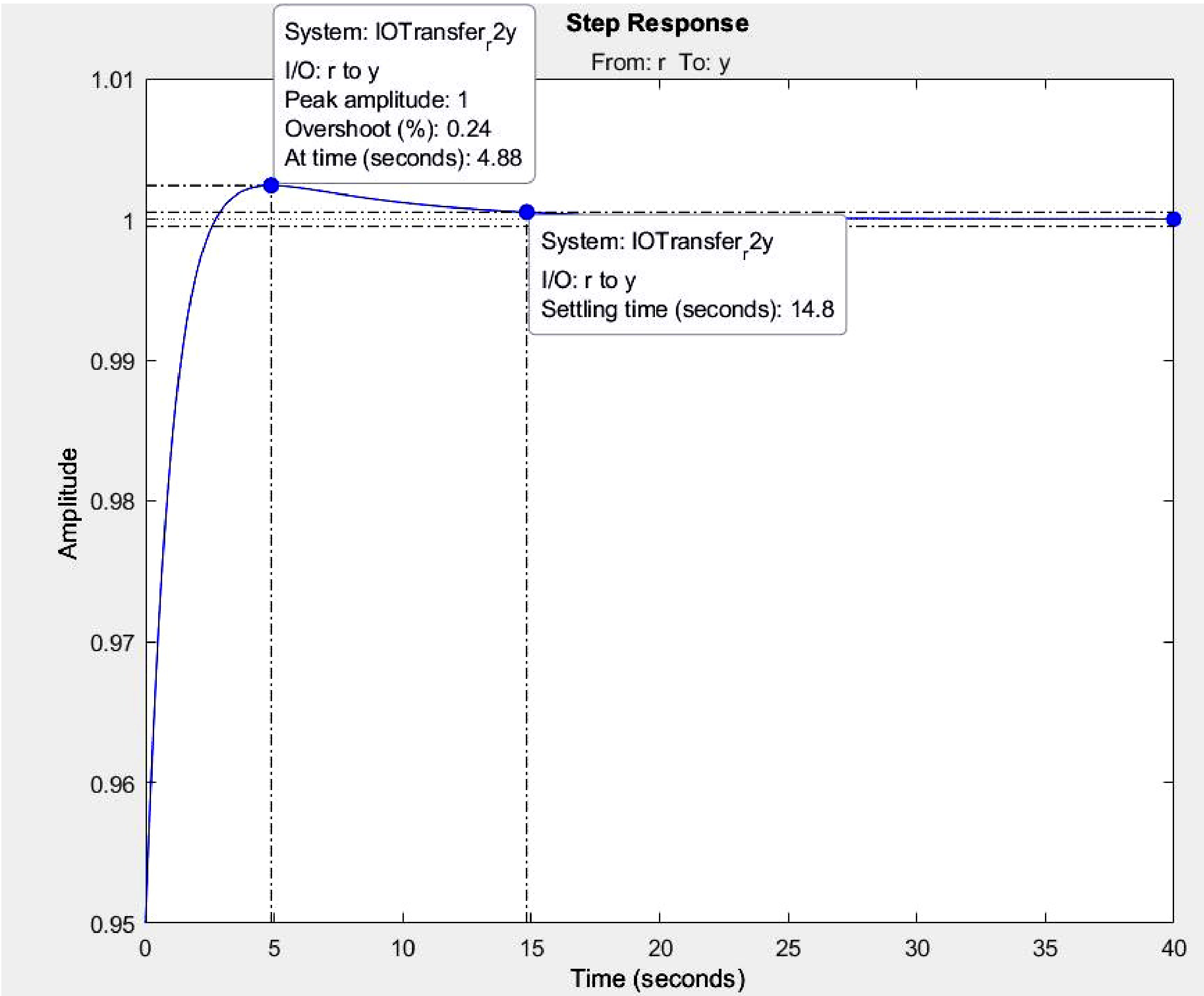
Kapen, P. T., Youssoufa, M., Foutse, M., Manfouo, H., & Mbakop, F. O. N. (2020). Design and prototyping of a low-cost, energy efficient eggs incubator in developing countries: A case study of cameroon. *Scientific African*, *10*(1), e00618. https://doi.org/10.1016/j.sciaf.2020.e00618

Khera, N., & Kohli, S. Comparison of fuzzy and on/off controller for winter season egg hatching incubator system. In: *2018 international conference on sustainable energy, electronics, and computing*

ISSN 1692-8407

Figure 11

PI controller time response.



systems (SEEMS). 2018, 1–6. https://doi.org/10.1109/SEEMS.2018.8687365.

Kokieva, G. E., Trofimova, V. S., & Fedorov, I. R. Greenhouse microclimate control. In: *Iop conference series: Materials science and engineering*. *1001*. (1). 2020, 012136. https://doi.org/10.1088/1757-899X/1001/1/012136.

Lata, S., Sah, R., Singh, S., & Jaiswal, S. Greenhouse monitoring using wsn and sensenuts nodes. In: *Aip conference proceedings*. *2294*. 2020, 030006. https://doi.org/10.1063/5.0031711.

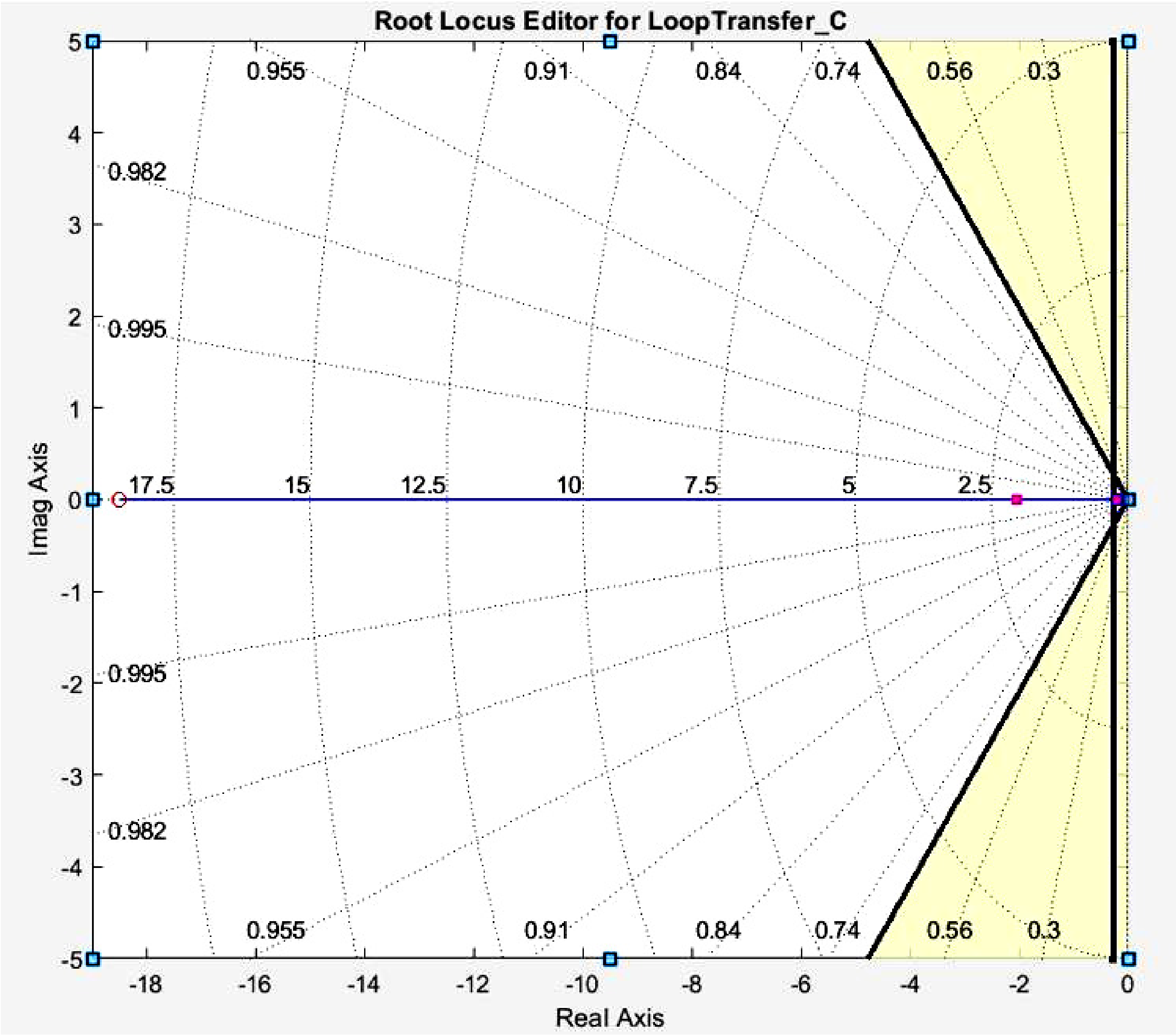
López, C. (2014). Performance evaluation of two bio-inspired optimization techniques: Genetic algorithms and particle swarm. *Tekhnê*, *11*(1), 49–58.

Maasoum, S. M. H., Mostafavi, A., & Sadighi, A. An autoencoder-based algorithm for fault detection of rotating machines, suitable for online learning and standalone applications. In: *2020 6th iranian conference on signal processing and intelligent systems (ICSPIS)*. 2020, 1–6. https://doi.org/10.1109\/ICSPIS51611.2020.9349574.

ISSN 1692-8407

Figure 12

System PI controller.



Marçal, T., Antunes, B., Ferreira, R., Correia, C., Pires, D., Matos, A., & Simões, J. N. (2015). Wireless multi-physiological signal monitor for clinical discharge and readmissions criteria setting and ambulatory usage. *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, 1–6.

Millette, S., Hull, C. E., & Williams, E. (2020). Business incubators as effective tools for driving circular economy. *Journal of Cleaner Production*, *266*(1), 121999. https://doi.org/10.1016/j.jclepro.2020.121999

Páramo, M., Romero, M., & Wilches, F. (2020). International journal of engineering research and technology. *International Journal of Engineering Research and Technology*, *13*(11), 4004–4010.

Roscoe, A., Knueppel, T., Silva, R. D., Brogan, P., Gutierrez, I., Elliott, D., & Campion, J.-C. P. (2020). Response of a grid forming wind farm to system events, and the impact of external and internal damping. *IET*

ISSN 1692-8407

Figure 13

Time response of the plant PI controller.

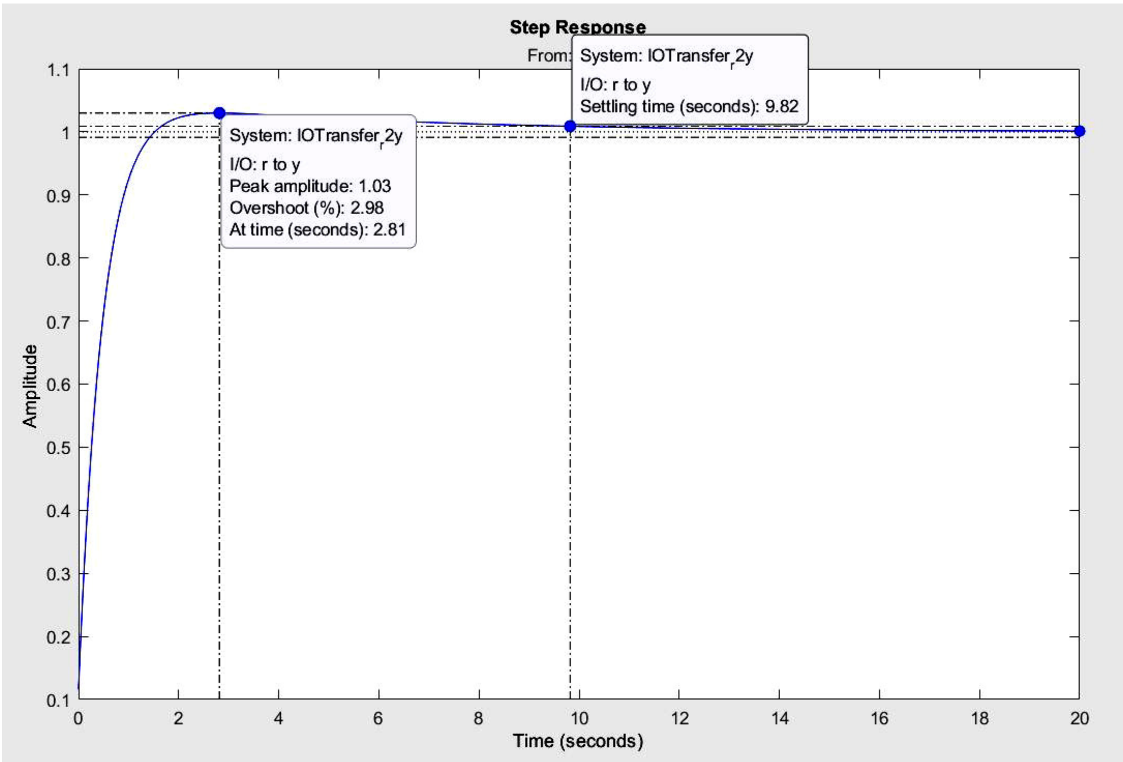
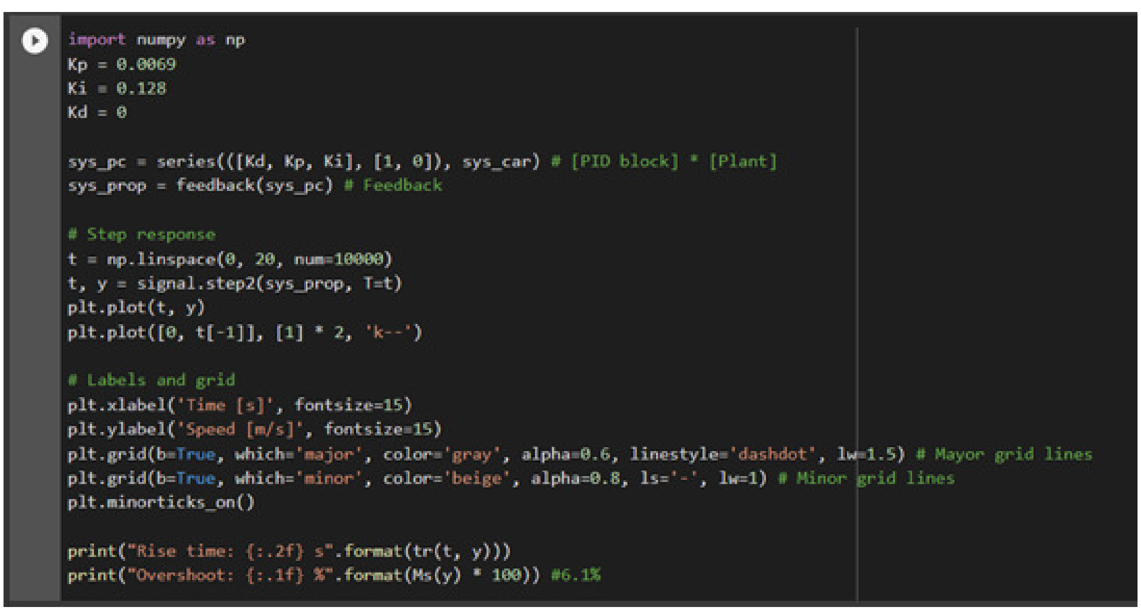


Figure 14

Python simulation code.



```
import numpy as np
Kp = 0.0069
Ki = 0.128
Kd = 0

sys_pc = series(([Kd, Kp, Ki], [1, 0]), sys_car) # [PID block] * [Plant]
sys_prop = feedback(sys_pc) # Feedback

# Step response
t = np.linspace(0, 20, num=10000)
t, y = signal.step2(sys_prop, T=t)
plt.plot(t, y)
plt.plot([0, t[-1]], [1] * 2, 'k--')

# Labels and grid
plt.xlabel('Time [s]', fontsize=15)
plt.ylabel('Speed [m/s]', fontsize=15)
plt.grid(b=True, which='major', color='gray', alpha=0.6, linestyle='dashdot', lw=1.5) # Mayor grid lines
plt.grid(b=True, which='minor', color='beige', alpha=0.8, ls='-', lw=1) # Minor grid lines
plt.minorticks_on()

print("Rise time: {:.2f} s".format(tr(t, y)))
print("Overshoot: {:.1f} %".format(Ms(y) * 100)) #6.1%
```

ISSN 1692-8407

Figure 15

PI controller response with Python.

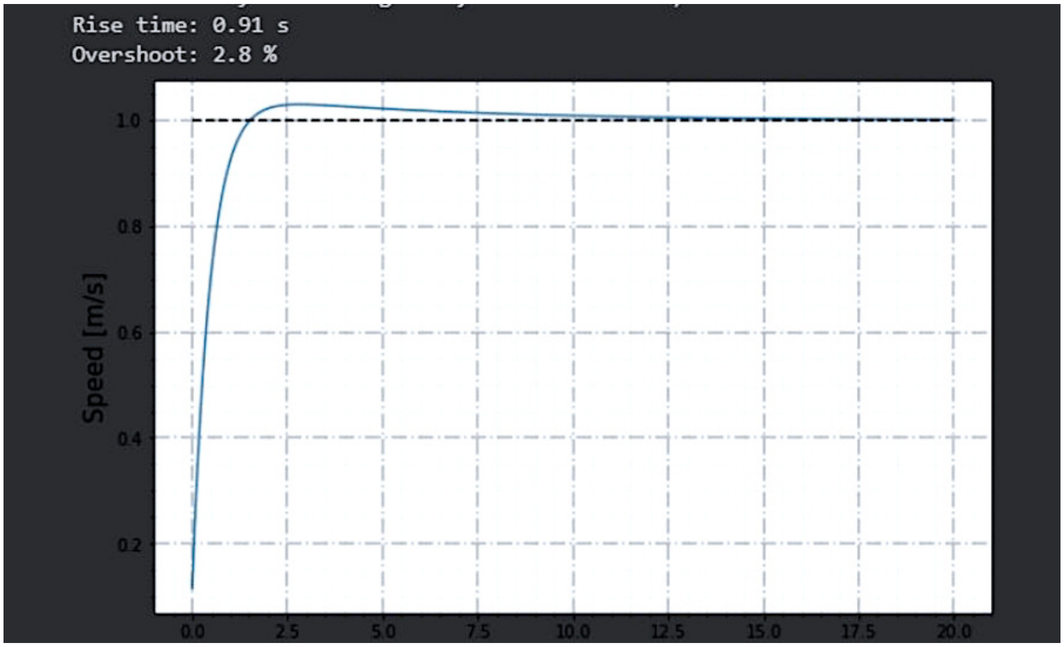
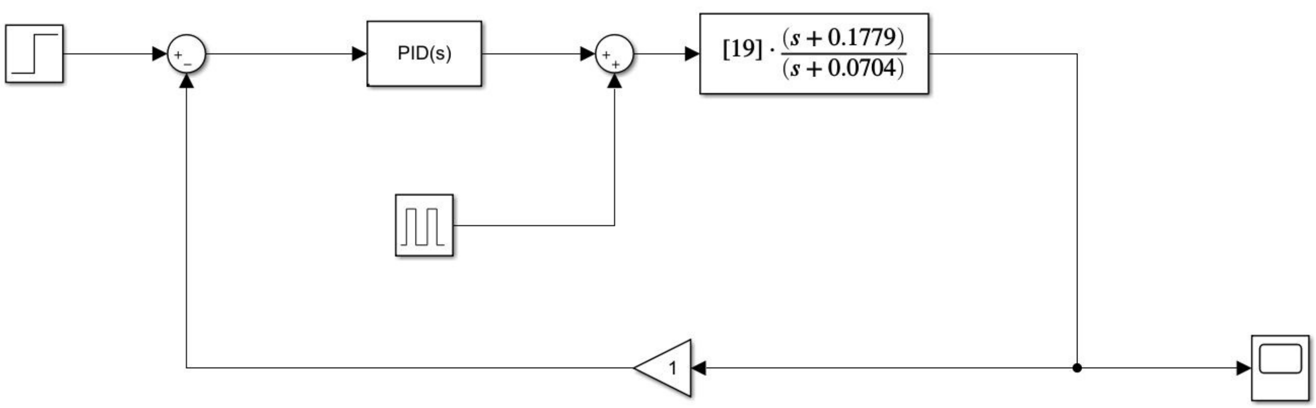


Figure 16

SIMULINK assembly.



Renewable Power Generation, *14*(19), 3908–3917. https://doi.org/10.1049\/iet-rpg.2020.0638

Sanjaya, W. S. M., Maryanti, S., Wardoyo, C., Anggraeni, D., Aziz, M. A., Marlina, L., Roziqin, A., & Kusumorini, A. (2018). The development of quail eggs smart incubator for hatching system based on microcontroller and internet of things (iot). *2018 International Conference on Information and Communications Technology (ICOIACT)*, 407–411.

Santoso, S. B., Adhy, S., Bahtiar, N., & Waspada, I. Development of the smart chicken eggs incubator based on internet of things using the object oriented analysis and design method. In: *2020 4th international conference on informatics and*

ISSN 1692-8407

Figure 17

PI response with SIMULINK.

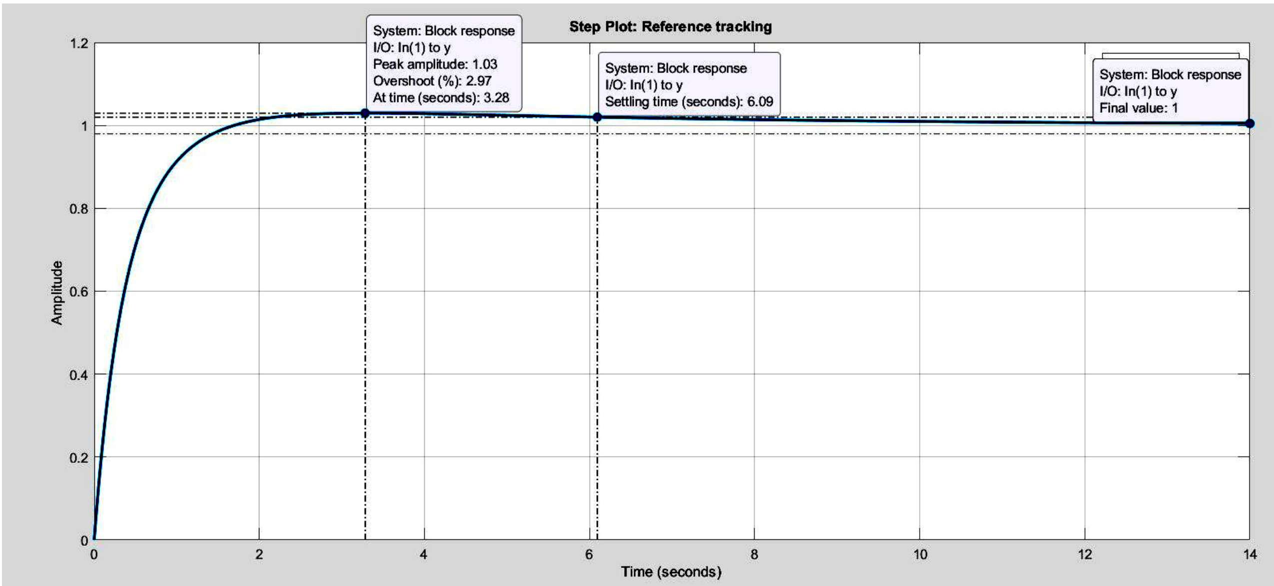


Figure 18

Behavior against disturbances.

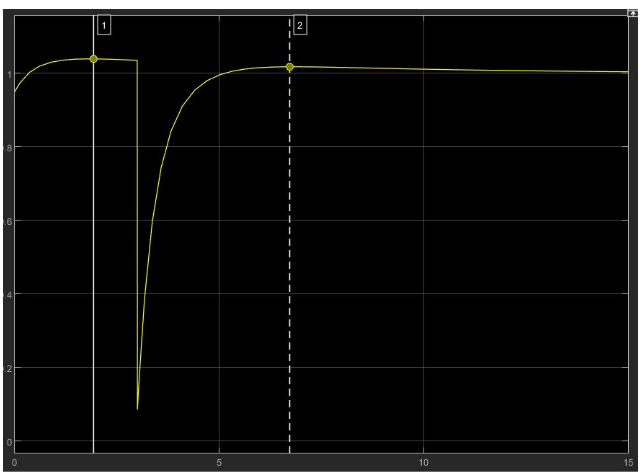
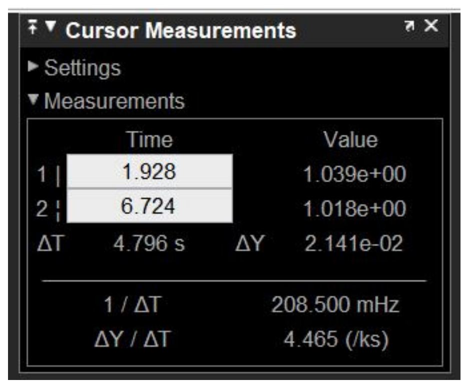


Figure 19

External disturbance test response.



computational sciences (ICICoS). 2020, 1–6. https://doi.org/10.1109/ICICoS51170.2020.9299000.

Shafiudin, S., & Kholis, N. (2018). Monitoring system and temperature controlling on pid based poultry hatching incubator. *IOP Conference Series: Materials Science and Engineering*, *336*(1), 012007.

Torres, J., & Ramírez, L. (2012). Adequacy of festo vacuum system belonging to the laboratoryof automation and control (le-mc-01) of the program inmechanical technology ud. *Tekhnê*, *9*(1), 69–78.

Zutta, J. (2005). Diseño, construcción y evaluación de prototipo mecánico para el lavado y sedimentación del almidón de arracaha (arracacia xantorriza). *Tekhnê*, *4*(1), 1–3.