ECE 788: Special Topics in Power

Lab 2: Vector Control of a PMSM

Connor Ricotta

## Introduction:

The purpose of this lab is to introduce the concept of current control of a PMSM. This is done by implementing a PI controller and seeing how it affects various parameters within the model. This like voltage, current, and motor speed are observed inn order to ensure proper operation. Throughout the experiment the tuning parameters are altered to see how it affects the response of the model, to further understand how to control it to achieve our desired outcomes.

## Section 2: Current Control Results

### Tr = 1ms:

Kpd = 3.1416 Kid = 188.5 Kpq =6.2832 Kiq = 188.5

A screen shot of a graph

Description automatically generated

After tuning the model responds very well, reaching steady state without any overshoot. After analyzing the response time, it seems to match well. The calculated response time is just under our expected of 1ms, sitting at 0.95ms. Looking at the graph above it appears to follow this exactly, reaching the 95% in this time frame.

A graph with a red line

Description automatically generated

A screen shot of a graph

Description automatically generated

The voltage also follows a similar response, spiking to its peak at the 0.1s, then returning to its nominal voltage within the expected response time.

A graph with a blue line

Description automatically generated

Observing the motor rpm, it seems to follow an identical curve of idq, however over a much larger time frame.

### Tr = 5ms

After using the equation wBW = 3/tr, wBW is calculated to be 600 with a response time of 5ms. The gains are as follows:

Kpd = 0.6 Kid = 36 Kpq =1.2 Kiq = 36

A screen shot of a graph

Description automatically generated

After running the model, and examining the data, the Iq response follows an identical curve when compared to the 1ms response. In this case, it still reaches the desired response, hitting 95% of the set-point in the expected 5ms.

A screen shot of a graph

Description automatically generated

Vq follows a similar change, reaching its peak and ten slowly returning within the 5ms. However, the peak is noticeably lower (50V), and it then begins to rise again after the 5ms.

A graph with a red line

Description automatically generated

As for the motor speed, it is identical to that of the 1ms response time.

Part 7:

For this section the initial values for Rs0, ld0, and Lq0 have been altered, and are being calculated with a 1ms response time.

Kpd = 6.2832 Kid = 282.74 Kpq =3.1416 Kiq = 282.74

A graph with a blue line

Description automatically generated

After changing these variables, the models performance has been greatly reduced. In the desired response time, Iq is only able to reach 80% of the desired set-point.

A screenshot of a computer

Description automatically generated

As for voltage, it follows a similar pattern, greatly slowing down the response, while also slowly rising. It also causes an anomaly at the beginning of the model causing it to slowly rise then come back down to the starting point.

A graph with a line going up

Description automatically generated

The motor speed however seems unaffected, with the exception of a similar anomaly as in the voltage graph.

## Discrete:

After Transferring to discrete time, the parameters have be changed to match that of the original 1ms response test.

A screen shot of a graph

Description automatically generated

After running the test, it is clear that the model is less granular than running in continuous, which is expected due to it only making changes when triggered by the 10khz clock. Despite this, the models performance seems unaffected. It is still able to reach the Iq setpoint in the desired response time of 1ms. However, due to being in discrete time, this could cause issues if the model is not running at a fast enough rate, where a reaction is delayed due to it not being triggered by the clock. This can be seen in this example (while having no negative affect) by the delay in the model reacting and not beginning to rise right as the step change is introduced.

A graph with a line drawn on it

Description automatically generated

Motor speed seems unaffected by this change other than the granularity being reduced.

A graph with a line going up

Description automatically generated

Voltage however sees many changes. The voltage initially begins fairly large once again and quickly returns to its expected value. After the peak returns it once again begins to rise reaching its steady state roughly at the same time as the motor speed, which is expected as the higher voltage can improve the power output. However, I would expect to see the motor speed also increase at the beginning with the voltage like seen in previous tests.

A screen shot of a graph

Description automatically generated

It is important to note however, that within the peak the response is nearly identical to that of the 1ms test, staying within this time frame.

# Conclusion:

After running the model and altering parameters, the current control model has proven to be an excellent method of achieving your desired outcomes. The equations used are straight forward and make tuning the PI controller much simpler and far more accurate compared to other methods and process’. Typically tuning is a long process and is far too inaccurate to complete without computer assistance and various other models running in parallel. The most important factor in this lab to consider is the choice to run in discrete time. While this does have potential to reduce accuracy and performance, it is very unlikely and typically will have minimal affects depending on the application. The benefit of runtime and computational power may be worth using depending on the application it is used for. For example, more intense process’ like an MPC where the computational load is far more apparent and can negatively affect your ability to control your motor, choosing a discrete model may be a much better alternative, so long as you are able to achieve acceptable accuracy with the lower granularity.