International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 20, Number 4 (2025) pp. 677-691

http://www.ripublication.com

Development of Non-Inverting Buck-Boost Converter with PID Controller

Mohd Badril Nor Shah[1]*, Nor Ain Syafiqah Nabila Che Jusoh[1], Muhammad Iqbal Zakaria[2] and Norhaslinda Hasim[1]

[1] *Faculty of Electrical Techonology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia*

[2] *School of Electrical Engineering, Universiti Teknologi MARA (UiTM), Shah Alam, Malaysia*

**Correspondence: Correspondence author*

Abstract

This paper presents the design and implementation of a non-inverting buck-boost DC-DC converter integrated with a Proportional-Integral-Derivative (PID) controller for improved voltage regulation under varying load and input conditions. The converter topology allows both step-up and step-down voltage conversion while maintaining a positive output polarity, making it suitable for microcontroller-based and sensor-driven applications. The control algorithm was implemented using an Arduino microcontroller to ensure cost-effectiveness and simplicity. System parameters, including PID gains, were manually tuned to minimize overshoot, fluctuation, and settling time. Experimental results demonstrate that the proposed system achieves stable and adjustable output voltages ranging from 3 V to 24 V, with minimal steady-state error. Although higher desired voltages exhibited slightly increased overshoot and oscillations, the overall performance confirmed reliable closed-loop regulation. The findings indicate that the developed non-inverting buck-boost converter with PID control provides an efficient, low-cost and flexible solution for applications requiring both step-up and step-down DC voltage conversion.

Keywords: Non-inverting buck-boost, PID controller

Introduction

In modern power electronics, the demand for efficient and stable voltage regulation under varying input and output conditions has increased substantially, particularly in renewable energy systems, battery-powered devices, and automotive applications.

The Buck-Boost converter, recognized for its capability to both step up and step down voltage, provides a versatile and effective solution to meet these dynamic requirements. However, maintaining a constant output voltage under diverse load conditions necessitates a closed-loop control approach.

To address these challenges, this project proposes the development of a Buck-Boost converter integrated with a Proportional-Integral-Derivative (PID) controller. The PID controller is widely utilized due to its robustness, adaptability to both slow and fast dynamic systems, applicability to linear and nonlinear systems, and relatively simple implementation. Moreover, its algorithm can be efficiently executed using a low-cost microprocessor, making it an attractive option for practical and cost-sensitive applications.

A non-inverting buck–boost converter has the same operation of normal buck-boost converter, except the output voltage is positive with respect to the ground. This feature is safer and more compatible for most loads, e.g., microcontrollers, sensors, chargers, etc, since they always been designed to receive positive voltage. However, non-inverting buck–boost converter require additional components to operate. Figure 1 shows the basic circuit of a buck-boost converter and a non-inverting buck–boost converter. From the figure, it can be seen that a non-inverting buck-boost converter require two switches and two diodes as
compared to inverting buck-boost converter.

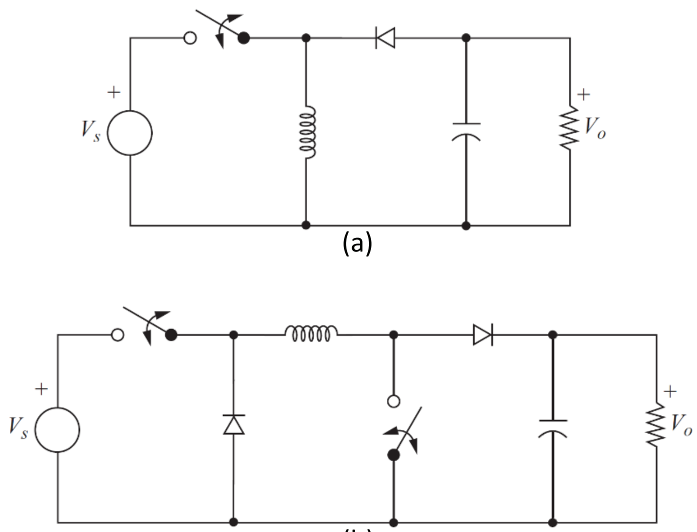


Figure 1. (a) An inverting buck-boost converter (b) A non-inverting buck-boost converter

There are a few works that has been done that are related to non-inverting buck-boost converter with a controller. The comparison between PI, one-cycle control (OCC), and hybrid OCC-PI controllers for a non-inverting buck-boost converter has been done in [1]. The hybrid OCC-PI controller achieved the fastest transient response and lowest steady-state error, especially under input voltage variation. In [2], the paper proposes a Direct Model Predictive Control (DMPC) algorithm for a non-inverting buck-boost converter to regulate output voltage with minimal ripple. Simulation shows faster convergence and reduced overshoot compared to classical PI control. The works in [3] is a combination of sliding-mode control with a Levenberg–Marquardt adaptive predictive algorithm to enhance dynamic stability and reduce steady-state error. The method improves voltage tracking and efficiency under load and source disturbances. An enhanced non-inverting buck-boost topology with improved voltage gain and reduced switch stress has been introduced in [4]. It employs a digital control loop to stabilize transitions between buck and boost modes, achieving more than 94% efficiency. In [5], a DSP-based digital PI and discrete-time controllers has been developed for a non-inverting buck-boost converter. Experimental validation demonstrates stable output regulation across wide input voltage variations.

This project involves the design and implementation of a non-inverting buck-boost converter integrated with a PID controller. The primary objectives are to develop a closed-loop non-inverting buck-boost converter circuit equipped with a PID controller capable of regulating the output voltage within the desired range of 3 V to 24 V, while maintaining acceptable levels of overshoot, settling time, and fluctuation. Additionally, the project aims to demonstrate a design that is both cost-effective and practical for integration into a wide range of DC-DC conversion applications.

Circuit Design

The proposed device comprises five main components: a microcontroller, a buck-boost converter circuit, an output voltage sensor, seven-segment displays, and potentiometers for user input. The microcontroller serves as the central processing unit of the system, providing automated control of the display, voltage measurement, and implementation of the control algorithm for the buck-boost converter. An Arduino-based microcontroller was selected due to its extensive library support, ease of programming, and broad hardware compatibility. The functional block diagram of the project is illustrated in Figure 2, where the potentiometers are used to manually adjust the desired output voltage and tune the controller gains.

Buck-boost converter is a combination of buck converter and boost converter that can step down (buck) or step up (boost) an input voltage to maintain the desired output voltage. The input buck boost converter circuit can be taken any DC source, for example, batteries, DC power adapter, photovoltaic, etc. To maintain a stable output voltage within the desired range, the circuit must be configured as a closed-loop system, in which the voltage sensor module and Arduino microcontroller play essential roles.

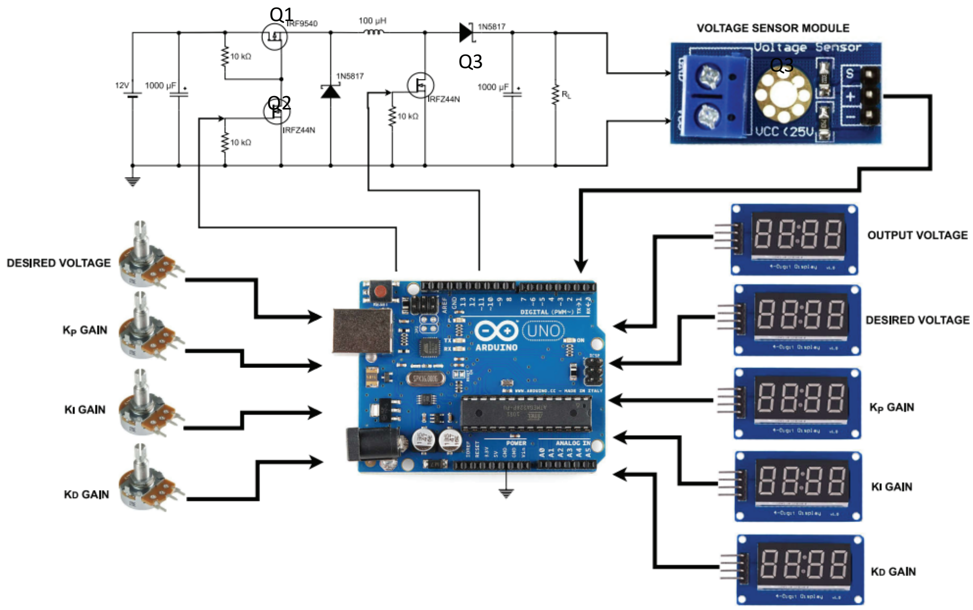


Figure 2. Field diagram of this project

The minimum value of inductor of a buck-boost converter is given by:

$$L_{min} = \frac{(1-D)^2(R_L)}{2f} \quad (1)$$

In this project, the smallest load resistance is $R_L = 510\ \Omega$ and the buck-boost converter is operated in range 0% to 90% duty cycle D. The maximum value of PWM frequency can be by Arduino UNO is 62.5 kHz [6,7]. By taking $D = 0$, therefore

$$L_{min} = \frac{(1-0)^2(510)}{2(62500)} = 4.08 \text{ mH} \quad (2)$$

Hence, the minimum value of inductor of the buck boost converter is 4.08 mH. However, an inductor with a value of approximately 4.08 mH is rarely available in the market, therefore a 10 mH inductor was selected instead, as it is more commonly available. Furthermore, it is recommended to select the inductor to be minimum 25% larger than L_{min} to ensure continuous current operation [8].

The capacitor C value can be calculated by using

$$C_{min} = \frac{D}{R_L r f} \quad (3)$$

where r is the ripple factor of the output voltage. If the value of r is set to 0.1%, then at $D = 0.9$, the value of capacitor C is

$$C_{min} = \frac{0.9}{500(0.1)(62500)} = 0.288\ \mu F \tag{4}$$

Therefore, the minimum value of capacitor of the buck-boost converter is $0.288\ \mu F$. The capacitor 1 μF with rating 50V is selected in this project since it is commonly available in market.

The switching devices in the circuit consist of a P-channel MOSFET (IRF9Z24N) and an N-channel MOSFET (IRFZ44N), denoted as Q1 and Q3, respectively. The IRF9Z24N MOSFET (Q1) is coupled with another IRFZ44N MOSFET (Q2) to facilitate its control. This configuration allows a voltage to be established across the gate and source terminals of the IRF9Z24N transistor, thereby activating it.

There are two ways to control the circuit:

a) Buck or boost mode:
   - Buck mode: Q3 always OFF, Q2 receives PWM signal
   - Boost mode: Q2 always ON, Q3 receives PWM signal

b) Buck-boost mode:
   - Q2 and Q3 operate complimentary. When Q2 turned on, Q3 turned off, and vice versa.

In this project, the buck-boost mode will be used to control the circuit since it is simple to be implemented in microcontroller [9].

The hardware prototype of this project is shown in Figure 3. The flowchart of the operation for the device is illustrated in Figure 4. The process begins with system initialization and setting the PWM frequency to 62.5 kHz. The system then reads the output voltage from the voltage sensor, the desired voltage from a potentiometer, and the PID gains (K_P, K_I, and K_D) from their respective potentiometers. These values are subsequently displayed on the seven-segment display by the microcontroller. By using the same inputs, the PID controller computes the control signal $u(k)$, which is also equivalent to duty cycle value of PWM signal, but limited to 90% duty cycle. This is due to the circuit unable to boost the voltage after duty cycle 90% due to inductor saturation and losses in MOSFET and diode [10]. The resulting duty cycle of PWM signals is applied to transistors Q2, but for the value of duty cycle that applied to transistor Q3 is $90\% - u(k)$ which reflect the inverse PWM signal that applied to Q2. The system continues to operate in this loop until the device is switched off.

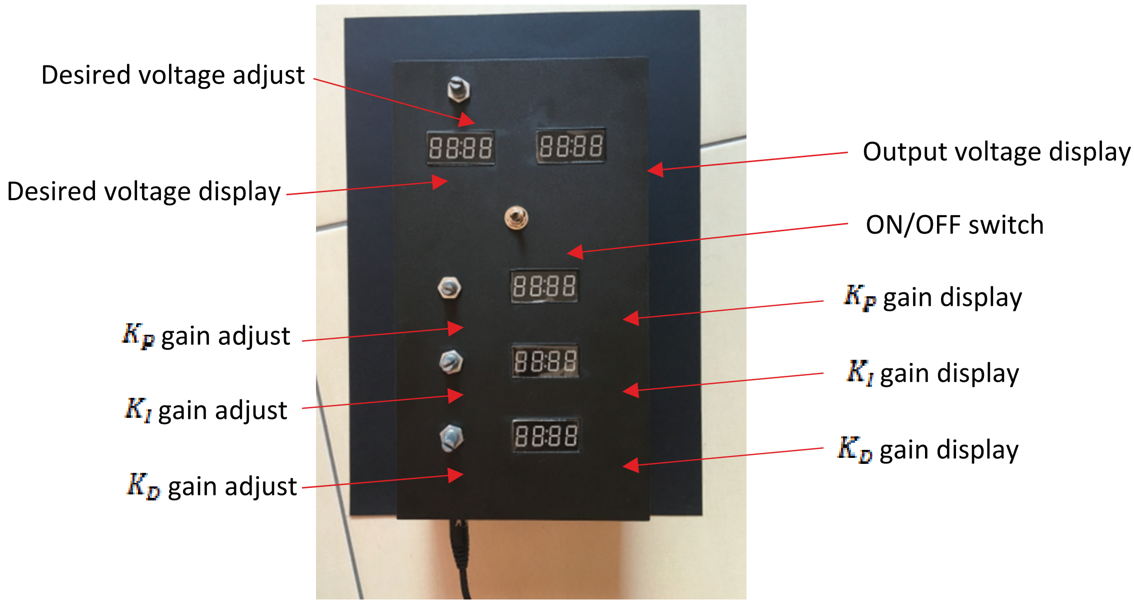


Figure 3. Hardware prototype of this project

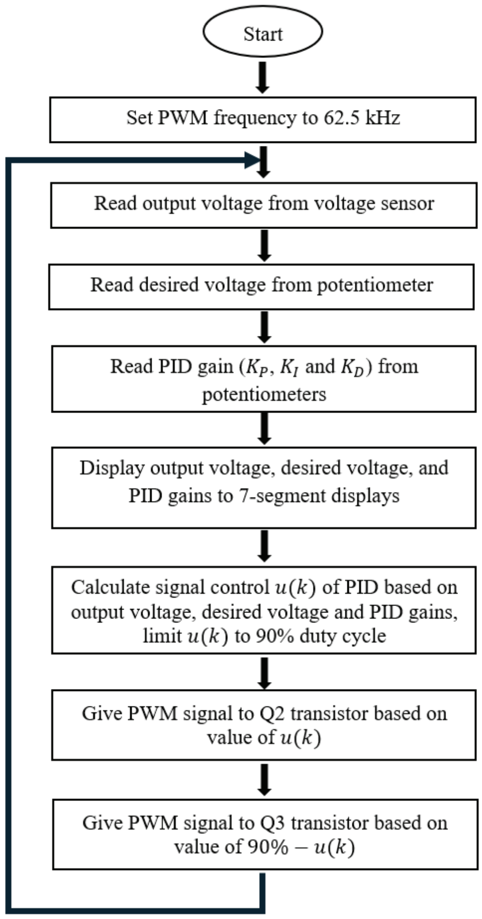


Figure 4. Flowchart of the operation for the developed device

PID Controller

PID is a proportional-integral-derivative controller, which controls loop feedback mechanism. It is commonly used in industrial control systems. The first PID controller was developed by Elmer Sperry in 1911 and it is used to automate a ship steering mechanism. On 1922, the first theoretical paper on PID control was develop by Nicolas Minorsky, which then was applied to automatic steering of ships [11]. Figure 5 is the block diagram of a system controlled by PID controller where $y(t)$ is controlled output, $r(t)$ is reference signal, $e(t)$ is error signal and $u(t)$ is control signal.

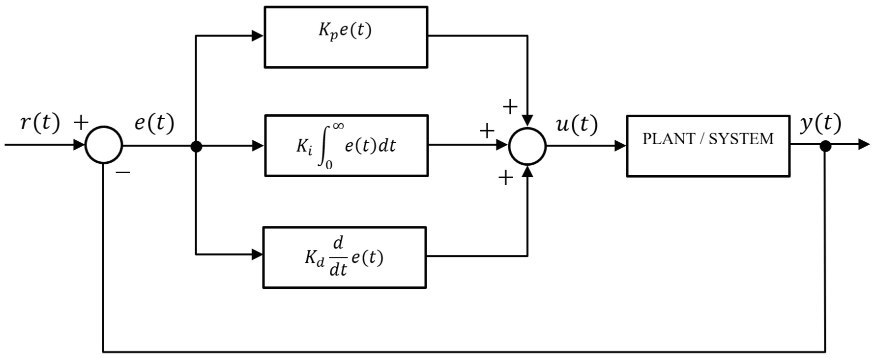


Figure 5. Block diagram of a system with PID controller

PID control signal is the sum of three terms which are based on the error measurement, that are

$$u(t) = K_P e(t) + K_I \int_0^{\infty} e(t)dt + K_D \frac{d}{dt} e(t) \quad (5)$$

where $e(t) = r(t) - y(t)$, K_P is proportional constant, K_I is integral constant and K_D is derivative constant. The value of K_P, K_I and K_D are the key in providing stable and desired transient response which can be obtained by using heuristic methods, analytical methods, frequency response method, optimization method and adoptive turning methods [12].

Nowadays, PID control algorithm is always implemented in microcontroller to enable compact design of a product. However, the integral and derivative operation in PID equation of (1) cannot be performed directly by microcontroller. Hence, the integral and derivative term of (1) need to be converted to discrete form. Starting with the derivative term, we can use the following difference equation as an approximation, such that;

$$u(k) = K_P e(k) + K_I \left(T_s \sum_{0}^{\infty} e(k) \right) + K_D \left(\frac{e(k) - e(k-1)}{T_s} \right) \tag{6}$$

The details derivation and more precise approximation of PID controller in discrete form can be found in [12, 13].

The kernel code of implementing discrete PID of (6) in Arduino IDE environment software can be found in [14]. Arduino IDE is a software designed for Arduino which allows users to write code in C language, compile, send/receive to/from microcontroller. K_P, K_I and K_D of PID are the parameters that need to be tuned. In this project, the heuristic approach is used to determine the appropriate values of PID gains. The procedure of determining the PID gains for this project is explained as follows:

(i) Set all gains to zero.

(ii) Increase the K_P gain until the output voltage steadily oscillates.

(iii) Increase the K_D gain until the oscillations significantly reduced.

(iv) Repeat steps (ii) and (iii) until the output voltage is stable with minimal oscillation.

(v)Increase the K_D gain to bring the output voltage to the desired voltage.

Results and Discussion

As explained in previous section, PID controller has gain that need to be tuned which are K_P, K_I and K_D. By using the procedure of PID tuning approach that has been explained, the gains of PID controller are determine when the output of the circuit is connected to 510 Ω resistor. The controller has been tuned when the desired voltage is set to 3 V, 9 V and 24 V and it is found the suitable PID can cater all the desired voltage are $K_P = 7$, $K_I = 3$ and $K_D = 31$.

The device is subsequently tested using 1 kΩ and 2 kΩ resistive loads at the same desired output voltage levels. For performance analysis, key parameters including overshoot magnitude, settling time, and maximum voltage fluctuation are recorded, with their definitions illustrated in Figure 6. The performance evaluation is conducted across nine distinct test cases.

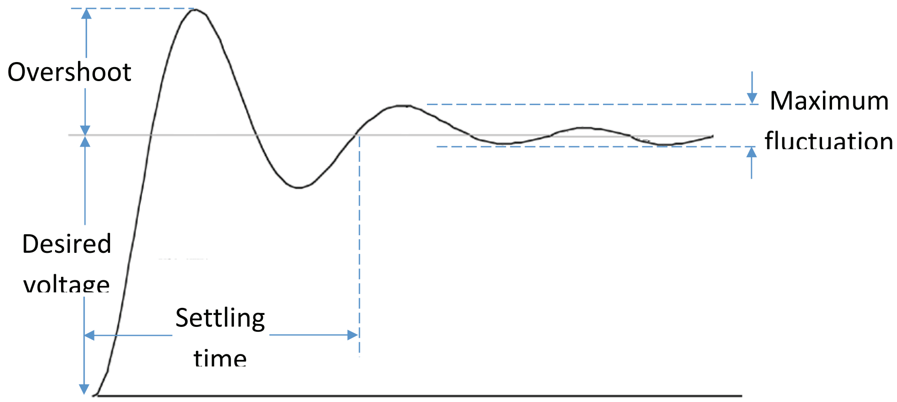
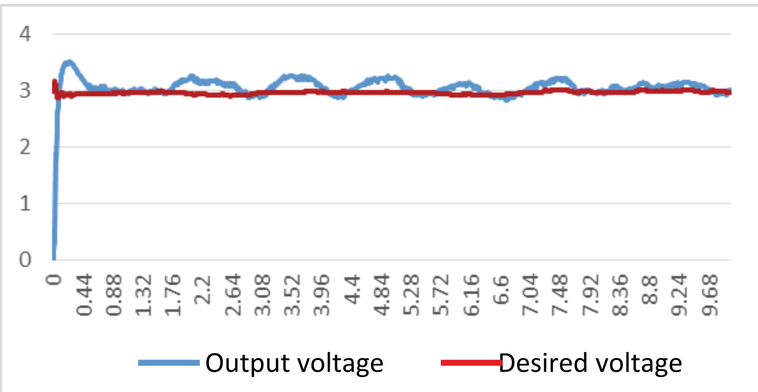
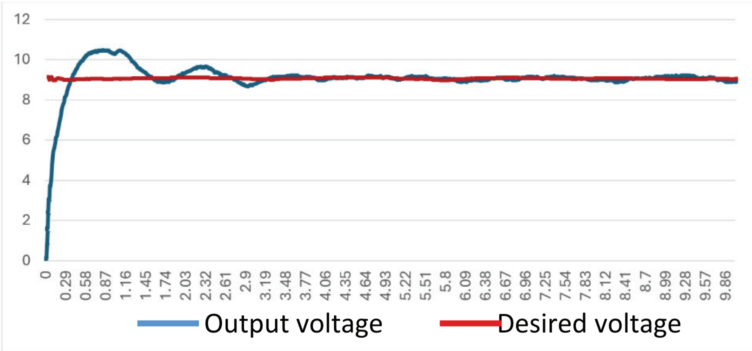


Figure 6. Definition of overshoot, settling time and maximum fluctuation in this project

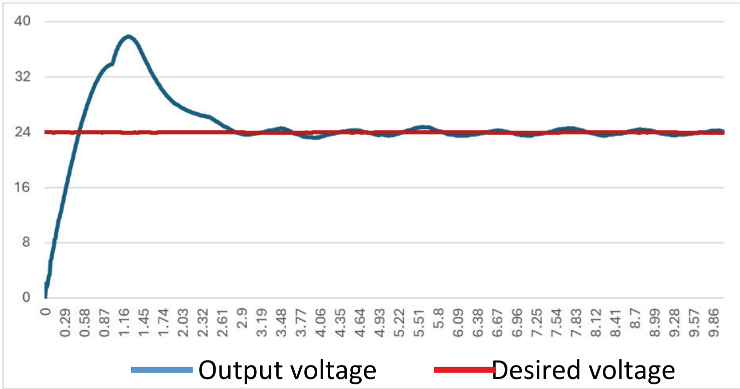
Figures 7(a) to 7(c) illustrate the output voltage response of the device when the load resistance is set to 510 Ω. The results show that the PID controller is capable of regulating the output voltage at the desired level, though with some degree of oscillation. The oscillation is more pronounced at the 24 V setpoint, with an amplitude of approximately 2 V, while at 9 V and 3 V, the maximum fluctuations are smaller, around 1 V and 0.3 V, respectively. In terms of overshoot, a larger overshoot of about 14 V is observed at the 24 V setpoint, whereas smaller overshoots of approximately 1.5 V and 0.5 V occur at 9 V and 3 V, respectively. The corresponding settling times for the desired voltages of 3 V, 9 V, and 24 V are approximately 0.5 s, 1.7 s, and 3.0 s, respectively.



(a)



(b)

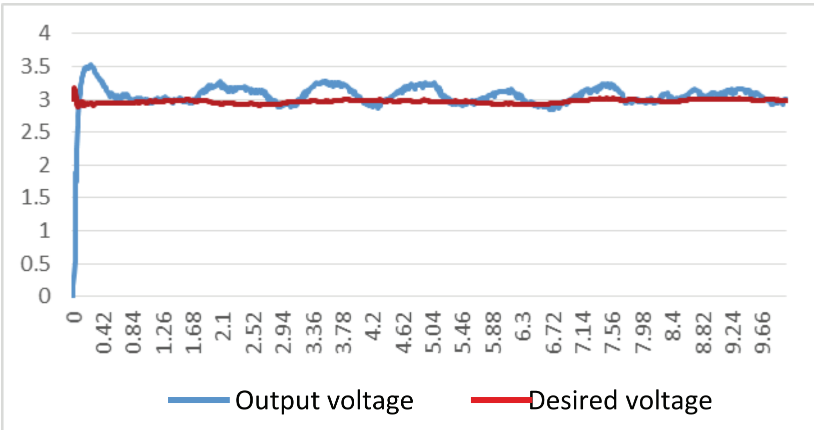


(c)

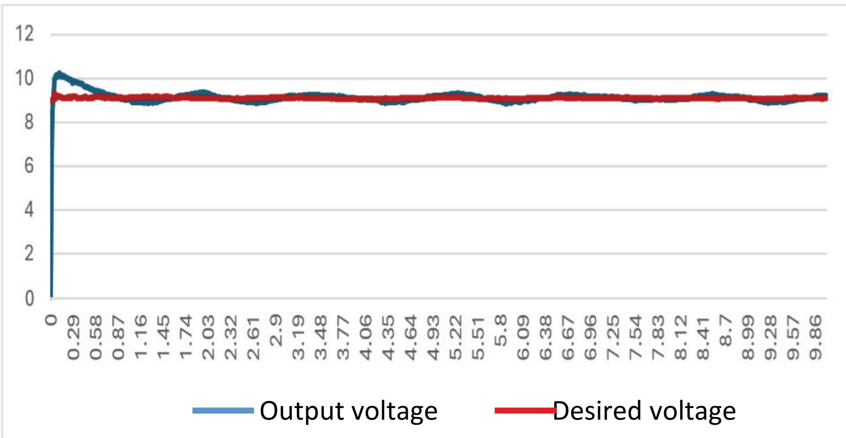
Figure 7. Response of output voltage when load resistance is 510 Ω and desired voltage is set at (a) 3 V (b) 9 V and (c) 24 V

Figures 8(a) to 8(c) illustrate the output voltage response of the device when the load resistance is set to 1 kΩ. The experimental results indicate that the oscillation is more pronounced at the 24 V setpoint, with an amplitude of approximately 3 V. In contrast, at the 9 V and 3 V setpoints, the maximum fluctuations are smaller, around 1 V and 0.3 V, respectively. In terms of overshoot, a larger overshoot of about 6 V is observed at 24 V, while smaller overshoots of approximately 2 V and 0.5 V occur at 9 V and 3 V, respectively. The corresponding settling times for the desired voltages of 3 V, 9 V, and 24 V are approximately 0.5 s, 0.8 s, and 2.0 s, respectively.

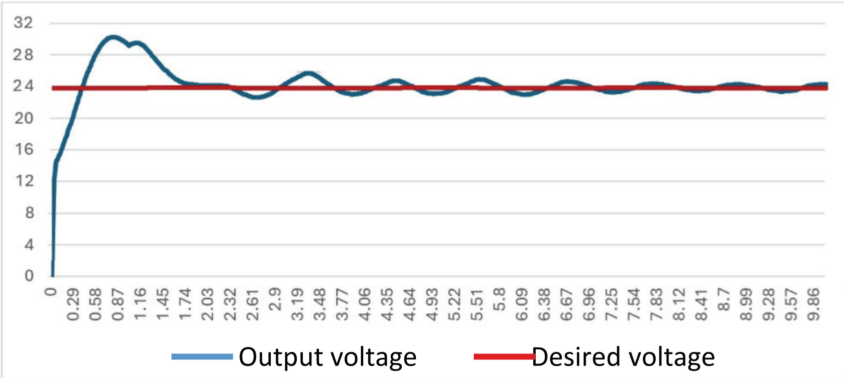
Figures 9(a) to 9(c) depict the output voltage response of the device when the load resistance is 2 kΩ. The experimental results show that the oscillation is most pronounced at the 24 V setpoint, with an amplitude of approximately 2 V. At the 9 V and 3 V setpoints, the maximum fluctuations are smaller, around 1 V and 0.4 V, respectively. Regarding overshoot, a larger overshoot of about 7 V occurs at 24 V, while smaller overshoots of approximately 1 V and 1.2 V are observed at 9 V and 3 V, respectively. The corresponding settling times for the desired voltages of 3 V, 9 V, and 24 V are 0.8 s, 0.8 s, and 1.7 s, respectively.



(a)

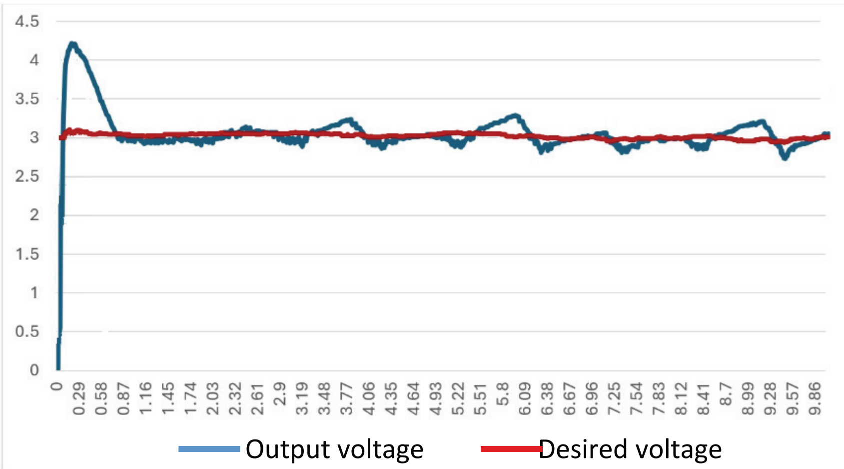


(b)

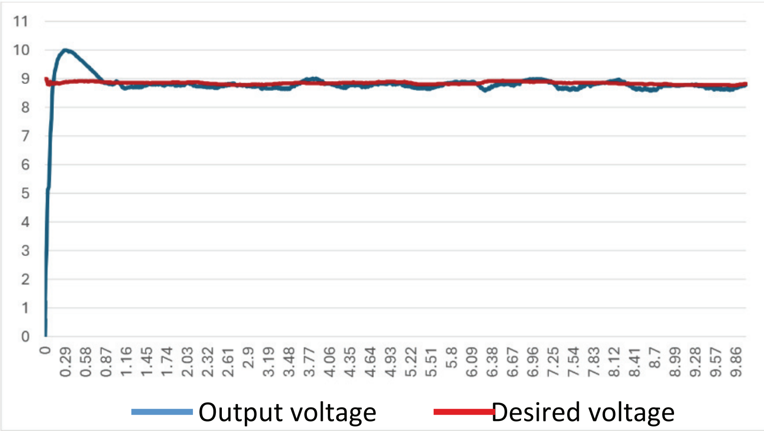


(c)

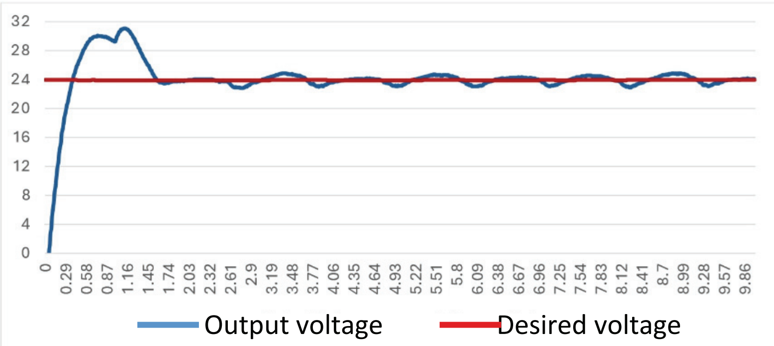
Figure 8. Response of output voltage when load resistance is 1 kΩ and desired voltage is set at (a) 3 V (b) 9 V and (c) 24 V



(a)



(b)



(c)

Figure 9. Response of output voltage when load resistance is 2 kΩ and desired voltage is set at (a) 3 V (b) 9 V and (c) 24 V

The summary of device performance in term of maximum fluctuations, overshoot and settling time is tabulated in Table 1. From the results, it is found that the PID gains $K_P = 7$, $K_I = 3$ and $K_D = 31$ are able to regulate the output voltage at the desired voltage set by the user ranging from 3V until 24V. However, note that the higher desired voltage will increase the fluctuation, overshoot and settling time of the output voltage.

Table 1. Performance summary for the output voltage of the developed device

Desired voltage (V)	Load Resistance (Ω)	Maximum fluctuation (V)	Overshoot (V)	Settling time (s)
3	510	0.3	0.5	0.5
9	510	1.0	2.0	1.7
24	510	3.0	6.0	3.0

3	1000	0.3	0.5	0.5
9	1000	1.0	2.0	0.8
24	1000	3.0	6.0	20.
3	2000	0.4	1.2	0.8
9	2000	1.0	1.0	0.8
24	2000	2.0	7.0	1.7

Conclusion

The development of the non-inverting buck-boost converter with a PID controller successfully achieved stable and adjustable DC output voltage across a wide range of load and input variations. The implementation of the PID algorithm in an Arduino microcontroller demonstrated effective voltage regulation with acceptable overshoot and settling times. Experimental results verified that the controller parameters $K_P = 7$, $K_I = 3$, and $K_D = 31$ provides satisfactory performance for desired voltages from 3 V to 24 V under varying load conditions. Although higher output voltages resulted in slightly greater overshoot and fluctuations, the converter maintained reliable and continuous operation. Overall, the project proved that a PID-controlled non-inverting buck-boost converter can offer an efficient, low-cost, and flexible solution for applications requiring both step-up and step-down DC voltage conversion.

Acknowledgement

The authors would like to thank UTeM for sponsoring this publication under Journal Publication Incentive 2025.

References

[1] A. Karaarslan and A. Shaibu, “Performance Analysis and Comparison of Non-Inverting Buck-Boost Converter Using PI, OCC, and Hybrid OCC-PI Control,” *Journal of Electrical Technology UMY (JET)*, vol. 6, no. 2, pp. 65–74, 2022.

[2] M. S. Reddy, P. K. Sahu, and K. P. Singh, “Direct Model Predictive Control of Non-Inverting Buck-Boost DC-DC Converter,” *International Journal of Emerging Science and Engineering (JESE)*, vol. 14, no. 9, pp. 42–48, 2023.

[3] O. Asvadi-Kermani, F. Zahraee, M. V. C. Prasad, and M. B. Bui, “Adaptive Regulation of a Non-Inverting Buck-Boost Converter with Levenberg–Marquardt Based Sliding-Mode Predictive Control Scheme,” *Scientific Reports*, vol. 15, no. 1, pp. 1–13, 2025.

[4] O. Abdel-Rahim, H. Hassan, and A. S. Ali, “An Efficient Non-Inverting Buck-Boost Converter with Improved Step-Up/Down Ability,” *Energies*, vol. 15, no. 13, pp. 4550–4565, 2022.

[5] M. Calzolari, “Advanced Control of a Non-Inverting Buck-Boost Converter,” M.Sc. thesis, Politecnico di Torino, Turin, Italy, 2022.

[6] “Lesson 37: Pulse Width Modulation in Arduino,” *MyPractic*, [Online]. Available: https://mypractic.com/lesson-37-pulse-width-modulation-in-arduino/

[7] A. Anbara, “Pulse Width Modulation (PWM) Example Code,” *GitHub Gist*,

2023. [Online]. Available: https://gist.github.com/anbara/e7e371d4fbdff896e8703fdb000fdaeb

[8] D. W. Hart, *Power Electronics*, 1st ed. New York, NY, USA: McGraw-Hill, 2010.

[9] H. Fan, "Design tips for an efficient non-inverting buck-boost converter," *Analog Applications Journal Industrial*, Texas Instruments, AAJ 3Q 2014, SLYT584, Apr. 2014.

[10] J. Monteiro, V. F. Pires, D. Foito, A. Cordeiro, J. F. Silva and S. Pinto, "A Buck-Boost Converter with Extended Duty-Cycle Range in the Buck Voltage Region for Renewable Energy Sources," *Electronics*, vol. 12, no. 3, Art. no. 584, Jan. 2023.

[11] S. Bennett, "Nicholas Minorsky and the Automatic Steering of Ships," *IEEE Control Systems Magazine*, vol. 4, no. 4, pp. 10–15, November 1984.

[12] K. Åström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*. Instrument Society ofAmerica, 1995.

[13] J. Charais and R. Lourens, "Software PID Control of an Inverted Pendulum Using the PIC16F684," *Application Note AN964, Microchip Technology Inc*., 2004.

[14] M. B. N. Shah, N. Zailany, A. F. Z. Abidin, M. F. Halim, K. Azha Annuar, A. H. Azahar, M. Haniff Harun and M. Faizal Yaakub, "PID-based temperature control device for electric kettle," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 1683-1693, Jun. 2019