UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Development of a test bench of electric generators

used in the context of renewable energy harnessing

Dissertation presented by
Ladislas D'HOOP DE SYNGHEM

for obtaining the Master's degree in
Electrical Engineering

Supervisor(s)
Emmanuel DE JAEGER

Reader(s)
Thierry DARAS, Bruno DEHEZ , Thomas MERCIER

Academic year 2016-2017

Abstract

In the global energy context, the use of renewable energy will become more and more predominating. One of the most mature technologies we have at our disposal is wind turbines. The optimisation of wind turbine controllers requires some plug and play systems in order to test new control ways. This experimental master thesis continues the work of others in order to develop a test bench of electrical machines. It contains a complete description of the control of a permanent magnet synchronous machine, starting from the machine equations and introducing the right mathematical tools in order to develop a robust field oriented control. A modelling part is explained as well as how to use the provided hardware interface. Finally, simulation results and physical measurements are presented to fully understand the behaviour of the machine.

Acknowledgements

My biggest thanks goes to the supervising team that brought this thesis to completion.

Professor Emmanuel De Jaeger, for his presence and guidance along the year, for his pertinent advices, and for helping me choose the right direction in my studies,

Thomas Mercier, that perfectly fulfilled his accompanying role, helping me day-by-day while being able to let me unravel situations to let me learn on my own,

Thierry Daras, that accompanied me during my journey in the lab, helping troubleshooting and providing technical support,

and Professor Emeritus Francis Labrique and Professor Bruno Dehez that took the time to answer my questions.

As this is the final work of my studies in electrical engineering, I would like to thank the teaching staff that made this redaction possible, in particular Professors Emmanuel De Jaeger and Bruno Dehez that gave the critical courses needed for this thesis.

I would also like to thank my family and especially my mother for being able to provide the environment, as well financially as psychologically, to let me complete my studies. My last thanks will go to my girlfriend that accompanied me as well in the last steps of this journey.

Contents

Nomenclature

$\bar{x}$	Vector x
δ	Distance between poles on the speed open loop response bode plot
ω_0	Unitary gain frequency of the speed loop
ω_c	Cut-off frequency of the current loop
ω_e	Electromechanical rotational speed, equals the mechanical rotational speed multiplied by the number of pole pairs
ω_s	Sampling pulsation, equals $2\pi f_s$
ω_{ref}	Speed setpoint
ψ_{pm}	Permanent magnet magnetic flux
ψ_{sa}	Phase a stator flux
ψ_{sb}	Phase b stator flux
ψ_{sc}	Phase c stator flux
ψ_{sd}	Direct component of the stator flux
ψ_{sq}	Quadrature component of the stator flux
τ	Time constant of the speed filter
$\|\bar{x}\|$	Norm of vector x
ξ	Position of the rotor magnetic axis with respect to the magnetic axis of phase a
f_s	Sampling frequency

I_{ref}	Current setpoint
I_{sa}	Phase a stator current
I_{sb}	Phase b stator current
I_{sc}	Phase c stator current
I_{sd}	Direct component of the stator current
I_{sq}	Quadrature component of the stator current
J	Rotor inertia
K_ω, $T_{i\omega}$	Coefficients of the speed PI controller with $C(s) = K_\omega \left(1 + \frac{1}{sT_{i\omega}}\right)$
k_s	Stator coefficient, a coefficient to compensate for the non-perfect sinusoidal shape of the magnetomotive force
K_v	Viscous friction coefficient
L_d	Direct component of the stator inductance
L_q	Quadrature component of the stator inductance
L_s	Stator inductance
L_z	Stator length
L_{s2}	Alternating term in the inductance value in the case of a salient pole machine
L_{xy}	Mutual inductance between phase x and y
n_s	Number of turns in the stator coils
P_0, P_2	Constant and alternating (only with salient poles) term of the magnetic permeance
$PMSM$	Permanent magnet synchronous machine
R_e	Stator radius
R_s	Stator resistance
T_r	Resisting torque
T_s	Sampling period

T_{em}	Electromechanical torque
V_{ref}	Voltage setpoint
V_{sa}	Phase a stator voltage
V_{sb}	Phase b stator voltage
V_{sc}	Phase c stator voltage
V_{sd}	Direct component of the stator voltage
V_{sq}	Quadrature component of the stator voltage
x_α, x_β	Components of x expressed in $\alpha\beta$ reference frame
x_a, x_b, x_c	Components of x expressed in abc reference frame
x_d, x_q	Components of x expressed in dq reference frame
K, T_i	Coefficients of the current PI controller with $C(s) = K\left(1 + \frac{1}{sT_i}\right)$
p	Number of pole pairs
PI controller	Controller with proportional and integral action
PWM	Pulse Width Modulation

Introduction

In the current global context of energy transition, it is more than ever important to improve the efficiency of renewable energy. With 153.7GW installed in Europe, and covering 10.4% of the European Union's electricity demand in 2016 [6], wind energy has an important share in that transition. Wind energy is a field that is in constant improvement, in the center of great research. It needs thus ways to test potential improvements.

Wind turbines use electrical machine in order to generate electricity. A good control of these electrical machines is key to have a good efficiency and energy production. There are many possible implementations including the use of gearboxes, power electronics and different electrical machines. An overview and comparison of the different implementations is given in [8].

It is in the scope of testing new controllers for renewable energy that the following test bench is realized. This thesis continues the work of Messrs. Bruno Chevalier and Hugo Templier [1]. In the test bench are two inverters, a DSpace interface, a synchronous machine and an asynchronous machine. More information can be found in appendix A.

While the previous work focused on the asynchronous machine control, this one will treat a field oriented control of the synchronous machine. The bigger picture is to have a complete test bench with each machine controlled and connected to the grid through a back-to-back converter.

This work will handle the machine control theory, beginning from the machine equations and will use the right mathematical tools in order to develop a controller. The first part will be dedicated to the machine theory, the equations and the development of the controller. The second part contains the models that were used for the simulations. The third part discusses the hardware specificities and what challenges the real world brings. Finally, we will discuss results with the simulation and reality in parallel in the fourth part.

Chapter 1

Machine control theory

Before jumping into the practical realization of the bench, we have to discuss some theory. In this chapter you will find the necessary theory to be able to understand from the equations of a permanent magnet synchronous machine all the control logic that has been developed in this thesis. The first section gives a short reminder about space vector theory, the next one contains the theoretical description of the machine. The last section will use these equations in order to derive a system from the point of view of linear control and develop the controllers.

1.1 Space vector theory

Before tearing into the equations of the machine, it is useful to recall a tool that will simplify them a lot. Considering we are in a three phase symmetrical system, the representation space of the variables of the machine is over dimensioned. We can use space vectors to reduce the dimension and perfectly represent those quantities using two degrees of freedom. In the case of an unbalanced system we would have the apparition of a third component, the zero sequence component. In fact, we can construct from three scalar quantities like the voltage, current or flux, a vector [10]:

$$\bar{x} = C\left(x_a + x_b e^{j2\pi/3} + x_c e^{j4\pi/3}\right) \tag{1.1}$$

Decomposing the vector in its real and imaginary part we have

$$\bar{x} = x_\alpha + j x_\beta \tag{1.2}$$

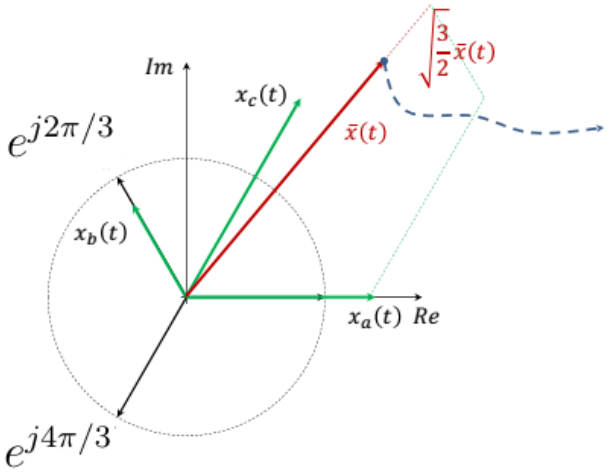

Figure 1.1: Representation of a space vector with $C = \sqrt{\frac{2}{3}}$ (adapted from [2])

Equating the real and imaginary part of both expressions we get

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = C \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{1.3}$$

If we take $C = \sqrt{\frac{2}{3}}$, we have a power invariant transformation, instead of amplitude invariant for $C = \frac{3}{2}$. Furthermore, taking the zero-sequence components into account, we have a square orthogonal matrix. Its transpose is thus the way to regain the abc components from the $\alpha\beta$, and is called the Concordia transform.

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = C \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{1.4}$$

We can go one step further in the transformation of our coordinates. For a three phase field rotating machine, it can be interesting to rotate the coordinates we have in the $\alpha\beta$ frame along with the rotor in a so called rotor frame. That way, we have quantities that are clear from the 50 Hz variations. We can write the rotation matrix along the rotor axis, with ξ being the rotor angle, i.e. the angle between the magnetic axis of the rotor and that of the

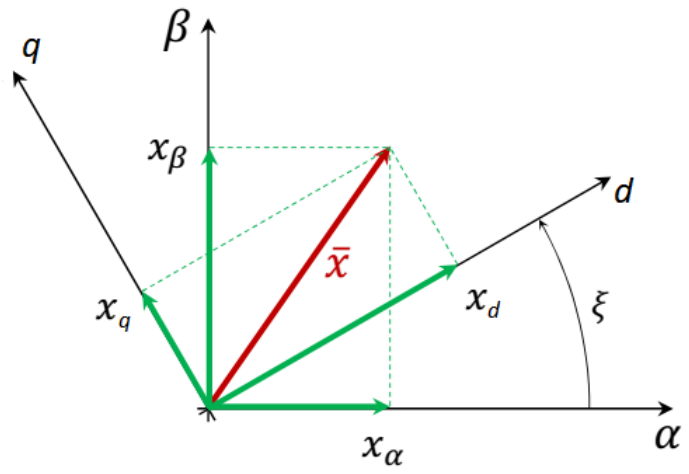

Figure 1.2: Representation of the dq-transform (adapted from [2])

phase a, if we take the phase a as reference (figure 1.2).

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} cos(\xi) & sin(\xi) \\ -sin(\xi) & cos(\xi) \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{1.5}$$

If we take the inverse rotation to go back to the $\alpha\beta$ frame, we have this:

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} cos(\xi) & -sin(\xi) \\ sin(\xi) & cos(\xi) \end{bmatrix} \begin{bmatrix} x_d \\ x_q \end{bmatrix} \tag{1.6}$$

This is also called the Park transform.

We can put everything together to go from the abc to the dq frame, meaning first from abc to $\alpha\beta$ and then pre multiply it by its rotation matrix we get what is called the dq-tranform. Everything from now on will be discussed in a dq-reference frame.

1.2 Machine equations

We will use these tools from previous section to derive a simple model describing the permanent magnet synchronous machine (PMSM) in the dq reference frame. Let's start with the electrical equations of the synchronous machine

with permanent magnets and without damper windings on the stator as there are no rotor equations:

$$V_{sa} = R_s I_{sa} + \frac{d\psi_{sa}}{dt} \tag{1.7}$$

$$V_{sb} = R_s I_{sb} + \frac{d\psi_{sb}}{dt} \tag{1.8}$$

$$V_{sc} = R_s I_{sc} + \frac{d\psi_{sc}}{dt} \tag{1.9}$$

The flux linkage equations are:

$$\psi_{sa} = L_{aa} I_{sa} + L_{ab} I_{sb} + L_{ac} I_{sc} + \psi_{pma} \tag{1.10}$$

$$\psi_{sb} = L_{ab} I_{sa} + L_{bb} I_{sb} + L_{bc} I_{sc} + \psi_{pmb} \tag{1.11}$$

$$\psi_{sc} = L_{ac} I_{sa} + L_{bc} I_{sb} + L_{cc} I_{sc} + \psi_{pmc} \tag{1.12}$$

We can develop the inductance expressions:

$$L_{aa} = L_s + L_{s2} cos\left(2\theta\right) \tag{1.13}$$

$$L_{bb} = L_s + L_{s2} cos\left(2\left(\theta - \frac{2\pi}{3}\right)\right) \tag{1.14}$$

$$L_{cc} = L_s + L_{s2} cos\left(2\left(\theta - \frac{4\pi}{3}\right)\right) \tag{1.15}$$

$$L_{ab} = -\frac{1}{2} L_s + L_{s2} cos\left(2\left(\theta - \frac{4\pi}{3}\right)\right) \tag{1.16}$$

$$L_{ac} = -\frac{1}{2} L_s + L_{s2} cos\left(2\left(\theta - \frac{2\pi}{3}\right)\right) \tag{1.17}$$

$$L_{bc} = -\frac{1}{2} L_s + L_{s2} cos\left(2\left(\theta\right)\right) \tag{1.18}$$

with $L_s = \frac{4}{\pi} L_z R_e k_s^2 n_s^2 P_0$ and $L_{s2} = \frac{4}{\pi} L_z R_e k_s^2 n_s^2 \frac{P_2}{2}$, L_z the stator length, R_e the radius, n_s the number of turns, k_s the stator coefficient, i.e. a coefficient to compensate for the non-perfect sinusoidal shape of the magnetomotive force and P_0 and P_2 the constant and alternating term of the magnetic permeance.

For the permanent magnet flux we have:

$$\psi_{pma} = \psi_{pm} cos\left(\theta\right) \tag{1.19}$$

$$\psi_{pmb} = \psi_{pm} cos\left(\theta - \frac{2\pi}{3}\right) \tag{1.20}$$

$$\psi_{pmc} = \psi_{pm} cos\left(\theta - \frac{4\pi}{3}\right) \tag{1.21}$$

Using the previously seen tools we can apply a transformation matrix which is the combination of the Park and Concordia transform, called the dq0 transform:

$$\begin{bmatrix} X_0 \\ X_d \\ X_q \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} cos\left(\theta\right) & cos\left(\theta - \frac{2\pi}{3}\right) & cos\left(\theta + \frac{2\pi}{3}\right) \\ -sin\left(\theta\right) & -sin\left(\theta - \frac{2\pi}{3}\right) & -sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \tag{1.22}$$

This will cause an apparition of a term $\frac{d\psi}{dt} = \frac{d\psi}{d\theta}\frac{d\theta}{dt}$ which finally gives the equations in dq reference frame of the PMSM:

$$V_{sd} = R_s I_{sd} + \frac{d\psi_{sd}}{dt} - \omega_e \psi_{sq} \tag{1.23}$$

$$V_{sq} = R_s I_{sq} + \frac{d\psi_{sq}}{dt} + \omega_e \psi_{sd} \tag{1.24}$$

with ω_e the electromechanical rotational speed, i.e. the mechanical speed multiplied by the number of pole pairs. We can develop the field expressions

$$\psi_{sd} = L_d I_{sd} + \psi_{dpm} \tag{1.25}$$

$$\psi_{sq} = L_q I_{sq} \tag{1.26}$$

with $L_d = \frac{3}{2}(Ls + L_{s2})$ and $L_q = \frac{3}{2}(Ls - L_{s2})$. As we are working on a round rotor PMSM, we have no alternating term in the magnetic permeance and thus $L_d = L_q = \frac{3}{2}L_s$. Note that $\psi_{dpm} = \sqrt{\frac{3}{2}}\psi_{pm}$ in the dq reference frame, with ψ_{pm} the permanent magnet flux. For the sake of clarity of the equations the d will not be mentioned and L_s will be written as such, as it is clear in which frame we work. We finally have:

$$V_{sd} = R_s I_{sd} + L_s \frac{dI_{sd}}{dt} - \omega_e L_s I_{sq} \tag{1.27}$$

$$V_{sq} = R_s I_{sq} + L_s \frac{dI_{sq}}{dt} + \omega_e (L_s I_{sd} + \psi_{pm}) \tag{1.28}$$

As for the torque we have

$$T_{em} = p(\psi_{sd} I_{sq} - \psi_{sq} I_{sd}) \tag{1.29}$$

With $L_d = L_q$ this reduces to

$$T_{em} = p\psi_{pm} I_{sq} \tag{1.30}$$

We can see that only I_{sq} has an effect on the torque, which is why I_{sd} is usually set at 0. It could be set to another value in the case of flux weakening

where it would act against the permanent magnet flux in order to reduce the induced electromotive force and increase the speed capability at the cost of a reduced torque. After having the expression of the torque we can express

the mechanical equation of the machine:

$$J\frac{d\omega}{dt} = T_{em} - T_{mec} \tag{1.31}$$

$$= T_{em} - K_v\omega - T_r \tag{1.32}$$

1.3 Control logic

We have seen in previous section that we can easily work in a reference frame where the quantities do not vary at the frequency of the machine. The controllers will act on the direct and quadrature components of the currents vector. This is called vector control or field oriented control. On figure 1.3 you will find the global control scheme, with the speed controller giving a quadrature current setpoint, the direct current being set at 0, the current controller and the mathematical transformations to work in the dq frame. We have thus two embedded loops, one controlling the speed and containing the current loop, and the other one receiving the current setpoint from the speed controller and controlling the current. The abc voltages are fed into a PWM inverter connected to the machine.

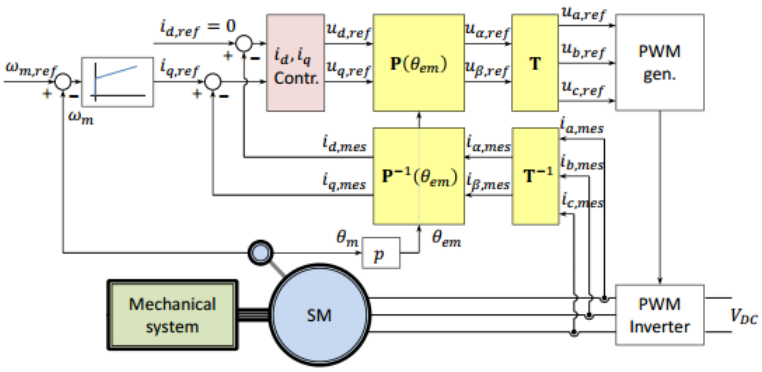

Figure 1.3: Global control scheme with speed controller and embedded current controllers (adapted from [2])

Current controller

A static error being unacceptable, it is necessary to use integral terms in the controllers. This is why PI controllers are discussed here. On figure 1.4, the control scheme is shown with the machine expressed in the dq frame.

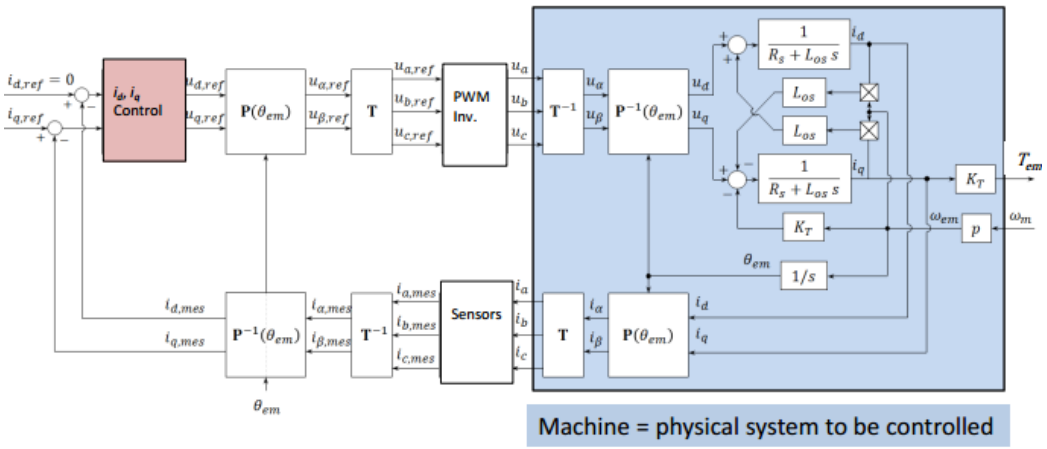

Figure 1.4: Current control scheme in dq frame [2]

In order to simplify the synthesis of the controllers, we can do a few hypotheses:

- The power electronics and measurement systems can be modelled with simple gains Gu and Gi.
- The mathematical transformations are done perfectly and instantaneously.
- Measurements are ideal.
- Parameters are perfectly estimated.

With those hypotheses, we can derive a simplified control scheme by decoupling the d and q axis and compensating the q-axis electromotive force for both d and q axis current controller. Those terms are added after the controllers. With the machine model in dq reference frame, we are then able to do some block algebra in order to cancel the compensating terms with internal coupling terms in the machine model, and doing so, remove the coupling between the two axes. Further information about that can be found

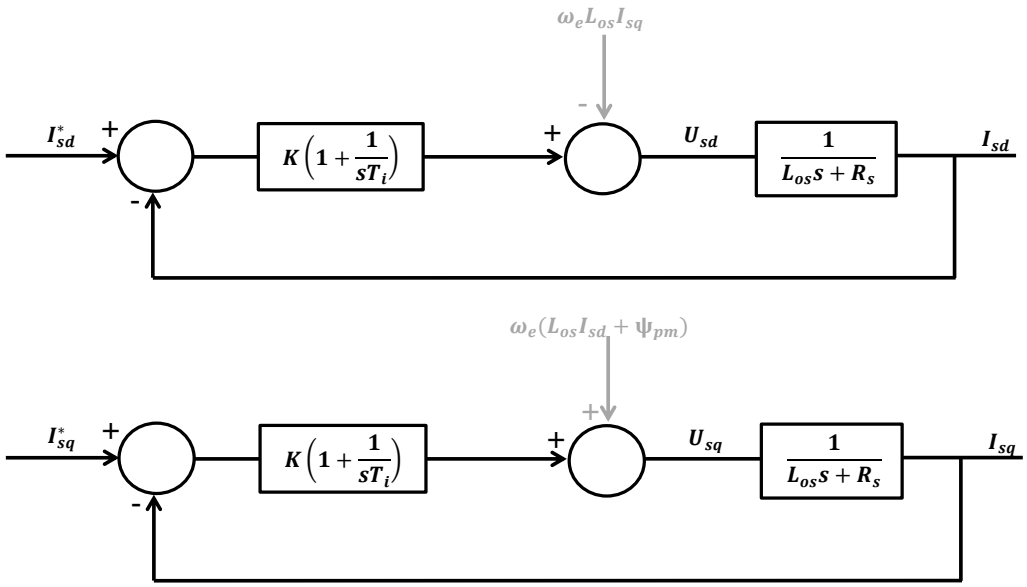

Figure 1.5: Simplified current control scheme for d and q axis

in [2]. Furthermore, we will make another valid assumption that allows us to treat current and speed controller independently: the current controller is embedded into the speed loop. We will have a speed feedback allowing to calculate an error, and that error will go into a PI controller. The PI controller will give a current setpoint that will be used for the q axis current controller. The current controller will give a voltage setpoint that will be applied to the machine, and we will use measurements of current and speed for the feedback. If we make the hypothesis that the current loop is much faster than the speed loop, we can synthesize them independently. The speed loop is much slower than the current loop, so we can consider the speed setpoint as constant while calculating the parameters of the current controller (figure 1.5). If we consider the current loop as much faster than the mechanical loop, we can consider that the current equals its setpoint. This is noted in figure 1.7. On figure 1.5 you will find the simplified control schemes for d

and q axis. The decoupling terms are shown in grey, as they should combine with terms inside the machine to give such a simple transfer function and thus not appear in this scheme, but considering they have to be added after the PI controller in reality, they are still mentioned.

We can write the complete transfer function between I_{ref} and I, valid for

both d and q axis:

$$
\begin{aligned}
i &= \frac{K\left(1+\frac{1}{sT_i}\right)}{R_s+sL_s}(i-i_{ref}) && (1.33)\\
\frac{i}{i_{ref}} &= \frac{\frac{K}{L_s}s+\frac{K}{L_sT_i}}{s^2+\frac{K+R_s}{L_s}s+\frac{K}{L_sT_i}} && (1.34)\\
&\approx \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} && (1.35)
\end{aligned}
$$

We can assimilate the system to a second order system without the zero. The observed dynamics should not be as good as what we impose on the controller with the effect of the zero. The slower the zero, i.e. the greater the T_i, the higher the effect of the zero on the system. The zero will have as consequence to increase the overshoot [5].

We can now equate the corresponding terms to find the relationship between natural frequency and damping factor and the parameters of the system. We have

$$
\begin{aligned}
K &= 2\zeta\omega_n L_s - R_s && (1.36)\\
T_i &= \frac{K}{L_s\omega_n^2} && (1.37)
\end{aligned}
$$

Another way to get rid of the problem of the zero is proposed in [1]. This procedure has been followed in our case. We can rewrite the transfer function like this:

$$
\frac{i}{i_{ref}} = \frac{sT_i+1}{\frac{T_iL_s}{K}s^2+T_i\frac{K+R_s}{K}s+1} \qquad (1.38)
$$

Now we can place our poles by equating the denominator of 1.38 with a generic form:

$$
\begin{aligned}
\frac{T_iL_s}{K}s^2+T_i\frac{K+R_s}{K}s+1 &= (1+As)(1+Bs) && (1.39)\\
&= 1+(A+B)s+ABs^2 && (1.40)
\end{aligned}
$$

That way we find

$$
\begin{aligned}
AB &= \frac{T_iL_s}{K} && (1.41)\\
A+B &= T_i+T_i\frac{R_s}{K} && (1.42)
\end{aligned}
$$

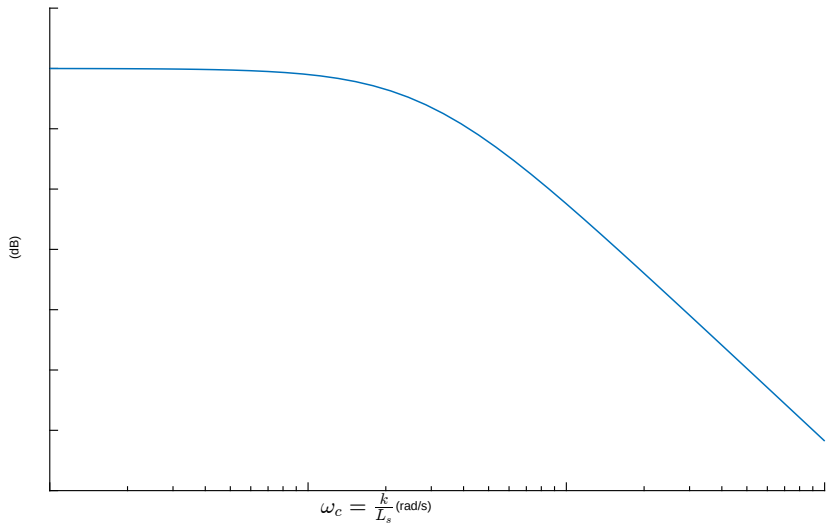

Figure 1.6: Bode diagram of $\frac{i}{i_{ref}}$, the transfer function of the current loop

If order to have a pole-zero cancellation, we impose

$$A = T_i \tag{1.43}$$

$$B = T_i \frac{R_s}{K} \tag{1.44}$$

$$\Rightarrow T_i = \frac{L_s}{R_s} \tag{1.45}$$

This gives us as transfer function a low pass filter with time constant $\frac{L_s}{K}$ or alternatively a cut-off frequency of $\frac{K}{2\pi L_s}$:

$$\frac{i}{i_{ref}} = \frac{1}{1 + s\frac{L_s}{K}} \tag{1.46}$$

We see that the bandwidth is directly proportional to the proportional constant (figure 1.6).

We will see after discussing the speed controller what the limits are to this parameter.

Speed controller

After the current controllers, we can do the speed controller. We already have the relation between T_{em} and I (equation 1.30). We can transpose the

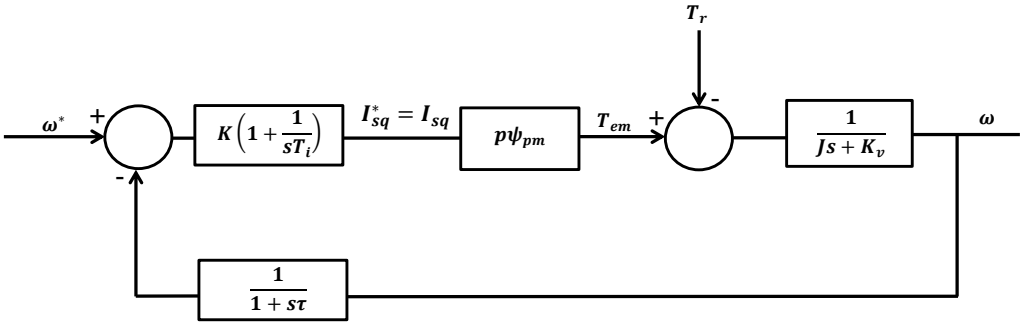

Figure 1.7: Simplified speed control scheme

mechanical equation (equation 1.31) into the Laplace domain:

$$Js\omega - K_v\omega = T_{em} - T_r \tag{1.47}$$

$$\frac{\omega}{T_{em} - T_r} = \frac{1}{Js + K_v} \tag{1.48}$$

Putting everything together in figure 1.7, we can compute the open loop response of the system. Note that the speed feedback is filtered, as this will be done in reality and have an influence on the dynamics.

$$OL(s) = K_\omega \left(1 + \frac{1}{T_{i_\omega} s}\right) (p\psi_{pm}) \left(\frac{1}{Js + K_v}\right) \left(\frac{1}{1 + s\tau}\right) \tag{1.49}$$

The viscous friction coefficient being nearly two orders of magnitude smaller than the inertia in our machine, we can neglect it. By regrouping everything together we have

$$OL(s) = \frac{p\psi_{pm}}{J} \frac{K_\omega}{T_{i_\omega}} \frac{1 + sT_{i_\omega}}{s^2(1 + s\tau)} \tag{1.50}$$

We have two poles at s=0, giving an attenuation of $40\frac{dB}{dec}$ in lower frequencies. For stable operation, the unity gain frequency ω_0 should be higher than the zero at $s = \frac{1}{T_{i_\omega}}$ and lower than the pole in $s = \frac{1}{\tau}$ [11]. Ideally, to achieve maximum phase margin, it should be right in the middle on the bode plot. After conversion from dB we have, with δ the distance on the bode plot (figure 1.8):

$$\omega_0 = \delta \frac{1}{T_{i_\omega}} \tag{1.51}$$

$$\frac{1}{\tau} = \delta\omega_0 \tag{1.52}$$

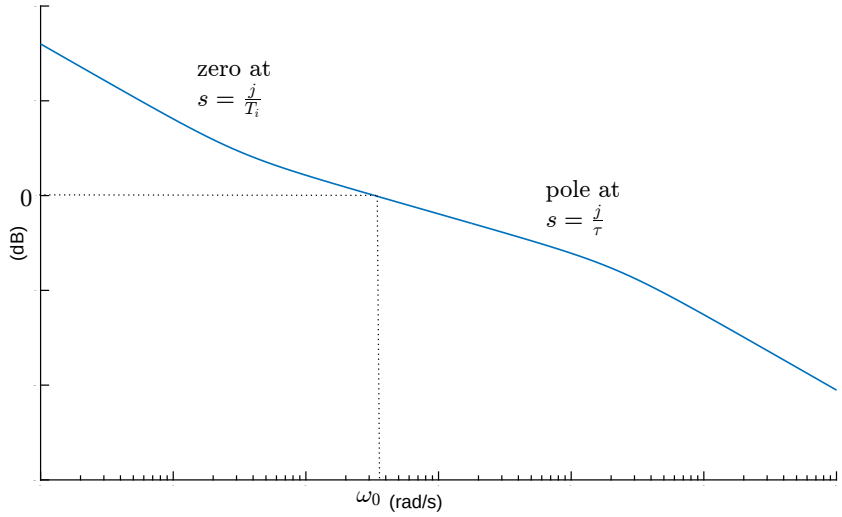

Figure 1.8: Bode diagram of OL(s), the open loop response of the speed loop

From there we can find an expression for T_{i_ω}

$$T_{i_\omega} = \delta^2 \tau \tag{1.53}$$

The meaning of δ is visible on figure 1.8: the higher the delta, the further apart the pole and the zero and the highest the phase margin can be. If $\delta = 1$, then we cancel the pole and the system has no stable region. To find the value of K_ω, we simply use the fact that the unity gain occurs at a frequency $\omega_0 = \frac{\delta}{T_{i_\omega}}$. By replacing it in the open loop response of the system and equating it to one, we can extract K_ω.

$$K_\omega = \frac{J}{\delta p \psi \tau} \tag{1.54}$$

We have seen that the system response depends on the placement of the zero of the integrator to have maximal phase margin, for a given response time. The δ parameter is a compromise between stability and speed.

Now that we have discussed the speed controller and found what values to give him, we can go back to the gain of the current controller. We had as result a low pass filter with the bandwidth determined by the gain. Ideally, we want a high bandwidth, to have fast responses. But a high bandwidth results on a stress on the machine, as it gives higher frequency current variations, that in extreme cases can even be heard from the machine. Let us see the limits of this parameter. Intuitively, we need to have a high bandwidth, but

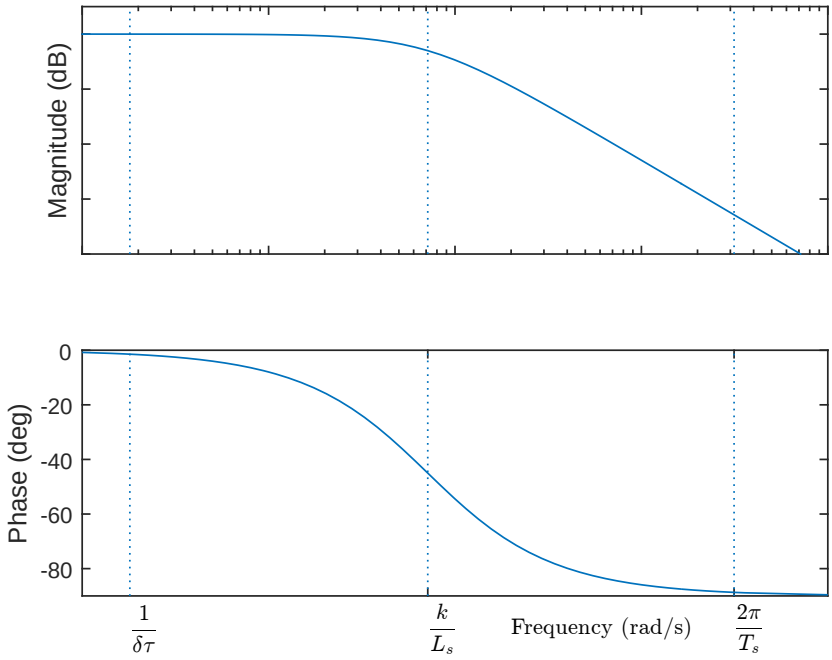

Figure 1.9: Bode diagram of $\frac{i}{i_{ref}}$ with the limits to its cut off frequency

not to high, and we are even limited by the hardware. Indeed, the Nyquist theorem tells us that we need to be at least at half the sampling frequency. A thumb rule says that if we are at maximum one tenth of it, everything should be fine [11]. We need to have

$$\omega_c < \frac{\omega_s}{10} \tag{1.55}$$

$$\frac{K}{L_s} < \frac{2\pi}{10T_s} \tag{1.56}$$

$$K < \frac{\pi L_s}{5T_s} \tag{1.57}$$

We also want to be one order of magnitude above the unitary gain frequency of the whole speed loop, to keep our assumption about the negligible influence of the current pole on the speed loop valid. Hence

$$10\omega_0 < \omega_c \tag{1.58}$$

$$\frac{10L_s}{\delta\tau} < K \tag{1.59}$$

In practice, as long as the stress is not too high on the machine, the highest K is chosen so that we have the highest response speed.

To summarize, the cut off frequency –actually its pulsation $\frac{k}{L_s}$ – of the current loop is clamped between two values (figure 1.9). On the left we have the speed filter pole $\frac{1}{\delta\tau}$, which we don't want to interfere with to keep the synthesis of the two loops independent, and on the right we have the sampling pulsation $\frac{2\pi}{T_s}$, from where we have to keep a reasonable distance to avoid unexpected effects due to the sampling.

We can then also study the influence of a resisting torque perturbation on the system, as seen on figure 1.7:

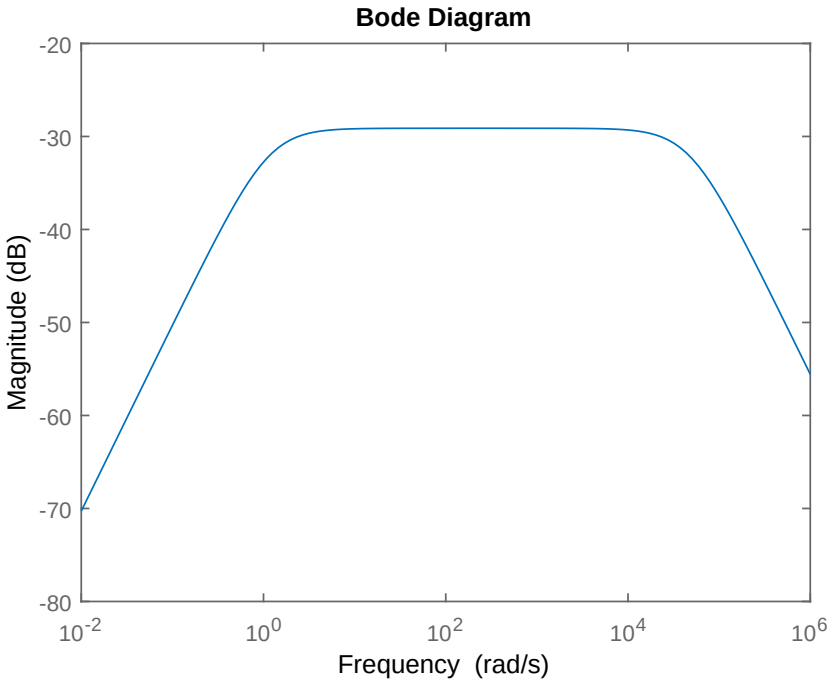

Figure 1.10: Bode diagram of $\frac{\omega}{T_r}$, the transfer function between torque perturbation and speed

$$\frac{\omega}{T_r} = \frac{s}{Js^2 + (p\psi_{pm}K + K_v)s + \frac{p\psi_{pm}K}{T_i}}$$

On figure 1.10 we can see that with the above-mentioned conditions the torque perturbations can be correctly attenuated.

Chapter 2

Numerical modelling

In this section we will discuss the models of the machine using Simulink and a specific library called Simscape - SimPowerSystems. This library contains models for power electronics, machines, sources, grid components, and so on. The whole chapter will discuss how different parts of the system were modelled.

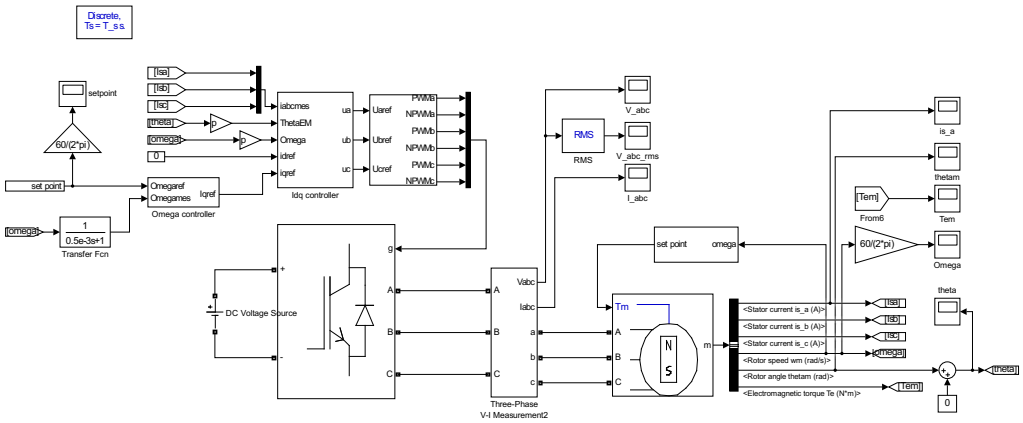

Figure 2.1: Main block of model used for simulation

On figure 2.1 you see the main block of the model. It contains the blocks for the physical elements like (from left to right) the DC source, the inverter, and the synchronous machine. On top of that you have simulink blocks that will do the control. From left to right you have the speed setpoint, the speed controller giving a setpoint to the current controller, the current controller will generate a voltage that will be converted into pulses for the switches of

the inverter through a PWM block. On the right of the power chain you have the measure blocks.

2.1 Speed controller

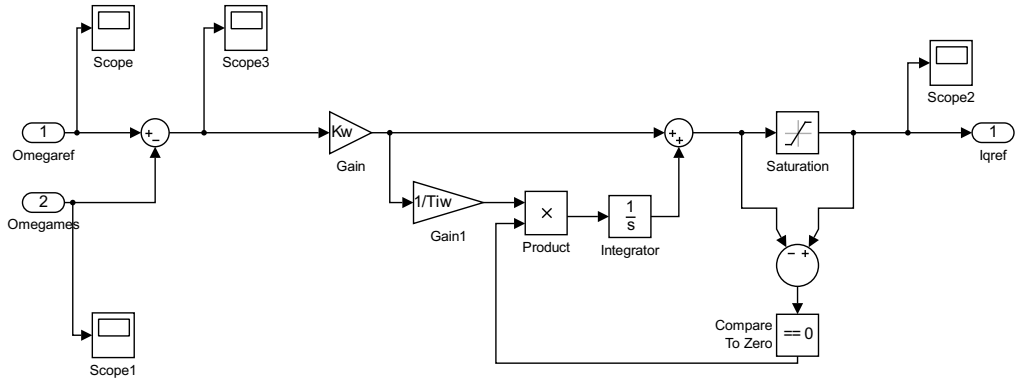

Figure 2.2: Speed controller block

The speed controller will take a speed setpoint and give a q axis current setpoint. The schematics is similar to the left part of figure 1.7. What is added before entering the next controller is a limitation to not damage the machine. This is the purpose of the saturation block. If we ask to much in the controller, the saturation will limit the output to guarantee a safe use. The use of a saturation raises a problem: if we are in saturation, the setpoint before the saturation will not stop to increase, due to the integral term of the PI. This is called windup. If now the asked setpoint is decreasing, before really decreasing the output, the integrator will have to un-integrate, i.e. lower its value before having a value corresponding to the setpoint, inducing a delay. This effect is counteracted by the clamping anti windup: we compare the values before and after the saturation, and if they are not equal, it means we are in saturation. Then we just multiply the error by zero in front of the integrator so it does not integrate. The values will be computed in 3.2

2.2 Current controller

The current controller is the most complex block of the simulation. It contains the implementation of the controller developed in 1.3. The simulink schematic contains the left part of figure 1.5, with the PI controller and the

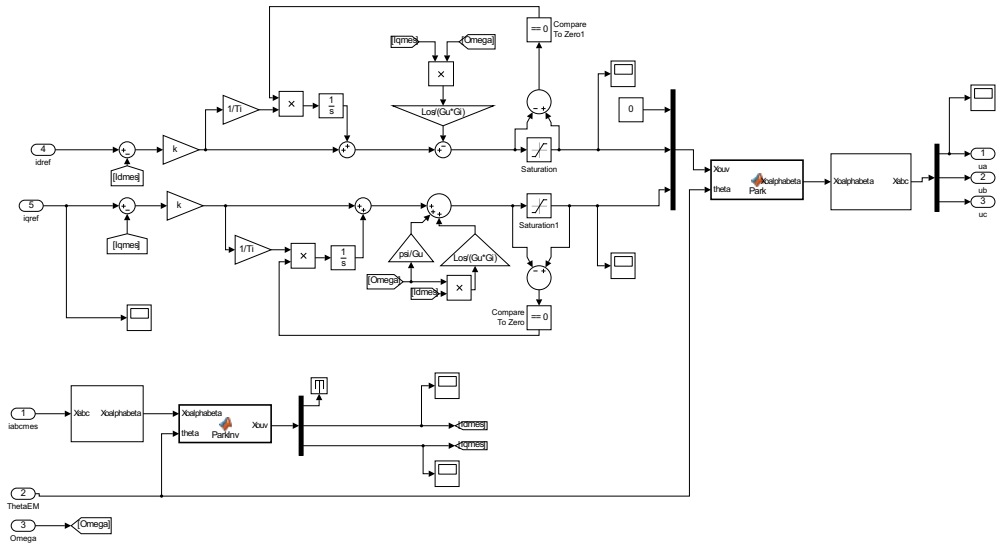

Figure 2.3: Current control block

compensation terms. On the top left you have the two entries of the d and q axis controllers. Then follows a PI on both axis with again a multiplication by zero in case of saturation from the anti windup. Then we add or subtract the compensation terms and we find again the saturations with anti windup. This gives a voltage setpoint in dq components. To apply them to the machine, we have to convert them back to the abc frame using Park's and Concordia's transforms. The blocks below are there to convert the current measurements into the dq frame.

2.3 PWM bloc

The last block before going into the inverter is the PWM block. It converts the abc values into pulses that will command the IGBT of the inverters. In order to do so we will use intersective pulse width modulation. We compare the value to apply to a triangular signal. If we are above, the output is 1, if we are below, the output is 0 2.4. These are the pulses to control the IGBT. U_{ref} having $U_{dc}/2$ as amplitude, we multiply the triangle by that to be able to go from 0 to 100% of the inverter capability. The difference is then compared to 0 in order to have the signal of opening and closing of one arm's two IGBT's. These signals are fed to the inverter.

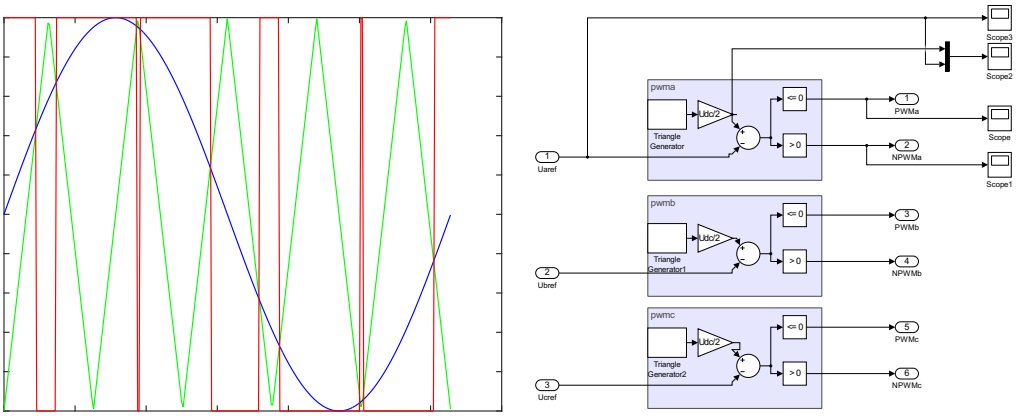

Figure 2.4: Illustration of intersective PWM modulation, in green the triangle signal, in blue the modulating signal and in red the output

Chapter 3

Hardware description and practical implementation

Once the simulations worked, it was time to implement the controller on the real machine. In this chapter, we will see through the application of a controller on a real permanent magnet synchronous machine. The first section will explain the hardware. We will then see how to compute the saturation values and the differences and difficulties that will be caused by working in the real world using DSpace and Simulink. One great difference is the measurement of the rotor angle, that is discussed in the next section. We will then detail the use DSpace and DSControlDesk as an interface to control the machine.

3.1 Hardware description

The modelled controller has to be implemented on the testbench. The machine used is a permanent magnet synchronous machine, Alsthom LC320TJ. It was supposed to be used as a brushless DC motor, so the nameplate contains equivalent DC values, as measured from the DC side of the inverter. It has three pair of poles and the windings are connected in star. Its nameplate follows here:

Made in FRANCE by Alsthom	
ABB	
Brushless Servo Motor	
Type LC620TJ R0010	
I.Cl. F	IP 65
Mn	5 Nm
In 10.8 A	Un 310 V
Nn	3900 min^{-1}
Tacho K_E	2 mV/min^{-1}

This machine is connected with one inverter, fed by a DC supply and controlled through the DSpace. The rotor position is measured by an absolute encoder.

3.2 DSpace and Simulink

One of the firsts differences between the simulation and the real world is the risk to cause damage to expensive material. This is why some precautions have been taken. The different quantities are protected by saturation, like we saw in section 2:

Speed The speed input has been locked beneath the maximum speed of the machine, i.e. 3900RPM.

Current The output of the speed controller is protected with a saturation and anti wind-up. The maximal DC current mentioned on the datasheet is 10.8A. In a brushless DC motor, this current is applied on a phase during 120^o, then 0 for 60^o, then -10.8A for 120^o and then 0 again for the remaining 60^o. The nominal RMS value of a sinusoidal current being the same as the square current applied, we have an AC value of $i = \sqrt{\frac{2}{3}}I = 8.82A$. Transposing this in the DQ frame it gives us the value set for the q axis saturation $|\bar{I}| = \sqrt{2}i = 12.47A$, as the d axis does not require current.

Voltage The voltage output of the d and q current controllers are also protected, but not in a symmetrical way. The DC value given is 310V. It is the peak value of the line voltage averaged on the $[-30^o, +30^o]$ interval from its maximum at nominal speed. This means we have $V_{\phi\phi} = \frac{\pi}{3\sqrt{2}}310 = 229.55V$. To have the amplitude of the voltage vector we take the peak value, like for the current $|\bar{V}| = \sqrt{2} * 229.55 = 324.63V$.

Now we need some voltage in the d axis in order to correctly monitor the current. To do this we will take a look at the machine equations (equation 1.27), neglect the resistances and suppose we are at constant nominal speed and I_{sd} is regulated at 0. Then we have

$$V_{sd} = -\omega L_s I_{sq} \quad (3.1)$$
$$V_{sq} = \omega \psi_{pm} \quad (3.2)$$
$$|\bar{V}|^2 = V_{sd}^2 + V_{sq}^2 \quad (3.3)$$

Entering the maximum current value and the machine data, we find a ratio $\frac{V_{sd}}{V_{sq}} \approx 0.1$. With the maximum voltage constraint we can find a value for $V_{sd} = 32.3V$ and $V_{sq} = 323.02V$

Another problem came with the speed filter, and caused a lot of debugging. The encoder gives a measure of the angle, which is differentiated to have a speed measure. That signal is filtered using a moving average. That moving average will introduce a great lag in the loop, impeding the speed controller's maximum speed. Originally at 100 points, it has been set to 10 points. Another factor that may cause trouble is the hardware limiting the sampling frequency. We saw in 1.3 that the current bandwidth is limited by the sampling frequency. In our case, a sampling frequency of $5kHz$ limited the current bandwidth to $500Hz$.

The DSpace allowed to create a simulink scheme for the whole software control that was directly interfaced with the hardware. This needed specific blocks from DSpace. The schematics is pretty much the same as what we had in simulation, only its signals are directly fed into the inverters and the measurements come from ADC's. More information about this can be found in appendix A. Some differences were made in order to ease the real time control of some variables, e.g. the controller parameters, a reset for the integrators and so on. The global schematic is found on figure 3.1.

3.3 Determination of the position of the rotor

In the simulation world, having a precise measure of the angle was something straightforward. A wire coming out of the Simulink machine block gave the exact position of the rotor. But in the real world, with the use of the absolute encoder at our disposal, one additional problem made its way to my mind. We had a precise and clean measure of the angle, but who said that the 0 value of the encoder matched the 0 value my controller needed ? In fact,

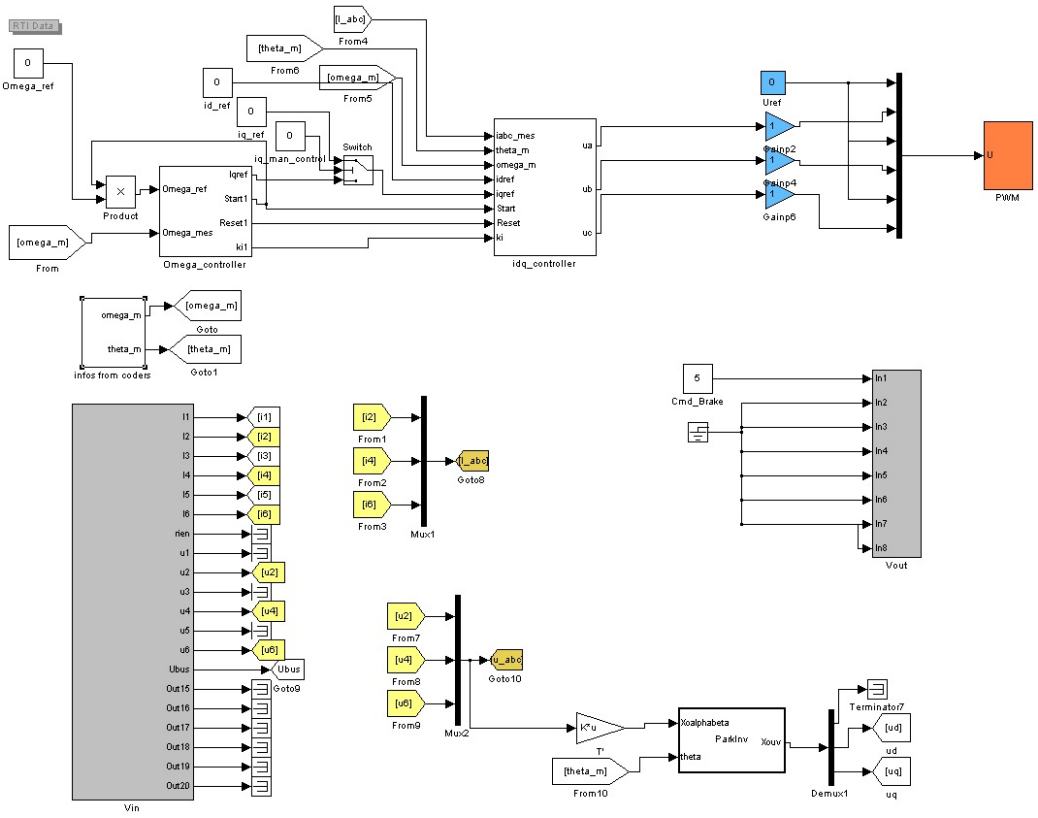

Figure 3.1: Global simulink schematics

there is a small block in the interface made by François Baudart that gives an offset to the value to make sure we can match the two. The question is: How can we match them? How can we precisely know that our reference for the dq-transform ξ_s (figure 3.2) will be the same as the reference for the encoder?

One thing to mention is the fact that the encoder is on 13 bits, meaning that we have a resolution of 0.044^o approximately and a maximal measure error of 0.022^o. On the following table 3.1 you will find a correspondence of encoder values and degrees.

In practice, we can try to put a DC source on one of the stator coils of the machine, causing a current and a magnetic field that will align the magnet with the created field. That way, we have the certainty that the magnet has a defined position in the machine and we can adapt our offset accordingly. While putting the plus terminal on one phase, and the minus terminal on

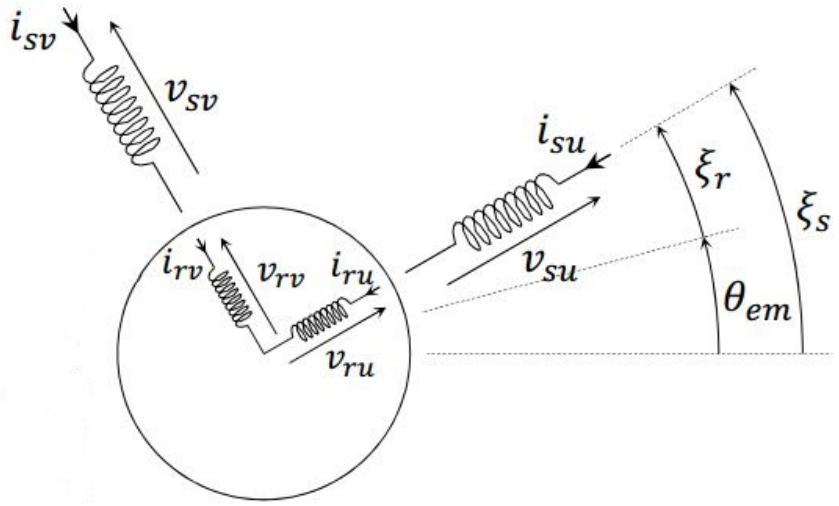

Figure 3.2: Illustration of the dq-transform [2]

deg	360^o	180^o	90^o	60^o	45^o	30^o
rad	2π	π	$\pi/2$	$\pi/3$	$\pi/4$	$\pi/6$
enc	8192	4096	2048	1365.33	1024	682.67

Table 3.1: Correspondence between encoder measures and values

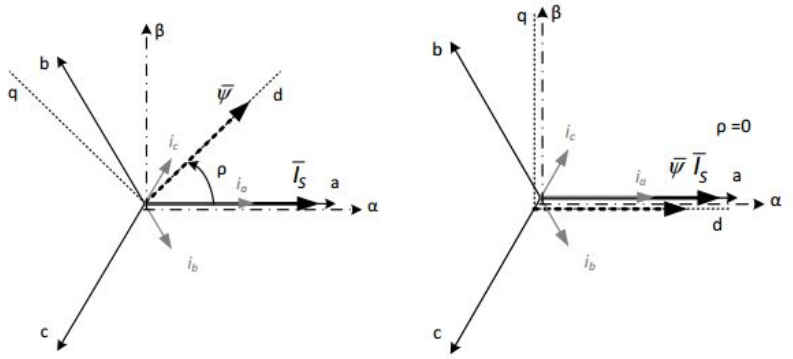

Figure 3.3: Alignment using a DC source[9]

a	b	c	θ
+	-	-	0^o
+	+	-	20^o
-	+	-	40^o
-	+	+	60^o
-	-	+	80^o
+	-	+	100^o

a	b	c	θ		
+	-	-	1461	4192	6922
+	+	-	1898	4628	7355
-	+	-	2368	5090	7820
-	+	+	81	2815	5550
-	-	+	530	3260	6000
+	-	+	990	3725	6455

Table 3.2: Simulations (left) and measurements (right) made with a DC source between phases

the two others, some field cancellation should occur, aligning the rotor on one phase. In fact, it is exactly what happens (figure 3.3). If we put the a phase at the plus, we will have a positive current, and with phase b and c at the negative terminal, a negative current. This cancels the every component that is not aligned with the a phase, giving the value of the 0 angle.

On table 3.2, you will find the values to which the rotor converges in simulation and in the reality. A 5V DC source has been applied between the terminals of the machine, + denoting the positive terminal and - the negative one. We see that the machine converges towards three different values, because it has three pole pairs. One value that could work as offset is 4192, or the two others at 0^o. We can distinctly see the offset of nearly 455 denoting 20 degrees between the different positions, as the offset of 2731, meaning 120 degrees to go from one pole pair to the other.

3.4 ControlDesk

With the simulink locked and loaded, we have to control the real-time program running on the DSpace. This is done using DSpace Control Desk. After importing the variable file generated with MATLAB, one can build a layout file to have its own interface. On the given interface figure 3.4, one can easily interact with the machine.

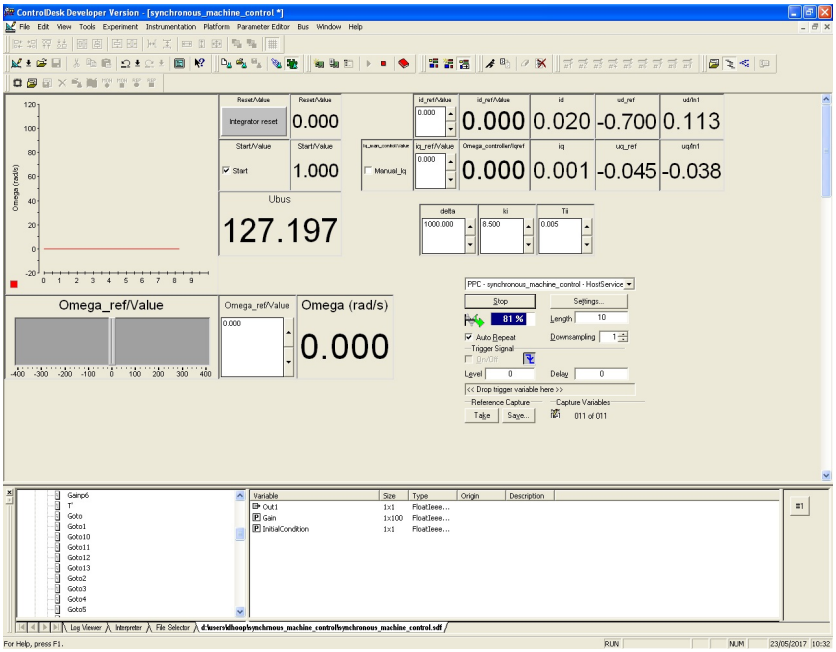

Figure 3.4: Control Desk interface

On the left you have the main display, with the speed in a graph and as numerical value. You can set it using the slider or the numerical value to set a precise value. There is a reset button should you cut off the voltage or the inverters while trying to impose a speed, so that we can reset the integrators. There is also a start value, to keep the speed at zero and prevent the integrators to integrate while not ready. Once everything is ready the start box can be checked. On the upper right corner you have two lines of values, one for the d and one for the q axis, with the current setpoint, the current, the voltage setpoint and the voltage. On the left of the line you can choose to impose a custom d axis current, or overwrite the speed controller value and give yourself a q axis current (torque) value. In order to do that, you have to check the box *manual_iq*. Beneath this you have the values of

the controllers, delta commanding the proportional and integral gains of the speed controller, and the two others for the current loop. The values we see now were used for a unloaded test and are not suited for a normal use. On the bottom right you have the capture window, where you can specify the capture length for your virtual instruments, start or stop and save the data in *.mat* files. Below all this, and not shown on this figure are some detailed plots of quantities and other debug controls.

Chapter 4

Results and discussion

It is time to discuss the final results of this work. A few tests have been made to validate the developed controller. First, some tests were made at no load: a speed step, a speed inversion and a low speed run. Then, we will see the machine reaction if we put a torque step while running. Finally, we will change the control parameters to study its influence. All the measurements are put in parallel with simulation results.

4.1 Test of the controller

In this test, the controller parameters have been kept constant, and the machine parameters are the right ones. We will examine the machine behaviour in different conditions. It is worth mentioning that the controller was developed in simulation and tested at first in a no-load situation. Even the coupling was removed from the machine, giving only the rotor inertia and internal friction as system parameters. The measurements have been made with the machine coupled to another PMSM, while the mechanical parameters have been taken from [1], in a situation close yet not equal to that one.

4.1.1 Speed step

The first way to quantify the speed controller is of course the step response. On figure 4.1 you can find the first test, asking the machine to go at 200 rad/s (approximately 1900 RPM) in one direction, in the opposite direction and then at standstill.

We can study the response from a little closer in simulation and reality. On figure 4.2 we can see a zoom on the speed step. We can deduce the

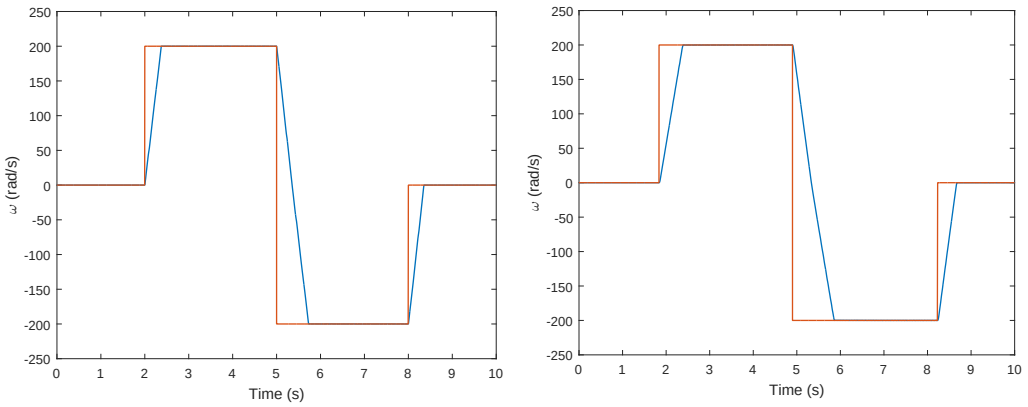

Figure 4.1: Global speed response of the system with the simulation (left) and the measurements (right) – Setpoint in red, response in blue

response time of the simulated system and in reality. It seems the simulation gives a smaller rise time of 90% approximately 350ms, while the real machine does it in half a second. This can be explained by a poor estimation of the rotor inertia, like said in the section introduction.

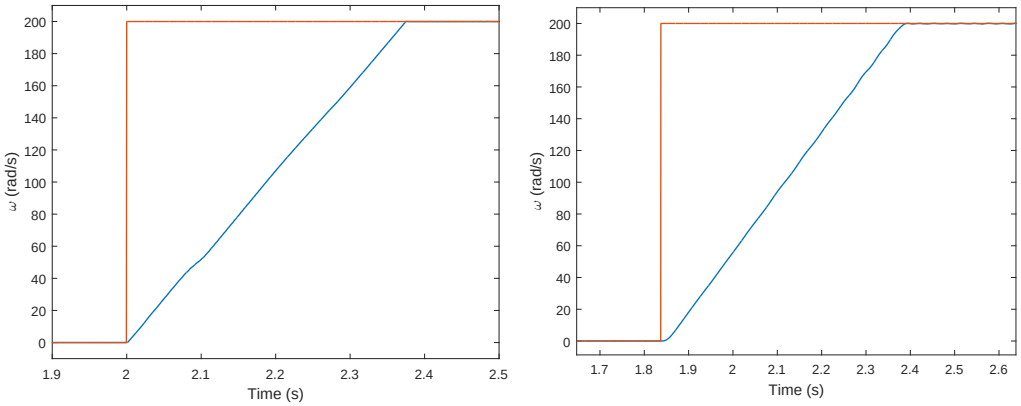

Figure 4.2: Zoom on the step response of the system with the simulation (left) and the measurements (right) – Setpoint in red, response in blue

The main limitation of the response of the system is not the controller, but rather the current limitation. It would be conceivable to put higher values than the nominal current for short times without overheating, as long as we respect the manufacturer's datasheet, to achieve better performances. There is no overshoot as the system is overdamped.

Another aspect to focus on is the steady state speed. On figure 4.3 we

can see a zoom of the speed curve. There are some oscillations in the simulation that are probably caused by the solver. In the measurements on the other hand, there is a phenomenon that is not seen in the simulation, with a distinct pattern. These oscillations are certainly caused by a bad alignment of the two machines or a rotor unbalance. They have an amplitude of 0.2 rad/s which gives 0.1% of the speed, it seems thus fairly reasonable. As for the frequency, we can count the number of periods in a time window and see if it is coherent with the machine speed. There is clearly one spectral component that dominates. Between times t=3s and t=4s, we have approximately 32 periods of that component. Now, if the origin of the oscillations is effectively a rotor unbalance or a bad alignment of the machines, we should find the rotational frequency as oscillation frequency. If we divide 200 rad/s by 2πrad/rotation we obtain 31.8 rotations/s, which is close enough to the 32 we counted.

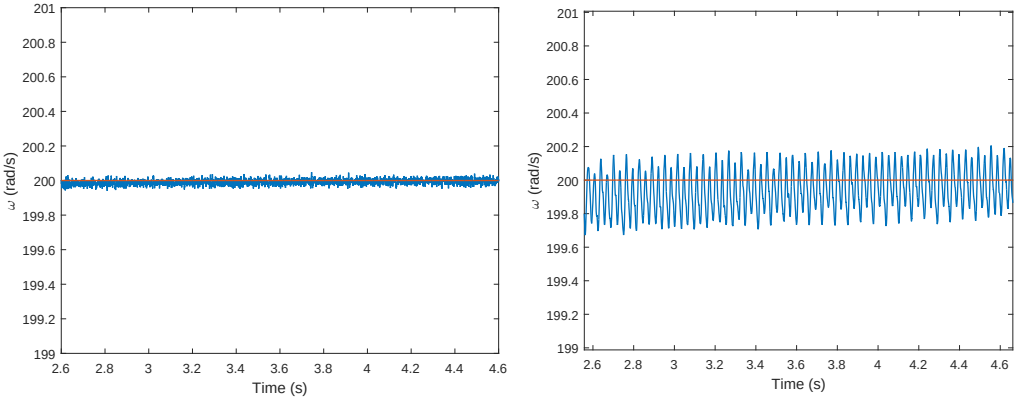

Figure 4.3: Steady state operating with the simulation (left) and the measurements (right) – Setpoint in red, response in blue

It is extremely complex to quantify the phenomenon and simulate it, but it is however possible to confirm this explanation by putting a oscillating resisting torque that would emulate the phenomenon, with a frequency equal to the machine rotational speed. On figure 4.4, the simulation has been rerun with a resisting torque of $T = 0.5sin(\omega t)$, and we can clearly see the apparition of the same phenomenon. As the quantification of the phenomenon is complex, no exact value has been found to match the same response. The rest of the simulations is done with this resisting torque.

Another quantity interesting to see is the q axis current, as it is directly proportional to the torque. We have on figure 4.5 the simulation and mea-

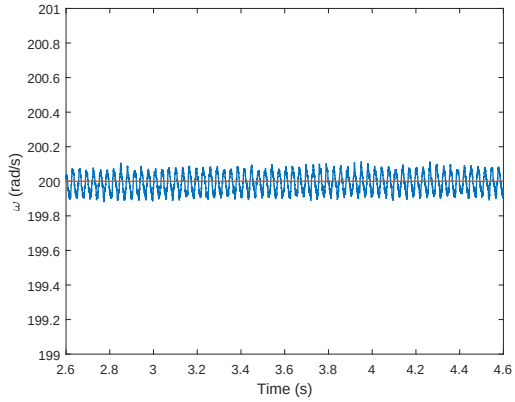

Figure 4.4: Simulation of steady state operating with the emulation of bad rotor alignment – Setpoint in red, response in blue

surement of the q axis current. We see that during the acceleration phase it is stuck at its maximal value, then it has a positive value on average to maintain the speed constant, but with quite large oscillations to counterbalance the bad alignment of the machines, lower in the simulation because the emulated effect is lower than the real one, as we can see on figures 4.3 and 4.4. Then during the deceleration and inversion the current is at the lower bound to develop the maximal opposite torque, then, same as before, keep the speed constant but with the oscillations. In the last part, the speed was null, yet we have some short current spikes even though the machine is not rotating apparently. It has moved for 0.005 radians in the last two seconds, which corresponds to 0.0025 rad/s. This is probably a residual static error that is decreasing thanks to the integral term. The presence of such spikes in order to completely decrease the value to zero might indicate that the controller is too aggressive.

Like the torque is an image of the q axis current, the speed is an image of the d axis voltage. Indeed, if we take a look at the machine equations (equation 1.27), while neglecting stator resistance and transients, we have a proportionality between V_{sd} and ω.

On figure 4.6 we have the values for V_{sd}. Indeed we have low voltage at low speed and high voltage at high speed on average, but there are some monstrous oscillations. Looking at the equations, we have a $-\omega L_s I_{sq}$ term, with the bad alignment being responsible for torque oscillations, giving I_{sq} oscillations, but the $L_s \frac{dI_{sd}}{dt}$ term has an influence too, as they cannot come from the $R_s I_{sd}$ term. One way to convince yourself is to remove the oscillating torque from the simulation and do another plot of the voltage.

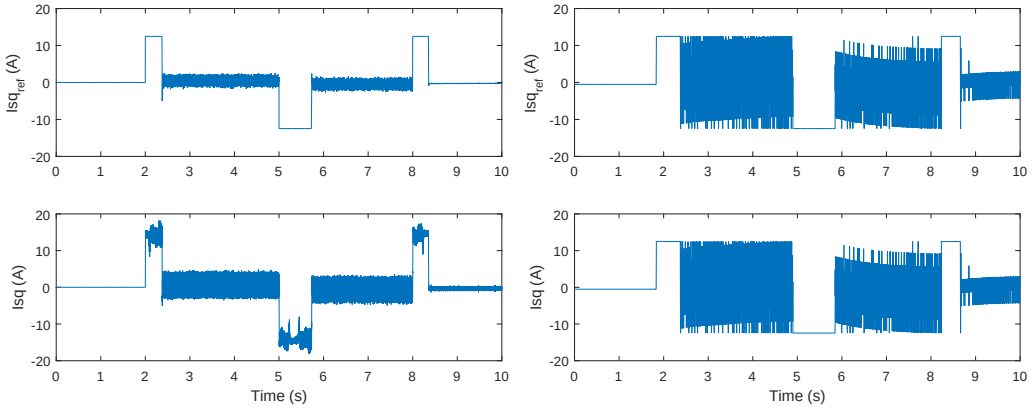

Figure 4.5: Q axis current with the simulation (left) and the measurements (right)

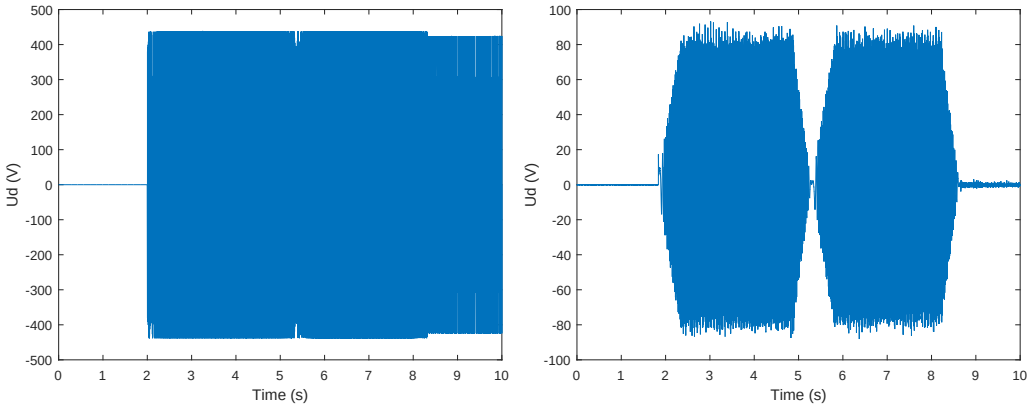

Figure 4.6: D axis voltage with the simulation (left) and measurements (right)

On figure 4.7 left, you can see that the oscillations are still present even without the oscillating torque. It comes most probably from the power electronics that cause quick transients on the d axis current, giving a high derivative and thus a high voltage. On figure 4.7 right, you have a simulation with an ideal controlled voltage source instead of the PMW inverter. We have a result much closer to the reality but some oscillations remain.

We can take a look at what is happening during an acceleration in one phase of the machine (figure 4.8). We clearly see the frequency for both voltage and current increasing as the machine speeds up. The voltage increases gradually with the speed and is maintained after the acceleration. The current is at its maximal value during the acceleration then reduces to keep the

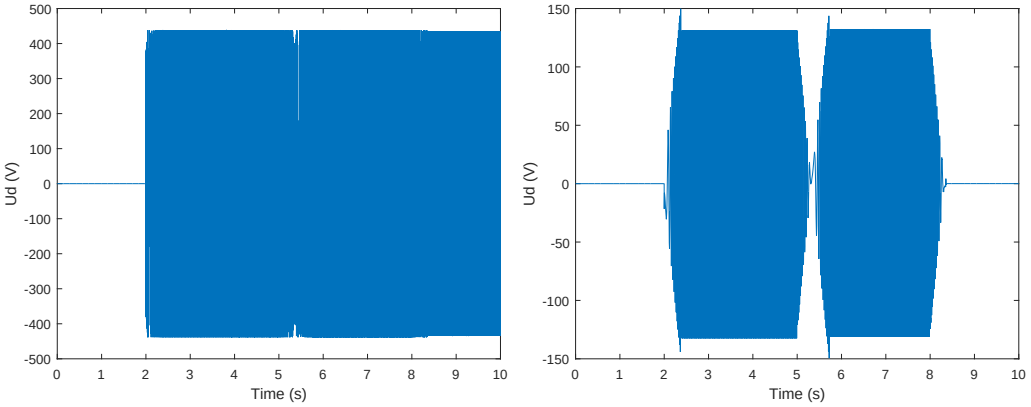

Figure 4.7: D axis voltage from simulation without the oscillating torque (left) and with an ideal voltage source (right)

speed constant and compensate for the oscillations.

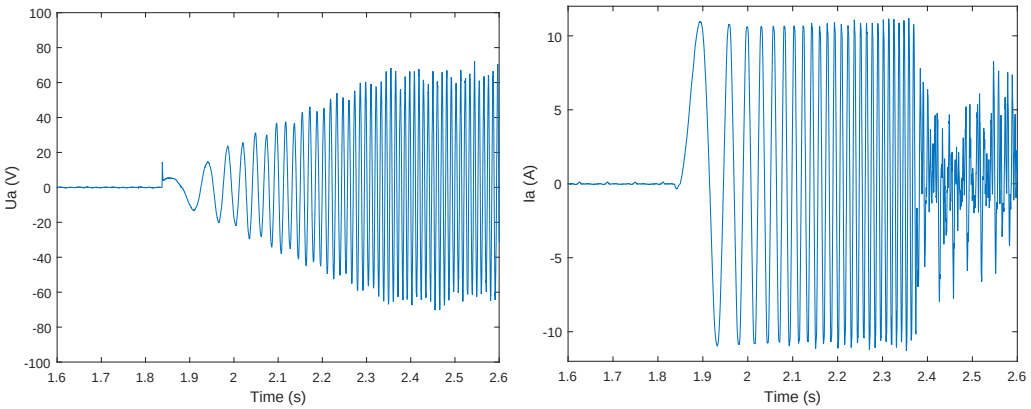

Figure 4.8: Phase a measurements with the voltage (left) and current (right)

4.1.2 Low speed test

We have seen the apparition of oscillations on the speed response due to the bad alignment of both machines at high speed, but it is important to quantify them at low speed too. On figure 4.9 you find the machine oscillating around 5 rad/s, which is roughly a little below 50 RPM. The oscillations are still present, but of course of much lower frequency. During 10 seconds, at 5 rad/s, the rotor will have moved 50 rad further, which corresponds to nearly 8 rotations: we can clearly see the 8 spikes in the oscillations. Also,

with the lower frequency, the controller has more time to counteract this effect. Formally, we are attenuating the perturbation much more on the bode diagram, figure 1.10. We have an amplitude of 0.4 rad/s for 5 rad/s, which gives 0.8%.

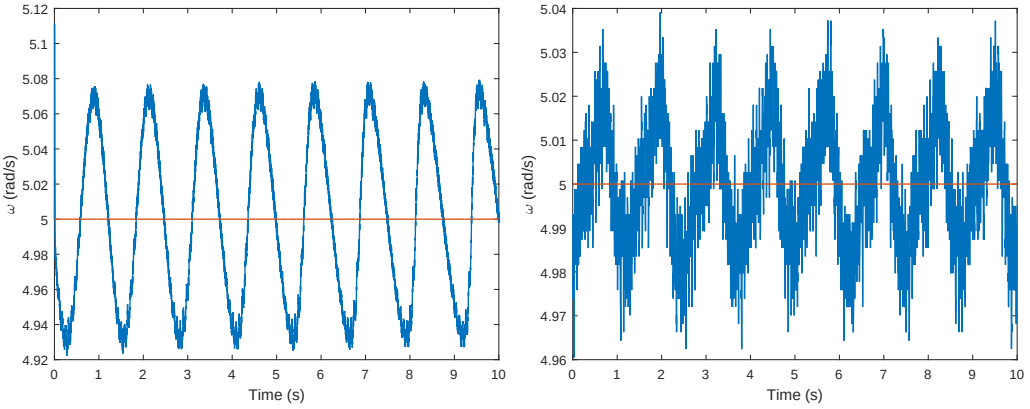

Figure 4.9: Steady state operating with the simulation (left) and the measurements (right) – Setpoint in red, response in blue

4.1.3 Torque step

In this test we made the machine turn at 100 rad/s and then switched a resistor on on the terminals of the other machine. At 100 rad/s, we plugged a resistor of 5Ω between two terminals of the other PMSM, which resulted in $19.44V_{RMS}$ and a current of $4A_{RMS}$. The load machine has a nominal voltage of $150V_{DC}$ with a current of $13.5A_{DC}$. The nominal torque is $5Nm$ at a nominal speed of 6000 RPM = 628.32 rad/s. It has thus an efficiency if 64.5%. If we compute the electrical power, divide it by the efficiency and divide it by the rotational speed we have a torque step of 1.5Nm.

We have again the recurring oscillations, but we also see a small speed drop before the controller can react, then it rises the electromagnetic torque and the speed stabilizes. We have the torque oscillations due to the bad alignment, and then we see the current rising to a higher level to develop torque (figure 4.11). As the step is only one third of the nominal torque, there is still room for it to develop torque before reaching the limit. However, if we go higher it will lose room to compensate for the pulsating torque.

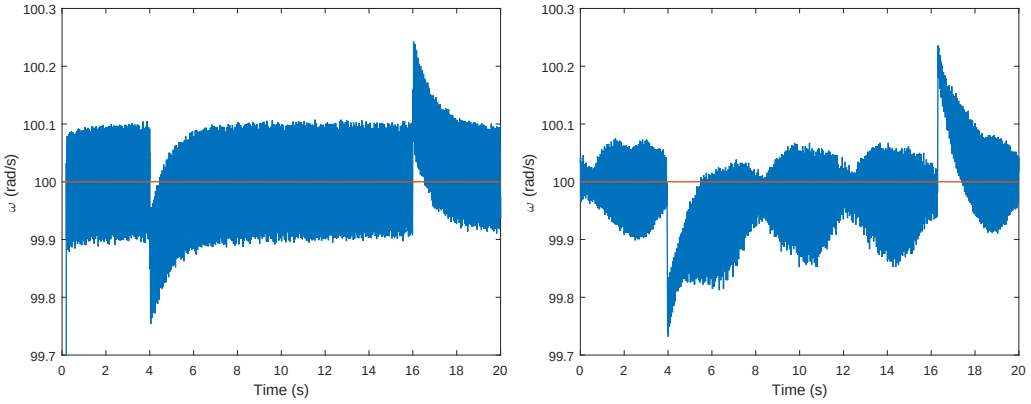

Figure 4.10: Torque step with the simulation (left) and the measurements (right) – Setpoint in red, response in blue

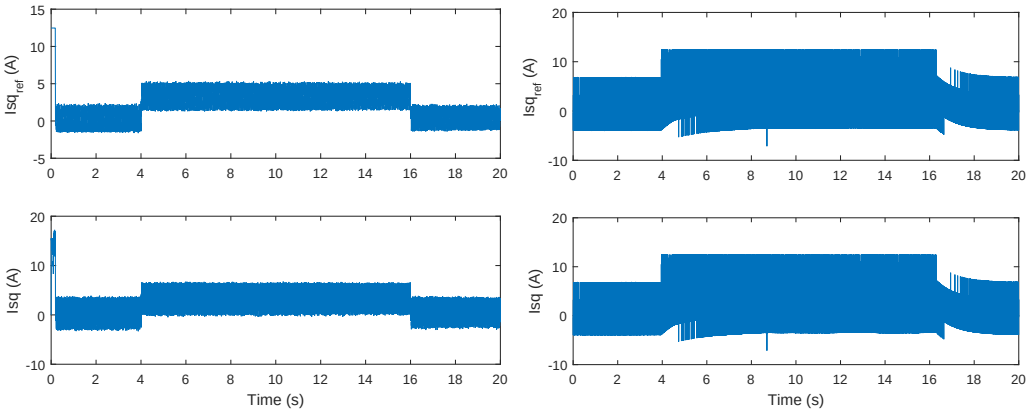

Figure 4.11: Q axis current after a torque step with the simulation (left) and the measurements (right)

4.2 Variation of controller parameter

We have seen it in section 1.3, the whole speed loop depends on a single parameter δ, which is the distance on the bode plot between the poles. We have also seen that the delta will enhance the stability or the response speed, depending if we place the poles far away from each other or close. If we take a low delta, the system will be fast but less stable than if we take a high delta. Taking a high delta also means reducing the speed bandwidth, reducing the natural frequency of the speed loop. On figure 4.12, it seems the value of delta has no influence on the system. In fact, as the most restrictive factor on the dynamics is the maximal current, speeding up or slowing down the speed controller has, to some extend, no effect.

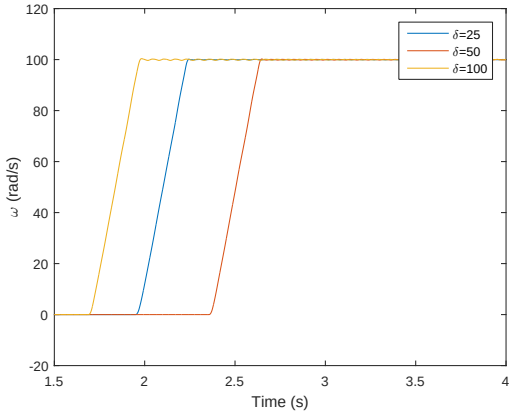

Figure 4.12: Speed response with different δ's

On figure 4.13 however, we can see an effect. As raising the value of δ means reducing the speed controller's natural frequency, the oscillations coming from the bad alignment of the machines become more perceptible.

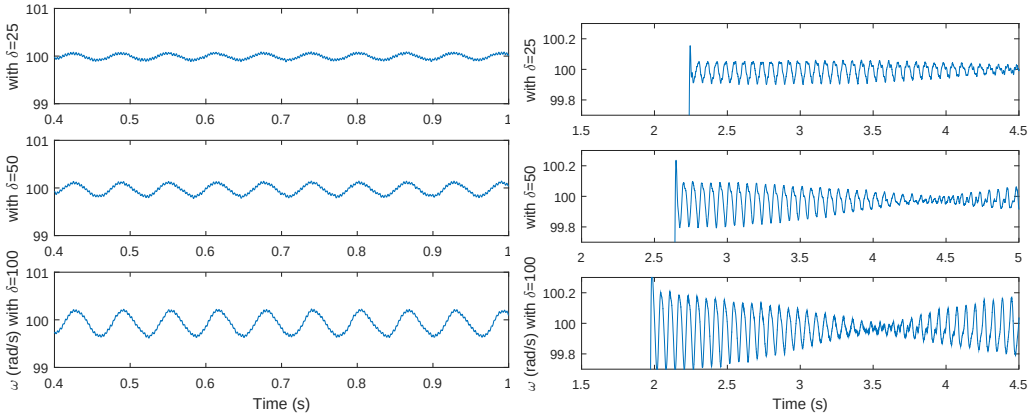

Figure 4.13: Constant speed test with different δ's with the simulation (left) and the measurements (right)

Conclusion

This report covered the development of a field oriented control of a permanent magnet synchronous machine. We have seen the necessary theoretical concepts in order to have a model of the machine. With the machine equations, we could derive a dynamic model in order to apply a control. The controller parameters have been calculated with the dynamic model of the machine and then it has been applied to a model of the machine, using Simulink's library Simscape. Once the model was validated through simulation, it was applied on the machine itself using a DSpace interface. Then it was debugged using the model again. The different results and comparisons between simulation and reality are presented in the last section.

The purpose of this thesis was the proficiency of the concepts needed to develop a machine control. It is the continuation of the work of last year, with as final goal a fully functional, modular test bench. Now the synchronous machine and the asynchronous machine have a dedicated controller. The next step would be to control them both at the same time using the two inverters. Another extension of the functionality of the test bench is to use one inverter on a machine and the other connected to the grid with a DC link between them. The full test bench needs some hardware upgrade before being complete. Two other inverters are needed to have the two back-to-back converters. Another upgrade that should be considered is a more powerful computer, as its enormous latency compromises the usability of the test bench.

One thing that an experimental master thesis puts forward is the importance of a good model in order to avoid surprises in reality. The simulations were critical to assess the controller before applying it to the machine. The controller parameters and the saturation values for instance were tested in simulation. Another aspect where the modelling comes to use is in debugging. Indeed, with the model we can start from the ideal case and then add factors to tend to the real situation. That way, it is not necessary to dive into a thousands possible factors that could cause errors, but they can be

ruled out one by one. It is that way that the importance of the speed filter was found, or the impact of the sampling frequency.

With this thesis, the reader should have an overview on field oriented control applied to permanent magnet synchronous machines. The deeper explanations and details are found in the references, so that it is possible to use it as a guide to control your own machine.

Bibliography

[1] Bruno Chevalier and Hugo Templier. Développement d'un banc de démonstration pour génératrices électriques, 2016. Université Catholique de Louvain.

[2] Bruno Dehez and Emmanuel De Jaeger. Lelec2313 - dynamic modelling and control of electromechanical converters. University Lecture, 2015.

[3] Damien Grenier, Francis Labrique, Hervé Buyse, and Ernest Matagne. *Électromécanique, convertisseurs d'énergie et actionneurs*. Dunod, 2009.

[4] Thomas Mercier. Participation of wind turbine generators to voltage control in power systems, 2013. Université Catholique de Louvain.

[5] Thomas Mercier. Field oriented control, designing pi controllers for first order plants, 2016. Université Catholique de Louvain.

[6] Aloys Nghiem and Ariola Mbistrova. Wind in power, 2016 european statistics. Technical report, Wind Europe, Rue d'Arlon 80, 1040 Brussels, Belgium, February 2017. Link.

[7] Dal Y. Ohm. Dynamic model of pm synchronous motors, 2000.

[8] Md. Rejwanur Rashid Mojumdar, Mohammad Sakhawat Hossain Himel, Md. Salman Rahman, and Sheikh Jakir Hossain. Electric machines & their comparative study for wind energy conversion systems (wecss). *Journal of Clean Energy Technologies*, Vol. 4(No. 4), July 2016.

[9] Viktor SLAPAK, Karol KYSLAN, Frantisek MEJDR, and Frantisek DUROVSKY. Determination of initial commutation angle offset of permanent magnet synchronous machine - an overview and simulation, 2014. Acta Electrotechnica et Informatica, Vol. 14.

[10] Andre Veltman, Duco W.J. Pulle, and R.W. de Doncker. *Fundamentals of Electrical Drives*. Springer, 2nd edition, 2007.

[11] Dave Wilson. Teaching your pi controller to behave, 2015. Texas Instruments.

Appendices

Appendix A

User guide

Université catholique de Louvain

Maxwell's didactic laboratory: Testbench

User guide

Author:
Ladislas d'Hoop de Synghem

Helpful people to contact:
Thierry Daras
Thomas Mercier
Emmanuel De Jaeger
Paul Sente

Academic Year:
2016-2017

A.1 Introduction

Welcome to this magnificent last-gen test bench of electrical machines ! You will find in this document all the required information to start using or configure it. On the front page you have the names of the people to contact should you need any help. The purpose of this testbench is to have an open system, easy to use, to have access to a synchronous or an asynchronous machine. Please read the whole notice before starting to make sure you do not do anything wrong.

A.2 Hardware

The testbench is composed of two electrical machines directly connected on the same axle. The machine the closest to you is a 1/3hp asynchronous machine, the other one is a Alsthom LC320TJ servomotor of 2kW (figure A.1).

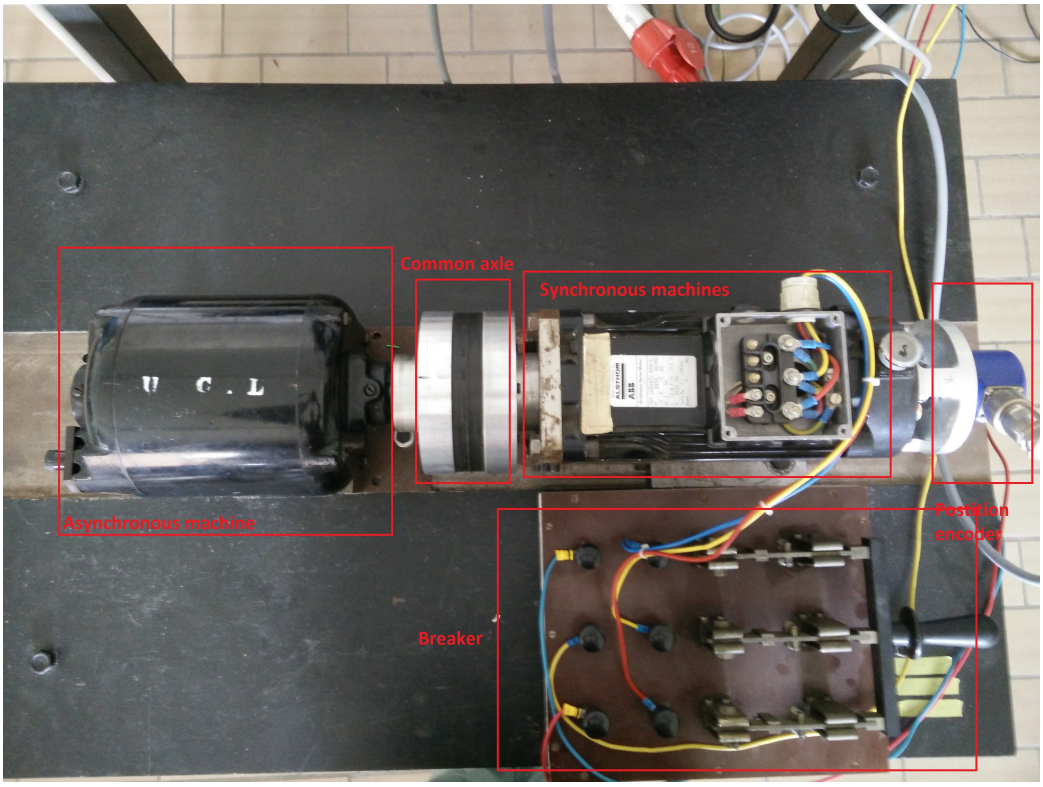

Figure A.1: The two machines

A.2.1 On the table ...

On figure A.2 you find a picture of the whole system. In front of you on the left you have a computer that serves as an interface with the system. Right to it you will find a 300V 10A DC source connected to a three phase plug. The box with the green switch is the DSpace DS1005 Real Time Interface, which will be the interface between a simulink real-time simulation and the hardware through a dedicated card in the computer. It gathers all the required information of the sensors and commands the two inverters. Right of the DSpace, there is a DC source supplying 15V and 5V for the position encoder of the rotor. Above all this, you have the two inverters of the testbench. Those receive PWM pulses from the computer through the DSpace.

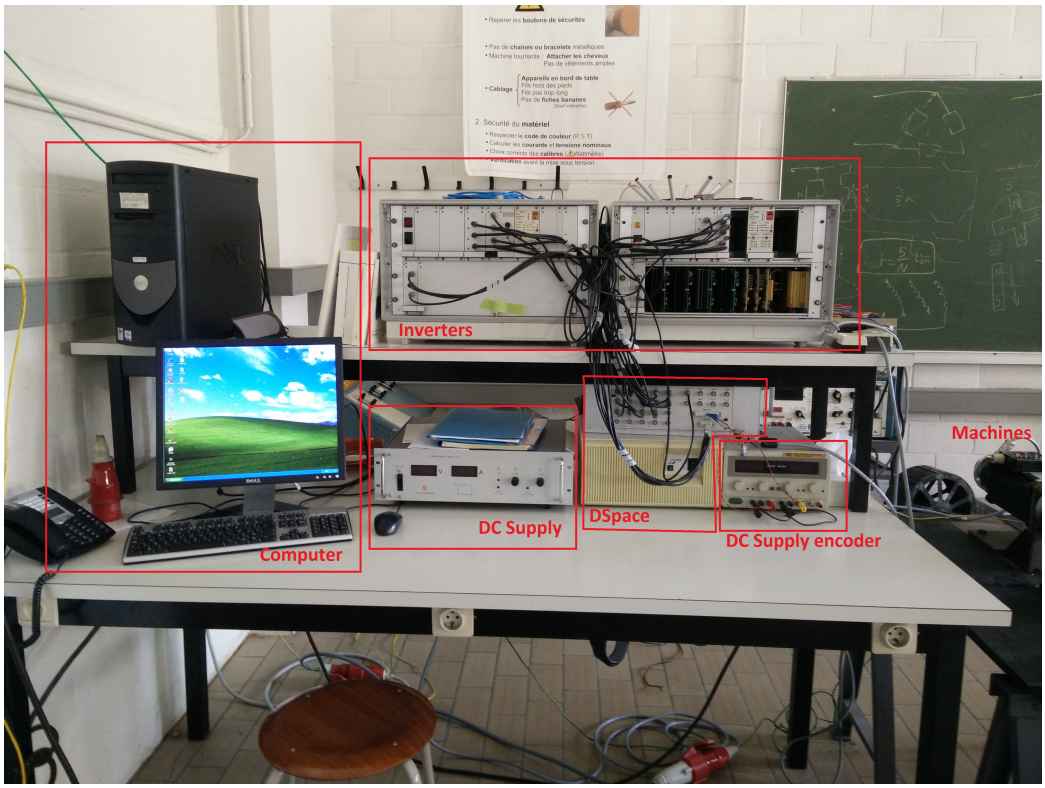

Figure A.2: All the material of the testbench

If you look at the back, figure A.3, you have on each inverter three connection points on the right for the three phases, on the left you have three access points to an internal three phase diode bridge, and in the middle an access to the internal DC bus between the two. Right now, those two inverters are coupled so that the interface with the computer can access six

phases, three per inverter. They also have a brake that can be disabled in the software.

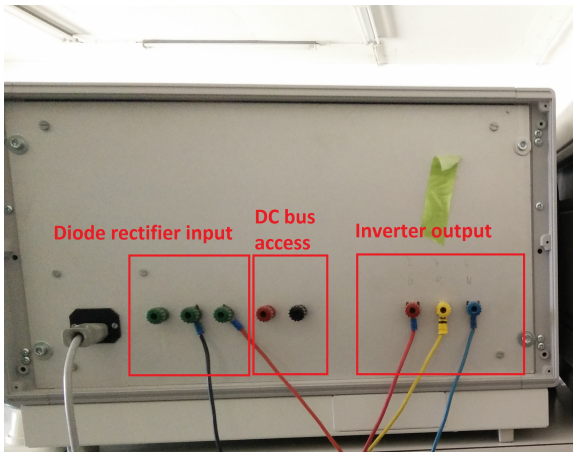

Figure A.3: Back of an inverter

A.2.2 Asynchronous machine

The asynchronous machine is made by Westinghouse Elec & MFG. Co. This is its nameplate:

Westinghouse AC motor type			
HP. 1/3	FRAME D145K	RPM 1725	CYC. 60
PH. 3	VOLTS 115	AMPS 2.2	^{o}C 55
HOURS CONT	SERIAL FN	STYLE 1178185B	ENCL

You see that it is supposed to run at 60Hz and that the nominal speed is 1725RPM. This is a strong indication that it has two pairs of poles.

A.2.3 Synchronous machine

The Alsthom LC320TJ was initially supposed to be fed in DC, so everything from the nameplate is given as values of a brushless DC motor. This means that the measured quantities of the datasheet are from the DC side of the supposed inverter. We then have to divide by two the given resistance or inductance. It is a permanent magnet three pairs of poles synchronous machine. The three windings are connected in star, even though you cannot access the neutral point. Connected to the rotor you will find an absolute encoder to have a precise value for the rotor position. Here you have its nameplate:

Made in FRANCE by Alsthom	
ABB	
Brushless Servo Motor	
Type LC620TJ R0010	
I.Cl. F	IP 65
Mn	5 Nm
In 10.8 A	Un 310 V
Nn	3900 min^{-1}
Tacho K_E	2 mV/min^{-1}

To convert those in AC value you have to think on the working of a brushless motor. With a little help from Pf. Francis Labrique and his book on electromechanical converters [3], we can find them:

$$V_{ph} = \frac{\pi}{3\sqrt{6}}U = 132.53V_{RMS} \quad \text{(A.1)}$$

$$I_{ph} = \sqrt{\frac{2}{3}}I = 8.8182A_{RMS} \quad \text{(A.2)}$$

$$Ls = \frac{1}{2}5.6mH = 2.8mH \quad \text{(A.3)}$$

$$Rs = \frac{1}{2}1.14\Omega = 0.57\Omega \quad \text{(A.4)}$$

A.3 Software

On the computer of the bench you have all that is required to run everything. Start the computer as soon as you are in the room, as it will take its time. Once it is started, you can access the software tools on the desktop. Use Matlab R2006b to run your simulink model, it is configured to work hand in hand with the DSpace. After the launch of Matlab it may prompt you to select the platform, choose RTI1005. Navigate to the folder of the simulation you want to run (where the .mbl is located) or create a new one. In D:/Users you will find the folder chevalier-templier for the control of the asynchronous machine, and the folder dhoop for the synchronous machine. Either way, the current directory of Matlab should be the one containing the simulation file. In your simulink you should pay attention to those settings: Stop time=inf and External simulation. The files already on the computer contain blocs specific to the hardware, made by Francois Baudart.

On figure A.4 you find the processing of the encoder input. On the left you have the 13 bits of the ADC. Those are converted in a decimal value and

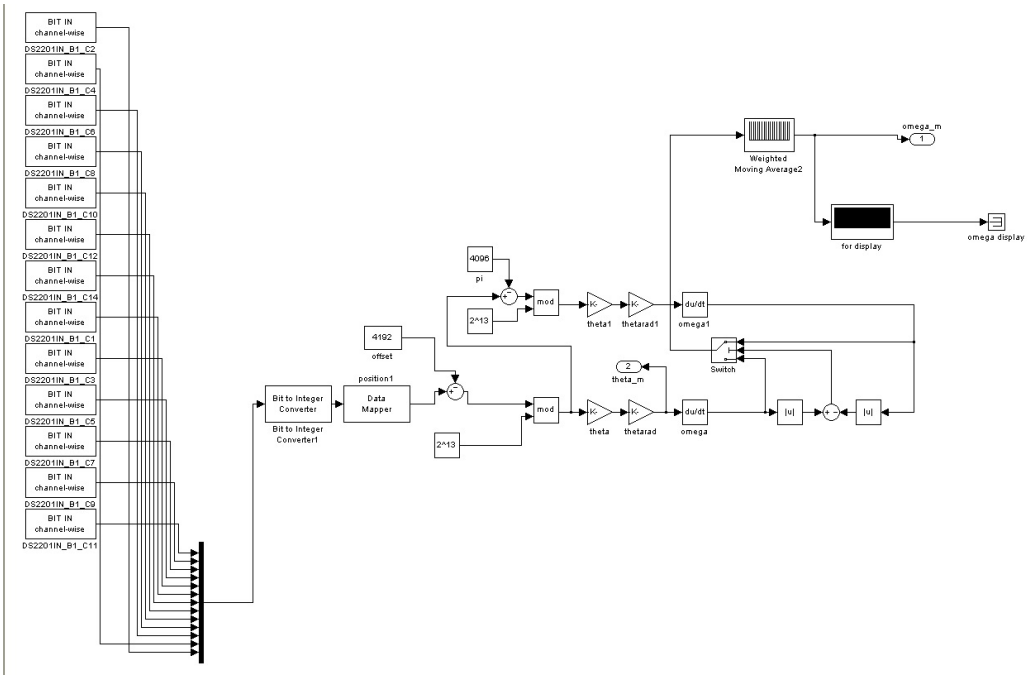

Figure A.4: Information from position encoder processing

then mapped from the Gray code they were in. The next step is to subtract an offset, which was needed for the machine alignment. After this we have a value between 0 and $2^{13} = 8192$. To have an angle value between 0 and 8192 after subtracting the offset we take the modulo. If we have a value of 200 and an offset of 500 we would have -300 as value. With the modulo we have $8192 - 300 = 7892$. We then convert that into radian using gains. That gives the value of the angle. We differentiate the angle to have the speed of the rotor. There is one last thing to mention: if we did only this, then every 2π there would be a singularity on the speed when we go from $2\pi - \delta$ to 0, i.e. just before closing the circle. To avoid that we compute another angle shifted by 180^o and differentiate that one too. We then take the speed that has the smallest absolute value to reject the singularity. Then that signal is smoothed. There are other blocks that are straightforward to understand. We have the measures of currents and voltages, with the corresponding numbers noted in the inverters in pencil, and also a brake command, that will deactivate the brake of the inverter (figure A.5). Be careful that if you do not use the latter, the output of the inverter will have a resistor connected in parallel that will dissipate heat.

Once the simulation is ready to run you can hit ctrl+B to do an incre-

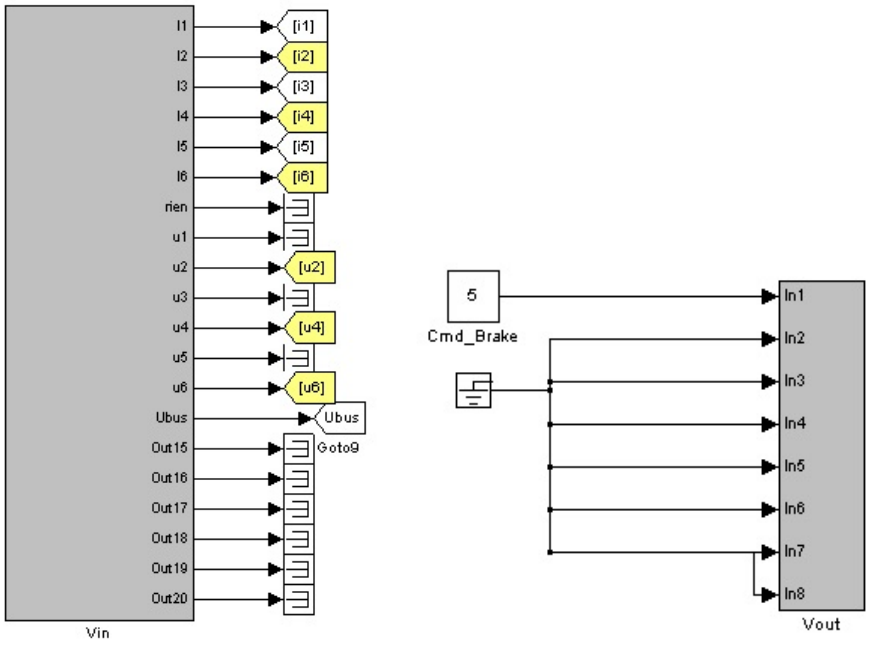

Figure A.5: Measure block (left) and brake command block (right)

mental build, but before that you should switch the inverters on and then the DSpace. Careful that the inverter on the right needs to be plugged. Go get a coffee while it is building. It should load the program on the DSpace. Now, everything is ready to run. Open DSpace ControlDesk (figure A.6), and open the variable file you just generated (file -> open variable file). The simulink variables should be visible on the lower block of the interface. Then you can either make a new layout or open an existing one (file -> open... -> *.lay). To build a new layout, you can select on the right panel an interface element like a slider, a numerical display, a plot of a variable, a numerical input and so on. Click on what you need and then draw the element in the main panel. To link it to the simulink variable, drag and drop it from the variable display at the bottom. You can control or display every simulink variable in real time.

To run the real time interface, press the play button above and select the animation mode for the variables to be displayed in real time. Make sure the inverters are turned on when the DSpace is running. Now you can interact with the testbench. Once it is running, you can start the inverters by pulling the first little switch on the left inverter to the right (switch C on figure A.7). If the *surcharge* led is on, press *réarmer* (button A).They should go

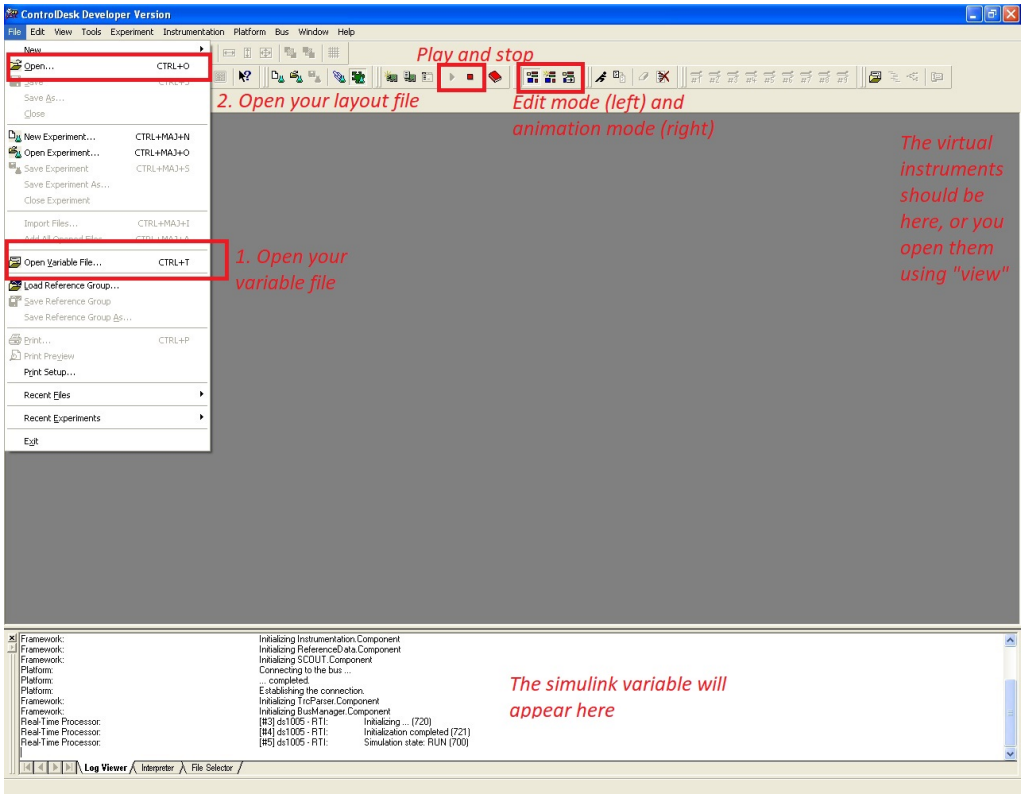

Figure A.6: ControlDesk interface

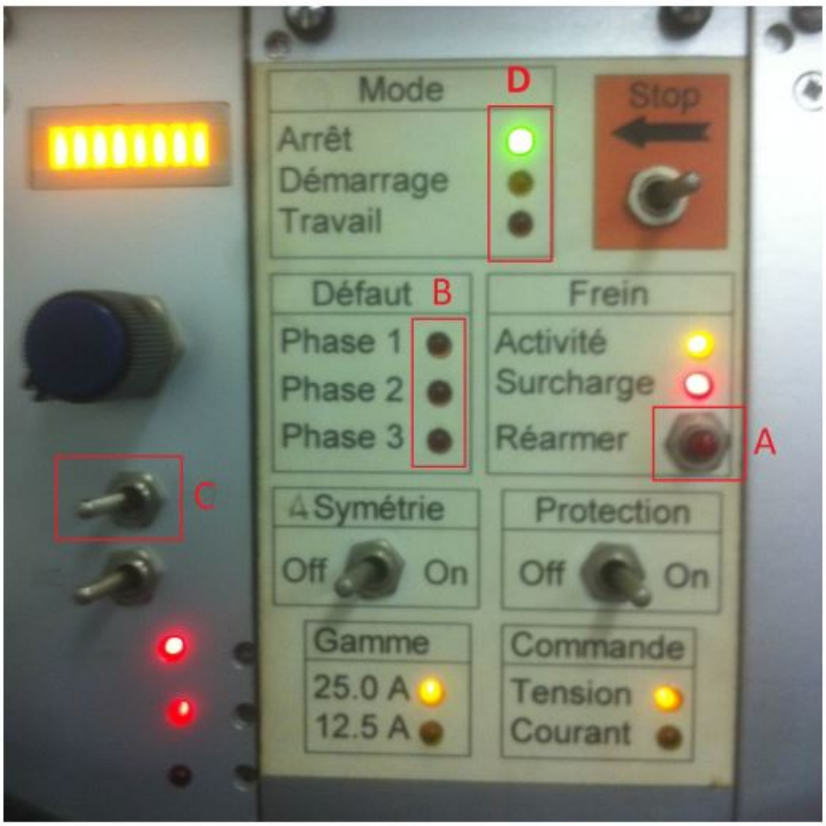

Figure A.7: Zoom of the left inverter panel (taken from the lab user guide made by Messr. Chevalier and Templier)

through a starting phase and then display *Travail* (leds D). You should see the evolution of your variables on the virtual instruments. Use the capture tools window to modify the display length or save the data.

A.4 Troubleshooting

The inverters won't start

Are you sure the program is running on the DSpace ? Check in ControlDesk that *play* is pressed. If the *surcharge* led is turned on, before pulling the switch, press both *réarmer* buttons. If they stay on, check that you have turned off the brake by putting à 5 signal in your simulink for the brake_cmd. Once the *surcharge* leds are off, you can pull the lever back left and right.

The inverters say default on a phase

Try to start it with no voltage on the DC bus.

ControlDesk says that no program is loaded

Did you build your simulink and has it loaded properly ? Check in the Matlab terminal that it says PROGRAM LOADED. Then try to press play.

Simulink cannot load the program

Is the DSpace turned on ? Check that there are no errors mentioned in your schematics on the Matlab terminal.

There is a weird noise coming from the DC supply

All this stuff dissipates power, so it needs to be ventilated. If you hear a new noise, you have probably been pumping a lot of current from the DC supply and it wants to cool down.

The machine makes some weird noises, even when it is not turning

Hold your horses ! Did you configure a too great bandwidth in your current controller ? The high vibrating noise comes probably from a high stress on the machine that comes from high frequency torque modifications.

A.5 Documentation

Figure A.8: Name plate of the synchronous (left) and asynchronous (right) machine

ALSTHOM
PARVEX
27,29 rue lucien JUY
21007 DIJON Cedex BP 249
Tel 80.41.81.18
Telex 350653

SERVOMOTEURS CA

TYPE LC620TJ

CARACTERISTIQUES UTILES

Couple permanent en rotation lente	Cr1	Nm	5
Couple nominal	Cn	Nm	5
Vitesse nominale	Nn	tr/min	3900
Puissance nominale	Pn	W	2000
Tension d'alimentation *	U	V	260
Courant nominal	In	A	10,8
Vitesse maxi pour C=2Cn		tr/min	3150
Couple impulsionnel	Cimp	Nm	19

CARACTERISTIQUES INTRINSEQUES

Fem par 1000 tr/min * (25°C)	Ke	V	53
Coef. de couple electromagnetique	Kt	Nm/A	0,506
Couple de frottement sec	Tf	Nm	0,1
Coef. de viscosite par 1000 tr/mim	Kd	Nm	0,01
Resistance du bobinage * (25°C)	Rb	Ω	1,14
Inductance du bobinage *	L	mH	5,6
Inertie rotor	J	Kgm²	0,0006
Acceleration theorique	γ	rad/s²	31700
Constante de temps mecanique	τm	ms	2,66
Constante de temps thermique	τth	min	20

Masse moteur (avec capteur)	Kg	9,4

* entre phase

Date:03.02.86	Type: LC620TJ	R...

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**