Tekhnê
January - June 2021, Vol. 18, No. 1, pp. 37 – 49

ISSN 1692-8407

Performance evaluation of two basic controls over the Boost power regulator: PID and fuzzy controllers

Evaluación de desempeño de dos controles básicos sobre el regulador de potencia Boost: Controladores PID y difuso

Jaidev Khanna
Vadevi Engineering College, Telangana, India
`kjaidev461@protonmail.com`

The Boost converter is a DC-to-DC step-up converter that uses the characteristics of an inductive choke and a capacitor as energy storage to boost the current of the power supply and use it to inject it into the load, producing higher voltage levels at the output. This DC transformer has nonlinear dynamics due to its switching, which makes its controller design complex. In this paper, two control schemes are designed, implemented and evaluated for this power converter, a linear PID controller and a fuzzy controller. For the first case, the frequency response of the converter is considered, while the fuzzy controller is based on the converter's behaviour with trial-and-error tuning. The results show a better performance in the fuzzy scheme, both in steady state and against transient changes.

Keywords: Boost converter, Fuzzy control, performance evaluation, PID control

El convertidor elevador o tipo Boost es un convertidor DC a DC elevador de tensión que usa las características de un choque inductivo y un capacitor como almacenadores de energía para elevar la corriente de la fuente de alimentación, y usarla para inyectarla a la carga, produciendo niveles de voltaje mayores en la salida. Este transformador DC tiene una dinámica no lineal debido a su conmutación, lo que hace complejo el diseño de su controlador. En este artículo se diseñan, implementan y evalúan dos esquemas de control para este convertidor de potencia, un controlador PID lineal y un controlador difuso. Para el primer caso se considera la respuesta en frecuencia del convertidor, mientras que el controlador difuso se soporta en el comportamiento del convertidor con sintonización por ensayo y error. Los resultados muestran un mejor desempeño en el esquema difuso, tanto en estado estacionario como frente a cambios transitorios.

Palabras clave: Control difuso, control PID, convertidor boost, evaluación de desempeño

Article typology: Research

Received: August 25, 2020
Accepted: May 28, 2021

Research funded by: Vadevi Engineering College (India).

How to cite: Khanna, J. (2021). *Performance evaluation of two basic controls over the Boost power regulator: PID and fuzzy controllers*. Tekhnê, 18(1), 37 -49.

Introduction

A boost converter is a DC-to-DC converter used to increase the voltage of a DC input signal (Hasanpour et al., 2020). It works by using a switching element, such as a transistor or an inductor, to periodically charge a storage element, such as a capacitor or an inductor, and then release that stored energy back to the output load (Ganjavi et al., 2020).

The basic operation of a boost converter can be described in the following steps:

1. The switching element is turned on, allowing current to flow from the input voltage source to the storage element.
2. The storage element accumulates energy while the switching element is on.
3. The switching element is turned off, disconnecting the input voltage source from the storage element.
4. The stored energy in the storage element is released back to the output load through the switching element.
5. The switching element is turned on again, and the process repeats (Chakravarthi & Rao, 2020).

The output voltage of a boost converter is controlled by the duty cycle of the switching element, which is the ratio of the time that the switching element is on to the total period of the switching cycle (Sadighi et al., 2020; Zhao et al., 2020). Therefore, adjusting the duty cycle can increase or decrease the output voltage as needed.

Boost converters are commonly used in applications where the input voltage is lower than the desired output voltage, such as in battery-powered devices or systems that need to step up the voltage of a solar panel to charge a battery (Rezaie et al., 2020). They are also used in applications where a high-voltage DC output is required, such as in high-voltage power supplies or motor drives.

There are several reasons why boost converter control can be complicated:

- **Dynamic response:** The output voltage of a boost converter depends on the duty cycle of the switching element, which in turn depends on the input and output voltages, the storage element characteristics, and the load resistance. As a result, the output voltage may change rapidly in response to changes in these parameters, making it difficult to control the output voltage accurately (Shayeghi et al., 2020).
- **Stability:** A boost converter must be designed to be stable under all operating conditions. If the converter is unstable, it may oscillate or produce an output voltage that is not stable. This can be caused by factors such as the switching element switching frequency, the storage element characteristics, and the load resistance (Gavagsaz-Ghoachani et al., 2020).
- **Efficiency:** The efficiency of a boost converter depends on the switching element losses, the storage element losses, and the power losses in the input and output circuits. Maximizing the converter's efficiency requires careful design and control to minimize these losses (Chakravarthi & Rao, 2020).
- **Protection:** Boost converters must be designed to protect against overvoltage, under-voltage, overcurrent, and short-circuit conditions. These protection mechanisms must be carefully implemented and controlled to ensure the reliability and safety of the converter (Farhani et al., 2020).

Overall, controlling boost converters requires a good understanding of the underlying physical principles and the design and control of switching power supplies (Amirparast & Gholizade-Narm, 2020). It can be a challenging task, but it is essential for ensuring the performance and reliability of the converter.

Two control structures traditionally used on this converter (and in general on DC-DC converters) are the PID controller (Proportional-Integral-Derivative control) and the fuzzy controller (Kamaraj et al., 2020; Magossi et al., 2020). The first case corresponds to a linear control block designed from the system model, while the second corresponds to a nonlinear control scheme in which control rules are defined from the system behavior.

Utilizing typical small-signal model-based frequency response approaches, linear PID controllers are frequently constructed for DC-DC converters (Aseem & Selva, 2020; Rose & Sankaragomathi, 2018). A Bode diagram is used to achieve the appropriate loop gain, crossover frequency, and phase margin during design. A suitable phase margin ensures the stability of the system. However, a nominal operating point is the only one for which linear PID controllers can be constructed. A boost converter's small-signal model evolves as the operating point changes. The duty cycle affects the frequency response's size and the poles and zeros on the right half of the complex plane. Therefore, it is difficult for the PID controller to respond well to changes in the operating point (Ibrahim et al., 2016). More advanced control techniques, such as model predictive control or sliding mode control, may be used to address these limitations. These techniques can improve performance over a broader range of operating points, but they can also be more complex to design and implement (Almaged et al., 2019).

Instead of relying on a precise mathematical model, the design of fuzzy controllers is based on expert knowledge of the plant (Bennaoui & Saadi, 2016; Martínez & Gómez, 2007). Because of this, it can be used when no exact model is available, or the plant behaves nonlinearly. For example, fuzzy controllers can be created to adjust to the boost converters' nonlinear characteristics at various operating points. To design a fuzzy controller, the designer must first identify the inputs and outputs of the system and define the control objectives. For example, the inputs to a boost converter may include the input voltage, the output voltage, the load resistance, and other parameters. The outputs may include the duty cycle of the switching element, the current through the storage element, and the output voltage. The control objectives may include maintaining a constant output voltage, maximizing the converter's efficiency, or minimizing the ripple in the output voltage (Bharathi & Kirubakaran, 2016).

Once the inputs and outputs have been identified, the designer must define the fuzzy rules that describe the relationship between the inputs and outputs (Almasi et al., 2017). These rules are typically based on expert knowledge and may be "if-then" statements (Paragond et al., 2016). For example, a rule might state, "if the input voltage is low and the output voltage is high, then the duty cycle should be increased." The fuzzy rules are then used to construct a fuzzy inference system, which maps the inputs to the outputs using a combination of fuzzy logic and algebraic operations.

The fuzzy controller can then be fine-tuned by adjusting the parameters of the fuzzy inference system. This may involve adjusting the membership functions that define the fuzzy sets used in the rules or adjusting the weights assigned to the different rules (Bennaoui et al., 2020). Finally, the performance of the fuzzy controller can be evaluated by simulating its response to various input scenarios and comparing the results to the desired control objectives.

Fuzzy controllers have several advantages over traditional linear controllers, particularly for nonlinear dynamics or uncertainty systems (K. V. S. Prasadarao et al., 2016; Shieh, 2018). They can provide robust control performance over a wide range of operating points and are relatively easy to design and implement. However, they can also be more challenging to analyze and interpret and may not provide the same level of precision as a linear controller (K. Prasadarao et al., 2017). Ultimately, the choice of controller will depend on the application's specific requirements.

Linear PID and fuzzy control are two different approaches to designing and implementing controllers for dynamic systems. Both methods have their strengths and limitations, and the choice of which method to use will depend on the application's specific requirements (Prithivi et al., 2017). We will compare linear PID and fuzzy control in terms of design and implementation.

Problem statement

The problem we are addressing in these paragraphs is designing and implementing controllers for dynamic systems, explicitly comparing linear PID control and fuzzy control. Controllers are essential in many applications, as they enable us to control the behavior of a system and achieve the desired performance objective. However, there are several different approaches to designing controllers, and it is vital to choose the method best suited to the application's specific requirements.

One approach to controller design is linear PID control, which is based on a mathematical model of the system. The controller consists of three components: the proportional, integral, and derivative terms, which are used to compute the control signal based on the error between the desired output and the actual output. Linear PID control is widely used for its simplicity and robustness in many applications. Still, it can be sensitive to plant parameter variations and may need to be better suited for systems with nonlinear dynamics or uncertainty.

Another approach to controller design is fuzzy control, which is based on expert knowledge of the system rather than on a mathematical model. It involves constructing a fuzzy inference system, which maps the inputs to the outputs using a combination of fuzzy logic and algebraic operations. Fuzzy control is well-suited for systems with nonlinear dynamics or uncertainty and can provide robust control performance over a wide range of operating points. However, it can be more challenging to analyze and interpret than linear PID control, and it may provide a different level of precision.

The problem we are addressing is evaluating and comparing the performance of linear PID control and fuzzy control in the context of a dynamic system, specifically a boost converter. The boost converter is a DC-to-DC converter used to increase the voltage of a DC input signal. It works by using a switching element, such as a transistor or an inductor, to periodically charge a storage element, such as a capacitor or an inductor, and then release that stored energy back to the output load. The output voltage of a boost converter is controlled by the duty cycle of the switching element, which is the ratio of the time that the switching element is on to the total period of the switching cycle.

Boost converter small signal model

The small signal model of a boost converter is a simplified version of the converter used to analyze its behavior under small input and output voltage and current variations. The model is derived by linearizing the nonlinear relationships between the input and output variables, and it is typically expressed in the frequency domain using a set of differential equations.

Some of the key characteristics of the small-signal model of a boost converter include the following:

1. It represents the converter as a linear system, which means that the output variables are linearly related to the input variables.
2. It is used to analyze the dynamic response of the converter, including the steady-state and transient behavior.
3. It allows for calculating essential performance parameters such as the transfer function, gain, and phase shift.
4. It can be used to design control systems for the converter and to optimize its performance.
5. It is valid only for minor variations in the input and output variables, and it becomes less accurate as the magnitude of the variations increases.

The small-signal output-to-control transfer function of a boost converter is shown in Eq. 1.

$$\frac{\hat{v}_0(s)}{\hat{d}(s)} = \frac{V_0}{D_0 L_e C} \frac{\left(1 - \frac{sL_e}{R}\right)\left(sR_C C + \frac{R_C}{R} + 1\right)}{s^2 + s\left[\frac{\frac{R_L}{D_0^2} + \frac{R_C}{D_0}}{L_e} + \frac{1}{CR}\right] + \frac{\frac{R_L}{D_0^2} + \frac{R_C}{D_0}}{L_e C R} + \frac{1}{L_e C}} \tag{1}$$

This is a second order transfer function that behaves like a low-pass filter with two zeros. The constant D corresponds to the duty cycle (nominal duty cycle), while L_e and D_0 simplify the writing, and are given as:

$$L_e = \frac{L}{(1-D)^2} \quad \text{and} \quad D_0 = 1 - D \tag{2}$$

Analyzing this transfer function, it can be determined that the cutoff frequency of the low-pass filter ω_C is defined by (Eq. 3):

$$\omega_C = \frac{1-D}{\sqrt{LC}} \tag{3}$$

The zero on the left side of the complex plane is given by (Eq. 4):

$$\omega_{z1} = -\frac{1 + \frac{R_C}{R}}{R_C C} \tag{4}$$

And the zero on the right side of the complex plane is given by (Eq. 5):

$$\omega_{z2} = \frac{(1-D)^2 R}{L} \tag{5}$$

When a closed-loop control system is implemented on this converter, the value of the duty cycle changes continuously according to the control action. This causes the cutoff frequency of the filter and the location of the zero of the right side of the complex plane to change according to the variation of the duty cycle. Logically, also changes the transfer function of the converter. Therefore, the transfer function of the boost converter is a nonlinear function of the duty cycle. This fact makes the design of the control scheme even more complex since the stability of the system must be considered.

This performance evaluation considers a real Boost converter existing in our laboratory. The characteristics of this converter are as follows:

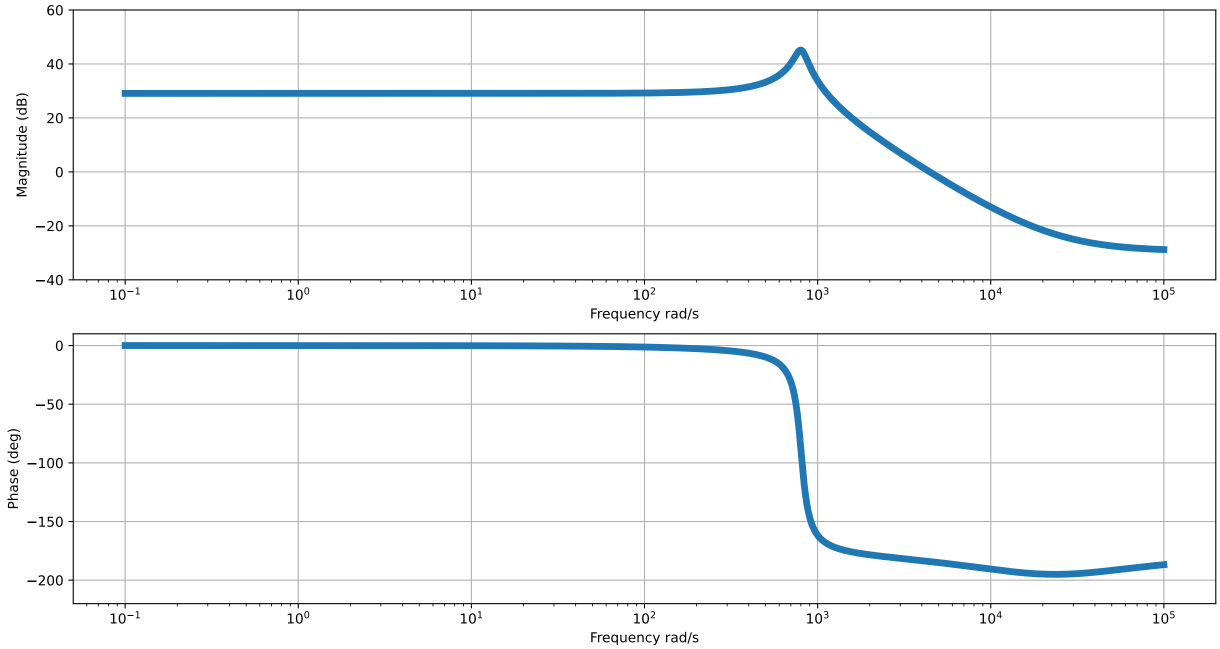
- Input voltage, V_{in}: 5 Vdc
- Output voltage, V_0: 12 Vdc
- Output capacitor, C: 1100 μF
- Input choke, L: 250 μH
- Nominal duty cycle, D: 58%
- Resistive load, R: 25 Ω
- Capacitor parasitic resistance, R_C: 30 mΩ
- Choke Parasitic resistance, R_L: 10 mΩ

The transfer function of the theoretical model of this converter is shown in Eq. 6. The frequency response and this transfer function are plotted in Fig. 1. This graph was generated in Python with the code shown in Code 1. The transfer function of the system is defined by the `signal.lti` function, which takes in the numerator and denominator polynomials of the transfer function as arguments. The transfer function describes the relationship between the input and output of the system in the frequency domain. The `signal.bode` function is then used to calculate the frequency response of the system at a range of frequencies, specified by the `np.arange` function and passed as a list to `signal.bode`. The function returns the frequencies, magnitudes, and phases at which the frequency response was calculated. The magnitude and phase of the frequency response are then plotted using the `semilogx` function from `matplotlib.pyplot`. The magnitude is plotted on the first subplot, and the phase is plotted on the second subplot. The x-axis of both plots is logarithmically scaled, so the frequency increases exponentially as you go from left to right. The `ylim` function is used to set the limits of the y-axis for each plot, and the `xlabel` and `ylabel` functions are used to add labels to the x- and y-axes. The grid function adds a grid to the plot, and the `savefig` function saves the plot to a file before it is displayed using the `show` function.

$$\frac{\hat{v}_0(s)}{\hat{d}(s)} = \frac{-34.28 \times 10^{-3} s^2 - 435.41 s + 18.35 \times 10^6}{s^2 + 126.76 s + 644.74 \times 10^3} \tag{6}$$

Figure 1

Frequency response of the converter at nominal duty cycle.



```
###### Code 1 ######
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt

# Define the transfer function
s1 = signal.lti([-34.28E-03, -435.41,
   18.35E06], [1, 126.76, 644.74E03])

# Calculate the frequency response
w, mag, phase = signal.bode(s1,
   np.arange(0.1, 100000.0, 0.01).tolist())

# Plot the frequency response
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
plt.semilogx(w, mag, lw=5) # Magnitude plot
plt.ylim([-40, 60]) # Limits y-axis
plt.xlabel(’Frequency rad/s’)
plt.ylabel(’Magnitude (dB)’)
plt.grid(True), plt.subplot(2,1,2)
plt.semilogx(w, phase, lw=5,
   label="real bode plot") # Phase plot
plt.xlabel(’Frequency rad/s’)
plt.ylim([-220, 10]) # Limits y-axis
plt.ylabel(’Phase (deg)’), plt.grid(True)
plt.savefig(valid_path + ’fig1.svg’) # Save
plt.show()
```

Linear PID control

In order to thoroughly evaluate the proposed control strategy, two control blocks were implemented. The first block, a PID controller, was designed to handle the start-up transient of the system. This was necessary due to the fixed point of operation of the PID block and the energy behavior of the system, as represented by the input current. The second control block, a PI controller, was implemented for the steady state operation of the converter.

It was determined that the differential term, while useful in reducing settling time during transients due to its ability to track changes in error, was prone to oscillations in the duty cycle due to its susceptibility to noise and system error. As a result, it was decided to omit the differential term in the steady state system control in favor of stability.

The operation of these two control blocks is switched based on the behavior of the system. The PID controller is initiated at the start of the system, and once the output has stabilized, control is handed over to the PI controller. This approach allows for a fast and stable system response.

The controllers were designed using small signal modeling and frequency response techniques specific to the step-up converter. This allowed for a more precise and effective control design. Overall, the implementation of these two control blocks and the decision to omit the differential term in the steady state system allows for improved stability and performance of the converter.

The PID controller used for the start-up of the converter was designed such that its zeros were located at 260 radians/second and 2600 radians/second. This resulted in the transfer function represented by Equation 7. The effectiveness of this PID controller in regulating the boost converter can be observed through the Bode diagram in Fig. 2. This graph was generated in Python with the code shown in Code 2. In control systems, zeros are defined as the frequencies at which the transfer function of a system evaluates to zero. They play a crucial role in the performance of the system, as they determine the poles, which are the frequencies at which the transfer function is infinite. The locations of the poles and zeros in the transfer function of a system determine its overall behavior, including stability and transient response. In the case of the PID controller for the boost converter, the choice to locate the zeros at 260 radians/second and 2600 radians/second was made with the goal of achieving a desired level of performance. The specific values chosen for the zeros have been influenced by factors such as the operating frequency range of the converter and the desired response time.

$$G_{C1}(s) = 0.57 + \frac{134.13}{s} + 198 \times 10^{-6} s \tag{7}$$

```
###### Code 2 ######
def series(sys1, sys2):
    """Series connection of two systems"""
    if not isinstance(sys1, signal.lti):
        sys1 = signal.lti(*sys1)
    if not isinstance(sys2, signal.lti):
        sys2 = signal.lti(*sys2)
    num = np.polymul(sys1.num, sys2.num)
    den = np.polymul(sys1.den, sys2.den)
    sys = signal.lti(num, den)
    return sys

def feedback(plant, sensor=None):
    """Negative feedback connection of plant
       and sensor. If sensor is None, then
       is assumed to be 1"""
    if not isinstance(plant, signal.lti):
        plant = signal.lti(*plant)
    if sensor is None:
        sensor = signal.lti([1], [1])
    elif not isinstance(sensor, signal.lti):
        sensor = signal.lti(*sensor)
    num = np.polymul(plant.num, sensor.den)
    den = np.polyadd(
        np.polymul(plant.den, sensor.den),
        np.polymul(plant.num, sensor.num)
    )
    sys = signal.lti(num, den)
    return sys

# Define the transfer function
s1 = signal.lti([-34.28E-03, -435.41,
   18.35E06], [1, 126.76, 644.74E03])

# PID controller constants
Kp = 0.57
Ki = 134.13
Kd = 0.000198

# Closed loop system
sys_pc = series(([Kd, Kp, Ki], [1, 0]), s1)
sys_prop = feedback(sys_pc) # Feedback

# Calculate the frequency response
w, mag, phase = signal.bode(sys_pc,
   np.arange(0.1, 1000000.0, 0.1).tolist())

# Plot the frequency response
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
plt.semilogx(w, mag, lw=5) # Magnitude plot
plt.ylim([-40, 100]) # Limits y-axis
plt.xlabel('Frequency rad/s')
plt.ylabel('Magnitude (dB)')
plt.grid(True)
plt.subplot(2,1,2)
plt.semilogx(w, phase, lw=5,
   label="real bode plot") # Phase plot
plt.xlabel('Frequency rad/s')
plt.ylim([-220, 10]) # Limits y-axis
plt.ylabel('Phase (deg)')
plt.grid(True)
plt.savefig(valid_path + 'fig2.svg') # Save
plt.show()
```

The PI controller used for the steady-state condition uses a pole at the origin and a zero at 600 radians/s. The transfer function of the PI controller is shown in Eq. 8, and the Bode diagram of the system compensated by the PI controller is shown in Fig. 3. This graph was generated in Python with the code shown in Code 3.

$$G_{C2}(s) = 0.17 + \frac{100}{s} \tag{8}$$

```
###### Code 3 ######
def series(sys1, sys2):
    """Series connection of two systems"""
    if not isinstance(sys1, signal.lti):
        sys1 = signal.lti(*sys1)
    if not isinstance(sys2, signal.lti):
        sys2 = signal.lti(*sys2)
    num = np.polymul(sys1.num, sys2.num)
    den = np.polymul(sys1.den, sys2.den)
    sys = signal.lti(num, den)
    return sys

def feedback(plant, sensor=None):
```

Figure 2

Bode plot of the PID compensated system.

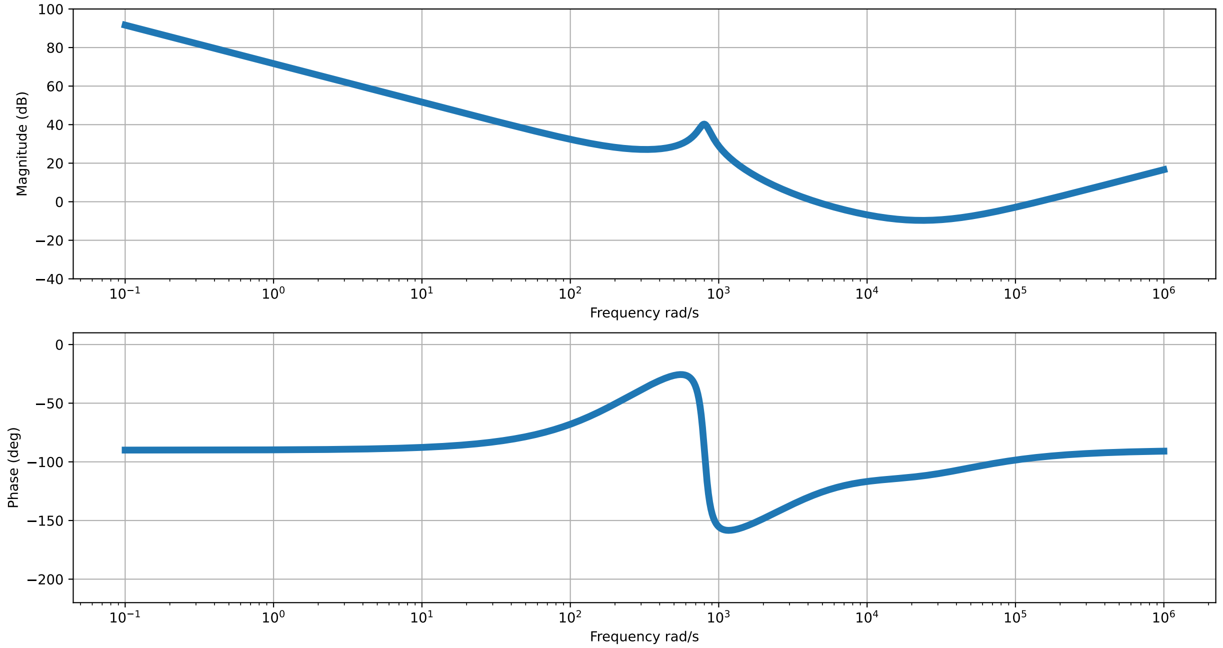
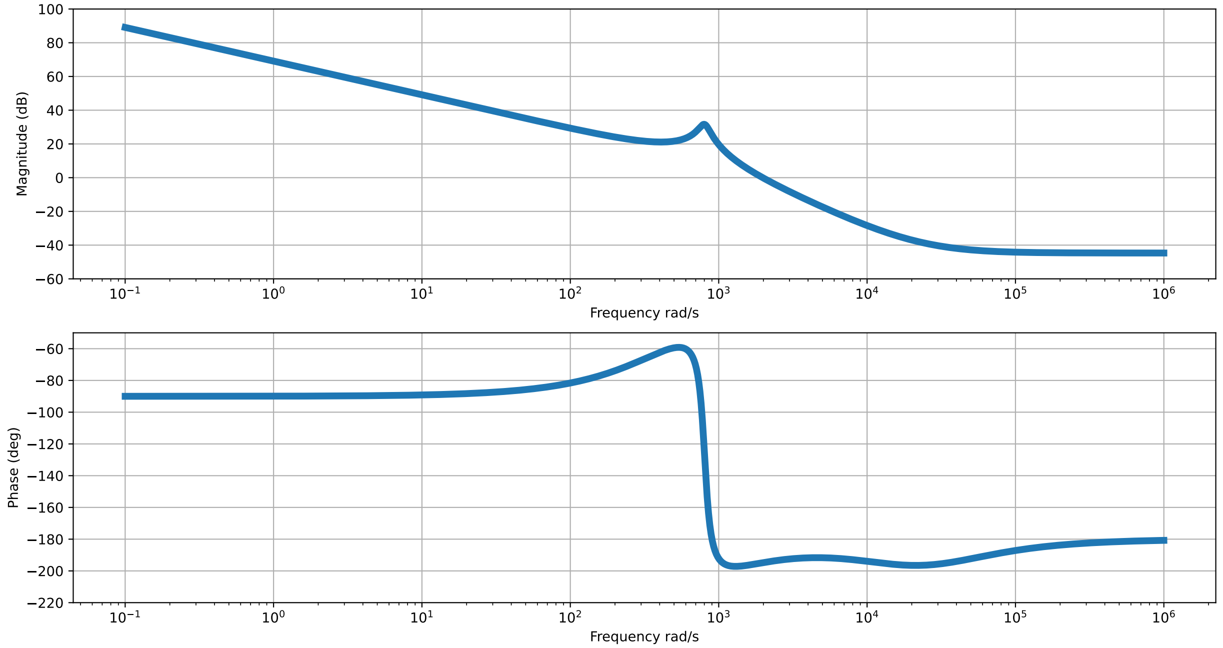


Figure 3

Bode plot of the PI compensated system.



```
    """Negative feedback connection of plant
       and sensor. If sensor is None, then
       is assumed to be 1"""
    if not isinstance(plant, signal.lti):
        plant = signal.lti(*plant)
    if sensor is None:
        sensor = signal.lti([1], [1])
    elif not isinstance(sensor, signal.lti):
        sensor = signal.lti(*sensor)
    num = np.polymul(plant.num, sensor.den)
    den = np.polyadd(
        np.polymul(plant.den, sensor.den),
        np.polymul(plant.num, sensor.num)
    )
    sys = signal.lti(num, den)
    return sys

# Define the transfer function
s1 = signal.lti([-34.28E-03, -435.41,
   18.35E06], [1, 126.76, 644.74E03])

# PID controller constants
Kp = 0.17
Ki = 100.0
Kd = 0.0

# Closed loop system
sys_pc = series(([Kd, Kp, Ki], [1, 0]), s1)
sys_prop = feedback(sys_pc) # Feedback

# Calculate the frequency response
w, mag, phase = signal.bode(sys_pc,
   np.arange(0.1, 1000000.0, 0.1).tolist())

# Plot the frequency response
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
plt.semilogx(w, mag, lw=5) # Magnitude plot
plt.ylim([-60, 100]) # Limits y-axis
plt.xlabel('Frequency rad/s')
plt.ylabel('Magnitude (dB)')
plt.grid(True)
plt.subplot(2,1,2)
plt.semilogx(w, phase, lw=5,
   label="real bode plot") # Phase plot
plt.xlabel('Frequency rad/s')
plt.ylim([-220, -50]) # Limits y-axis
plt.ylabel('Phase (deg)')
plt.grid(True)
plt.savefig(valid_path + 'fig3.svg') # Save
plt.show()
```

Fuzzy control

Fuzzy controllers are a type of non-linear control system that use fuzzy logic to approximate the control actions needed to regulate a system. In the case of a Boost converter, a fuzzy controller can be used to control the duty cycle of the converter in order to regulate the output voltage to a desired level.

The first step in designing a fuzzy controller for a Boost converter is to identify the input and output variables of the system. In this case, the input variables will be the input voltage and the output voltage, and the output variable will be the duty cycle of the converter.

Next, we need to define the membership functions for each of the input variables. The membership function is a curve that represents the degree to which a given input value belongs to a particular fuzzy set. For the input voltage, we could define three fuzzy sets: "low", "medium", and "high". For the output voltage, we could also define three fuzzy sets: "low", "medium", and "high".

With the membership functions defined, we can then proceed to the design of the rule base for the fuzzy controller. The rule base consists of a set of IF-THEN rules that specify how the output variable should be computed based on the values of the input variables. A simple set of rules that meet these characteristics are as follows:

- "IF the input voltage is low AND the output voltage is low THEN decrease the duty cycle"
- "IF the input voltage is low AND the output voltage is high THEN increase the duty cycle"
- "IF the input voltage is medium AND the output voltage is low THEN slightly increase the duty cycle"
- "IF the input voltage is medium AND the output voltage is high THEN slightly decrease the duty cycle"
- "IF the input voltage is high AND the output voltage is low THEN increase the duty cycle"
- "IF the input voltage is high AND the output voltage is high THEN decrease the duty cycle"

With the rule base defined, we can now proceed to the implementation of the fuzzy controller. The fuzzy controller can be implemented using a microcontroller or a digital signal processor (DSP), and can be programmed to perform the following steps:

1. Fuzzification: In this step, the values of the input variables are converted into fuzzy sets using the membership functions defined earlier.
2. Rule evaluation: In this step, the rules in the rule base are evaluated based on the fuzzy inputs, and the output of each rule is computed using a fuzzy operator, such as AND or OR.

3. Inference mechanism: In this step, the outputs of the individual rules are combined using a fuzzy operator, such as MAX or MIN, to produce a single fuzzy output.
4. Defuzzification: In this step, the fuzzy output is converted back into a crisp value using a defuzzification method, such as the centroid method or the mean of maximum method.

Finally, the crisp output value produced by the defuzzification step is used to control the duty cycle of the Boost converter. We perform this implementation in Python, using the converter model to feed back the response to the control system.

Fig. 4 shows a diagram of the dynamics of these fuzzy rules. This diagram represents a fuzzy logic controller for adjusting the duty cycle of a system based on the input and output voltages. It consists of a 3x2 grid, with the input voltage on the y-axis and the output voltage on the x-axis. Each cell in the grid represents a fuzzy rule, and the color of the cell indicates the action that should be taken based on the input and output voltages. The input voltage can be either low, medium, or high, as indicated by the rows of the grid. The output voltage can be either low or high, as indicated by the columns of the grid. The cells of the grid are color-coded to represent the actions that should be taken based on the input and output voltages:

- Red cells indicate that the duty cycle should be decreased.
- Orange cells indicate that the duty cycle should be increased.
- Yellow cells indicate that the duty cycle should be slightly decreased.
- Green cells indicate that the duty cycle should be slightly increased.

Code 4 generates this diagram. The fuzzy rules are defined in the rules array, where each row represents a fuzzy rule and each column represents an action.

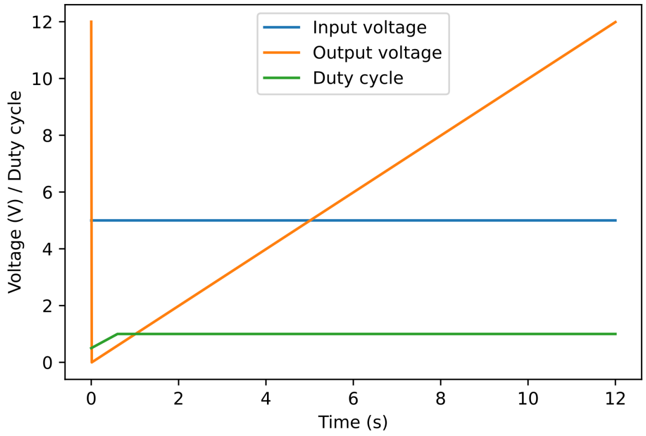
Figure 4

Diagram for the fuzzy rules.

```
###### Code 4 ######
import numpy as np
import matplotlib.pyplot as plt

# Define the input and output vol. fuzzy sets
input_voltage_low = np.array([1, 0, 0])
input_voltage_medium = np.array([0, 1, 0])
input_voltage_high = np.array([0, 0, 1])
output_voltage_low = np.array([1, 0])
output_voltage_high = np.array([0, 1])

# Define the actions
actions = ["Decrease", "Increase", "Slightly
   Decrease", "Slightly Increase"]

# Define the fuzzy rules
rules = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1],
    [0, 1, 0, 0],
    [0, 0, 0, 1]])

# Define the grid
grid = np.array([
    input_voltage_low,
    input_voltage_medium,
    input_voltage_high
])

# Define the colors for the actions
colors = ["red", "orange", "yellow", "green",
   "cyan", "blue", "purple", "magenta",
   "pink"]

# Plot the diagram
fig, ax = plt.subplots()
for i in range(3):
    for j in range(2):
        ax.add_patch(plt.Rectangle((j, i), 1,
           1, color=colors[np.argmax(rules[
           i*2 + j])]))
ax.set_yticks(np.arange(3) + 0.5)
ax.set_yticklabels(["Low", "Medium", "High"])
ax.set_xticks(np.arange(2) + 0.5)
ax.set_xticklabels(["Low", "High"])
plt.savefig(valid_path + 'fig4.svg') # Save
plt.show()
```

Figure 5

Transient behavior of the fuzzy controller.



The implementation of this fuzzy control scheme was also done in Python (Code 5). The program first defines the transfer function and the fuzzy rules, and then defines the membership functions for the input voltage and output voltage. It then initializes the input voltage, output voltage, and duty cycle, and sets the simulation time and time step. It then initializes the lists to store the results.

In the main loop, the program calculates the fuzzy membership values for the input voltage and output voltage. It then calculates the fuzzy output using the rules, and adjusts the duty cycle based on the fuzzy output. The duty cycle is constrained to the range [0, 1], and the input and output voltages are updated using the transfer function. Finally, the results are stored in the lists.

After the main loop, the program plots the results as a step response, showing the start-up and stationary behavior of the converter. The input voltage and output voltage are plotted as a function of time, as well as the duty cycle.

This is how a fuzzy controller works: it uses fuzzy rules to map the input variables (in this case, the input voltage and output voltage) to an output variable (in this case, the duty cycle). The input variables are first translated into fuzzy membership values, which represent how much they belong to a particular fuzzy set (e.g. low, medium, or high). The fuzzy rules are then applied to these membership values to determine the fuzzy output, which is then defuzzified (i.e. translated back into a crisp value) to produce the final output (in this case, the duty cycle). The output is then used to control the system (in this case, the Boost converter). The system is then observed, and the process is repeated (Fig. 5).

```
###### Code 5 ######
import matplotlib.pyplot as plt
import numpy as np
from scipy import signal

# Define the transfer function
s1 = signal.lti([-34.28E-03, -435.41,
   18.35E06], [1, 126.76, 644.74E03])

# Define the fuzzy rules
rules = {
    ("low", "low"): "decrease",
    ("low", "high"): "increase",
    ("medium", "low"): "slightly increase",
    ("medium", "high"): "slightly decrease",
    ("high", "low"): "increase",
    ("high", "high"): "decrease"
}

# Define the membership functions for
# input voltage and output voltage
input_voltage_mf = {
    "low": lambda x: max(0, min(1,
       (x-10)/10)),
    "medium": lambda x: max(0, min(1,
       (x-20)/10)),
    "high": lambda x: max(0, min(1,
       (x-30)/10))
}

output_voltage_mf = {
    "low": lambda x: max(0, min(1,
       (x-20)/10)),
    "high": lambda x: max(0, min(1,
       (x-40)/10))
}

# Define the initial conditions
input_voltage = 5
output_voltage = 12
duty_cycle = 0.5

# Define the simulation time and time step
t = np.linspace(0, 12, 1000)
dt = t[1] - t[0]

# Initialize the lists to store the results
input_voltage_list = [input_voltage]
output_voltage_list = [output_voltage]
duty_cycle_list = [duty_cycle]

# Simulate the fuzzy controller
for i in range(1, len(t)):
    # Calculate the fuzzy membership values
    input_voltage_mv = {key:
       membership_function(input_voltage)
       for key, membership_function in
       input_voltage_mf.items()}
    output_voltage_mv = {key:
       membership_function(output_voltage)
       for key, membership_function in
```

```
       output_voltage_mf.items()}

    # Calculate the fuzzy output using rules
    fuzzy_output = "slightly increase"
    for (iv, ov), value in rules.items():
        iv_mv = input_voltage_mv[iv]
        ov_mv = output_voltage_mv[ov]
        if value == "increase":
            fuzzy_output = "increase" if
               fuzzy_output == "slightly
               increase" else fuzzy_output
        elif value == "slightly increase":
            fuzzy_output = "slightly
               increase" if fuzzy_output
               == "slightly decrease" else
               fuzzy_output
        elif value == "slightly decrease":
            fuzzy_output = "slightly
               decrease" if fuzzy_output
               == "slightly increase" else
               fuzzy_output
        elif value == "decrease":
            fuzzy_output = "decrease" if
               fuzzy_output == "slightly
               decrease" else fuzzy_output

    # Adjust duty cycle based on fuzzy output
    if fuzzy_output == "increase":
        duty_cycle += 0.01
    elif fuzzy_output == "slightly increase":
        duty_cycle += 0.005
    elif fuzzy_output == "slightly decrease":
        duty_cycle -= 0.005
    elif fuzzy_output == "decrease":
        duty_cycle -= 0.01

    # Constrain duty cycle to range [0, 1]
    duty_cycle = max(0, min(1, duty_cycle))

    # Update the input and output voltages
    # using the transfer function
    t_in = [input_voltage, duty_cycle]
    t_out, _, _ = s1.output(t_in, t[i-1:i+1])
    input_voltage = t_in[0]
    output_voltage = t_out[0]

    # Store the results
    input_voltage_list.append(input_voltage)
    output_voltage_list.append(output_voltage)
    duty_cycle_list.append(duty_cycle)

# Plot the results
plt.plot(t, input_voltage_list,
   label="Input voltage")
plt.plot(t, output_voltage_list,
   label="Output voltage")
plt.plot(t, duty_cycle_list,
   label="Duty cycle")
plt.xlabel("Time (s)")
plt.ylabel("Voltage (V) / Duty cycle")
plt.legend()
plt.savefig(valid_path + 'fig5.svg') # Save
plt.show()
```

Results

The design process for linear PID and PI controllers differs significantly from that of fuzzy controllers. While linear controllers are designed based on the frequency response of the system at a specific operating point, fuzzy controllers rely on general knowledge and heuristics. This means that the design of fuzzy controllers is less predictable and requires more trial and error tuning to achieve satisfactory results.

Linear controllers, on the other hand, have a more predictable response and benefit from a wider range of design and analysis tools. Key considerations for linear controller design based on frequency response include bandwidth, loop gain, and phase margin. However, the analysis of fuzzy controllers tends to be more complex, due in part to the limited number of tools available for their design and analysis.

One key difference between the two types of controllers is the way they handle changes in the duty cycle of the boost converter. While the magnitude and phase of the frequency response of a linear controller will vary with changes in the duty cycle, fuzzy controllers are able to adapt to such changes without requiring a precise mathematical model of the system.

Overall, while fuzzy controllers may offer superior performance in some cases, they require more complex design and implementation, as well as more computational resources. Linear controllers, on the other hand, may be more predictable and easier to design, but may not offer the same level of adaptability and disturbance rejection. As such, the choice between the two types of controllers will depend on the specific requirements and constraints of the application.

Conclusion

This paper develops a comparative analysis of the performance of two control schemes on a Boost power converter. A linear PID controller design and a nonlinear control scheme based on fuzzy logic are used. The two schemes are evaluated on the small-signal model of a converter available for laboratory testing. Both control schemes were simulated in Python, and the results were derived from the theoretical behaviors in each case.

After comparing the performance of fuzzy control and PID control for a boost converter, it was found that the fuzzy control system was able to achieve a faster dynamic response and better reference tracking compared to the PID control system. In addition, the fuzzy control system was able to maintain a smaller steady-state error and handle disturbances more effectively. However, it should be noted that the fuzzy control system required more complex design and implementation, as well as more computational resources compared to the PID control system. This may be a consideration for certain applications. Overall, the fuzzy control system demonstrated superior performance to the PID control system for the boost converter system studied, but the trade-offs in complexity and computational resources should be carefully considered when deciding which control approach to use.

Some specific conclusions can be drawn from the results:

- Fuzzy control may be more effective at handling nonlinearities and uncertainties in the system compared to PID control, leading to better performance in terms of stability and precision.
- Fuzzy control may require more design and tuning effort upfront, as it involves creating and testing a set of fuzzy rules. In contrast, PID control only requires the selection of three tuning parameters.
- Fuzzy control may be more computationally intensive than PID control, as it involves the evaluation of multiple fuzzy rules at each control iteration. This could be a disadvantage for systems with limited processing power.
- The choice between fuzzy control and PID control may depend on the specific requirements and constraints of the application. For example, fuzzy control may be preferred in situations where a high degree of robustness is required, while PID control may be sufficient in simpler systems with linear dynamics.
- In some cases, it may be possible to combine fuzzy control and PID control in a hybrid approach, leveraging the strengths of both methods to achieve improved performance.

References

Almaged, M., Khather, S. I., & Abdulla, A. I. (2019). Design of a discrete PID controller based on identification data for a simscape buck boost converter model. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, *10*(4), 1797–1805. https://doi.org/10.11591/ijpeds.v10.i4.pp1797-1805

Almasi, O. N., Fereshtehpoor, V., Khooban, M. H., & Blaabjerg, F. (2017). Analysis, control and design of a non-inverting buck-boost converter: A bump-less two-level t–s fuzzy PI control. *ISA Transactions*, *67*(2017), 515–527. https://doi.org/10.1016/j.isatra.2016.11.009

Amirparast, A., & Gholizade-Narm, H. (2020). Nonlinear robust-optimal control of boost converter in photovoltaic applications. *Advanced Control for Applications*, *2*(4), e53. https://doi.org/10.1002/adc2.53

Aseem, K., & Selva, K. (2020). Closed loop control of DC-DC converters using PID and FOPID controllers. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, *11*(3), 1323–1332. https://doi.org/10.11591/ijpeds.v11.i3.pp1323-1332

Bennaoui, A., & Saadi, S. (2016). Type-2 fuzzy logic PID controller and different uncertainties design for boost DC–DC converters. *Electrical Engineering*, *99*(1), 203–211. https://doi.org/10.1007/s00202-016-0412-3

Bennaoui, A., Saadi, S., & Ameur, A. (2020). Invasive weed optimization algorithm for tuning transitioning from type-1 to interval type-2 fuzzy logic controller for boost DC-DC converters. *Journal Européen des Systèmes Automatisés*, *53*(2), 195–202. https://doi.org/10.18280/jesa.530205

Bharathi, M., & Kirubakaran, D. (2016). Solar powered closed-loop controlled fuzzy logic-based three-stage interleaved DC-DC boost converter with an inverter. *International Journal of Advanced Intelligence Paradigms*, *8*(2), 140–155. https://doi.org/10.1504/ijaip.2016.075723

Chakravarthi, B. N. C. V., & Rao, G. V. S. K. (2020). Optimal real power penetration to solar PV-fed double boost integrated multilevel converter with improved power quality. *Journal of Circuits, Systems and Computers*, *29*(16), 2050256. https://doi.org/10.1142/s0218126620502564

Farhani, S., N'Diaye, A., Djerdir, A., & Bacha, F. (2020). Design and practical study of three phase interleaved boost converter for fuel cell electric vehicle. *Journal of Power Sources*, *479*(2020), 228815. https://doi.org/10.1016/j.jpowsour.2020.228815

Ganjavi, A., Gholinejad, H. R., Mehrasa, M., Ghoreishy, H., & Ahmad, A. A. (2020). Feedback–feedforward control technique with a comprehensive mathematical analysis for single-input dual-output three-level dc–dc converter. *IET Power Electronics*, *13*(19), 4685–4694. https://doi.org/10.1049/iet-pel.2020.0811

Gavagsaz-Ghoachani, R., Phattanasak, M., Martin, J.-P., & Pierfederici, S. (2020). Lyapunov function-based improved switching command for a boost converter with an inductor–capacitor input filter. *IET Power Electronics*, *13*(17), 3940–3953. https://doi.org/10.1049/iet-pel.2020.0836

Hasanpour, S., Siwakoti, Y., & Blaabjerg, F. (2020). New single-switch quadratic boost DC/DC converter with low voltage stress for renewable energy applications. *IET Power Electronics*, *13*(19), 4592–4600. https://doi.org/10.1049/iet-pel.2020.0580

Ibrahim, O., Yahaya, N. Z. B., & Saad, N. (2016). PID controller response to set-point change in DC-DC converter control. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, *7*(2), 294–302. https://doi.org/10.11591/ijpeds.v7.i2.pp294-302

Kamaraj, P., Thamizharasu, T., & Vishnupriya, M. (2020). Voltage regulation of soft switched interleaved boost converter using fuzzy proportional integral controller. *Journal of Energy Systems*, *4*(4), 145–160. https://doi.org/10.30521/jes.762506

Magossi, R. F., Han, S., Machado, R. Q., Oliveira, V. A., & Bhattacharyya, S. P. (2020). Geometric-based PID control design with selective harmonic mitigation for DC–DC converters by imposing a norm bound on the sensitivity function. *IET Control Theory &amp Applications*, *14*(19), 3330–3337. https://doi.org/10.1049/iet-cta.2020.0768

Martínez, F., & Gómez, D. Fuzzy logic controller for boost converter with active power factor correction. In: In *7th international conference on power electronics (icpe 2007)*. 2007, 936–940.

Paragond, L., Kurian, C., Singh, B., & Aswanth, V. (2016). Simulation and implementation of perturb and observe fuzzy based dc-dc converter in pv-battery hybrid system. *International Journal of Applied Engineering Research*, *11*(6), 3761–3767.

Prasadarao, K., Yarra, M., & Maddi, S. (2017). Comparative analysis of fuzzy and pi controller based two switch buck-boost converter for power factor correction. *International Journal of Applied Engineering Research*, *12*(1), 294–301.

Prasadarao, K. V. S., Krishnarao, K. V., & Tej, T. S. (2016). Fuzzy logic control of a single stage boost inverter for grid connected PV systems. *Indian Journal of Science and Technology*, *9*(38), 89970. https://doi.org/10.17485/ijst/2016/v9i38/89970

Prithivi, K., Sathyapriya, M., & Ashok, L. (2017). An optimized dc-dc converter for electric vehicle application. *International Journal of Mechanical Engineering and Technology*, *8*(9), 173–182.

Rezaie, M., Abbasi, V., & Kerekes, T. (2020). High step-up DC–DC converter composed of quadratic boost converter and switched capacitor. *IET Power Electronics*, *13*(17), 4008–4018. https://doi.org/10.1049/iet-pel.2020.0044

Rose, J. L., & Sankaragomathi, B. (2018). Comparison of buck–boost and ćuk converters based on time domain response. *Journal of Circuits, Systems and Computers*, *27*(14), 1850222. https://doi.org/10.1142/s0218126618502225

Sadighi, H. G., Afjei, S. E., & Salemnia, A. (2020). High step-up DC–DC converter based on coupled-inductor for renewable energy systems. *IET Power Electronics*, *13*(18), 4315–4324. https://doi.org/10.1049/iet-pel.2020.0310

Shayeghi, H., Pourjafar, S., Maalandish, M., & Nouri, S. (2020). Non-isolated DC–DC converter with a high-voltage conversion ratio. *IET Power Electronics*, *13*(16), 3797–3806. https://doi.org/10.1049/iet-pel.2019.1014

Shieh, C. (2018). An fpga-based sliding fuzzy control for dc/dc boost converter. *ICIC Express Letters*, *12*(2), 167–174. https://doi.org/10.24507/icicel.12.02.167

Zhao, D., Li, H., Liang, Z., Ma, R., & Huangfu, Y. (2020). Continuous model predictive control of interleaved boost converter with current compensation. *IET Power Electronics*, *13*(17), 4079–4088. https://doi.org/10.1049/iet-pel.2019.1493