ECE451/551

Modern Control Design of a DC-DC Voltage Regulator System

Aim: To apply the techniques taught in this class in the design of a controller for a DC-DC voltage converter regulator system. Extensive use of Matlab will be made in the design process. Effectiveness of the design will be verified using the PECS circuit simulator before a hardware prototype is implemented.

The system considered in this project is the D4 dc-to-dc voltage converter. The function of this system is to convert a dc voltage level to a lower dc voltage level with high power efficiency. A state space model is obtained using the method of the State Space Averaging (SSA) which is widely known in the power converter industry and is described in the separate document on the course webpage.

Your aim is to write a report which documents/explains the design procedure. This should be broken up into several sections. The sections are explained below and are related to the Matlab code scripts provided where the design is handled. At each step of the design process plots are provided which provide intermediate results. This should be incorporated into your report. Many of the plots show the output and control effort responses to a step input disturbance. (Please use the step response function provided by the instructor as it more clearly shows the response at the switching instant). Other plots show loop transfer function Bode plot. Students are free to make changes to the Matlab code to show other, different or better results.

Section 1: System Modelling

Matlab function: D4_matrices.m (This is not provided. Students will use their own script.)

Purpose of code: Forms the state matrices of the system and outputs the SSA model.

The function has the following outputs:

```
function [A,B,C,E,Bd,Ed,Vo,D] = D4_matrices
```

Vo and D are the steady state output voltage and steady state duty ratio, respectively.

Section 2: Controllability and observability tests and open loop response

Matlab script: D4_system.m

Purpose of code: Checks the controllability and observability requirements of the system and provides the output voltage response to a unit input voltage step change.

Section 3: Adding integral control

Matlab function: FSFI_matrices.m

Purpose of code: Forms the state space model of the system with integral control.

Section 4: LQR Weighting matrix determination

Matlab script: PSO_LQR_Q_optimization.m, which calls the function, sim_lqr_Q.m

Purpose of script: To obtain a desired Q weighting matrix that is used in the LQR (Linear Quadratic Regulator) design. This matrix is optimized to provide the desired unity gain loop crossover frequency (which should be in the range 10 kHz to 15 kHz (you can choose the value) while also minimizing the output response of the system to a unit input voltage step change.

This code obviates the need to perform trial and error selection to obtain the Q weighting matrix needed in LQR design. In essence, the PSO (Particle Swarm Optimization) runs through a myriad of possibilities to obtain Q which optimizes the metrics specified.

Section 5: Full state feedback

Matlab script: D4_lqri.m

Purpose of script: Undertakes the LQR design (with integral control) using full state feedback. This produces the TLF (Target Loop Function) used in the subsequent estimator designs.

Section 6: Full Order Estimator (FOE) design

Section 6a:

Matlab script: FOE_LTR.m

Purpose of script: Performs Loop Transfer Recovery (LTR) of the full order estimator using various values of *q* (a multiplier used in the process). Of the various responses one is chosen.

Section 6b:

Matlab script: FOE_LTR_for_given_q.m

Purpose of script: Finalize the FOE/LTR design with the chosen *q* value. This leads to a compensator design. This then goes through an order reduction process leading to the final design.

A PECS implementation and simulation results should be provided.

Section 7: Reduced Order Estimator (ROE) design

Section 7a:

Matlab script: ROE_LTR.m

Purpose of script: Performs Loop Transfer Recovery (LTR) of the reduced order estimator using various values of *q* (a multiplier used in the process). Of the various responses one is chosen.

Section 7b:

Matlab script: ROE_LTR_for_given_q.m

Purpose of script: Finalize the ROE/LTR design with the chosen *q* value. This leads to a compensator design. This then goes through an order reduction process leading to the final design. The order of this controller will be lower than the FOE/LTR design and therefore better facilitate practical implementation.

A PECS implementation and simulation results should be provided. The op-amp compensator configuration to be used is shown on the next page.

Section 8: Hardware prototype

A hardware prototype using the ROE/LTR design should be implemented.

To be provided: The final circuit schematic of the full system. Obtained responses should be recorded. Examples of screen captures from an oscilloscope are shown on the next page. Please adopt the format shown there. (Please use the screen capture feature rather than taking a photograph. If you need help with this please refer to the following video: https://www.youtube.com/watch?v=XxCETWyGYxk).

There are two types of disturbance excitation: 1) load steps, and 2) input voltage steps.

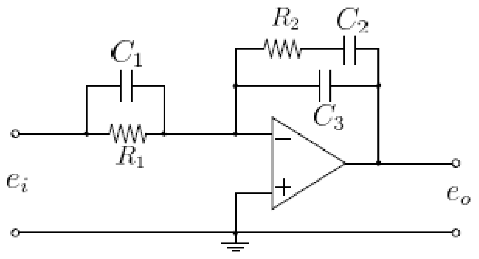
Section 9: Conclusion

Write a suitable conclusion to your report. Provide discussion as to how close your hardware observed step responses match those obtained from Matlab and PECS simulations.

Your report should illustrate your understanding of the underlying theory used in the scripts. Models used should be derived in your report. The block diagrams used to derive the models are: 1) Open loop plant, 2) Plant with integral control and full state feedback, 3) plant with full order estimator and state feedback, and 4) plant with reduced order estimator and state feedback. Students are encouraged to use the provided Latex files for figures used in their reports.

Submit an electronic version of your report along with all your *.m files in a zipped file. Email to tymerski@ee.pdx.edu.

Compensator used for ROE/LTR design:

Integrator with Lead	$-\frac{\omega_0}{s}\frac{(1+\frac{s}{\omega_1})(1+\frac{s}{\omega_2})}{(1+\frac{s}{\omega_3})}$ where: $\omega_0 = \frac{1}{R_1(C_2+C_3)}$ $\omega_1 = \frac{1}{R_2C_2}, \quad \omega_2 = \frac{1}{R_1C_1}$ $\omega_3 = \frac{1}{R_2\frac{C_2C_3}{C_2+C_3}}$	 R_2 C_2 C_1 C_3 R_1 e_i e_o − +

Desired oscillogram formats:

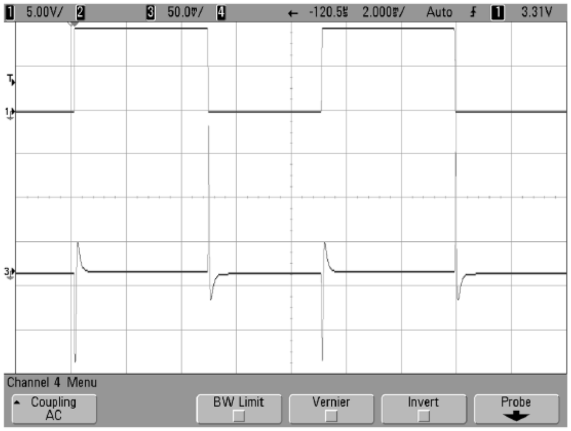


Figure 9. Output voltage response due to step load changes. The load changes from 5 ohms to 5 ohms in parallel with 10 ohms. Top curve: gate drive signal to the IRF530 Mosfet of the load stepping circuit, vertical scale: 5 V/div. Bottom curve: output voltage changes, vertical scale: 50 mV/div.

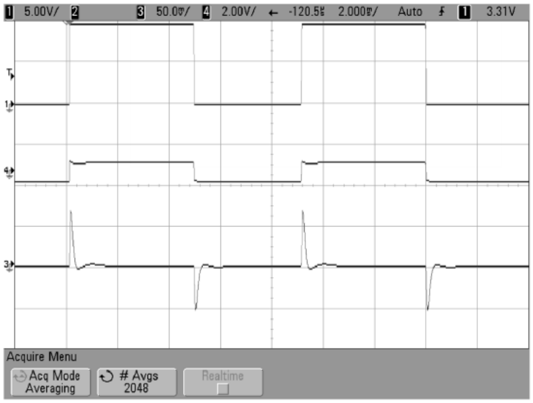

1 5.00V/ 2 3 50.0mV/ 4 2.00V/ ← -120.5s 2.000s/ Auto 1 3.31V
Acquire Menu
Acq Mode Averaging
# Avgs 2048
Realtime

Figure 8. Output voltage response due to step input voltage change. Input voltage V_g is stepped from 10 V to 11 V and then back to 10 V. Top curve: gate drive signal to the 2N7000 Mosfet of the voltage stepping circuit, vertical scale: 5 V/div. Middle curve: input voltage to the C1 converter, showing the step changes between 10 V and 11 V, vertical scale: 2 V/div. Bottom curve: output voltage changes, vertical scale: 50 mV/div.