Desalegn Melaku
ECE510
Challenge #10
Summary Report: Q-Learning Bottlenecks and DC-DC Converter Integration

1. Q-Learning Bottleneck in FrozenLake

The primary computational bottleneck in the FrozenLake Q-learning algorithm lies in the repeated update of the Q-table. This involves selecting the maximum Q-value for the next state and computing the new Q-value using the Bellman equation. Profiling and LLM analysis both identify this as the most time-intensive operation.

> ***Q[state, action] += α * (reward + γ * max(Q[next_state][:]) - Q[state, action])***

2. Hardware Acceleration Proposal

A dedicated hardware unit (coprocessor) is proposed to accelerate this computation. The hardware block consists of a finite state machine (FSM), on-chip memory (BRAM) for storing the Q-table, and a logic unit for computing the Bellman update. This enables parallel updates and faster convergence.

3. SystemVerilog Implementation

The FSM-based Q-update unit is written in SystemVerilog and handles Q-table reading, max value extraction, Q-value computation, and writing back the updated value. A testbench simulates the update process over a few clock cycles.

4. Simulation Waveform

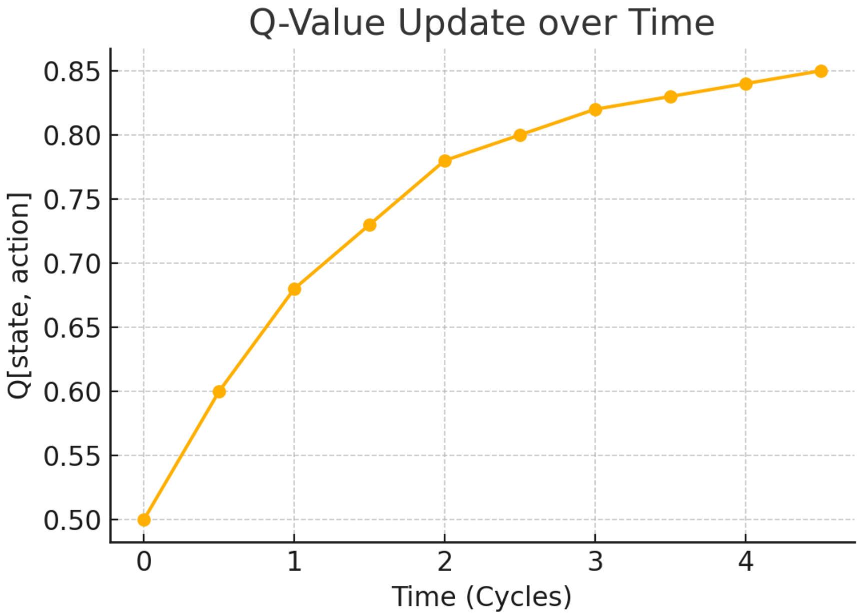

Figure: Q-value updates shown over several clock cycles.

5. Application to DC-DC Converters

The same Q-update FSM can be used to control the duty cycle in DC-DC converters like Buck, Boost, Buck-Boost, and Ćuk. The reinforcement learning controller adjusts the PWM to maintain stable output voltage by learning optimal switching policies.

States are derived from sensed voltage/current; actions are PWM duty cycles; and rewards are based on output voltage regulation.

6. RL-Boost Converter Integration Diagram

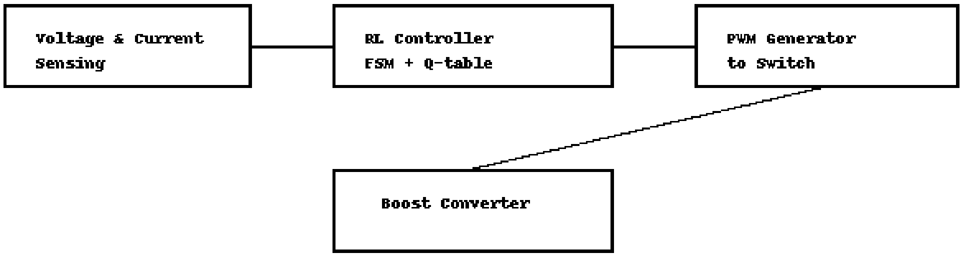

Figure: Hardware RL Controller Integration with Boost Converter.