**ECE 730**

**Control of Adjustable Speed Drives**

Faculty of Engineering – McMaster University – Graduate Studies

Winter 2025

Professor: Babak Nahid-Mobarakeh

**Lab 3 Torque Control of Permanent-Magnet Synchronous Motors**

**Report**

**Submitted by**: HaoJun(Henry) Zhao, zhaoh52, 400179164

**Submission Date**: March 23, 2025

Contents

[Introduction 3](#_Toc193745889)

[Methodology 4](#_Toc193745890)

[Simulation & Discussion 6](#_Toc193745891)

[Maximum Torque Per Ampere (MTPA) – ZeroId 6](#_Toc193745892)

[Maximum Torque Per Ampere (MTPA) – ZeroId Discussion 7](#_Toc193745893)

[Maximum Torque Per Ampere (MTPA) 8](#_Toc193745894)

[Maximum Torque Per Ampere (MTPA) Discussion 9](#_Toc193745895)

[Maximum Torque Per Ampere (MTPA) + 20% error 9](#_Toc193745896)

[Maximum Torque Per Ampere (MTPA) + 20% error Discussion 11](#_Toc193745897)

[Maximum Torque Per Ampere (MTPA) + Step Changes 12](#_Toc193745898)

[Maximum Torque Per Ampere (MTPA) + Step Changes Discussion 14](#_Toc193745899)

[Maximum Torque Per Ampere (MTPA) + Step Changes 14](#_Toc193745900)

[Maximum Torque Per Ampere (MTPA) + Step Changes Discussion 16](#_Toc193745901)

[DB-MPCC + Step Changes 17](#_Toc193745902)

[18](#_Toc193745903)

[DB-MPCC + Step Changes Discussion 18](#_Toc193745904)

[Appendix 20](#_Toc193745905)

# Introduction

Previous labs have completed the modeling for PMSM and the first steps of indirect torque control using a PI controller. This lab is mainly focused on the next step of indirect torque. The main objective of this lab is to implement and analyze advanced indirect torque control strategies for PMSMs, specifically the Maximum Torque Per Ampere (MTPA) technique and a Deadbeat Model Predictive Current Controller (DB-MPCC) which are used a lot in current EV applications. It also includes evaluating control robustness against parameter variations and comparing the performance of classical PI-based controllers with modern predictive algorithms including torque ripple, total harmonic distortion (THD), battery power consumption, and controller computational load are analyzed to benchmark performance. The goal of this lab is to have a deeper understanding of the PMSM indirect torque strategy. The modeling of this lab at a high level is shown in *Figure 1 Indirect Torque Control of PMSM*.

A diagram of a block diagram

AI-generated content may be incorrect.

Figure 1 Indirect Torque Control of PMSM

# Methodology

This lab mainly included three parts. The first part is the implementation of the MTPA strategy. Lab 1 gives us a basic model of PMSM which can be used here directly. The current control loop is developed in Lab2 which is key in indirect torque control but only step torque is applied in Lab 2. Furthermore, stays constant in Lab 2 which is not the case in MTPA + flux weakening. The main objective for the first part is to develop two look-up tables that find and are possible. The MTPA algorithm can be derived from the following formula:

Where Is  the number of pole pairs, is the permanent magnet flux, and are -axis inductances. The optimal and values minimize for a given torque. The second part of this lab is to implement a deadbeat model predictive current controller (DB-MPCC). The equation for DB-MPCC is the following:

Also, there is a notable point is the Third Harmonic Injection PWM is used here therefore, the voltage values have to be limited for the system to work properly. The requirement for third harmonic injection PWM is the following:

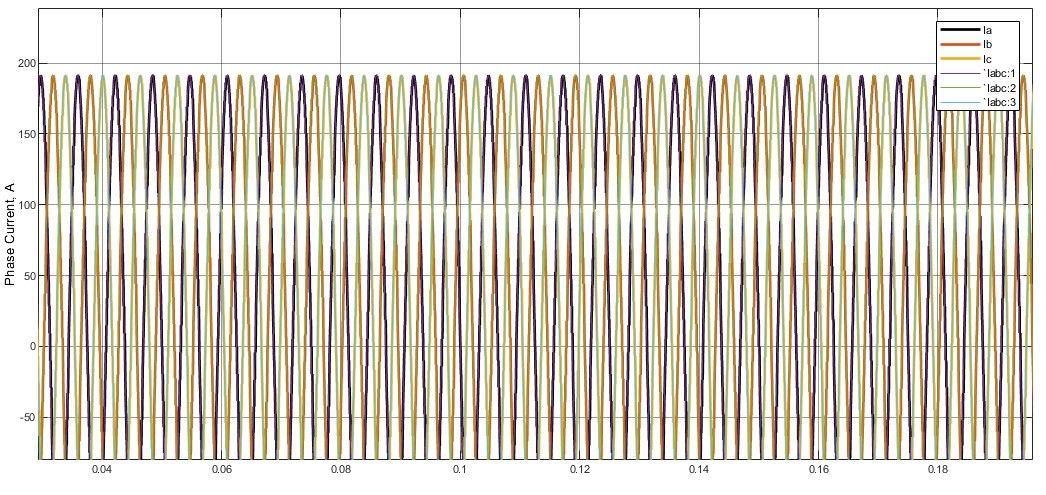
Therefore, if the applied and are greater than this value, the limitation needs to be applied.

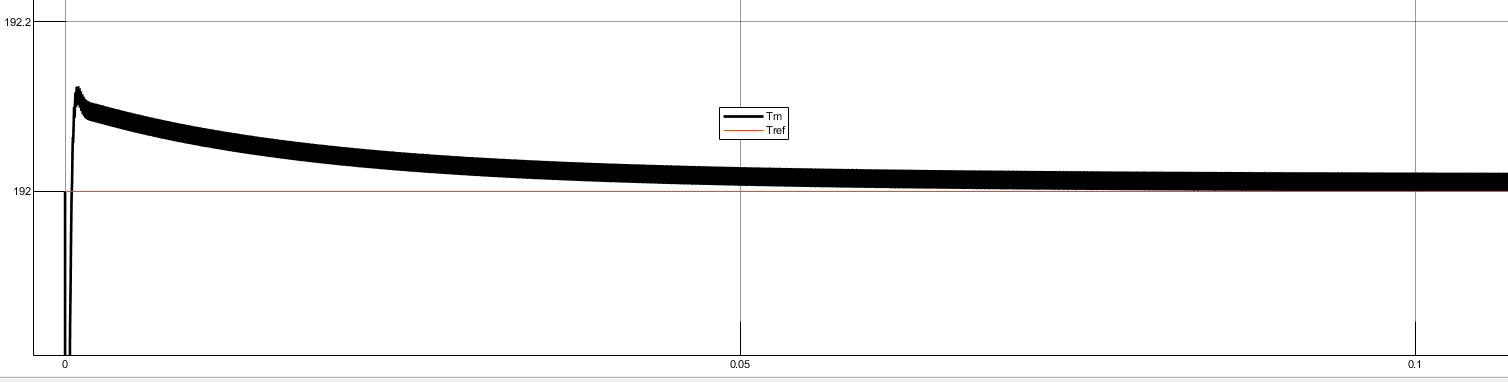
# Simulation & Discussion

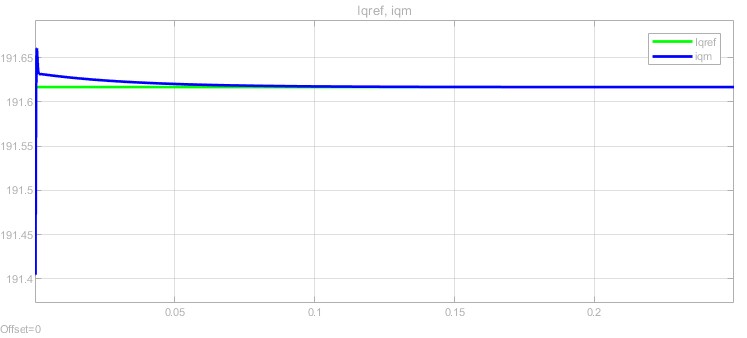
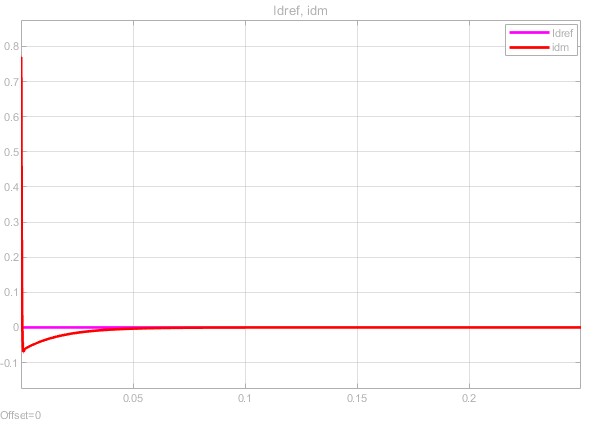
## Maximum Torque Per Ampere (MTPA) – ZeroId

The key parameter values can be concluded in the following table:

|  |  |
| --- | --- |
| Motor Efficiency (%) | 93.59 |
| Battery Power(kW) | 51.56 |
| Steady-state Average Torque (Nm) | 192 |
| Steady-state torque ripple (%) | 0.004078 |
| Micro-controller load | 0.001288 |
| Current THD (%) | 0.05156 |
| Torque Tracking Error (Nm) | 0.0136 |







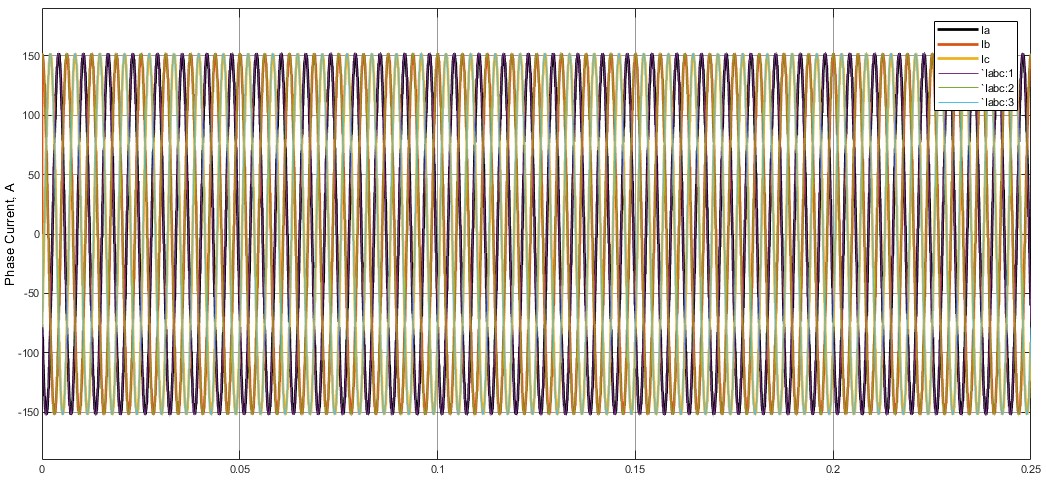
## Maximum Torque Per Ampere (MTPA) – ZeroId Discussion

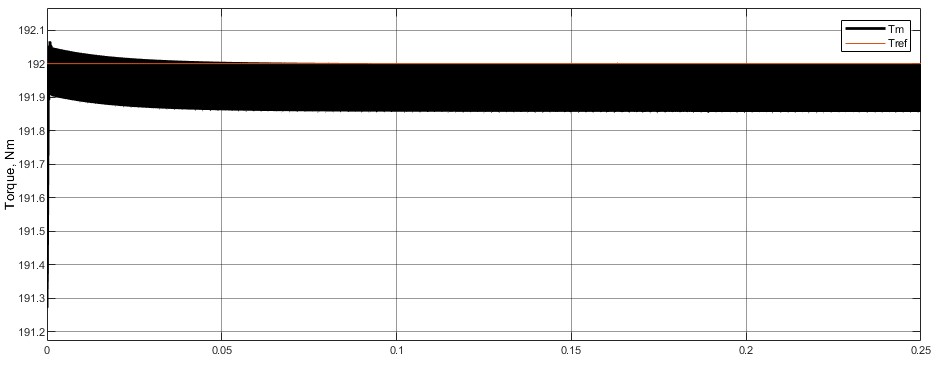
These simulation results are for zero Id strategy with only a simple PI controller. There is no MTPA control involved in this section and it works as a baseline of the whole lab. For , it can be observed that it has some initial disturbance at the start, it is due to the controller reacting to an initial synchronization. This is a natural characteristic of PMSM. Since is 0, the steady state for Is stabilized at 0. For , it is a similar reason behind the PI controller takes its effect to regulate to after around . The rise time matches exactly in the PI controller design. The shows classic three phases of machine behavior which is expected but which leads to three phases currently exhibiting higher amplitude to support torque generation. Torque value follows similar behavior. The small initial deviation is due to the transient response of the PI controller but is quickly corrected within the expected. rise time. The smooth torque response confirms that the PI controller is correctly tuned, providing good tracking performance with minimal delay.

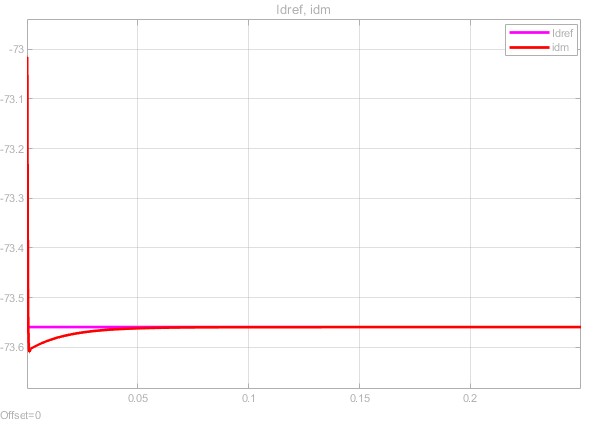
## Maximum Torque Per Ampere (MTPA)

The key parameter values can be concluded in the following table:

|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.86 |
| Battery Power(kW) | 50 |
| Steady-state Average Torque (Nm) | 191.9 |
| Steady-state torque ripple (%) | 0.02251 |
| Micro-controller load | 0.002302 |
| Current THD (%) | 0.04192 |
| Torque Tracking Error (Nm) | 0.1048 |





A graph with a line

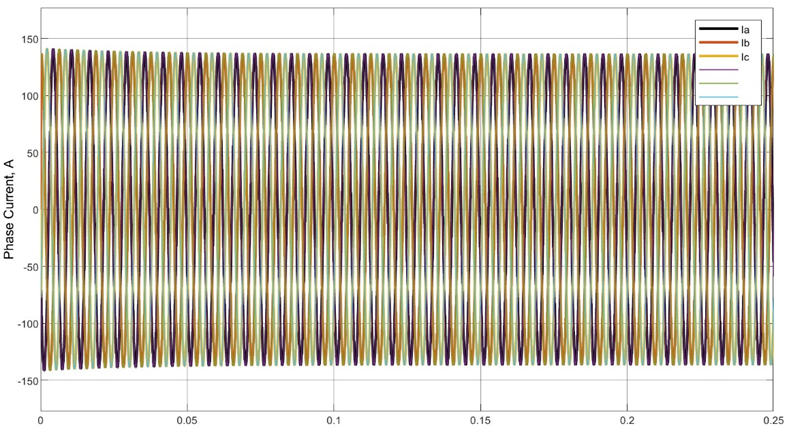
AI-generated content may be incorrect.

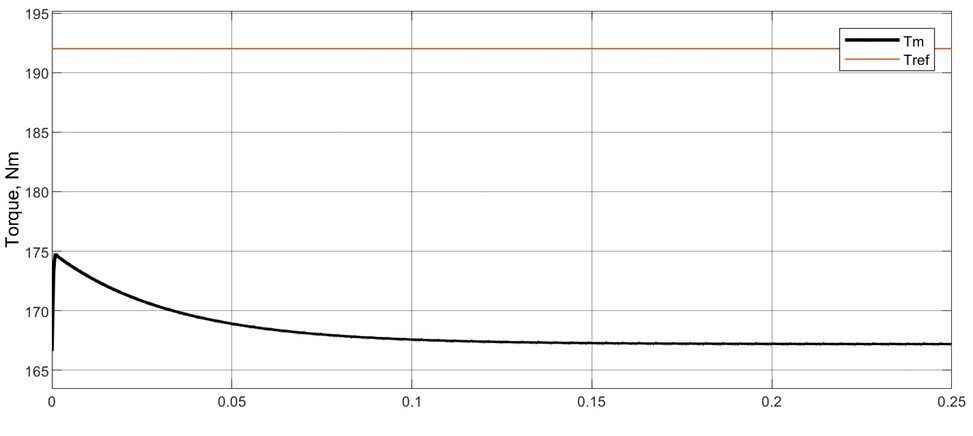
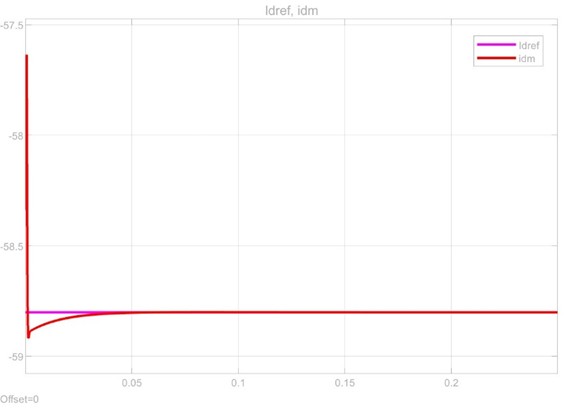
## Maximum Torque Per Ampere (MTPA) Discussion

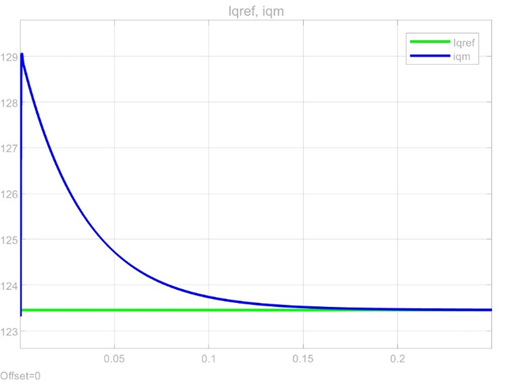
By looking at the metrics between MTPA and zero . There are lots of differences. Firstly, the controller load for MTPA is 0.002302 and for zero is 0.001288. Although this load is based on computer processor performance, it can show the computation expenses for different algorithms. The MTPA strategy requires real-time calculations for searching the values in the look-up table to find optimal and values which can make the algorithm more complex. On the other hand, the zero approach is simpler just regulating values. Therefore, the MTPA controller load is higher than zero . Secondly, the motor efficiency for MTPA is higher than zero strategy which is expected because that is the whole point of MTPA. The derivation behind MTPA is to lower copper losses by finding optimal and values. In contrast, keeping zero keep all the time, which means is fully responsible for torque generation, which can lead to higher phase currents and higher copper losses. For battery power, MTPA is lower than zero . For a similar reason, the motor efficiency increases and phase currents decrease which leads to less power consumption. The steady-state average torque is almost the same, meaning MTPA can regulate torque pretty well. The biggest drawback of MTPA is the torque ripple and torque tracking error. By introducing non-zero make the disturbance in torque is much higher than zero . For the same reason can cause tracking error to be higher than the zero . From THD% point of view, MTPA performs a little better than the zero strategy due to the smaller magnitude in three phases current.

## Maximum Torque Per Ampere (MTPA) + 20% error

|  |  |
| --- | --- |
| Motor Efficiency (%) | 96.14 |
| Battery Power(kW) | 43.7 |
| Steady-state Average Torque (Nm) | 167.2 |
| Steady-state torque ripple (%) | 0.02447 |
| Micro-controller load | 0.004211 |
| Current THD (%) | 0.04697 |
| Torque Tracking Error (Nm) | 24.71 |





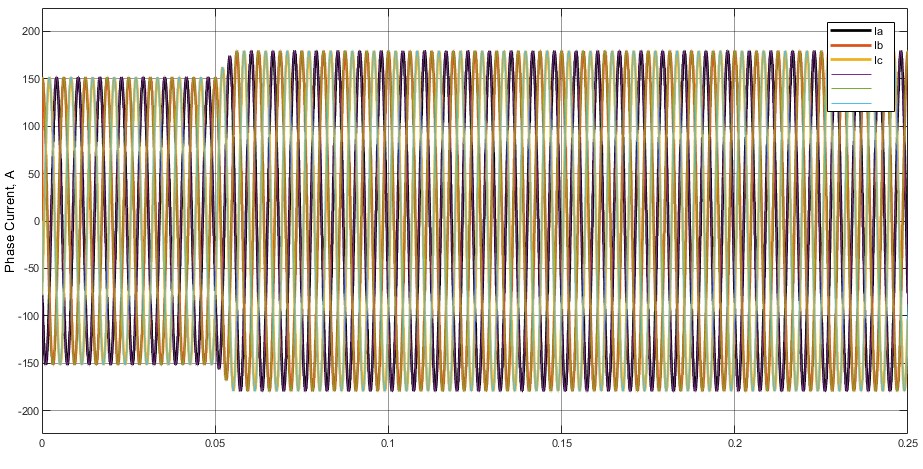


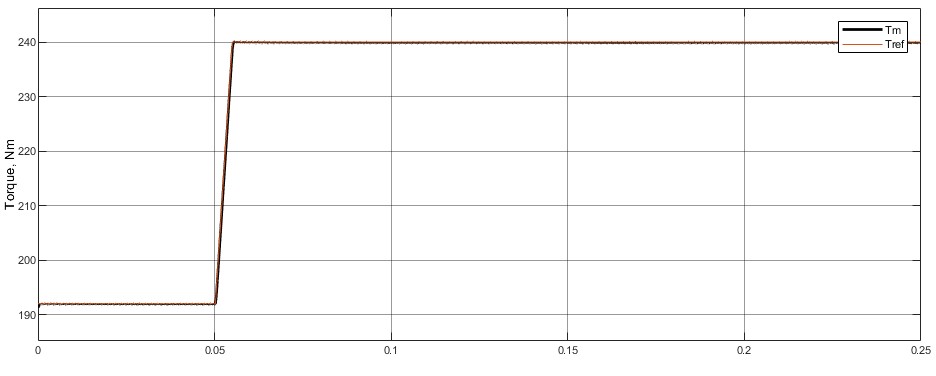
## Maximum Torque Per Ampere (MTPA) + 20% error Discussion

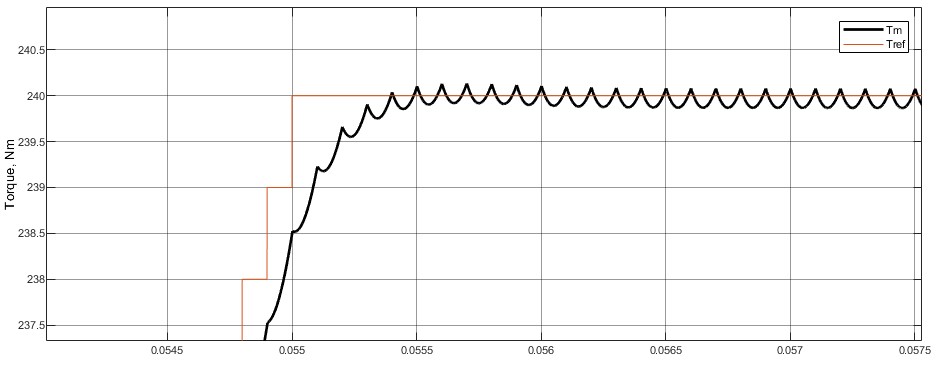
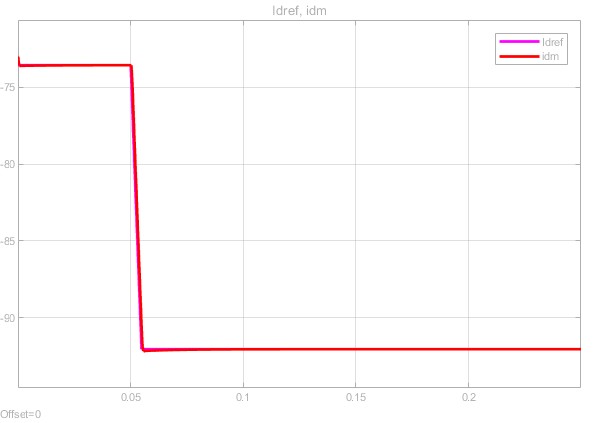
In this step, there is a 20% error on . The performance metrics have some changes compared to Step 2. The motor efficiency is increased compared to the regular MTPA strategy. This is because high cause less total magnitude of phase currents. MTPA look-up table adjusts and to match . The lower current causes fewer copper losses. That is the reason why the motor efficiency increases here. Furthermore, lower current also causes less power consumption. The steady-state torque drops significantly because there is an error in . This is a similar situation in Lab 2. The parameter error can cause the PI controller to fail to regulate the reference value. Also, from a mathematics point of view, an increase in may cause drops unnecessarily which can lead to lower torque output. The error in does not affect the torque ripple value that much but there is an increase in torque ripple. The error causes MTPA to look up table interpolation and select not perfect value which can introduce more fluctuations in the torque response. The microcontroller load also increases. This can be caused by different reasons. First of all is the algorithm issue, a 20% error in parameters may need more adjustment in the controller to compensate for the error. Secondly, it may be because I used my personal laptop to run this part instead of my PC. The torque track error increases significantly as well. It is hard for the controller to regulate precise torque values because the error in make the MTPA strategy select the wrong and values.

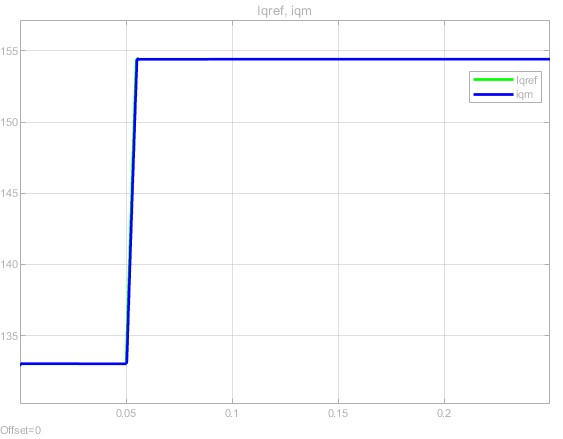
## Maximum Torque Per Ampere (MTPA) + Step Changes

|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.39 |
| Battery Power(kW) | 63.19 |
| Steady-state Average Torque (Nm) | 239.9 |
| Steady-state torque ripple (%) | 0.02619 |
| Micro-controller load | 0.001724 |
| Current THD (%) | 0.0374 |
| Torque Tracking Error (Nm) | 0.02754 |







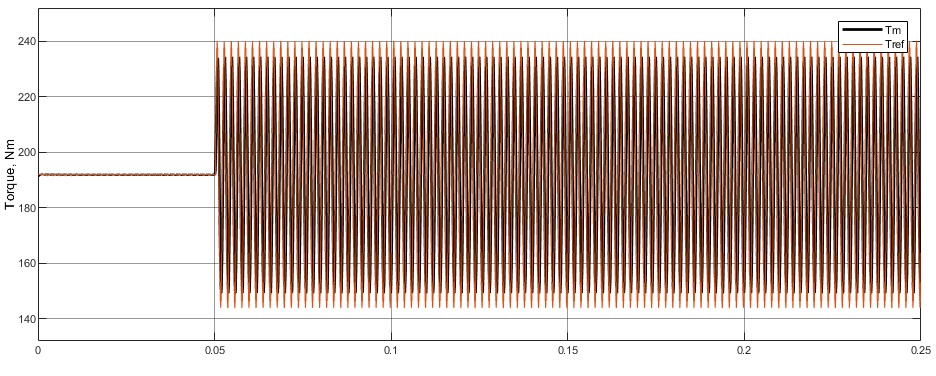
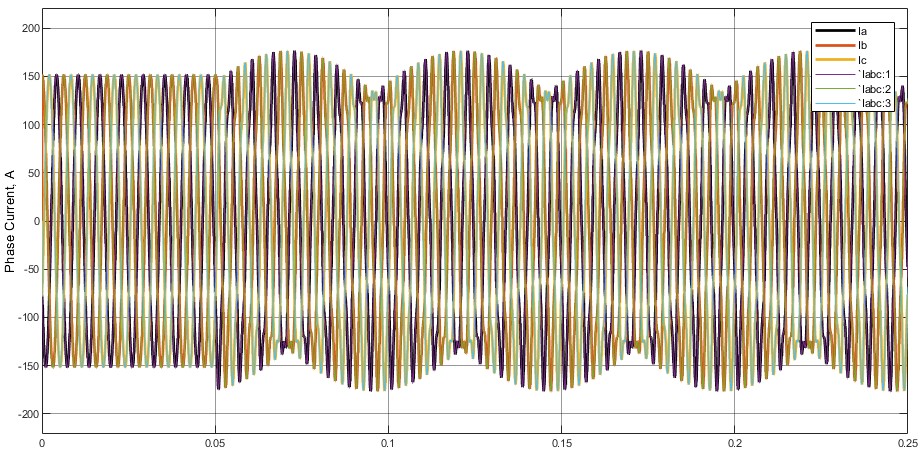


## Maximum Torque Per Ampere (MTPA) + Step Changes Discussion

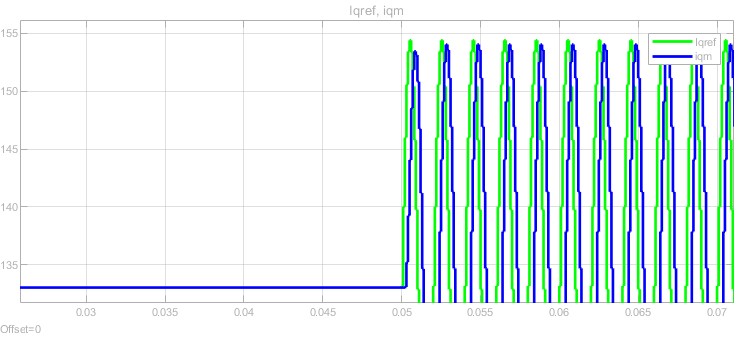
In this step, 20% error on is removed. Only reference torque is changed from constant to step change. From the simulation result, the measured torque followed the reference well. It matches the PI controller design parameters. The small disturbance and errors can be caused by sudden demand increases. This is the situation that happened in Lab 2 as well. has a sudden drop when a step change happens that also matches the MTPA control strategy, the whole process is smooth without any overshoot which represents controller is well-tuned. Similar behavior for , in contrast to decreasing, Increases due to the MTPA look-up table. The magnitude of phase currents increases when step changes happen which is also expected. For the value of the metric, all the values are reasonable. The motor efficiency drops around 0.5% because the step change torque reference is higher than the constant torque reference which requires higher current, leading to increased copper losses. Battery power also increases because of higher torque reference. Torque ripple increases slightly which means the system is a little hard to maintain torque status. The controller load is similar to the base model which also makes sense because the only thing change is the type of reference torque which should not make any difference from the controller's point of view.

## Maximum Torque Per Ampere (MTPA) + Sinusoidal Change

|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.73 |
| Battery Power(kW) | 49.65 |
| Steady-state Average Torque (Nm) | 239.9 |
| Steady-state torque ripple (%) | 0 |
| Micro-controller load | 0.002348 |
| Current THD (%) | 12.51 |
| Torque Tracking Error (Nm) | 11.74 |





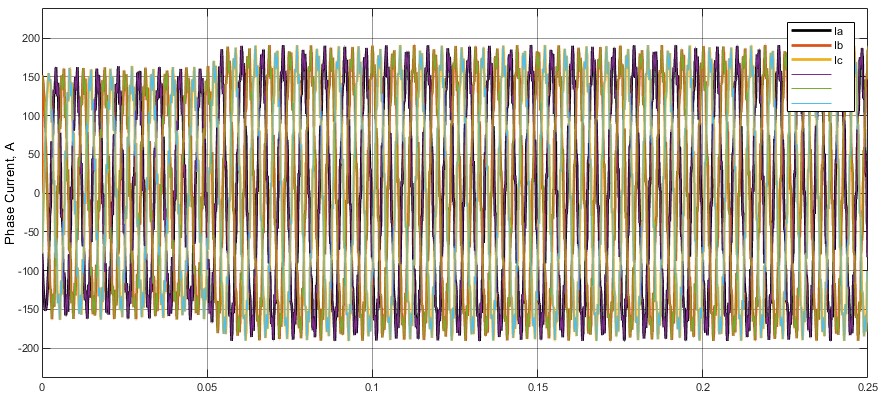


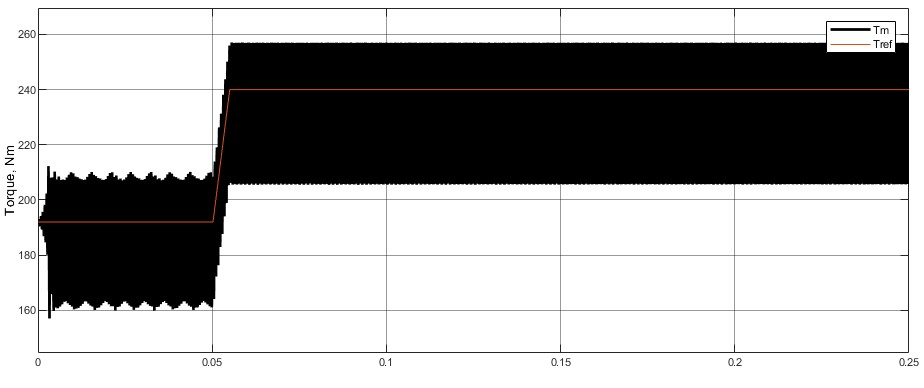
## Maximum Torque Per Ampere (MTPA) + Sinusoidal Changes Discussion

In this section, the reference torque changed to sinusoidal reference instead of step change. In this step, the noticeable torque tracking error exists in the system which indicates that the system struggles to maintain the sinusoidal torque reference. This is expected because MTPA is optimized for steady-state torque conditions not for sinusoidal behavior. Furthermore, are all have some delays in their response which also due to the sinusoidal input makes the PI controller hard to regulate. The phase currents follow sinusoidal torque reference fully which also makes sense. The motor efficiency increases in this case which is caused by current instability. THD value also has a significant change because Are not stable.

## DB-MPCC + Step Changes

|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.49 |
| Battery Power(kW) | 61.24 |
| Steady-state Average Torque (Nm) | 232.2 |
| Steady-state torque ripple (%) | 7.338 |
| Micro-controller load | 0.002734 |
| Current THD (%) | 9.783 |
| Torque Tracking Error (Nm) | 176.7 |





# 

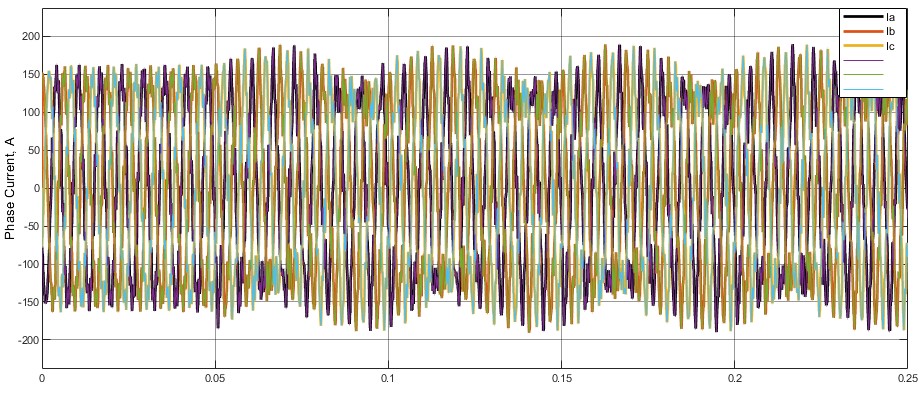
## 

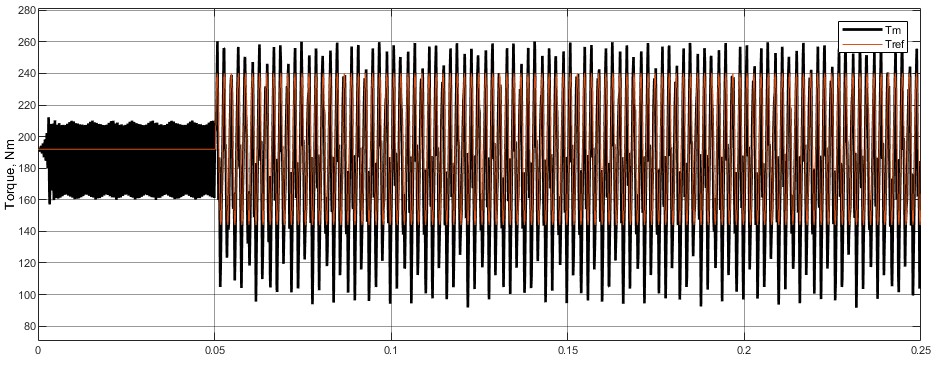
## DB-MPCC + Step Changes Discussion

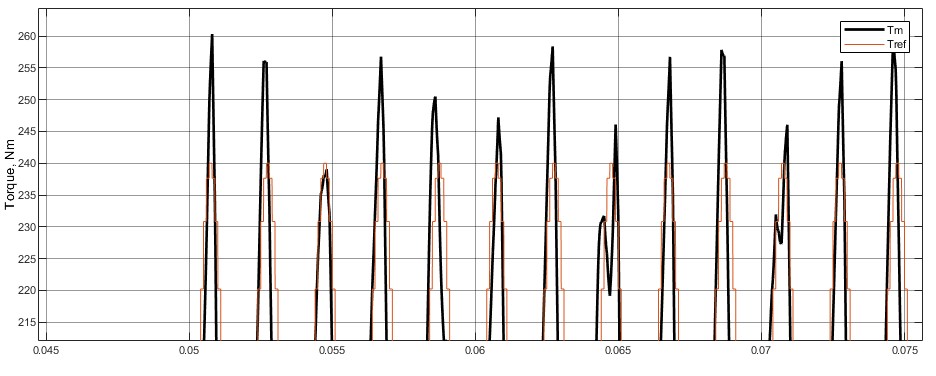
For the PI current controller, the data shows the low torque ripple and current THD is low. On the other hand, the DB-MPCC shows a much higher torque tracking error and torque ripple compared to the PI current controller. The current THD and computational load perform a little better than the PI current controller. It potentially shows that MPCC has potential higher response speed than PI's current controller. Looking into the formula more, there is no complex recursion part. Furthermore, battery power consumption for DB-MPCC is much higher than PI current controller which makes sense because the DB-MPCC selected value is not values from MTPA. Since Value is not optimal, it causes the whole system to have much higher losses. From the implementation point of view, deadbeat-MPCC must have a voltage limiter for Third Harmonic Injection PWM which also limits its performance. The deeper reason behind the higher torque ripple and higher torque tracking error is that deadbeat MPC is an open loop control and its prediction horizon is only one which means it is really sensitive to the step change. Performance matches expectations, it remains a relatively small value and increases and step changes for Get involved. The shape is close enough to a sine wave. and Performance is relatively bad compared to the PI controller because there is no decoupling term in DB-MPCC.

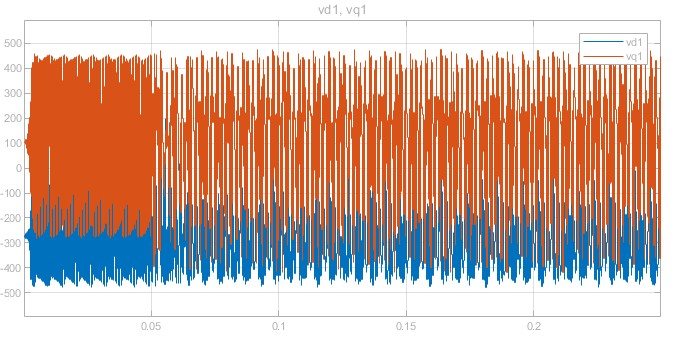
## DB-MPCC + Sinusoidal Changes

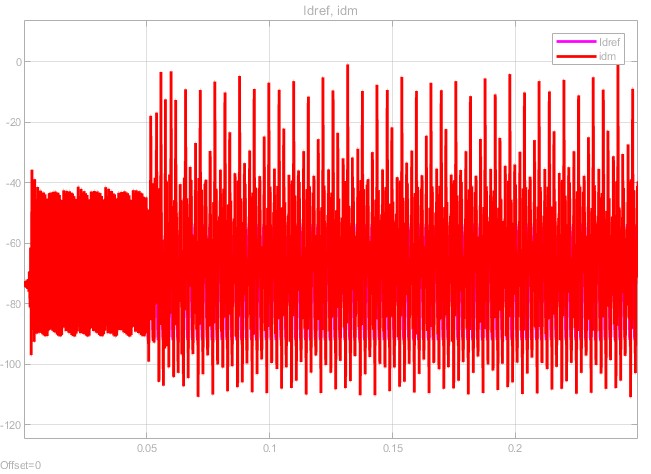
|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.49 |
| Battery Power(kW) | 61.24 |
| Steady-state Average Torque (Nm) | 180.1 |
| Steady-state torque ripple (%) | 0 |
| Micro-controller load | 0.002996 |
| Current THD (%) | 20.78 |
| Torque Tracking Error (Nm) | 22.22 |

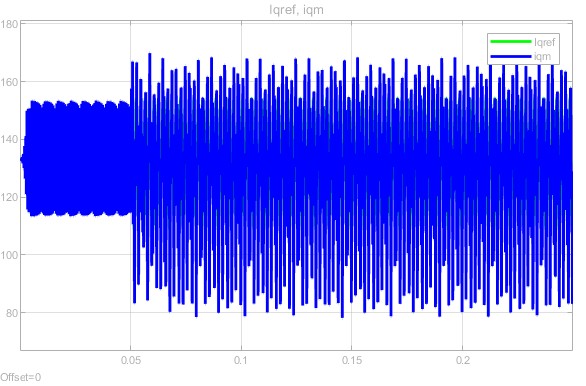










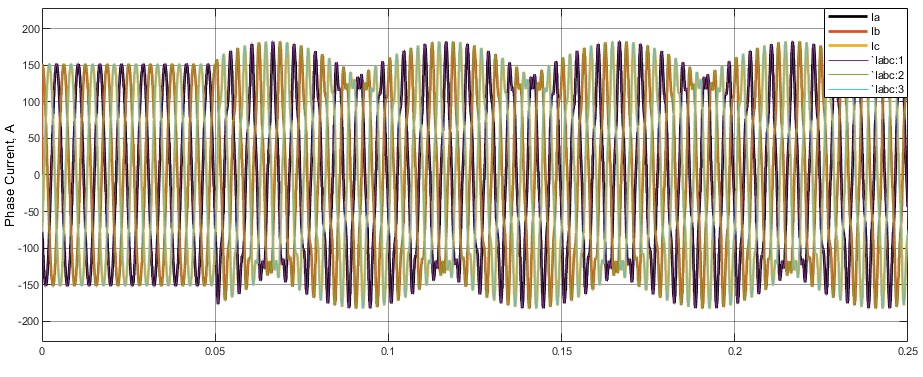


DB-MPCC + Sinusoidal Changes Discussion

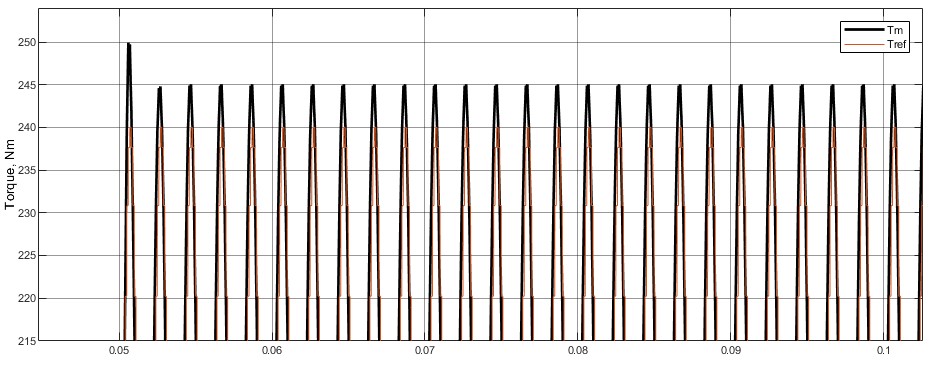
The deadbeat model predictive control with sinusoidal changes has significant differences compared to MTPA with sinusoidal changes. First of all, the three phases current for both condition shows shape of sine waves, but MTPA one has less disturbance and its Current THD is much lower than DB-MPCC. Secondly, and shows instability in DB-MPCC and can not stay in the steady state. In comparison, MTPA controller ensures a straightforward decoupling. and measurements are more ideal. DB-MPCC generates dynamic voltage commands due to its discretion in the formula. The switching for MPC control is not fixed value which can put extra stress on the inverter. By looking at the performance indicators, both control methods show similar motor efficiency. DB-MPCC need more power with the similar reason in DB-MPCC with step changes.

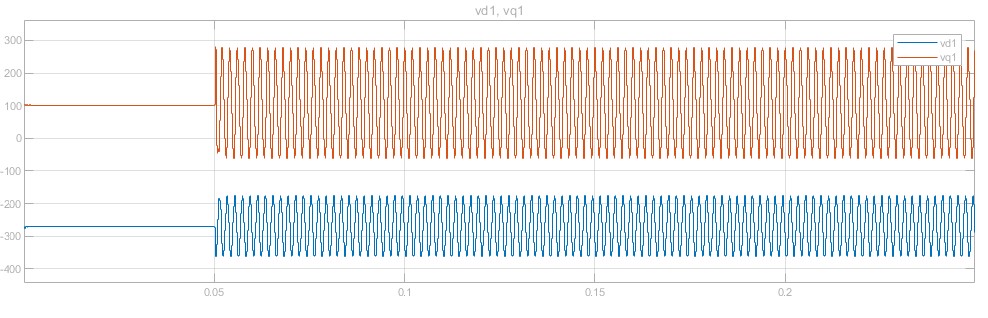
## DB-MPCC + Sinusoidal Changes + negative

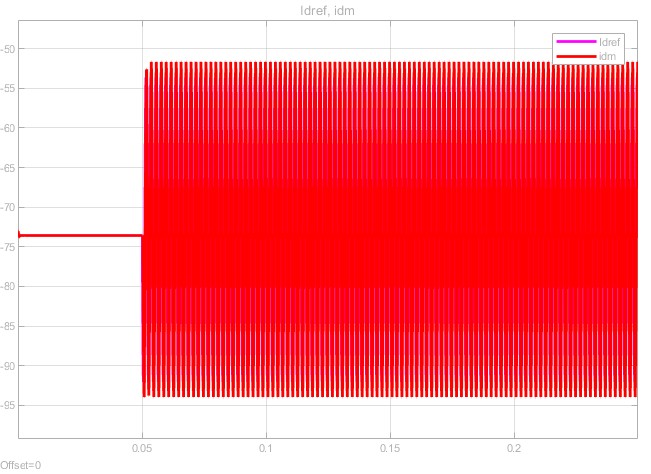
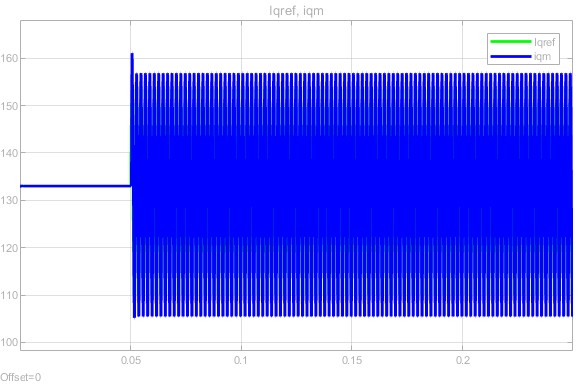
|  |  |
| --- | --- |
| Motor Efficiency (%) | 95.76 |
| Battery Power(kW) | 49.7 |
| Steady-state Average Torque (Nm) | 191.2 |
| Steady-state torque ripple (%) | 0 |
| Micro-controller load | 0.002286 |
| Current THD (%) | 15.61 |
| Torque Tracking Error (Nm) | 11.33 |



A graph of a graph showing a number of lines

AI-generated content may be incorrect.





## DB-MPCC + Sinusoidal Changes + negative Discussion

From tracking error point of view, negative PI controller gain improves the performance of the controller. All critical performance indicator decreases including battery power, microcontroller load, current THD and torque tracking error. The motor efficiency also increases at the same time but it can not conclude negative gain can help with the performance because it still lacks of proof for negative gain under sudden step changes.

# Appendix