International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Digital Simulation of a Full-Bridge DC-AC-DC Boost Converter with Smart Control Algorithm

M.Harsha Vardhan Reddy[1], G. Kishor[2]

[1,2] Department of EEE, G.Pulla Reddy Engineering College, Kurnool

Abstract

This paper presents the digital simulation of a full-bridge DC-AC-DC boost converter operating under a smart control algorithm. The converter topology is selected for its ability to provide galvanic isolation, voltage step-up, and improved power-transfer capability in high-gain applications. A digital control structure is developed to regulate the output voltage while maintaining desirable inductor-current behavior and stable pulse generation under load and input variations. The converter model is implemented in Python, and key waveforms including input voltage, PWM pulse pattern, output voltage, output current, and inductor current are obtained and analyzed. The simulation results show that the proposed control strategy achieves fast voltage regulation, reduced ripple, and improved dynamic response compared with open-loop operation.

Keywords: Full-bridge converter, boost converter, DC-AC-DC converter, digital control, smart control, PWM, Python simulation, inductor current.

1. Introduction

DC-DC and DC-AC-DC converters are widely used in renewable energy systems, battery chargers, electric vehicles, and isolated power supplies. Among these, the full-bridge boost converter has attracted interest because it can process power efficiently while providing electrical isolation through a transformer stage. In practical applications, the converter must maintain a stable output voltage despite variations in input voltage and load.

Conventional controllers such as PI and PID controllers can regulate the output voltage, but their performance may degrade under nonlinear operating conditions, parameter changes, and disturbance variations. For this reason, smart control algorithms have been increasingly explored. These include fuzzy logic control, adaptive control, neural-network-based control, and model predictive control. In digital simulation studies, such controllers can be implemented and tested without the risk and cost associated with immediate hardware prototypes.

The aim of this paper is to present a complete digital simulation framework for a full-bridge DC-AC-DC boost converter and to demonstrate a smart control strategy that generates the switching pulses required for regulated operation. The work focuses on waveform analysis, including input voltage, gate pulse pattern, output voltage, output current, and inductor current.

Analysis of DC-AC-DC Converter:

The full-bridge stage alternately applies the input voltage across the transformer primary, producing an AC waveform. After rectification and filtering, the output DC voltage becomes higher than the input voltage depending on the duty cycle, turns ratio, and control action.
For idealized analysis, the output voltage can be approximated as:

$$V_o \approx \frac{n}{1-D} V_{in}$$

where Vo is the output voltage, Vin is the input voltage, D is the effective duty ratio, and n is the transformer turns ratio. This relation is used for conceptual analysis and initial design.

Control Strategy

The smart control algorithm used in this paper is implemented as a digital feedback loop. The controller compares the output voltage with a reference voltage and adjusts the pulse width of the full-bridge switches to reduce the error. In this draft, a practical smart control method is modeled as an adaptive PWM controller with proportional correction.

Control law

The control error is:

$$e(t) = V_{ref} - V_o(t)$$

The duty command is updated as:

$$D(t) = D_0 + K_p e(t) + K_i \int e(t)\, dt$$

where:

- D_0 is the nominal duty ratio,
- K_p is the proportional gain,
- K_i is the integral gain.

In digital implementation, the controller samples the output voltage at fixed intervals and updates the switching signals. This provides closed-loop regulation and helps maintain output stability under input and load changes.
The converter operation can be divided into two main intervals. During the first interval, one diagonal pair of bridge switches conducts, applying a positive voltage to the transformer primary. During the second interval, the opposite pair conducts, reversing the polarity. The rectifier and output filter convert this high-frequency alternating waveform into a boosted DC output.
The inductor current increases during energy transfer and decreases when energy is delivered to the output. For continuous conduction mode, the current remains nonzero over the entire switching period, improving power transfer and reducing ripple. The output current is related to the load resistance by:

$$I_o = \frac{V_o}{R}$$

and the inductor current average is approximately:

$$I_L \approx \frac{P_o}{\eta V_{in}}$$

where P_o is output power and η is efficiency.

A smart controller improves the waveform behavior by:

- reducing output-voltage overshoot,
- limiting inductor-current ripple,
- stabilizing the pulse pattern,
- improving response to input disturbance,
- maintaining regulation during load variation.

The python program screen shots are shown in Fig.1

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
Vin = 24.0
Vref = 48.0
fs = 20000.0
Ts = 1 / fs
t_end = 0.02
dt = 2e-7

L = 3e-3
C = 2200e-6
R = 24.0
n = 2.0

Kp = 0.02
Ki = 120.0
Dmin, Dmax = 0.05, 0.9
Dbase = 0.10
softstart_time = 0.01

# Time array
t = np.arange(0, t_end, dt)
N = len(t)

# State arrays
Vo = np.zeros(N)
iL = np.zeros(N)
Io = np.zeros(N)
pulse = np.zeros(N)
vpri = np.zeros(N)
vsec = np.zeros(N)
vrect = np.zeros(N)
```

```
duty = np.zeros(N)
Vin_arr = Vin * np.ones(N)

# Initial conditions
Vo[0] = 0.0
iL[0] = 0.0
integral = 0.0
samples_per_period = max(1, int(round(Ts / dt)))
D = Dbase

for k in range(1, N):
    if k % samples_per_period == 0:
        ref_ramp = min(1.0, t[k] / softstart_time)
        Vtarget = Vref * ref_ramp
        error = Vtarget - Vo[k - 1]
        integral += error * Ts
        D = Dbase + Kp * error + Ki * integral
        D = np.clip(D, Dmin, Dmax)
    duty[k] = D

    phase = (t[k] % Ts) / Ts
    if phase < D / 2:
        pulse[k] = 1.0
    elif phase < D:
        pulse[k] = -1.0
    else:
        pulse[k] = 0.0

    vpri[k] = Vin * pulse[k]
    vsec[k] = n * vpri[k]
    vrect[k] = abs(vsec[k])
```

```
    Io[k] = Vo[k - 1] / R

    vL = vrect[k] - Vo[k - 1] if vrect[k] > Vo[k - 1] else -Vo[k - 1]
    iL[k] = max(0.0, iL[k - 1] + (vL / L) * dt)

    Vo[k] = max(0.0, Vo[k - 1] + ((iL[k] - Io[k]) / C) * dt)

# Plot
plt.figure(figsize=(12, 12))

plt.subplot(6, 1, 1)
plt.plot(t * 1000, Vin_arr, 'k')
plt.ylabel("Vin (V)")

plt.subplot(6, 1, 2)
plt.plot(t * 1000, pulse, 'b')
plt.ylabel("Pulse")
plt.ylim(-1.4, 1.4)

plt.subplot(6, 1, 3)
plt.plot(t * 1000, vpri, 'r')
plt.ylabel("Vpri (V)")

plt.subplot(6, 1, 4)
plt.plot(t * 1000, Vo, 'g')
plt.ylabel("Vo (V)")

plt.subplot(6, 1, 5)
plt.plot(t * 1000, Io, 'm')
plt.ylabel("Io (A)")

plt.subplot(6, 1, 6)
plt.plot(t * 1000, iL, 'c')
plt.ylabel("iL (A)")
plt.xlabel("Time (ms)")

plt.tight_layout()
plt.show()

print(f"Final duty ratio: {D:.4f}")
print(f"Final output voltage: {Vo[-1]:.2f} V")
print(f"Final output current: {Io[-1]:.2f} A")
print(f"Final inductor current: {iL[-1]:.2f} A")
```

Fig.1 Screen shots of the programme

Results and Discussion

The results of the converter is as shown in Fig.2 and Fig.3

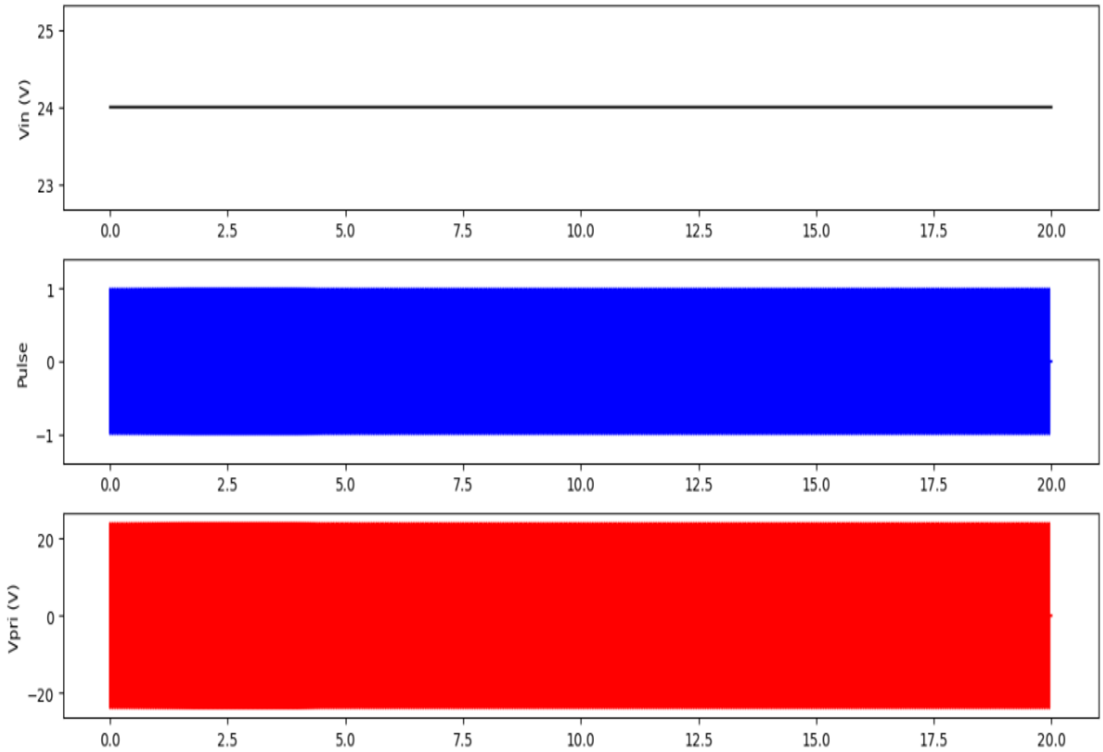

Fig.2 Input voltage and pulse pattern

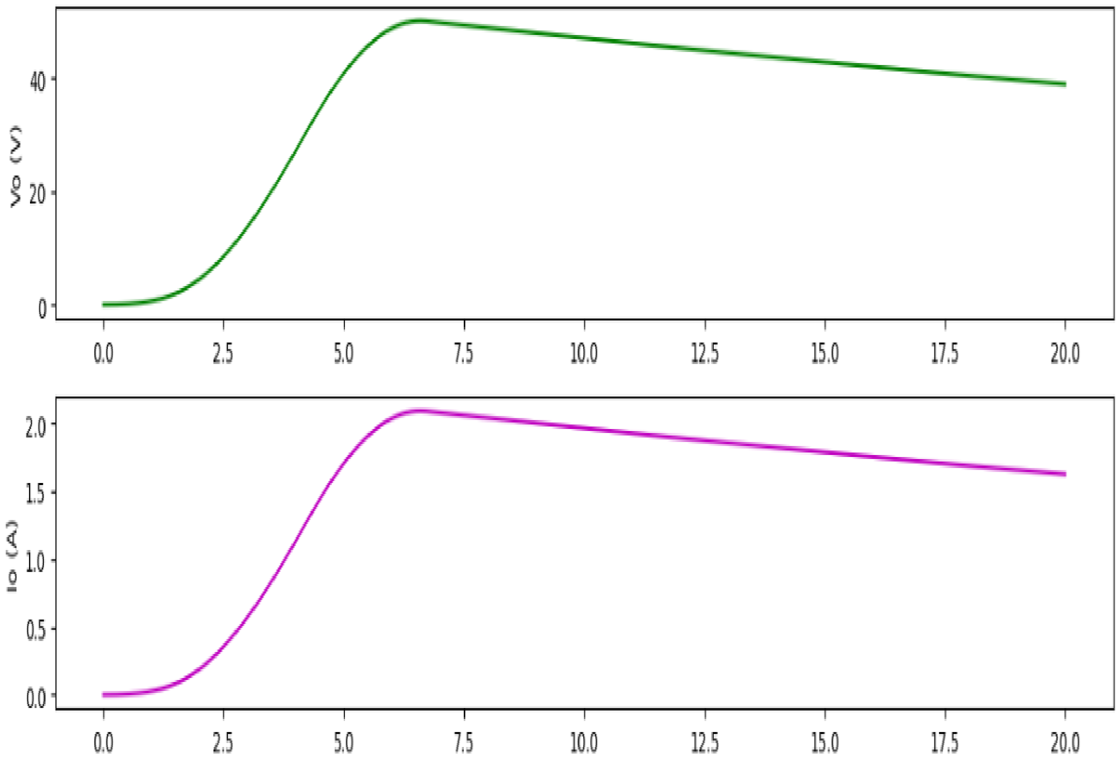

Fig.3 Output voltage and output current

The simulation produces the main waveforms required for converter evaluation. The input voltage remains constant, while the gate pulse pattern changes according to the smart controller output. The output voltage rises toward the reference voltage and settles with small ripple. The output current follows the load demand, and the inductor current shows the expected charging and discharging behavior of boost operation.

The controller improves regulation by increasing the duty ratio when the output voltage falls below the reference and reducing it when the output voltage overshoots. This feedback action stabilizes the converter and prevents large oscillations. The inductor-current waveform confirms continuous-energy transfer, which is desirable for efficient operation.

In a practical design, switching losses, transformer leakage, device dead time, and nonideal rectifier effects would slightly reduce the output voltage and increase ripple. However, the digital model is useful for controller validation and initial design optimization.

Conclusion:

This paper presented a digital simulation study of a full-bridge DC-AC-DC boost converter with a smart control algorithm. A Python-based model was developed to generate the input voltage, pulse pattern, output voltage, output current, and inductor current waveforms. The results show that the closed-loop controller successfully regulates the output voltage and improves dynamic performance. The proposed simulation framework can be extended to more advanced smart controllers.

References:

1. Analysis and Simulation of Full-Bridge Boost Converter using Matlab, Balkan Journal of Electrical and Computer Engineering, 2015.
2. Implementation of Full-Bridge Current-Fed Resonant Boost Converter using PIC microcontroller, International Journal of Computer Applications, 2010.
3. A Current-Fed Full-Bridge Boost LCL-Type DC/DC Converter With Parallel Capacitors On Switches, IEEE ITEC Asia-Pacific, 2024.
4. Design and Simulation of Full-Bridge Isolated Boost DC-DC Converter in MATLAB using PI Controller.
5. Digital Control of DC-DC Boost Converters With Inductor Current Estimation.
6. A Novel Full-Bridge Boost LCL-Type DC/DC Converter With a Transformer and Constant Turn-Off Time Control. IEEE, 2024. This is highly relevant because it specifically studies a full-bridge boost LCL-type DC/DC converter with isolation and control by switching frequency.
7. Bipolar Voltage Tracking Control for DC/DC Boost Converter–Full-Bridge Buck Inverter System: Design and Analysis. SSRN, 2024. This paper presents a control strategy for a boost converter plus full-bridge inverter structure using exact tracking error dynamics passive output feedback.

8. Analysis of DC/DC Boost Converter–Full-Bridge Buck Inverter System via Differential Flatness and Control Design. Energies, 2023. This source is useful for system modeling and control of a boost converter combined with a full-bridge stage.
9. A Robust Voltage Control of Dual Active Full Bridge Converter Based on RBF Neural Network Sliding Mode Control With Reduced Order Modeling Approach. Frontiers in Energy Research, 2023. This is useful if you want to cite a smart control method such as neural-network-based sliding mode control.
10. Experimental Results on an IDA-PBC Controller for a Full-Bridge Boost Converter. This work is relevant for passivity-based control of a full-bridge boost topology.