# EEE6203 COURSEWORK ASSIGNMENT

**Name:** Benjamin Griffiths

**UCARD:** 160159871

# Contents

# Introduction

This report will cover the use of MATLAB and Simulink to design, model and simulate a control system to drive a permanent magnet (PM) brushless AC (BLAC) machine. The control system will be capable of driving the machine in both torque and speed control modes.

The specification of the motor for the control system to drive is shown below in table 1.

| Parameter | Value |
|---|---|
| Machine topology | Surface mounted permanent magnet |
| Number of pole-pairs | 7 |
| Connection | Star |
| Continuous/peak power (kW) | 5.0/10.0 |
| Phase resistance @ 120°C (mΩ) | 22.2 |
| Synchronous inductance (mH) | 0.344 |
| Flux linkage per phase (mWb) | 39.6 |
| Torque constant ($Nm_{peak}$/A) | 0.415 |
| Back-EMF constant ($V_{peak}$/(rad/s) | 0.277 |
| Base speed (rpm) | 1350 |
| Maximum speed (rpm) | 5500 |
| Continuous current ($A_{peak}$/$A_{rms}$) | 85/60.5 |
| Maximum current ($A_{peak}$/$A_{rms}$) | 170/121 |
| Moment of inertia ($kgm^2$) | 0.008 |
| DC link voltage (V) | 270 |

*Table 1: Machine parameters.*

The first section of the report will cover the modelling of the machine drive, including modelling the machine in the dq and ABC reference frames. The model will then be simulated in torque and speed control modes to assess the performance of the control systems.

# 1. Modelling Synchronous Permanent Magnet Machine Drive

## 1.1 Modelling Machine in dq Reference Frame

It is necessary to model the machine in the direct-quadrature (dq) reference frame in an effort to simplify analysis, since it allows 3Φ time varying quantities to be modelled as DC signals. Typically, the variables in the differential equations that describe the behaviour of 3Φ machines are time varying, such as the flux linkages and currents, which will vary during the machines motion. This makes mathematical modelling and analysis of a 3Φ machine complex, thus mathematical transformations can be utilised to transform the 3Φ quantities into other reference frames, allowing the equations to be solved. [1]

To model the machine with the parameters given in table 1, the model is split into two sub-models that simulate the electrical and mechanical characteristics.

To model the electrical characteristics, we must first define the dq quantities with a pair of simultaneous equations shown in equation 1 and 2.

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Psi_d \\ \Psi_q \end{bmatrix} + \omega \begin{bmatrix} -\Psi_q \\ \Psi_d \end{bmatrix} \quad 1$$

$$\begin{bmatrix} \Psi_d \\ \Psi_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \Psi_m \\ 0 \end{bmatrix} \quad 2$$

Where $\Psi_d/\Psi_q$ are the dq axis flux linkages, $\Psi_m$ is the magnet flux linkage, ω is the mechanical speed, $L_d/L_q$ are the dq axis inductances and $i_d/i_q$ are the dq axis currents.

These equations can be solved in Simulink to find $i_d$ and $i_q$ by using an ordinary differential equation solver, as seen in figure 1.

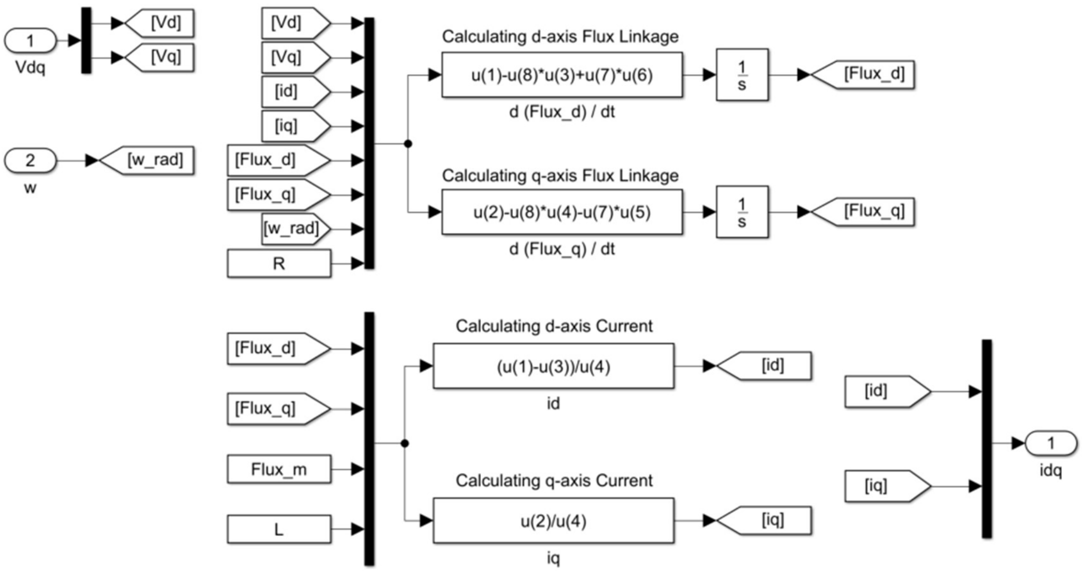


*Figure 1: Solving dq-axis differential equations.*

Now that the dq-axis flux linkages and currents are known, the electromagnetic torque ($T_e$) can be calculated using equation 3.

$$T_e = \frac{3p}{2}\left(\Psi_d i_q - \Psi_q i_d\right) \quad 3$$

The Simulink implementation of equation 5 is shown in figure 2.

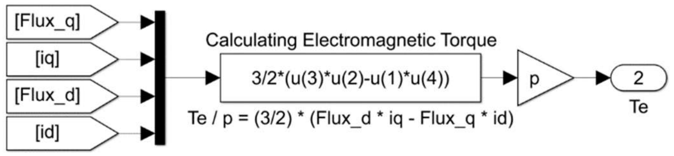


*Figure 2: Solving electromagnetic torque equation.*

The electrical model is now complete, and the electromagnetic torque calculated can be passed into a mechanical model of the machine, which will describe how the rotational movement is influenced by factors such as load torque, viscous friction and inertia.

The mechanical output speed of the machine can be solved using equation 4.

$$\omega_m = \int \left( \frac{T_e - T_m - B\omega_m}{J} \right) dt \qquad 4$$

Where $T_e$ is the electromagnetic torque calculated in the machines electrical model, $T_m$ is the mechanical load torque, B is the viscous friction coefficient and J is the total inertia.

The Simulink implementation of equation 4 is shown in figure 3.

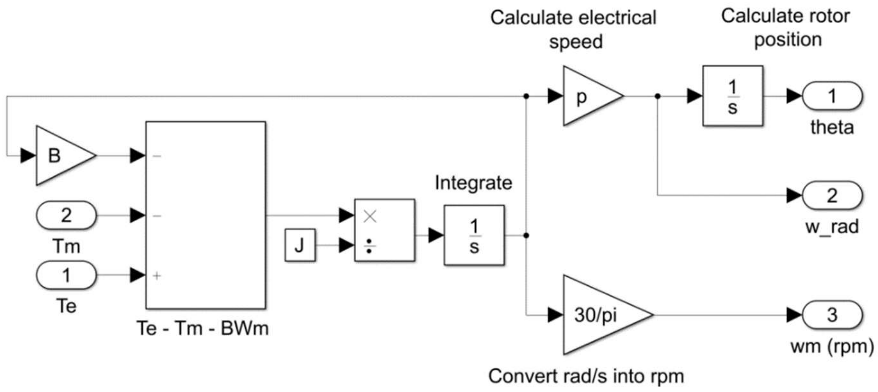


*Figure 3: Calculating machine speed and position.*

## 1.2 Modelling Machine in ABC Reference Frame

The machine model created is in the dq reference frame, however, the phase voltages and currents required to drive the machine are in the ABC reference frame. Fortunately, the 3Φ quantities can be transformed into the dq frame by first applying an ABC-αβ transformation, otherwise known as Clark transform, followed by an αβ-dq transformation, known as the Park transform.

The output dq quantities from the machine model can then be decomposed back into ABC quantities later on by applying an inverse transformation.

The ABC-dq transformation is achieved using equation 5.

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \qquad 5$$

To convert the 3Φ voltages into the dq frame for the machine model, a Simulink implementation of equation 5 is used, shown in figure 4.

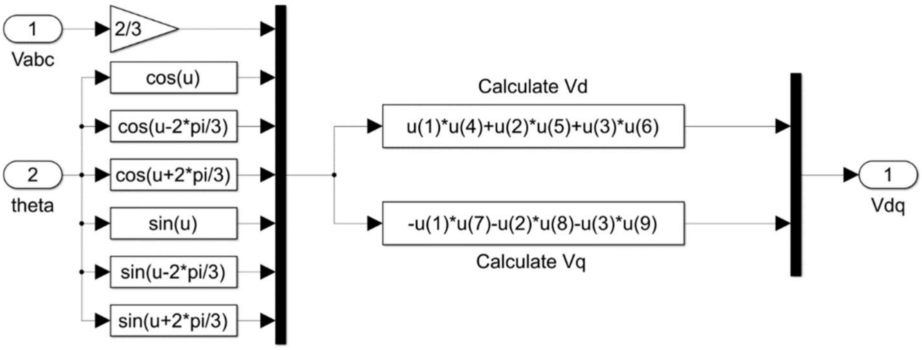


*Figure 4: Vabc - Vdq transformation.*

The inverse transformation to re-obtain the ABC quantities is given by equation 6.

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \end{bmatrix} \qquad 6$$

A Simulink implementation of equation 6 shown in figure 5.

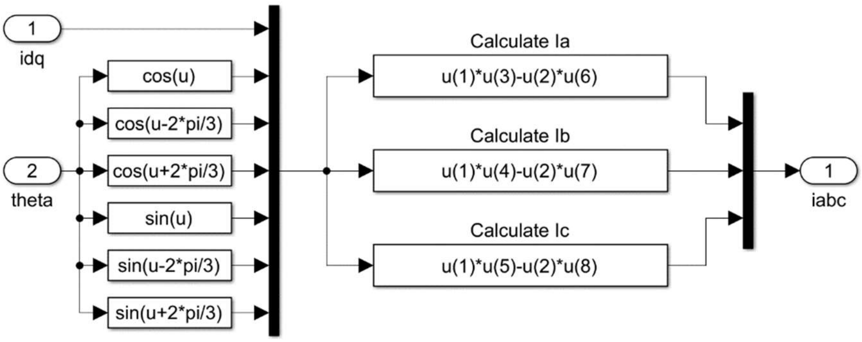


*Figure 5: dq to ABC transformation.*

## 1.3 Design of an $i_{dq}$ PI Controller

A control system is used to set the dq axis quantities to maintain a desired output speed or output torque and react to system disturbances. A proportional-integral (PI) controller will be used for this task, with a desired bandwidth of 800Hz. The proportional part of the control system influences how aggressively the output changes based on an error signal, whereas the integral part affects the controller action depending on the time duration an error has existed.

By looking again at equation 1, we can see that $i_{dq}$ can be controlled by varying $V_{dq}$. However, it can also be seen at there is some cross-coupling action between the d and q axes, namely, the terms that contain electrical speed.

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Psi_d \\ \Psi_q \end{bmatrix} + \omega \begin{bmatrix} -\Psi_q \\ \Psi_d \end{bmatrix}$$

Cross-coupling terms

The cross coupling between these axes will act to reduce the speed of the dynamic current response, hence eliminating these will enhance the performance of the current controller. This can be achieved by decoupling these terms. [2]

Expanding equation 1, we get $V_d$ and $V_q$ with undesirable coupling terms.

$$v_d = Ri_d + \frac{d}{dt}\Psi_d - \omega\Psi_q$$

$$v_q = Ri_q + \frac{d}{dt}\Psi_q + \omega\Psi_d$$

Coupled terms

The previously designed electrical model for the machine requires the change in dq axis flux, which is obtained by rearranging equation 1, shown in equation 7 and 8.

$$\frac{d\Psi_d}{dt} = V_d - Ri_d + \omega\Psi_q \qquad 7$$

$$\frac{d\Psi_q}{dt} = V_q - Ri_q - \omega\Psi_d \qquad 8$$

Here, we can see that the cross-coupling term is added for $V_d$ and subtracted for $V_q$. The effect of the undesirable coupling terms on the PI controller can be eliminated by modifying $V_d$ and $V_q$.

By letting $V_d'$ and $V_q'$ equal the desired dq-axis voltages from PI controller output, the cross-coupling term for $V_d$ can be subtracted, and the coupling term for $V_q$ added so that they cancel out in the motor model, shown in equation 9 and 10.

$$V_d = V_d' - \omega\Psi_q \qquad 9$$

$$V_q = V_d' + \omega\Psi_d \qquad 10$$

The Simulink implementation of the current controller with decoupling is shown in figure 6.

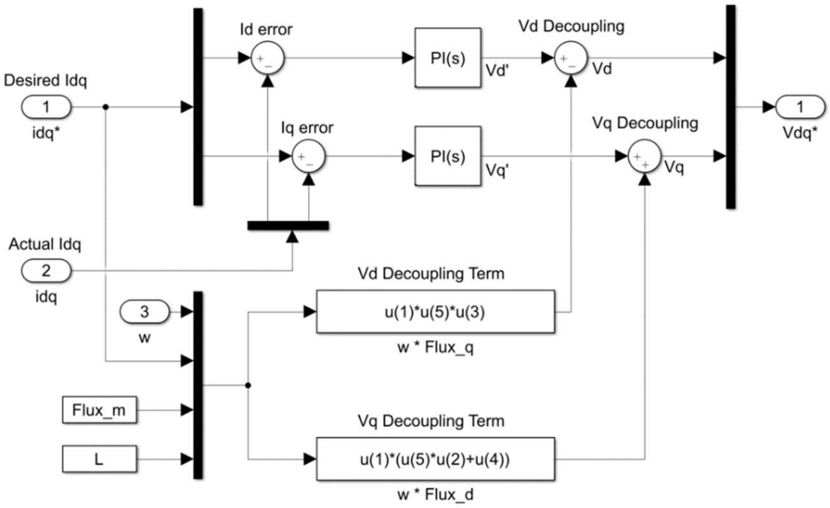


*Figure 6: Idq PI Controller Simulink implementation.*

## 1.4 Over-modulation & Anti-windup Phenomenon

The previously designed controller has no constraint on the output voltage it can demand, and may try to apply a voltage beyond what is available, known as over-modulation or saturation. The DC-link voltage that would power a drive system is typically fixed, particularly in applications such as electric vehicles which are powered with batteries. Therefore, the maximum voltage that the control system can demand should not exceed the maximum available voltage. To prevent this, the output of the current controller is fed into an over-modulation block that will limit or clamp the maximum output voltage.

The Simulink implementation of over-modulation prevention is shown in figure 7.

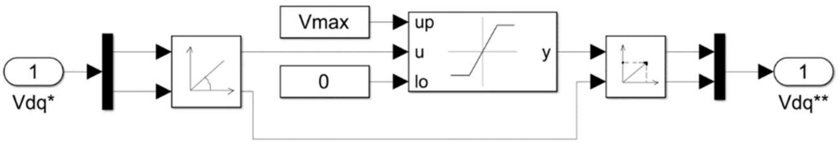


*Figure 7: Over-modulation prevention block.*

Where Vmax is the maximum phase voltage that can be applied across a motor phase, given by equation 11.

$$V_{max} = \frac{V_{DC}}{\sqrt{3}} \quad 11$$

The inclusion of this over-modulation block introduces a problem in the control system. When the controller outputs a voltage that exceed the maximum permissible, the over-modulation block will act to limit this voltage. However, the integrator will continue to act on the error between the voltage received by the motor and the output of the controller, causing the output of the controller to continue rising, known as integrator wind-up.

To prevent this, the controller should stop integrating when the output saturates, which is achieved by comparing the voltage before and after the over-saturation block ($V_{pre}$ and $V_{post}$). If $V_{pre}$ is equal/less than $V_{post}$, the controller's output is not saturated and the integrator can continue to operate as normal. However, if $V_{pre}$ is greater than $V_{post}$, the controller's output is clearly saturated, and the integrator should be stopped.

This functionality is implemented into Simulink by modifying the existing current controller with a custom PI block, shown in figure 8.

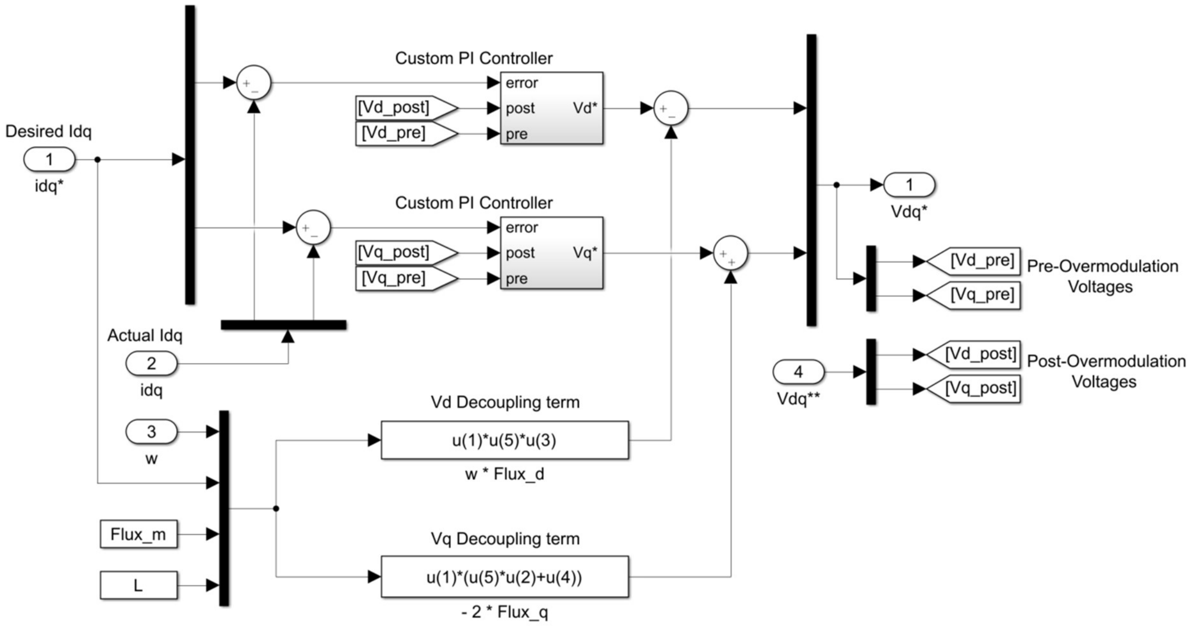


*Figure 8: Modified current controller.*

The custom PI controller is shown in figure 9, where the integrators input is switched to zero if the output is saturated ($V_{post}<V_{pre}$).

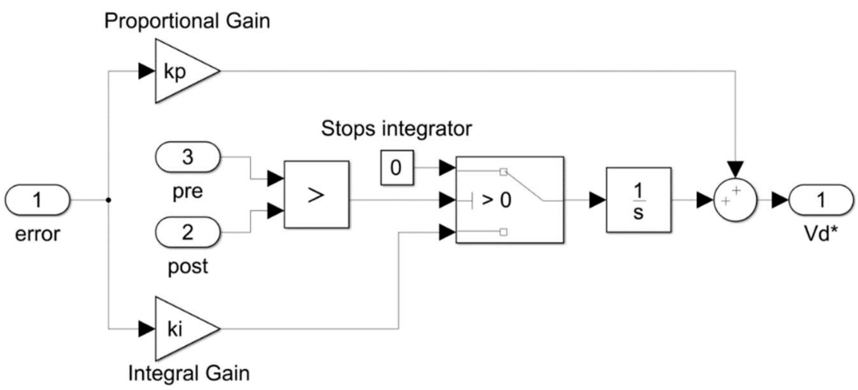


*Figure 9: Custom PI controller.*

## 1.5 Integration of current controller into machine model in ABC reference frame

The key components required to drive the machine have been designed and can now be connected together accordingly to simulate the machines performance. The model starts with the dq-axis current controller, with a desired current as the input. The output dq-axis voltage from the current controller is then filtered by the over-modulation block. Since the motor model block takes 3Φ quantities as inputs, the dq-axis voltages are transformed into the ABC reference frame. The complete system is shown in figure 10.

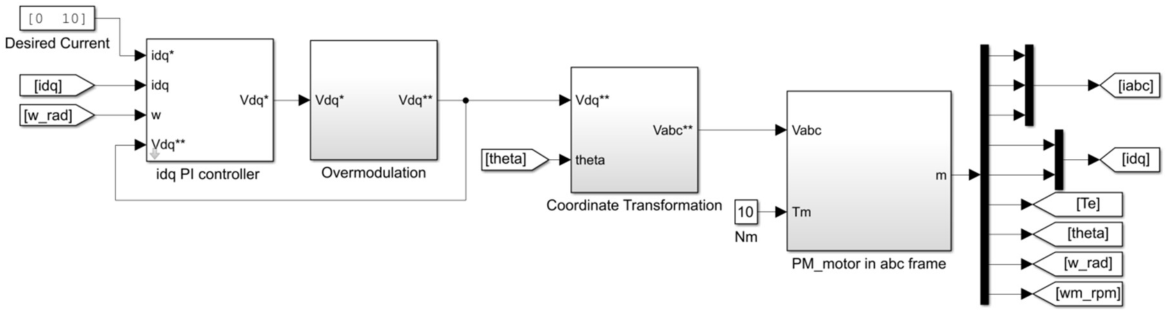


*Figure 10: Complete drive control system.*

# 2. Simulation Results

## 2.1 Simulation of machine in torque control mode

In torque control mode, the control system will act to maintain a desired output torque by adjusting the output speed. The output torque can be related to the dq-axis current by equation 12.

$$T_e = K_t i_q \ or \ i_q = \frac{T_e}{K_t} \tag{12}$$

Where $K_t$ is the torque constant of the machine, given by equation 13.

$$K_t = \frac{3}{2} p \Psi_m \tag{13}$$

Figure 11 shows a diagram of the machine rotor with the dq-axis. The d-axis is coincident with the flux linkage of the rotor from the permanent magnets, and the q-axis is defined to be perpendicular to the d-axis. It can be seen that rotational torque is maximised when the dq-axis flux vector is perpendicular to the rotors flux vector. However, if the resultant dq-axis flux vector is coincident to the rotors flux vector, the rotor magnet will feel an outward force towards to stator, and will not produce any rotational torque, wasting power.

Therefore, to maximise the output torque for a given current, and hence maximise the efficiency of the machine, the d-axis current demand should be set to zero and the q-axis current should be modulated so that the resultant current phasor is always orthogonal to the d-axis. [3]

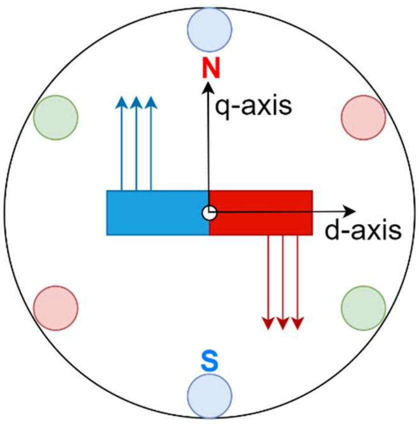


*Figure 11: Maximum torque resultant flux vector.*

A step change in torque demand using equation 12 can be modelled in Simulink as shown in figure 12.

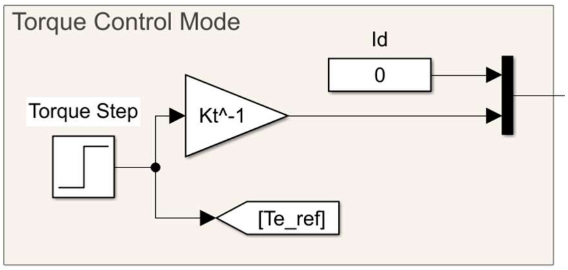


*Figure 12: Step change in torque demand.*

Using J = 1, an initial mechanical speed of 1350rpm and a mechanical load torque of 10Nm, the simulation results from a step change in electrical torque from 5 – 15Nm is shown in figure 13.

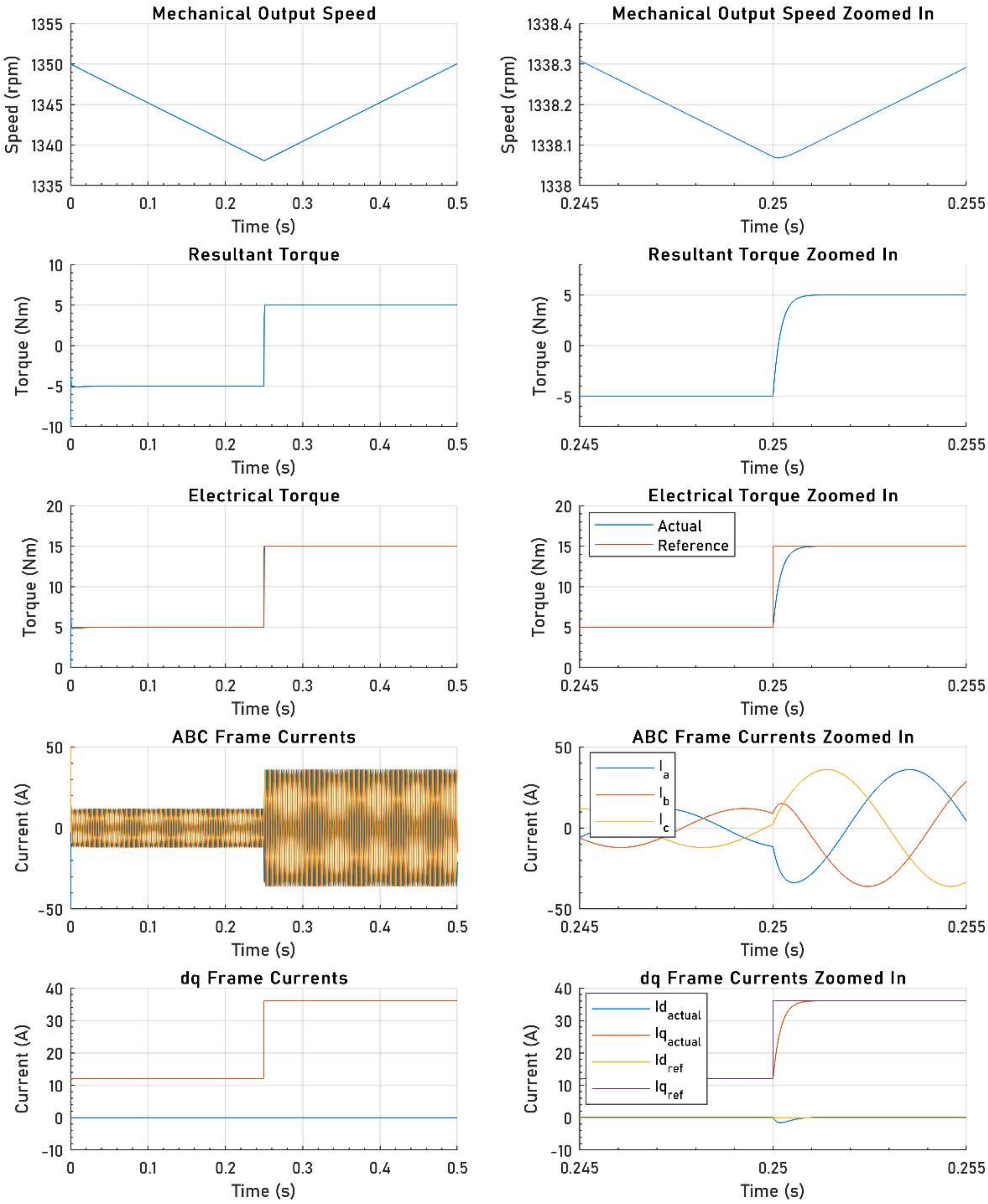


*Figure 13: Step change in torque simulation results.*

From t = 0 to t = 0.25, there is an electrical torque of 5Nm, resulting in a net torque of -5Nm against the direction of rotation, causing the mechanical speed to reduce as seen in the first graph. At t = 0.25, the electrical torque changes from 5Nm to 15Nm, giving a new net torque of +5Nm in the direction of rotation, causing an increase in mechanical speed.

As shown in the zoomed in graphs, the electrical torque and dq-axis currents track the reference signals with a rise time of 416µs.

## 2.2 Simulation of machine in speed control mode

To operate the machine model in speed control mode, a speed controller must be designed and implemented to set the dq-axis reference current based on the desired and actual mechanical output speed.

This functionality can be realised by designing a PI controller to set the q-axis current based on an error between a defined demand speed and the actual output speed.

A closed loop speed controller is represented by the block diagram shown in figure 14.

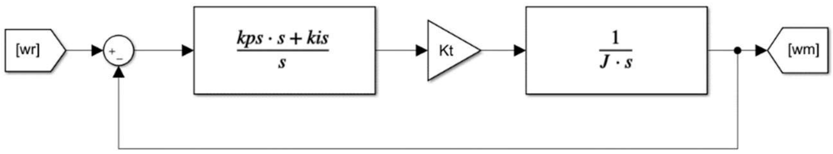


*Figure 14: Speed controller block diagram.*

The proportional and integral gains required to achieve a bandwidth of 50Hz can be obtained from the closed loop transfer function, shown in equation 13.

$$\frac{\omega_m}{\omega_r} = \frac{\frac{(k_i + k_p s)k_T}{s^2 J}}{1 + \frac{(k_i k_p s)k_T}{s^2 J}}$$

Rearranging into the 2nd order transfer function standard form:

$$\frac{\omega_m}{\omega_r} = \frac{(k_i + k_p s)k_t}{s^2 J + (k_i + k_p s)k_t}$$

$$\frac{\omega_m}{\omega_r} = \frac{(k_i + k_p s)\frac{k_t}{J}}{s^2 + \frac{k_p k_t}{J}s + \frac{k_i k_t}{J}} \qquad 13$$

Where:

$$\omega_n^2 = \frac{k_i k_t}{J}$$

$$2\zeta\omega_m = \frac{k_p k_t}{J}$$

Since $\omega_n = 50$, $J = 0.008$, $k_T = 0.415$ and $\zeta = 1$ for critical damping, the proportional and integral gains can now be calculated.

$$k_i = \frac{J\omega_n^2}{k_t} = \frac{0.008 \cdot (50 \times 2\pi)^2}{0.415} = 1902.6$$

$$k_p = \frac{2\zeta\omega_n J}{k_t} = \frac{2 \times 1 \times 50 \times 2\pi \times 0.008}{0.415} = 12.1$$

The Simulink implementation of the speed control system is shown in figure 15.

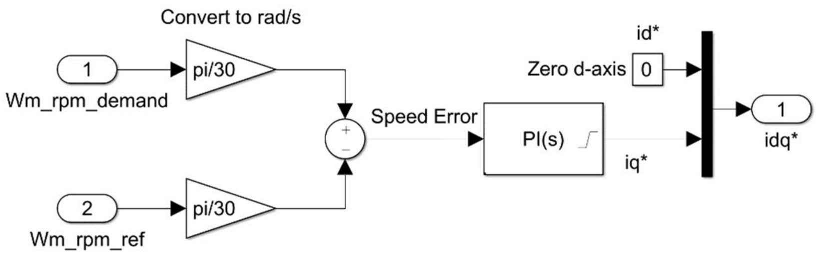


*Figure 15: Speed controller.*

A step change in speed demand can be modelled in Simulink by using the speed controller to set the dq-axis currents, shown in figure 16.

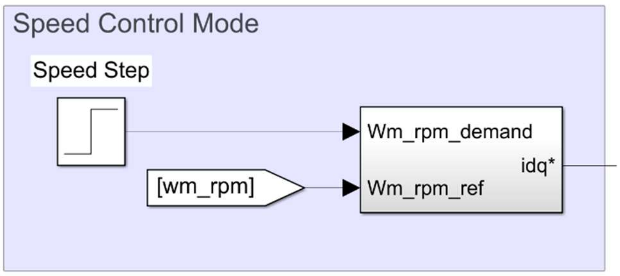


*Figure 16: Step change in speed demand.*

Using J = 1 and a load torque of 10Nm, the simulation results from a step change in speed demand from 1350-5000rpm is shown in figure 17.

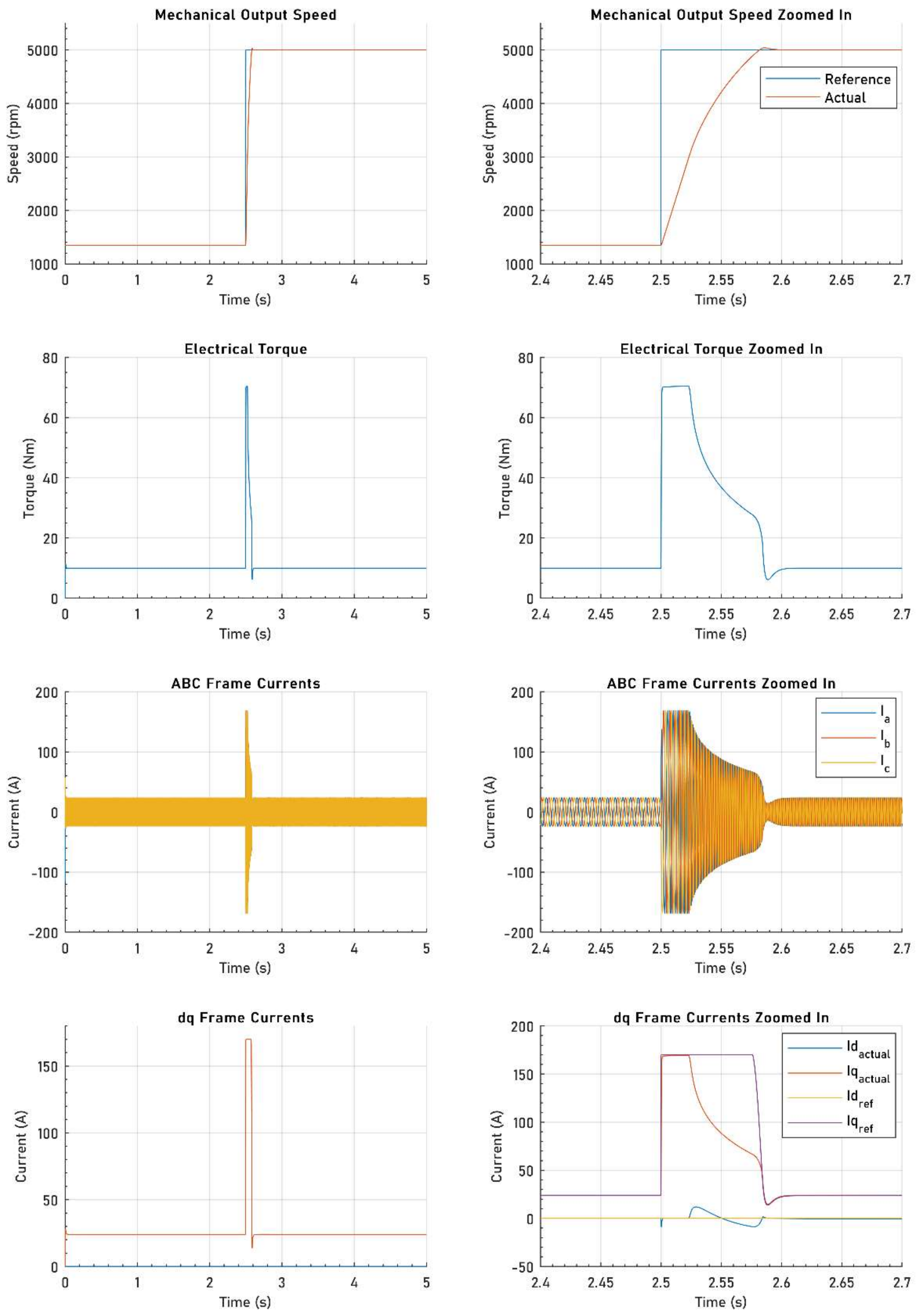


*Figure 17: Simulation results from step change in speed demand.*

As expected, the step change in speed demand causes a spike in the electromagnetic torque until the new speed is reached. The actual output speed follows the reference speed nicely with minimal overshoot and a rise time of 59.7ms.

The simulation results of a linear ramp in speed demand from 0 – 5500rpm (the maximum theoretical speed of the motor) is shown in figure 18.

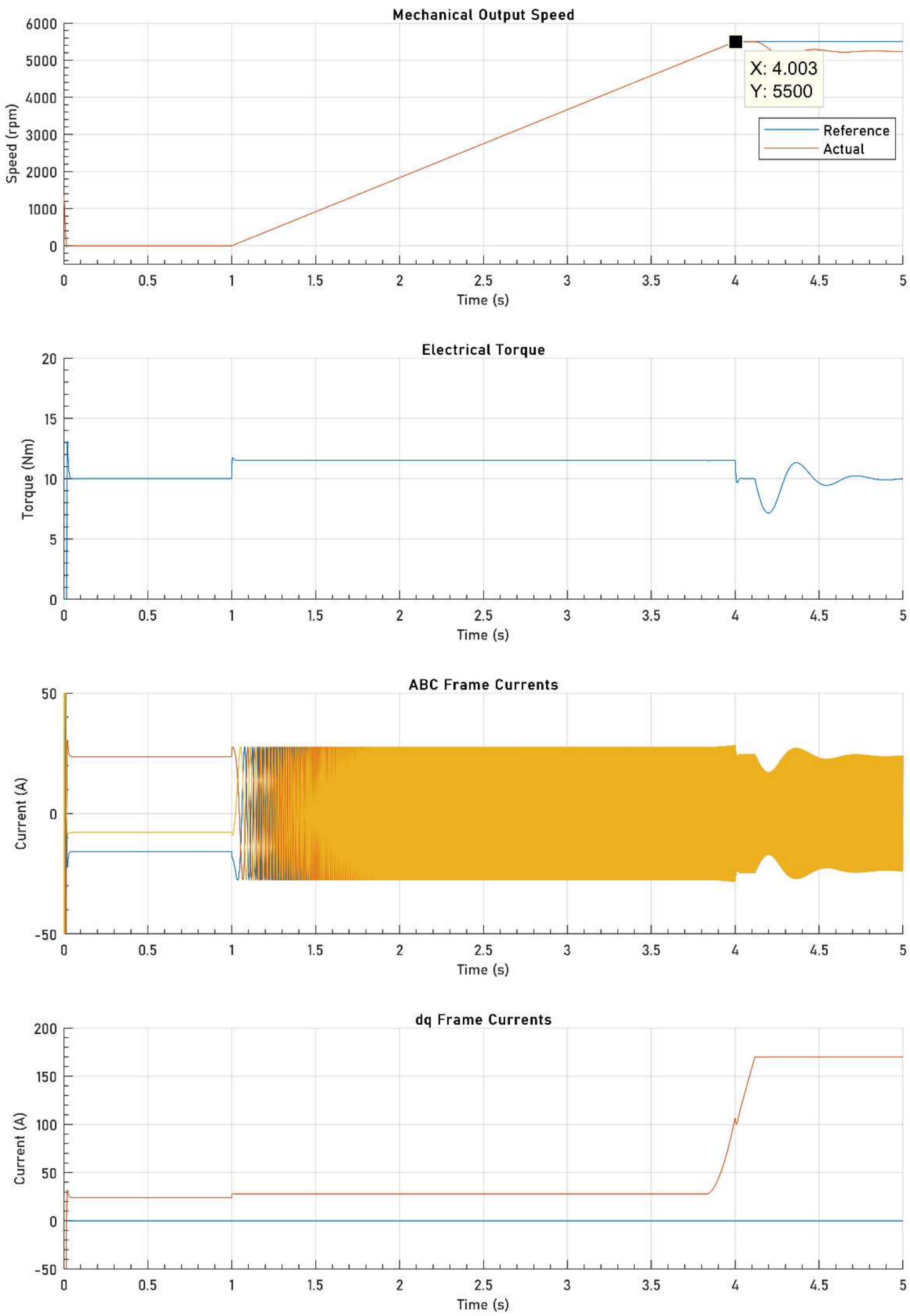


*Figure 18: Speed ramp results.*

As seen in the first graph, the speed tracks the reference speed from 0 – 5500rpm accurately, although the theoretical maximum mechanical speed cannot be maintained by the model, hence the strange behaviour once this speed is reached.

# 3. Summary

Overall, this report has covered the process and results of modelling and simulating a permanent-magnet brushless AC synchronous machine in the Simulink environment using sets of equations that describe the machine behavioural characteristics.

Firstly, the process of building an electrical and mechanical model of the machine was demonstrated based of a set of characteristic equations, including a discussion on the reasoning for applying the necessary axis transformations in an effort to simplify the model. This was followed by the design of a suitable current PI controller to output reference dq-axis voltages to achieved reference dq-axis currents. The reasoning and motivation to decouple the cross-coupling terms in the current controller was explained, and the process to implement decoupling into the Simulink model was demonstrated. Next, the windup phenomena attributed to integral controllers was explained, and a method to prevent it was designed and applied to the model.

Once the model was complete and working, testing was performed in both torque and speed control modes.

The torque control mode was tested by applying a step change in torque, which was converted to a step change in demand current using the torque constant of the machine. Initially, while the output torque was constant, the output speed of the motor linearly decreased because the load torque was greater than the electromagnetic torque being provided by the machine. However, when the step change in torque was applied, the torque provided by the machine exceeded the load torque, hence a linear increase in output speed was observed. The actual machine torque tracked the reference torque precisely, with a small rise time due to the characteristics of the machine, showing that the current controller performed well.

The speed control mode was tested by designing a PI controller that would output a reference dq-axis current based on a speed error signal. The required proportional and integral gains were extracted from the closed-loop transfer function of the generic speed controller block diagram and were fed into the Simulink model. The speed controller was first tested by applying a step change in reference speed from an initial operating speed. The actual output speed tracked the reference speed well with minimal overshoot and a short rise time due to the machine characteristics. As expected, the machine torque spiked whilst the machine was accelerating, but died down once the machine had reached the desired speed.

The speed control was further tested by performing a linear ramp in reference speed up to the theoretical maximum speed of the motor. The actual speed of the motor tracked the reference speed perfectly up to the maximum speed of 5500rpm, but the model was unable to maintain this speed for long. As expected, the electrical torque provided to achieve the linear ramp in speed was more or less constant.

In conclusion, the Simulink model designed of the machine operated as expected and achieved good performance in both torque and speed control modes.

# 4. References

[1] Microsemi, “Park, Inverse Park and Clark Transformations”, Microsemi, USA, CA, 2013. [Online]. Available: https://www.microsemi.com/document-portal/doc_view/132799-park-inverse-park-and-clarke-inverse-clarke-transformations-mss-software-implementation-user-guide

[2] S. Zhou, J. Liu, L. Zhou, Y. Zhang, “DQ Current Control of Voltage Source Converters with a Decoupling Method Based on Pre-processed Reference Current Feed-forward,” IEEE Transactions on Power Electronics, vol. 32, no. 11, pp. 8904-8905, Nov. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7812790

[3] Unknown Author, “Field Oriented Control (FOC) Made Ultra Simple.” Roboteq. https://www.roboteq.com/index.php/applications/100-how-to/359-field-oriented-control-foc-made-ultra-simple (Accessed: Dec. 18th, 2019)