Journal of Computer Science and Technology Studies (JCSTS))
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

Simulation and Implementation PID Controlling Buck Converter DC

I Wayan Raka Ardana[1] ✉ **Lalu Febrian Wiranata**[2] **and Ida Bagus Irawan Purnama**[3]
[123]*Politeknik Negeri Bali, Indonesia*
✉ **Corresponding Author:** I Wayan Raka Ardana, **E-mail**: rakawyn@pnb.ac.id

ARTICLE INFORMATION

Received: 04 October 2021
Accepted: 21 November 2021
Published: 07 December 2021
DOI: 10.32996/jcsts.2021.3.2.7

KEYWORDS

Buck Converter, PID, Simulation, Matlab, Proteus

ABSTRACT

Regulating the output voltage based on the desired set point is useful for many applications. However, getting the optimal value using fast computation with minimal error is still challenging. This paper aims to design, simulate, and implement a second-order Buck-Boost DC-DC converter circuit so that the voltage result according to the desired set point can be achieved. Initially, testing is conducted using Matlab Simulink. Then, Proteus is used to test the computation of the program on embedded systems in which the result is implemented in C. In low voltage power electronics applications, this approach has never been used to determine the output form. To determine the value of Kp, Ki, dan Kd, PID, Ziger Nichos (Guo, 2002). method is used. Meanwhile, tuning is done through Matlab. For simulation on Proteus, the output is tested by setting the setpoint values of 3.0, 2.5, and 1.7 volts. This aims to see the pattern of changes in the simulation. The simulation results with Proteus show that they have similar peak values but with different overshoot values. This is because the simulation must pass the reference voltage before it drops to the desired setpoint value. Proteus simulation can also help to prove embedded system programs are running correctly. On the other hand, the value of 1.7 volts is used as a setpoint in device implementation. This is due to the determination that the setpoint voltage in the implementation does not exceed the value of the source/power supply. The results show that for the rise time value of 378,770 ms, Overshoot and settling time are 11.798% and 0, respectively. This means the result produces an optimal value which is a return to the initial target. The optimal factor is assessed from the ability to minimize existing errors as well as having the shortest possible computational process.

1. Introduction

The main objective of this research is to design a DC buck converter electronic component simulation using Matlab and Proteus tools. This is then followed by implementing this design to get live data of the output form. Simulation with Matlab is carried out to easier determine the values of Kp, Ki, Kd from Matlab tuning. Meanwhile, simulation with Proteus is carried out to test the output form against the setpoint. This also aims to find out whether the programming algorithm on embedded systems is running correctly. The second-order system, which is the main target, is first reviewed in depth both when the state is off and when the state is on. A better analysis is expected to be carried out to understand the output form of the second-order system used in both conditions (Shirazi, 2009).

To determine the value of Kp, Ki, and Kd, the Ziger-Nichos method is used. However, in Matlab, it can be searched by simulation programming. A DC-DC Buck Converter is used for the main system, which has a high level of efficiency [3]. It also describes in detail the conditions when the system is in the state off and state on and the form of modelling when the switching process occurs. This makes it is easier to analyze the shape of the state space program used for simulation and implementation. For the results, first tested by simulation using Matlab to get the parameters Kp, Ki, and Kd, and to see the resulting output form. Before being implemented into embedded systems, it is first tested using Proteus to be a reference for the success rate in determining the algorithm in its implementation.

2. Literature

*T*he literature review relates to some of the basic concepts used in this study. Some references and analytical processes from several standard journals and books are presented. Among them are the basic concepts of PWM (Pulse Width Modulation) in regulating the duty cycle, a buck converter to adjust the output speed of the Mosfet so that it is possible to set the desired setpoint, PID to determine the magnitude of the parameter, as well as the stability of the desired output.

2.1 Parameter Step Response

Fig. 1 describes the response parameters used as benchmarks for determining the quality of the output system. These parameters include RiseTime, which is the rising time required when the response increases from 10% to 90% or from 0% to 100% of the peak value, Overshoot which is the required time to reach the first peak value or maximum peak value, and SettlingTime which is the time required by the response output to enter and reach an output range (usually between 2% and 5%) (Ogata, 1997).

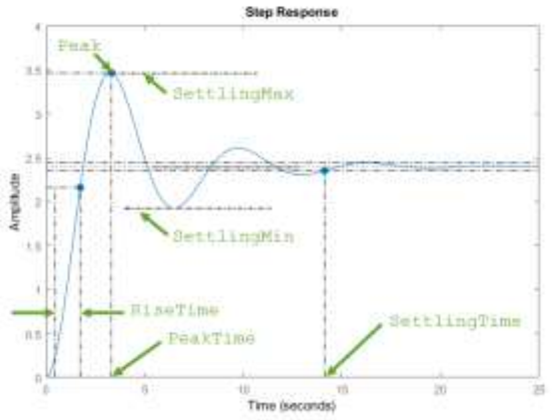

Fig. 1. Response Parameter (Mathwork, 2021)

2.2 PWM (Pulse Wide Modulation)

PWM is a technique for substituting the modulation by changing the pulse width (duty cycle) without changing the frequency and amplitude values. The generated pulse condition has a high and low zone. The pulse width given in this modulation is in the form of a percentage (%) from 0% to 100%, representing the pulse width. If 100%, then the signal will continue to be in a high condition or can be found at 50% pulse width, which means the duty cycle is in a balanced on-off position at a certain frequency. Usually, this PWM signal is easily generated by the MCU (microcontroller unit) for various purposes, especially to control the output of the MOSFET, which has a very fast opening and closing control when traversed by voltage or current. The pulse width of the PWM can be adjusted as needed. In this paper, the frequency of 62500 Hz is used (Celanovic, 2000).

Figure. 2 describes the working principles for calculating the duty cycle at a frequency of 50 Hz so that the percentage of the duty cycle used in a certain time range is obtained. This relates to the opening and closing speed of the MOSFET valve itself.

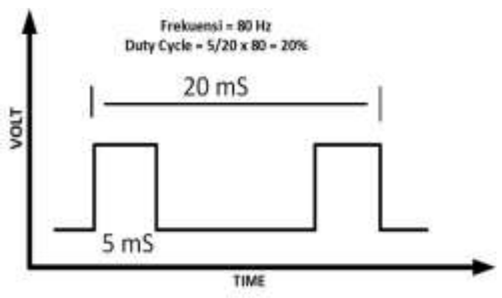

Fig. 2. Pulse Width Modulation (PWM)

2.3 Buck Converter

Buck Converter is a device that contains several electronic components and has the characteristic of lowering the voltage. This device also has high efficiency (Liu, 2009). Usually, the Buck Converter consists of Mosfet, Inductor, Capacitor, Diode, and Resistor. The working principle of the Buck Converter is to regulate the PWM output voltage on the MCU. When it gets a signal, the Mosfet will open the current valve and cause the current to fill the inductor and store it in a magnetic field form. This state also causes the capacitor to be able to charge enough voltage to supply the output in a short amount of time.

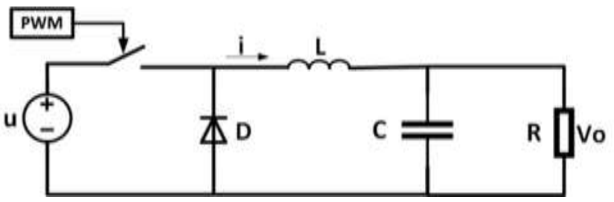

Fig. 3. Buck Converter Circuit

Fig. 3 shows the Buck Converter circuit when it is ON (switch-on) or when the input voltage passes through the component. While the OFF state describes when there is a switch-off from the Mosfet, the inductor releases the current stored in the component. In this case, the values of A, B, C, D are the system matrix, *x* is the state variable, *x* is the form of the state variable, *u* is the system input, and *y* is the system output. The relations of these variables are written in equations (1) and (2).

$$x = Ax + Bu \tag{1}$$

$$y = Cx + Du \tag{2}$$

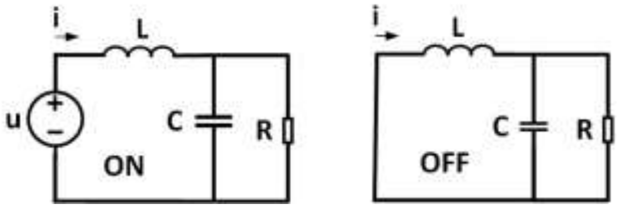

Fig. 4. Buck Converter during ON and OFF state

For state variables in voltage and current on the Buck Converter is written as V_C and i_L. During ON state V_C and i_L, it can represent with equations (3) and (4).

$$V_c = u - L\frac{di}{dt} \tag{3}$$

$$i_L = C\frac{dv}{dt} + \frac{V_c}{R} \tag{4}$$

With $i_L = x_1$ and $V_c = x_2$, so we can write the derivative on the equation $x_1{}'$ and $x_2{}'$ and the equation become (5) and (6). The state-space matrix form when the buck converter is ON is shown in equation (7).

$$x_1{}' = -\frac{1}{L}x_1 + \frac{1}{L}u \tag{5}$$

$$x_2{}' = \frac{1}{C}x_1 + \frac{1}{RC}x_2 \tag{6}$$

$$\begin{bmatrix} x_1{}' \\ x_2{}' \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \tag{7}$$

The OFF mode where u is 0 can be explained by equation (8), which becomes the derivative value, while the state space form is shown in equation (9).

$$x_1{}' = -\frac{1}{L}x_2 \tag{8}$$

$$\begin{bmatrix} x_1{}' \\ x_2{}' \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{9}$$

To combine the values of the shape of the state space matrix when the ON and OFF modes of the buck converter occur, it is necessary to add a switching duty cycle *d*, which is shown in equations (11) and (13).

$$\bar{A} = A_{(ON)}d + A_{OFF}(1-d) \quad (10)$$

$$\bar{A} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} d + \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}(1-d) = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \quad (11)$$

$$\bar{B} = B_{(ON)}d + B_{OFF}(1-d) \quad (12)$$

$$\bar{B} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{d}{L} \\ 0 \end{bmatrix} \quad (13)$$

To complete the form of the matrix state-space system, equations (11) and (13) can be substituted into equation (1).

$$\begin{bmatrix} x_1{}' \\ x_2{}' \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (14)$$

So, that the form of the output system from the Matrix state-space V_C and i_L, is written in the form of equation (15), which becomes the form C and D in the matrix output system [9, 10].

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (15)$$

2.4. PID (Proportional Integral Derivative) Controller

PID control is one of the classic controls which is quite popular used in the industrial world. Almost all industries have implemented this type of control because, besides being easy to understand, this type of control is easy to implement in its application. In this paper, we try to control the PID type to control the duty cycle speed of the MOSFET opening and closing process so that it is possible to apply it to the valve opening and closing. The PID control has slightly different specifications because of the proportional control itself.

PID control consists of proportional, integral, and derivative. This type of proportional control calculates the difference (error) between the setpoint and the control variable each time the sampling is taken.

$$u_p(t) = K_p e(t) \quad (16)$$

Derivative control is used to fix all existing errors quickly and immediately eliminate these existing errors.

$$u_d(t) = \frac{d}{dt}e(t) \quad (17)$$

A closed-loop system can also be written with a combination of P, I, D written in equation (18) so that it becomes equation (19). Here, the closed-loop system changes from the t domain to the s domain.

$$u\,(t) = K_p e(t) + K_p K_i \int e(t)\,dt + K_p K_d \frac{de(t)}{dt} \quad (18)$$

$$\frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i S} + T_d S\right) \quad (19)$$

Kp is the proportional gain, Ki is integral gain, Ti is the integral time constant, and Td derivative time.

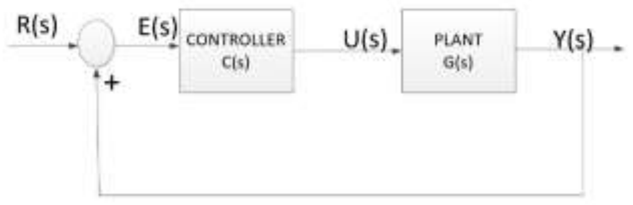

Fig. 5. Close Loop Buck Converter

Fig. 5 shows R(s) input, E(s) error, U(s) control signal, and Y(s) output in close loop buck converter. The Ziegler-Nichols method is used to find the value of an existing parameter. This method can be used in any way. Set the Kp value to get the form of an oscillation or an oscillating output wave (Kcr); besides that, the value of Ki, Kd given is 0. Fig. 6 shows the setting of the Kp value to oscillate. Then, after the oscillating value is obtained, the next step is to calculate the peak-to-peak distance (Pcr) value of the Pcr multiplied by the PID parameter value. Kp = 0.6 Kcr, Ki = 0.5 Pcr and Kd = 0.125 Pcr. (Astrom, 2010).

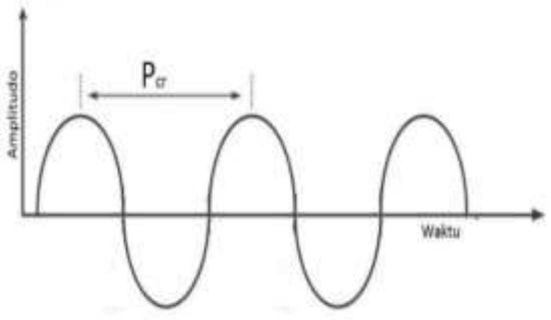

Fig. 6. PID ZN Tuning explains the manual process of getting P*cr* value

3 Results and Discussion

This paper presents an analysis of simulation using Matlab Simulink tool to find the PID tuning value and use Proteus simulation to run the application in a more realistic direction. Accordingly, it can be implemented by C language before testing its form.

3.1 Matlab Simulation Results

This section further discusses the second-order systematic form of the buck converter, which is the main design in analyzing the output. As for the scope, it only discusses the second-order concept in the simulation. The result is analyzed using Simulink Matlab in the form of power output based on some predetermined parameters. The stability obtained from the simulation can be seen in the course of Overshoot, settling time, and rise time. Fig. 7 depicts the system block diagram of the buck converter. The composing of the buck converter system where this system describes the process of measuring the sensor output will be measured using a current and voltage sensor which is the reference for controlling the output of the buck converter. However, the application only shows the form of control on the buck converter.

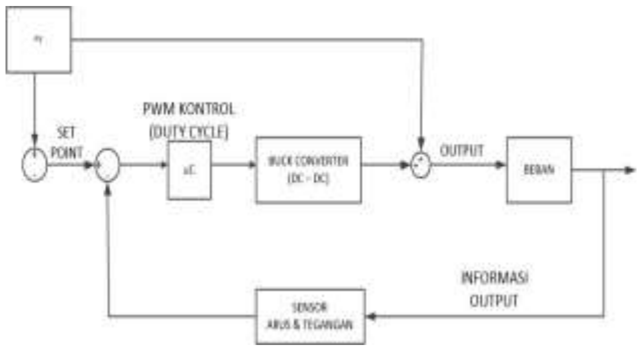

Fig. 7. Close Loop Buck Converter System

The expected process of the close loop in Fig. 7 has a stable output at 12V. In addition, the desired result is that the output has a fast rise time value, as small as possible Overshoot, and does not have a steady-state error. The concept offered is in the form of a Matlab Simulink simulation. First, the V_{in} reference value for the input of the buck converter is presented. Second, the application of the equation form in Matlab Simulink is reviewed the Mosfet's output when switching occurs and other things considered necessary. Third, the application of PID in the buck converter so that the output form is seen according to the desired setpoint. Forth, the application of the equation in Fig. 8 for the buck converter system, where the values used are listed in Table 1. In this subsection, the Matlab switching process is also listed in Fig. 9.

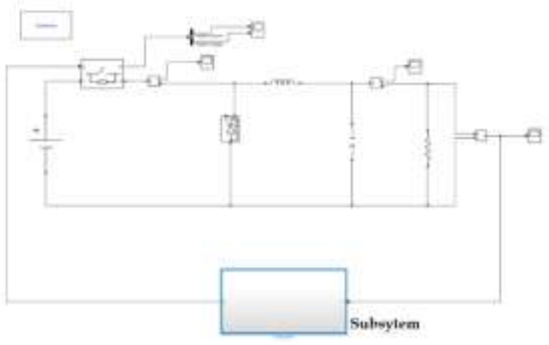

Fig. 8. Buck Converter System

Fig. 8 describes a tested schematic in Matlab Simulink. The used schematic directly tests the PID parameter input form. In principle, when the input voltage is sourced from PV (simulated with 20.38V input from the battery), it will provide the buck converter system voltage and current. The Mosfet will regulate the incoming voltage, which acts as a faucet or valve that allows electronic components such as capacitors and inductors to be filled with their respective charges. Then, from the Mosfet, the DC voltage will start to be stored in the capacitor so that it can withstand some of the load used for a while. This mechanism also applies to inductors that can store electric current in the form of a magnetic field. They can then supply current to the load in a few moments. This aims if there is a demand for a load, there will be no short circuit that causes a spark. In addition, the presence of electronic components makes the current and voltage output more efficient.

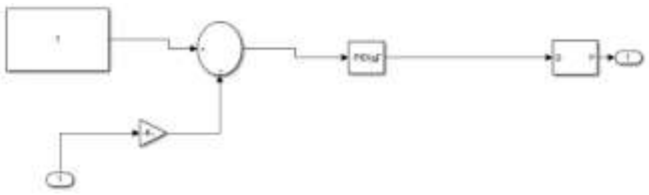

Fig. 9. Buck Converter Sub-System

Fig. 9 buck converter subsystem is a signal generation system at a frequency of 62500 Hz and explains the set point used, which is 12V. The setpoint value is entered into the Gain K with the step system input. This system becomes feedback from the information obtained at the output of the buck converter, which regulates how much voltage and current is required so that it will be able to supply energy into the load more efficient. However, this paper does not simulate the used load and only focuses on the PID concept on the buck converter. The concept of duty-cycle control is described in switching Mosfet, Fig. 10.

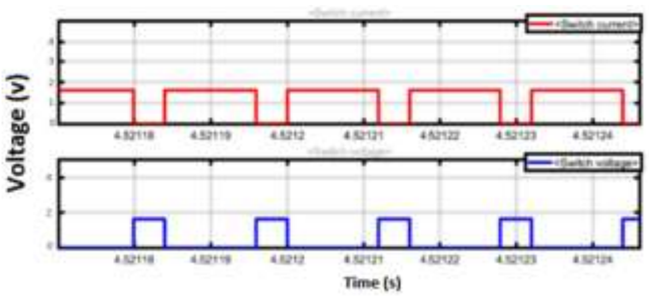

Fig. 10. Switching Mosfet

The simulation results show that the voltage and current of the inductor differ in phase by 90^o due to the switching process in the Mosfet component, when the t_{on} current becomes zero and conversely when the t_{off} current reaches its maximum. The results explain that there is a phase difference of 90^o. This happens because of the switching process in the Mosfet component, which results in the current at the t_{on} being zero while at t_{off} the current becomes maximum. Meanwhile, at the voltage, it happens vice versa. The parameter values are shown in Table 1. The implementation of the PID tuning process results; first, the tuning process is carried out with the standard/default values. The results can be seen in Fig. 11, where the PID value is entered into the buck converter system, Fig. 8.

Table 1. Parameter value of Buck Converter

Parameter	Value
Vin	5.00 V
Resistor (R1)	12 ohm
Capacitor	0,22 F
Inductor	0,22 H
Freq PWM	62500 Hz

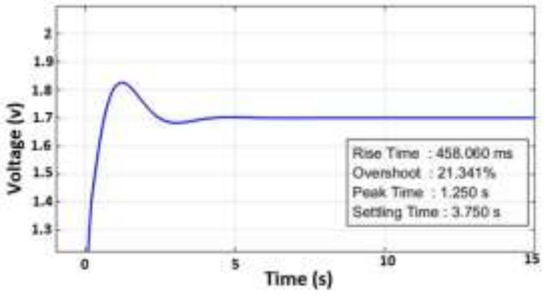

Fig. 11. PID Value in Default Tuning

Fig. 11 explains the data collection process using default values of proportional 2, integrated 5, and derivated 1. This 2, 5, 1 is the standard value of Kp, Ki, and Kd, which is used for the initial determination process. This is very helpful because this basic value can be used as a reference for estimating the PID amount, which is usually not far beyond the basic value used. The results are quite good, and it looks like the setting time is stable after getting it. The ZN PID tuning process is carried out directly by Matlab on the Control Estimation Tool Manager (CETM) tools. This is done to maximize the results, where if done manually, the process is too large for the Kp value itself.

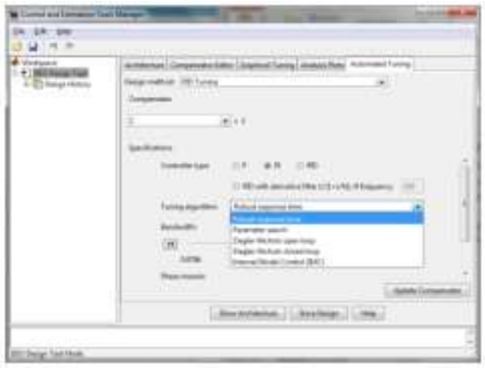

Fig. 12. PID Tuning Parameters

The block-like Fig. 13 PID Simulink tuning can be used for the tuning process or determining PID parameters.

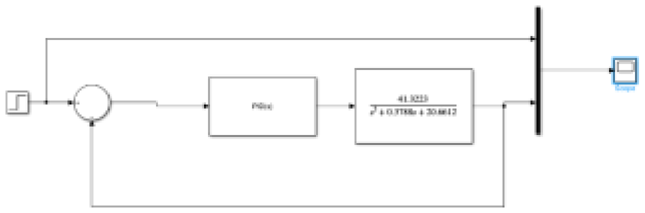

Fig. 13. Simulink Tuning PID

Fig. 14 describes the results for the parameters P = 2,966, I = 7,749, D = 0.

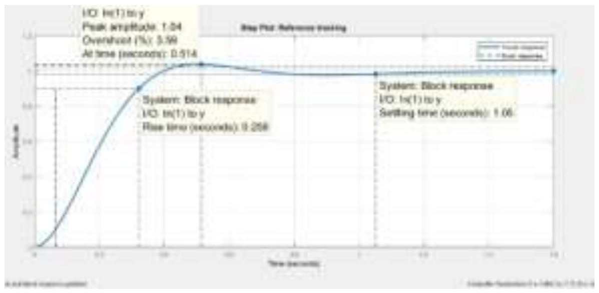

Fig. 14. PID Tuning Result

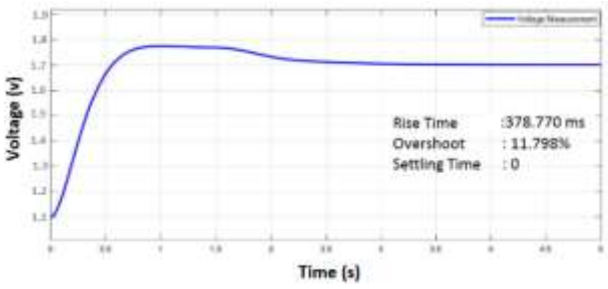

Fig. 15. Buck Converter Simulation Result

Fig. 15. is obtained by entering the parameter values into Fig. 8 of the buck converter system so that the rise time is 378,770 ms, Overshoot 11.798%, and settling time 0. The difference between the three values (rise time, Overshoot, settling time) is quite bigger than the standard value of Fig. 11, also show different results because the parameter values P, I, D have been entered into the test system.

3.2 Proteus Simulation Results and Implementation

Fig. 16 shows the test results for the 1.7V setpoint where in this simulation, three setpoint values are tested, which are 3.0, 2.5, and 1.7V. The configuration form with the implementation is made as close as possible.

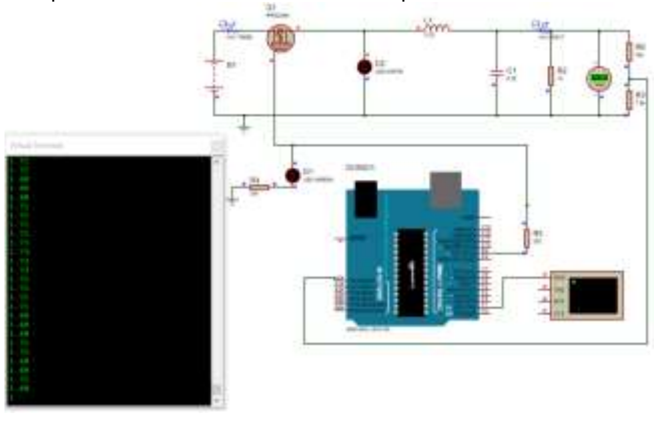

Fig. 16. Buck Converter Simulation Result

Fig. 17 shows the results of different setpoints. It can be seen that the overshot value has the same peak but a different setpoint. This state happens because in the simulation with Proteus, the voltage must decrease from the reference voltage, which in this case is tested at 5V, and only occurs in the simulation. This also proves that both the program made on the embedded system and the hardware can follow the desired setpoint value in simulation.

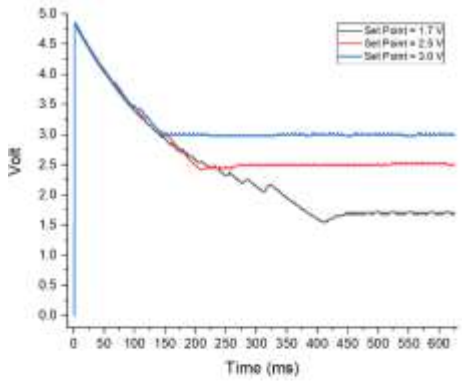

Fig. 17. Result of Implementation

Table 2 lists the results of the obtained setpoint values, where this value is achieved by entering the results from the simulation and processing using Matlab.

Table 2. Simulation Results with Matlab

Set-Point	Rise Time	SettlingTime 0.05%	Overshoot	Peak Time (ms)
3.0	0.5590	Nan	61%	2
2.5	0.4568	186.5	93.2%	2
1.7	0.3148	Nan	185.88%	2

Fig. 17 shows the results of the implementation with a setpoint of 1.7V. This value is tested to match the charging voltage of 4-5 AA batteries. The results are relatively good.

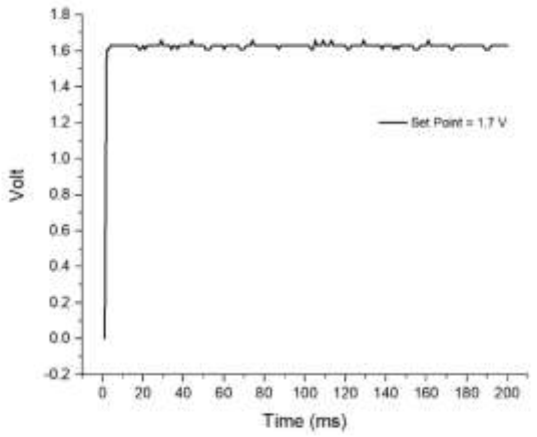

Fig. 17. Implementation Results of Buck Converter

In contrast to the Proteus simulation results, the implementation results provide better data because the voltage does not first pass the reference value but goes directly to the desired setpoint value of 1.7V. It can be seen in Table 3; there is no overshoot. This is because the winding used is quite large with a small setpoint. The average output is 1.625V, which has a difference of about 0.075V from the 1.7V setpoint.

Table 3. Implementation Results

Set-Point	Rise Time	Settling Time 0.05%	Overshoot	Peak Time (ms)
1.7	2.1972	191.25	0%	29

The average value of the relative error after steady-state is obtained, which is about 4.61% of the 1.7 V setpoint. It can be seen from Table 3 that there is no overshoot. This state is due to the use of supercapacitors and large coils. It can be seen from Table 3 that there is no overshoot. This state is due to the use of supercapacitors and large coils.

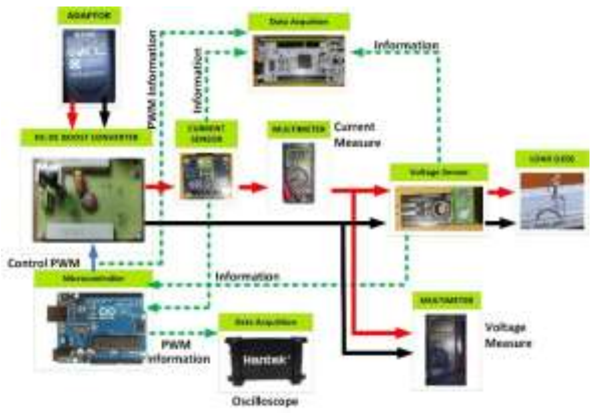

Fig. 18. Buck-Boost Component and Data Acquisition

4. Conclusion

The results obtained in the Matlab simulation are very good used to find the parameter values of Kp, Ki, and Kd. These are used to run the simulation and implementation. In this case, there are differences in the values of rising time, settling time, and Overshoot from each simulation and the experimental results. However, this is not a problem because all the outputs have a fairly stable output value after the Overshoot, especially in the implementation results that have been applied. The results of the implementation are shown in Table 3, where the system has a fairly small peak time and rising time. Meanwhile, the overshoot value is 0%. This result is obtained because the implementation uses a type of capacitor with a value of 0.22 F, which is referred to as a supercapacitor in the market designation and juxtaposed with the inductor value of 0.22 H. This makes the second-order system has two balanced stability components.

References

[1] Astrom, K. (1995). PID controllers. *theory, design and tuning*.

[2] Åström, K. J., & Hägglund, T. (2001). The future of PID control. *Control engineering practice*, *9*(11), 1163-1175.

[3] Ali, A., & Majhi, S. (2009). PI/PID controller design based on IMC and percentage overshoot specification to controller setpoint change. *ISA transactions*, *48*(1), 10-15.

[4] Celanovic, N., & Boroyevich, D. (2000). A comprehensive study of neutral-point voltage balancing problem in three-level neutral-point-clamped voltage source PWM inverters. *IEEE Transactions on power electronics*, *15*(2), 242-249.

[5] Chander, S., mod Agarwal, P., & Gupta, I. (2010, December). FPGA-based PID controller for DC-DC converter. In *2010 Joint International Conference on Power Electronics, Drives and Energy Systems & 2010 Power India* (pp. 1-6). IEEE.

[6] Grassi, E., & Tsakalis, K. (1996, December). PID controller tuning by frequency loop-shaping. In *Proceedings of 35th IEEE Conference on Decision and Control* (Vol. 4, pp. 4776-4781). IEEE.

[7] Gowda M (2014). Modelling of a buck DC-DC converter using Simulink, IJIRSET, 3.

[8] Shirazi, M., Zane, R., & Maksimovic, D. (2009). An autotuning digital controller for DC–DC power converters based on online frequency-response measurement. *IEEE Transactions on Power Electronics*, *24*(11), 2578-2588.

[9] Kwok, K. E., Ping, M. C., & Li, P. (2000). A model-based augmented PID algorithm. *Journal of Process Control*, *10*(1), 9-18.

[10] Liu, Y. F., Meyer, E., & Liu, X. (2009). Recent developments in digital control strategies for DC/DC switching power converters. *IEEE Transactions on Power Electronics*, *24*(11), 2567-2577.

[11] Mathwork,https://www.mathworks.com/help/control/ref/lti.stepinfo.html accessed on May 20, 2021

[12] Ogata, K., & Yang, Y. (2002). *Modern control engineering* (Vol. 4). India: Prentice hall.