ECE 730
Control of Adjustable Speed Drives

Lab 3

Torque control of Permanent-Magnet Synchronous Motors

Due date: Mar. 23, 2025 (before 11:59PM)
Please upload report on Avenue to Learn:
Assessments> Assignments> Lab 3

Late submissions won't be marked.

Objective

- Simulating a voltage-source inverter, developing a deadbeat current control algorithm and testing a maximum torque per Ampere (MTPA) strategy for a three-phase Permanent-Magnet Synchronous Machine using MATLAB-Simulink.

Note- You are supposed to be familiar with MATLAB-Simulink and are able to run your simulation either remotely or on your computer.

Procedure

Figure 1 shows the block diagram of the current control loop in the framework of vector control of sinusoidal Permanent-Magnet synchronous machines (PMSM).

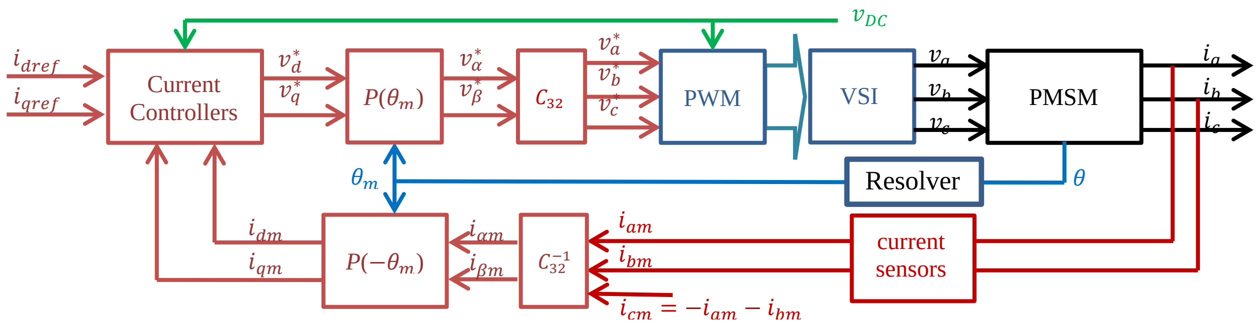

Figure 1- Vector control of PMSM: current control loop.

In this figure, the block PMSM has been already developed in Lab 1. Rotor position and phase current measurements are supposed to be accurate, *i.e.*, $\theta_m = \theta$ and $i_{abc_m} = i_{abc}$. Transformation matrices in the controller (blocks in red) are the same as those already used in the modeling of PMSM in Lab 1. The stator dq-currents control has been realized in Lab 2. In this lab, the aim is to implement a maximum torque per Ampere (MTPA) control in simulation under more realistic conditions. In the meantime, to make comparisons more effective, we assume that the motor speed is kept constant by setting "Dyno" equal 1 in the MATLAB script. This corresponds to setting the dyno speed constant in experimental setup (dyno is an electric machine mechanically coupled to the electric motor under test).

I. Maximum Torque Per Ampere (MTPA)

We would like to implement an indirect torque control using the MTPA strategy. Two 1D MTPA lookup tables (LUTs) are generated in the MATLAB script with "Tref" as input variable and "Idopt" and "Iqopt" as output variables for respectively optimal i_d and i_q (optimal in the sense of MTPA). These optimal currents will be used as reference currents for current controllers. Before running the simulation to test the effectiveness of the MTPA strategy, do the following:

– Configure "Idopt" and "Iqopt" LUTs settings in Simulink>Control (set breakpoints and table data in 1-D LUTs).
– Set "Strategy" switch in Simulink>Control to "ZeroId" position. This disables MTPA for the first test. Later in Step 2 (see below), we will set this manual switch to "MTPA" to turn on the MTPA strategy.

– Set the torque type (torqueREF, top-right) to “Constant” and the current control method (Control method, top-right) to “PI + decoupling” (see Figure 2).
– In this section, we do not consider the voltage-source inverter (VSI). Therefore, set “Switch” on the main Simulink page (bottom-right) to “Vavg” position (see Figure 2). We do not care about the total efficiency under this condition as the VSI is not in the loop.

Then, work through the following steps:

1) Run the simulation and read motor efficiency (“Motor_Efficiency%”), power extracted from the battery (“Pbat”), steady-state average torque (“SS_Average_Torque”), steady-state torque ripple (“SS_Torque_Ripple”), the micro-controller load (“uCtrlLoad”) and the current THD (“THDi%”). Show and comment “Tm” and “Tref”, “Iabc” and “Idqm”.
2) Set the “Strategy” switch in Simulink>Control to “MTPA” position and re-do Step 1. To avoid initial discrepancies, you may set the variable “MTPA” to 1 in the MATLAB script and re-load the parameters by running the script. After running the simulation, compare the metrics under “MTPA” and “ZeroId” strategies (Step 1).
3) Assume an error of $+20\%$ on Ψ_f. Generate new MTPA LUTs by running the MATLAB script after setting Psif0=1.20*Psif. Then, run the simulation while the “Strategy” switch is set to “MTPA” position and compare the metrics with Step 2.
4) Set Ψ_f to its nominal value (Psif0=1.0*Psif) and run the MATLAB script. Then, select “Step change” for the torque type and run the simulation. Read torque tracking error (“Torque_Tracking_error”) in addition to the metrics in Step 1. Show and comment “Tm” and “Tref”, “Iabc”, “Idqm” and “Vdq*”.
5) To evaluate the torque tracking performance of the PI+decoupling controller, set the torque type to “Sinusoidal” and re-do Step 4.

II. Deadbeat Model Predictive Current Controller (DB-MPCC)

We would like to compare the control performances of a PI+decoupling current controller and a deadbeat model predictive current controller. From Topic 3, we have the following equations for the control voltages with a DB-MPCC:

$$v^*_{d_k} = R_s i_{d_k} - \omega_k L_{q_k} i_{q_k} + \frac{\ell_{d_k}}{T_s}\left(i^*_{d_{k+1}} - i_{d_k}\right) + K_{pd}\left(i^*_{d_{k+1}} - i_{d_k}\right)$$
$$v^*_{q_k} = R_s i_{q_k} + \omega_k\left(L_{d_k} i_{d_k} + \Psi_f\right) + \frac{\ell_{q_k}}{T_s}\left(i^*_{q_{k+1}} - i_{q_k}\right) + K_{pq}\left(i^*_{q_{k+1}} - i_{q_k}\right)$$

6) Write the code for the DB-MPCC in Simulink>Control>Ctrl. We assume the DB-MPCC is the control method #2.
7) Set K_{pd} and K_{pq} respectively to 3 and 6 and run the simulation while the torque type is “Step change”. Read motor efficiency, power extracted from the battery, steady-state average torque, steady-state torque ripple, the micro-controller load and the current THD. Show and comment “Tm” and “Tref”, “Iabc”, “Idqm” and “Vdq*”. Do you see any significant change in efficiencies and the battery power compared to PI+decoupling current controller under the same torque type (see Step 4)?
8) Set the torque type to “Sinusoidal” and re-do Step 7. Compare the results to Step 5.
9) Set K_{pd} and K_{pq} respectively to -3 and -6 and re-do Step 8. Compare the tracking error with Step 8.

III. Voltage-Source Inverter (VSI)

In previous labs, the VSI has been modeled using a constant gain G_{VSI}. In this section, a more realistic model for a three-phase VSI is implemented. The theoretical model is available on the course website (see A2L>Lectures>Topic 2). To test the motor drive with the VSI, do the following:

- Set “Switch” to “Vvsi” position (see Figure 2, bottom right).
- Set the manual switch in Simulink>Efficiency to “Pbat” (see Figure 2).
- Select “PI + decoupling” as the current control method and “Step change” as the torque type (see Figure 2).
- Set “Strategy” switch in Simulink>Control to “MTPA” position.
- Reset all the motor parameters to their nominal values in the MATLAB script.

Then, work through the following steps:

10) Run the simulation and read motor efficiency, total efficiency (“Total_Efficiency%”), power extracted from the battery, steady-state average torque, steady-state torque ripple, torque tracking error, the micro-controller load and the current THD. Compare these results with Step 4. Show and comment “Tm” and “Tref”, “Iabc”, “Idqm”, “Vdq*”, “dabc*”, “V123” and “Vabc”. Zoom in on phase a voltage over one fundamental period.
11) Change the switching (and sampling) frequency to 5 kHz first, then 20 kHz and for each case, re-do Step 10. For changing f_{sw}, you can change “Tp” in the MATLAB script.
12) Set $f_s = f_{sw} = 10\ kHz$ and divide the battery voltage by 2 in Simulink>Supply>Vbat. Run the simulation and comment your results.

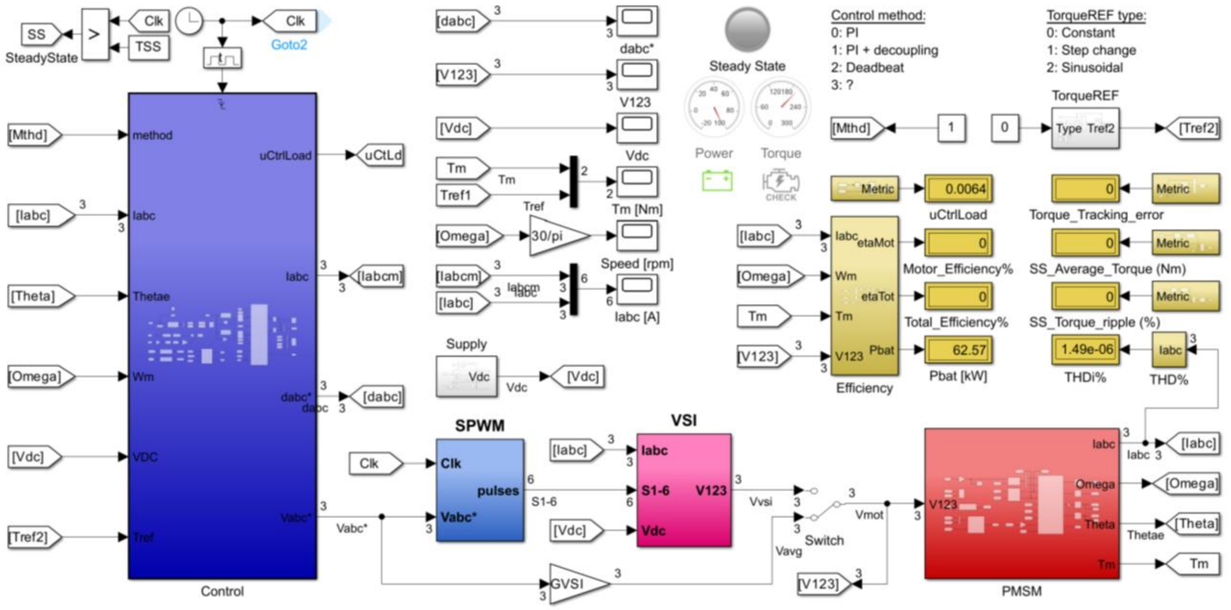

Figure 2- Simulink program.