Switched Mode Power Converter

Alex Karonen (0566770) and Basanta Khanal (

Lappeenranta–Lahti University of Technology LUT

2025

Table of contents

Contents

1 Introduction

The report demonstrates the execution of a DC-DC power converter control system through PI controller implementation along with PWM output and state machine operation to optimize efficiency. The PI controller maintains stable output voltage by modifying PWM duty cycle controls to achieve accurate power conversion. The state machine operates through different operational modes (IDLE, MODULATING, CONFIG, UART_COMMS, PRINTING) to optimize both system responsiveness and task execution.

The task scheduler directs essential procedures through user input management as well as system parameter updating and monitoring operations. UART communication remains conflict-free through the implementation of a semaphore mechanism. The structured control system enables real-time power converter operation which delivers improved performance and reliability benefits.

2 PI Controller

The Proportional-Integral (PI) Controller serves as a feedback control system for the power converter which minimizes errors in automation applications. The controller contains two operational elements: the proportional (P) term uses current error values for fast response and the integral (I) term tracks accumulated errors to prevent systematic deviations. The controller's performance depends on the precise tuning of Kp (proportional gain) and Ki (integral gain) parameters because raising Kp leads to faster response but produces overshoot and elevating Ki eliminates steady-state error yet creates oscillations. Proper adjustment of the system parameters leads to both stability and measurement precision.

3 State Machine

A state machine functions as a programming construct that moves systems automatically between defined states through user inputs and system conditions. Each state in the model serves a particular operational purpose as the system moves between states through logic-ruled transitions. Embedded systems employ state machines to run operations efficiently by avoiding unnecessary tasks throughout their operations. This project uses a state machine structure for managing system operational modes which enables tasks execution according to current system state.

4 Program Code

The program is written in C for a microcontroller (STM32F411RE Nucleo Board).

4.1 Functions Used in the Code:

- resetModulation(): It resets the PI controller state by clearing the control signal (u1), output voltage (u_out), and state vector (x). This prevents instability when unexpected values occur.
- idle(): It sets the PWM duty cycle to zero, effectively turning off the output, and marks the IDLE task as completed.
- modulate(): This runs the PI controller (runPI), updates the output voltage, and adjusts the PWM duty cycle. If an invalid control signal (e.g., NaN) is detected, it resets modulation to maintain system stability.
- configurate(): This function resets modulation and sets the PWM duty cycle to its maximum, signalling entry into configuration mode.
- PassingSemaphore() & ReleaseSemaphore(): These functions manage critical sections by disabling and re-enabling interrupts, ensuring safe updates to shared resources.

- askForParams(): THese handles user input via UART, allowing adjustments to u_ref in MODULATING mode or Kp/Ki in CONFIG mode. It ensures valid input before applying changes.
- printInfo(): This displays the system's state, control signal, and PI controller gains, providing real-time feedback for monitoring and debugging.
- LoopTasks(): The task scheduler iterates through tasks and executes them as needed, ensuring efficient and responsive operation.
- setup(): This initializes all hardware peripherals, including GPIO, clocks, timers, and interrupts, making the system ready for operation.
- main(): Main function itializes tasks and hardware, then continuously calls LoopTasks() in an infinite loop, ensuring real-time processing and state transitions.

4.2 States in the Code:

In this code, the state machine manages different operational modes: The system operates in three states: IDLE leaves everything static with a zero duty cycle and MODULATING enables active output control and duty cycle adjustment and CONFIG lets users modify controller parameters at the expense of modulation resetting. UART_COMMS operates through UART to manage user input enabling modifications to u_ref, Kp, and Ki values but using semaphore mechanisms to stop button interruptions. Default values for the PI-controller parameters are as follows: Kp = 1.0, Ki = 0.1. The reference voltage is asked always before starting modulation. The last functionality of the system is PRINTING which shows system information to provide real-time monitoring and debugging capabilities. The structured state-based methodology maintains a systematic system with quick responses and modular capabilities.

5 Conclusion

The controller of the DC-DC power converter control system using a PI controller, PWM modulation, and a state machine with C programming code in microcontroller STM32F411RE board has been achieved and it ensures stable and efficient power conversion. The PI controller effectively minimizes errors through precise Kp and Ki tuning, while the state machine optimizes task execution based on system conditions. A structured task scheduler enhances responsiveness, and semaphore-based UART communication prevents conflicts. The modular design enables real-time operation, adaptability, and ease of debugging. Overall, the system achieves reliable performance with improved efficiency and stability, making it suitable for embedded power control applications.