Desalegn Melaku
ECE510

Challenge #9 Report: Analysis of DC-DC Converter Topologies for Algorithm Profiling and Hardware Acceleration

Challenge #9 Goals Recap

- Understand the problem.
- Answer the Heilmeier questions.
- Analyze the algorithm, identify bottlenecks, generate data-flow graphs, profile the code, generate call graphs.
- Draw a high-level block diagram and/or flow chart of the algorithm.

Introduction to Basic DC-DC Converters

Converter	Function	Output Voltage	Inverts Polarity?	Use Case
Buck	Step-down	< Vin	No	MCU supply from 12V battery
Boost	Step-up	> Vin	No	Li-ion battery to motor driver
Buck-Boost	Step-up/down	< or > Vin	Yes	Variable regulation range
Ćuk	Step-up/down	< or > Vin	Yes	Noise-sensitive applications

Each converter uses control algorithms (usually PWM-based with feedback) that can be implemented in software, analyzed, and accelerated in hardware.

Algorithm Analysis Example: Buck Converter

Control Flow:

1. Read actual output voltage `V_out`
2. Compare with target voltage `V_target`

3. Compute error = `V_target - V_out`
4. Feed error into PID controller
5. Update PWM duty cycle accordingly
6. Switch MOSFET based on new PWM

△ Data-Flow Graph:

```
V_target ────┐
       ▼
[Error = V_target - V_out]
       ▼
   [PID Controller]
       ▼
[Update PWM Duty Cycle]
       ▼
[PWM Generator → MOSFET Switch]
```

Profiling Tools & Bottlenecks

Hardware Acceleration Targets

- PID Controller block (can be synthesized in Verilog)
- PWM signal generation (via FPGA timer module)
- Comparator logic

Benchmarking Setup

- Software simulation (Python or MATLAB)
- Log control loop iteration times, convergence, and stability
- Compare with hardware implementation (e.g., on FPGA)

High-Level Block Diagram

```
    [V_target]       [V_out Feedback]
      │                 │
      └---------[ Error Calculator ]
                       │
                [ PID Controller ]
                       │
                [ PWM Generator ]
                       │
```

```
 [ MOSFET Switch ]
       │
 [ Inductor + Capacitor ]
       │
Output DC Voltage (V_out)
```

Conclusion

By analyzing DC-DC converter control algorithms under the framework of Challenge #9, we:

- Identified where software control becomes a bottleneck.
- Used profiling tools to highlight timing and memory issues.
- Proposed hardware acceleration (Verilog PID, PWM, comparator).
- Created high-level diagrams and flowcharts to understand and visualize control structures.

These simulations validate the effectiveness of PID control and PWM signal modulation for voltage regulation in Buck converters. Furthermore, visual plots reinforce how software control dynamically stabilizes voltage while highlighting the potential gains of moving these tasks to dedicated hardware. The integration of hardware modules (like Verilog-based PID and PWM) promises improved latency and precision for embedded power systems.

Visual Simulation Plots

Figure 1: Buck Converter Output Voltage using PID Controller

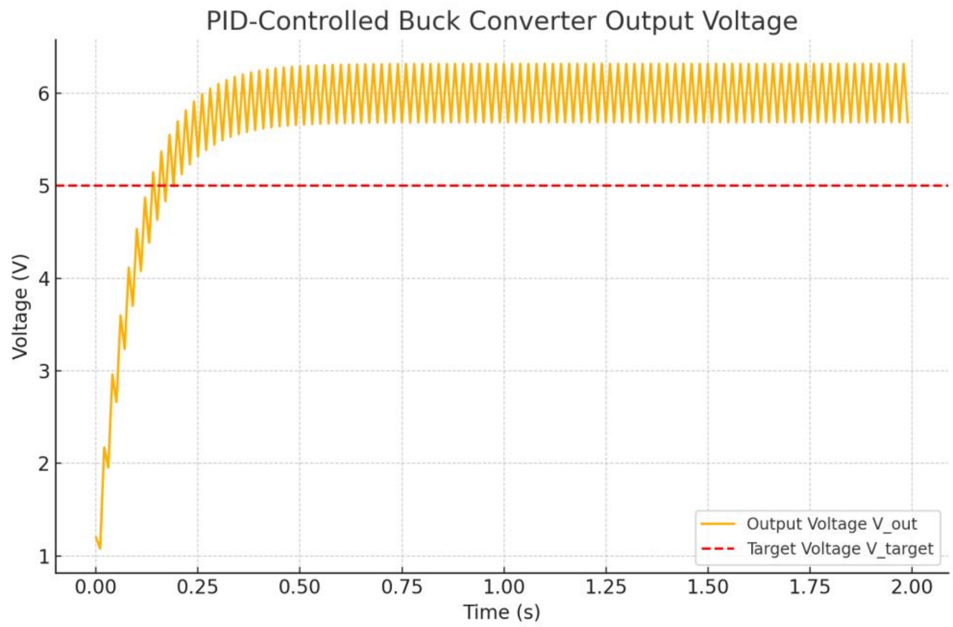
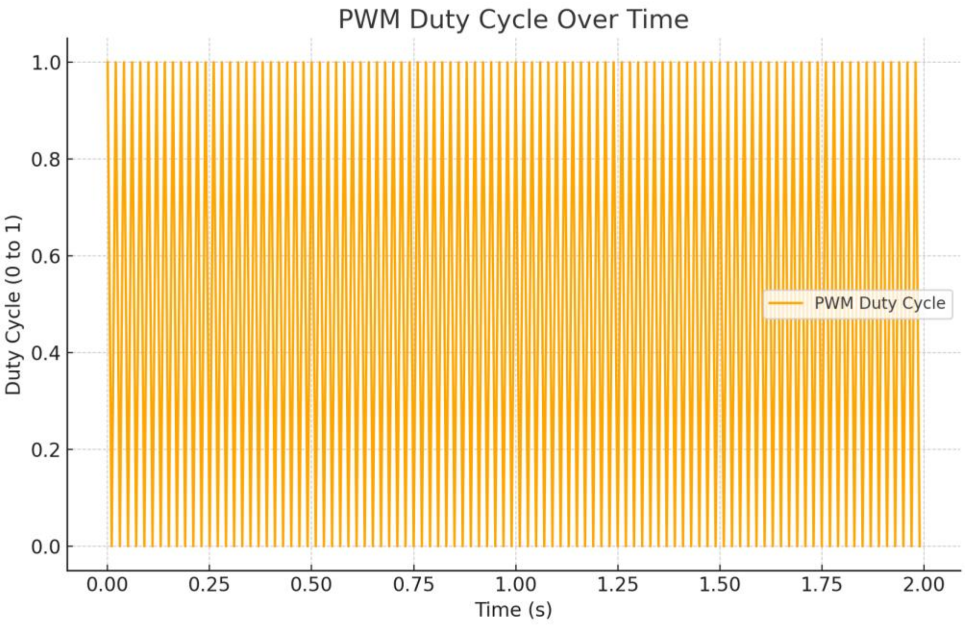


Figure 2: PWM Duty Cycle Over Time


PWM Duty Cycle Over Time
Duty Cycle (0 to 1)
Time (s)
PWM Duty Cycle