ISSN 0280–5316
ISRN LUTFD2/TFRT--5551--SE

Speed Control of a Switched Reluctance Motor

Henrik Stendahl

Department of Automatic Control
Lund Institute of Technology
February 1996

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
February 1996

Document Number
ISRN LUTFD2/TFRT--5551--SE

Author(s)
Henrik Stendahl

Supervisors
Ulf Jönsson, Karl-Erik Årzén and
Christer Andersson (Emotron)

Sponsoring organisation

Title and subtitle
Speed Control of a Switched Reluctance Motor

Abstract

This thesis gives an introduction to the switched reluctance motor. A model is developed in Simulink to investigate its nonlinear characteristics. A controller is designed for the Simulink model. The major concern for the controller is to obtain a smooth torque. Normally the current is measured for control, but in this case the current is not used, which means that an inner loop is lost. The following gain scheduling principle for the speed controller is proposed. Operating points for the speed and the torque are defined, around which the system is linearized. Two parallel controllers are implemented. A shape controller to minimize the torque ripple, and a PI-controller to handle reference changes and load disturbances. The shape controller produces a voltage shape at a level determined by the PI-controller. The parameters of the controllers are tuned according to the current operating point.

Key words
Switched Reluctance Motor, Speed Control, Nonlinear Models, Gain Scheduling, Computer Simulation

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
73

Recipient's notes

Security classification

Acknowledgments

I would like to thank my supervisors *Ulf Jönsson* and *Karl-Erik Årzén* at LTH and *Christer Andersson* at Emotron for all help and guidance they have provided. I would also like to thank *Lars Sjöberg* from the department of electrical engineering and automation, for the model, and for useful hints and advice. I am also grateful to the help *Mikael Johansson* has provided with MatLab Simulink. Finally I would like to thank the staff at the department for making my time here very pleasant.

Lund, February 1996

Henrik Stendahl

Contents

1. Introduction

The switched reluctance motor is a very old motor, but only a couple of decades ago it was almost unknown. In the last years the motor has been 'reborn', and has been implemented in an increasing number of applications. One of its many advantages relative to other more common motors, as the DC and AC drives, is the low manufacturing cost. One of the problems with the motor is its nonlinear characteristics, which makes control hard. The torque tends to ripple very much. The usual strategy to obtain smooth torque is to measure the instantaneous current and then implement a current controller. This is an expensive approach, since the current controller and the sensor need to be very fast. In this thesis a cheaper solution is discussed, which does not require the instantaneous value of the current.

A model is developed in Simulink, and the design of the controller is based on results from simulations on this model.

In the second chapter the motor is described. First a brief introduction is given, and after that a more detailed description of its characteristics follows. A short description of the converter and the speed sensor is also covered in this chapter.

The model is presented in chapter three. First the development of the model is described, and then the control loop is shown. Finally the implementation in Simulink is described.

In Chapter four the control strategy is introduced. Here we will also see how the controller used today works. The control strategy is based on shaping the voltage in order to get a smooth torque.

In the fifth hapter some simulation results are shown. For comparison some simulations are done with the controller that is used today.

In the last chapter conclusions are given. A number of suggested improvements are also listed here.

2. The Switched Reluctance Motor

2.1. Introduction

This chapter gives a description of the switched reluctance motor, and some of its characteristics needed to understand the control strategies presented in Chapter 4. For further reference, see [5], and [6].

History

The first switched reluctance motor (SRM) was built already in 1838, in Scotland by Davidson. The motor was used to propel a locomotive on the Glasgow–Edinburgh railway. It was not a complete success; the top speed was less than what could be achieved by one man pushing. The development of the motor was interrupted with the invention of the DC motor which was superior as an electro magnetic energy converter. In the mid 1960's the research was continued, but only in the last 15 years, with the development of power electronics and computer aided electro magnetic design, the commercial potential has been realized. Today the performance of the SRM is comparable, and in many aspects even better than the classical DC and AC drives.

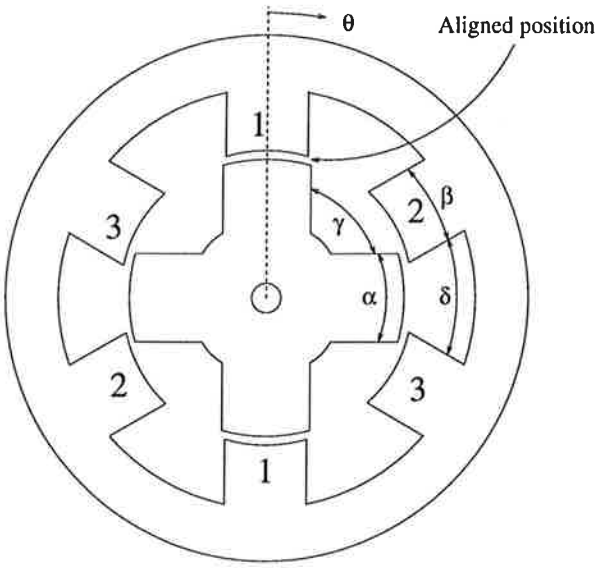
A Brief Description

The SRM is one of the simplest motors today. Fig. 2.1 shows the cross section of a 6-4 motor, which means that it has six stator poles and four rotor poles. 6-4 and 8-6 are the most usual setups, others exist, but are less common. The poles are salient and only the stator poles (normally) are winded. The windings are arranged in couples, called phases, so that two opposite windings always carry the same current. This means that the 6-4 motor has three phases. The two windings in a phase should cooperate in such a way that the produced fluxes flow in the same direction through the rotor. Both the stator and the rotor are laminated to decrease the eddy currents.

In Fig. 2.1 some useful notation is defined, which will be used more or less frequently throughout the thesis. The position of the rotor is denoted θ. This is defined separately for each phase. In the picture, the angle, θ for phase 1 is shown. A more detailed description of the how the angle is defined follows in Section 2.2. The terms aligned and unaligned are used separately for each phase. In Fig. 2.1 the rotor is in the aligned position for phase 1. In the unaligned position the stator pole should be exactly midway between two rotor poles. Neither for phase 2 nor phase 3 is the rotor in the unaligned position.

The simplest form of a reluctance motor is shown in Fig. 2.2.

When current is forced into the winding the rotor tends to align with the stator, where the reluctance is minimized. Minimized reluctance is equal to maximized inductance. The direction of the current does not matter as long as the two windings cooperate. In this case the rotor would move clockwise until it reaches the aligned position (vertical) where it would stop. If we look



α : Rotor pole arc
β : Stator pole arc
γ : Rotor inter polar arc
δ : Stator inter polar arc
N_r : Number of rotor poles = 4
N_s : Number of stator poles = 6
q : Number of phases = 3 (numbered)
θ : Rotor angle

Figure 2.1 Cross section of a 6-4 SRM.

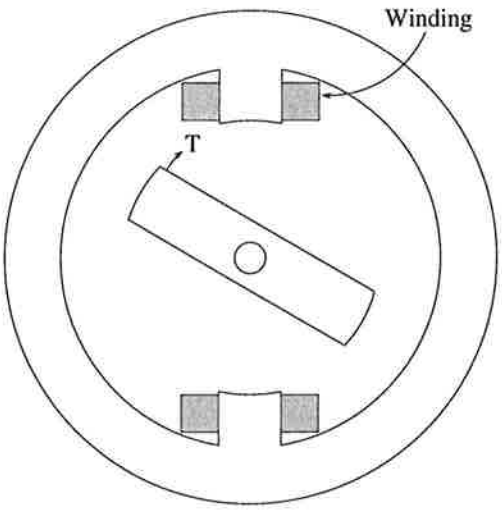


Figure 2.2 Simplest form of a SRM.

at Fig. 2.1 again we can see what happens if we force current into the different windings. If current flows in phase 1 where the rotor already is in the aligned position, the current does not produce any torque if the rotor is standing still. If it rotates however, the current produces a torque that brakes the rotor (whichever direction it moves in). If current flows in phase 2 or phase 3 a torque is produced that tends to move the rotor in the anti clockwise and clockwise direction respectively. If the different phases are turned on and off

in the appropriate order, the rotor will rotate continuously.

Comparison

A lot of work has been done to compare the SRM with other electro magnetic drives. Because of the recent commercial break-through there is still a lot of research on the motor. It is very hard to make a fair comparison, because the most advantages give corresponding drawbacks. Of course the application is essential, in order to determine which motor should be used for a certain task. Some of the most important advantages of the SRM are the following :

- The motor is very simple, and has therefore a low manufacturing cost.
- In most applications the major losses take place in the stator (the rotor carries no winding), which is easy to cool.
- The motor is brushless, which reduces the maintenance costs.
- The torque/moment of inertia-ratio is high.
- The speed range is very broad.

The two most important disadvantages are :

- The torque ripple is very high. For a narrow speed range it can be limited, but especially for high speed, the minimization of torque ripple is hard.
- The noise is high, especially for large machines.

To make a fair comparison the converter should also be included. Converters for the SRM (see Section 2.4) can be made simple, since the torque is independent of the direction of the current.

2.2. The Theory of the Motor

In this section we will go more into detail with the operation of the motor. First we are going to look at some useful definitions. Then we will approach the motor from the electrical point of view. After that we will look at the mechanical characteristics. Then we put the theory together and look at the torque production. We will also see how the speed of the rotor affects the torque. Then we will have a look at the efficiency and the effectiveness of the SRM. Finally, we will look at the losses in the motor.

All simulations done in this chapter are based on the model described in Chapter 3. The motor is run at a speed of 0.8 $\times$ nominal speed. The units in the graphs are explained in Section 2.3. Time and angles are in seconds and radians, but other quantities are not in SI-units.

Definitions

Here we explain some useful quantities, which will be used in the thesis.

The Position of the Rotor θ is used to describe the position of the rotor. Each phase has its own angle, θ, defined. When we use θ, we have to know whether it is the electrical angle or the mechanical angle. The mechanical angle, θ_{mech}, ranges from 0, when one rotor pole is aligned with a stator pole, to 2π when the same rotor pole again is aligned with the same stator pole. That means that θ_{mech} ranges from 0 to 2π in one revolution. In the electrical case, the angle θ_{el} ranges from 0, at the aligned position, over π, at the unaligned position to, 2π at the aligned position for the next rotor pole. All for the same phase. That is

$$\theta_{el} = (N_r \theta_{mech}) \bmod 2\pi$$

where N_r is the number of rotor poles. The value of θ always lies between 0 and 2π.

The relation between the angles for different phases in a motor with 3 phases is as follows:

$$\theta_{phase2} = \theta_{phase1} - \frac{2\pi}{3}$$
$$\theta_{phase3} = \theta_{phase1} + \frac{2\pi}{3}$$

When speed is mentioned, it is always the angular velocity, that is meant. This is denoted ω, and the nominal angular velocity is denoted ω_n. The speed can, as the position, also be defined from the electrical or mechanical point of view. For each mechanical revolution all rotor poles passes a certain phase. For each rotor pole passing a certain phase one electrical lap is fulfilled. This means that the electrical speed is N_r times the mechanical speed. This will be explained more carefully in Section 2.3.

Where not explicitly noted the electrical speed and position is meant.

The Phases The number of phases for a motor is denoted q. This is equal to the number of stator poles divided by two.

Where not specially noted we will look at only one phase at a time. The mutual effects between the phases are then neglected.

The Inductance When the rotor turns, the inductance for a specific phase changes. The smallest inductance appears when the rotor is in the unaligned position, and the largest when the rotor is in the aligned position. These two inductances are named L_u and L_a respectively.

Electrical Equivalent Scheme

Before we go into detail with the SRM, let us look at a separately excited DC-motor. Fig. 2.3 shows the equivalent scheme of such a motor.

Here R and L are constants and u_0 depends on the speed of the rotor.

$$u_0 = k_v \frac{d\theta}{dt}$$

where k_v is a constant. This gives the differential equation :

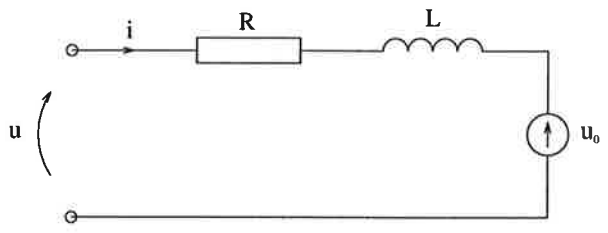


Figure 2.3 Equivalent scheme of a separately excited DC-motor.

$$u = i \cdot R + L\frac{di}{dt} + k_v\frac{d\theta}{dt}$$

If we multiply both sides with i, the first term represents the electrical losses, the second term the energy that is stored in the inductor per unit of time and the third term the power that produces torque. The stored energy returns to the source when the source is shut off.

The SRM has a slightly different equivalent scheme. One phase is illustrated in Fig. 2.4.

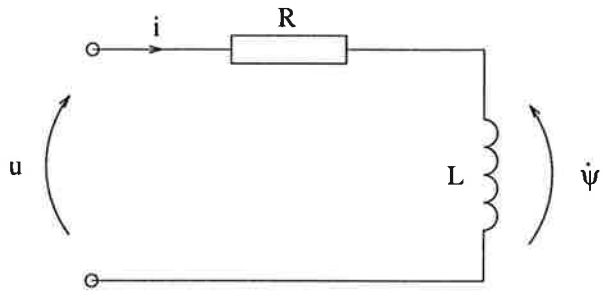


Figure 2.4 Equivalent scheme of a SRM.

Here Ψ is the flux linkage, which is the total flux through the two poles. Ψ has the unit Vs, and thus the time derivative, $\dot{\Psi}$, is a voltage. The equation becomes:

$$u = i \cdot R + \frac{d\Psi(\theta, i)}{dt} = i \cdot R + \frac{\partial\Psi(\theta, i)}{\partial i} \cdot \frac{di}{dt} + \frac{\partial\Psi(\theta, i)}{\partial\theta} \cdot \frac{d\theta}{dt} \qquad (2.1)$$

Fig. 2.5[1] shows the relation between Ψ and i for different angles, θ. These curves are the magnetization curves for the motor. The slope of the curves decreases as the flux increases. This is due to the saturation of the iron. When the iron saturates it looses its magnetical characteristics, and for total saturation, the iron behaves like vacuum (from the magnetical point of view). For angles near π (at the unaligned position) the air gap is large, and the magnetical characteristics for the air dominate the total magnetical characteristics. This explains why these curves are almost flat.

The definition of the instantaneous inductance is :

[1]The data needed to plot this is taken from the model described in Chapter 3

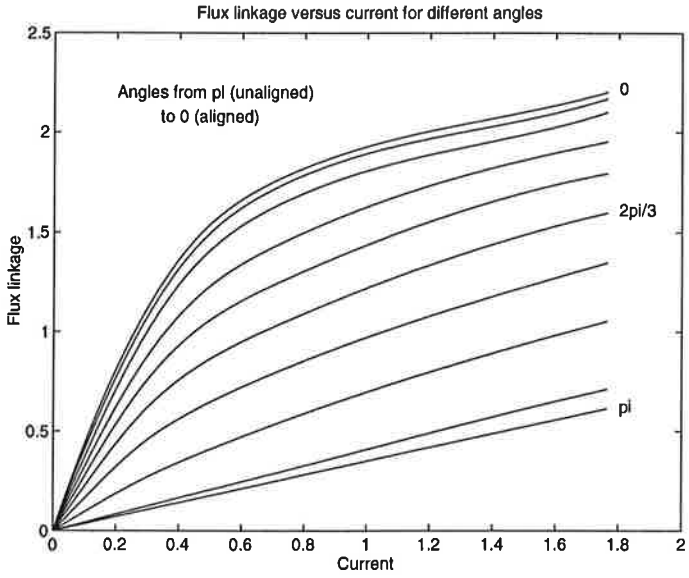


Figure 2.5 The relation between Ψ and i.

$$L(\theta, i) = \frac{\partial \Psi(\theta, i)}{\partial i}$$

That means the slope of the $\Psi - i$ curve. There is also another form of the inductance that is interesting. This is the global inductance, which is defined as :

$$\overline{L}(\theta, i) = \frac{\Psi(\theta, i)}{i}$$

In (2.1) we could also replace the time derivative of the angle by the angular velocity, w.

The equation now becomes :

$$u = i \cdot R + L(\theta, i) \cdot \frac{di}{dt} + \frac{\partial \Psi(\theta, i)}{\partial \theta} \cdot \omega \tag{2.2}$$

This is a general expression and could be used in the saturated case. If we look at the linear, non-saturated case, the curves in Fig. 2.5 become straight lines. This means that the instantaneous inductance and the global inductance are the same. Also, the inductance does not depend on the current anymore. We get the equation:

$$u = i \cdot R + L(\theta) \cdot \frac{di}{dt} + i \cdot \frac{dL(\theta)}{d\theta} \cdot \omega \tag{2.3}$$

The third term in (2.2) and (2.3) is the electro motoric force (EMF), built up by the movement of the rotor. This term is like u_0 for the DC motor except that it depends not only on the speed, but also on the position of the

rotor and the current. Equation (2.2) is very hard to solve analytically in the saturated case, Equation (2.3) however, is solvable if we do some further small approximations. Let us say that L grows linearly with θ from the unaligned ($\theta = \pi$) to the aligned position ($\theta = 2\pi$) (which is almost true for small currents), and ω is constant. We introduce the slope of the inductance k_L, and the minimal inductance L_u for the unaligned position. t is replaced by $\frac{\theta}{\omega}$, and $\frac{di}{dt}$ by $\frac{di}{d\theta}\omega$. The interesting case starts at the unaligned position, where $\theta = \pi$. This means that θ should be replaced by $\theta - \pi$. The differential equation becomes :

$$\frac{di}{d\theta} + \frac{R + k_L\omega}{\omega(k_L(\theta - \pi) + L_u)} i = \frac{u}{\omega(k_L(\theta - \pi) + L_u)} \qquad \theta \geq \pi$$

with the solution

$$i(\theta) = \frac{a}{c}\left(((\theta - \pi) + b)^c - b^c\right)((\theta - \pi) + b)^{-c} \qquad \theta \geq \pi$$

where

$$a = \frac{u}{k_L\omega}$$
$$b = \frac{L_u}{k_L}$$
$$c = \frac{R + k_L\omega}{k_L\omega}$$

and $i(\pi)$ is set to zero.

The step response for the current when the voltage is turned on at $\theta = \pi$ is shown in Fig. 2.6.

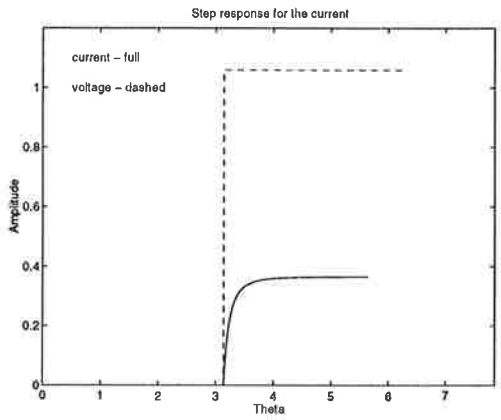


Figure 2.6 The step response for the current.

The values of the parameters are taken from the motor RRA-90L, explained in Section 2.3. The motor is driven at rated duty.

The Mechanical Characteristics

Already in Section 2.1 we could see how torque was produced. Now we are going to calculate how much. From the torque, the speed is easy to calculate. In this section, only mechanical quantities are used.

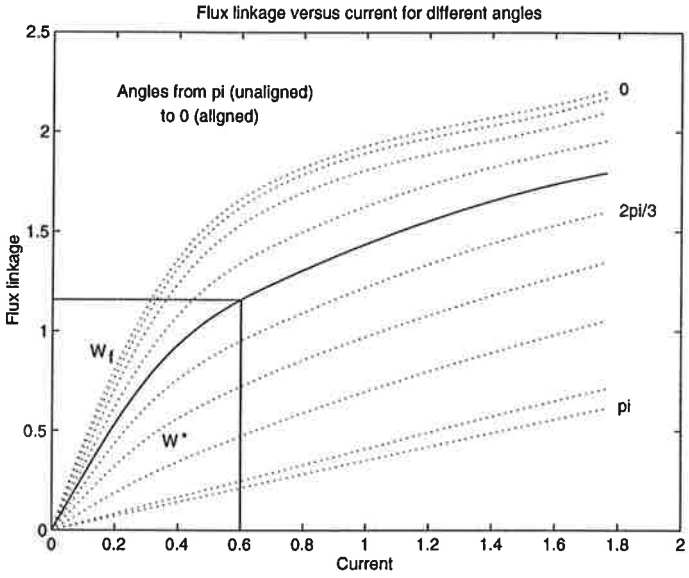


Figure 2.7 The relation between Ψ and i.

Calculation of the Torque As in the electrical description we will look at only one phase at a time. To get the total torque we will just sum up the torque from the different phases.

To calculate the torque we use the coenergy. The instantaneous torque is given by :

$$T = \left.\frac{\partial W'}{\partial \theta}\right|_{i=\text{const}} \tag{2.4}$$

where W' is the coenergy defined as :

$$W' = \int_0^i \Psi \, di \tag{2.5}$$

The first equation is the most general expression for the torque. Both equations are valid in the case with saturation.

As in the previous chapter we will look at the $i - \Psi$ diagram, shown in Fig. 2.7. The area under the curve is the coenergy at a given angle.

If we neglect the saturation the curves become straight lines with the slope $L(\theta)$. This gives us:

$$\Psi = L(\theta) \cdot i$$

and

$$W' = \frac{1}{2} L i^2$$

therefore

$$T = \frac{1}{2} i^2 \frac{dL(\theta)}{d\theta}$$

Note that this formula is valid only if the current is constant and the iron does not saturate.

There is another interesting energy, the stored field energy, W_f. This returns to the source when the voltage is shut off.

$$W_f = \int_0^{\Psi} i \, d\Psi$$

W_f is also shown in Fig. 2.7.

Calculation of the Speed To get the acceleration of the rotor we use the following equation :

$$J \cdot \ddot{\theta} = T$$

Here, J is the moment of inertia for the axis and load (if any) and T is the total torque after the load torque is subtracted. Often the load is proportional to the speed. In that case we get the differential equation :

$$J \cdot \ddot{\theta} + d \cdot \dot{\theta} = T \tag{2.6}$$

where d is called the viscous damping. The speed can be obtained by solving this differential equation.

Torque Production

Here we will put the electrical and mechanical characteristics together.

The Inductance We start to look at the inductance L as a function of θ for the linear case. See Fig. 2.8

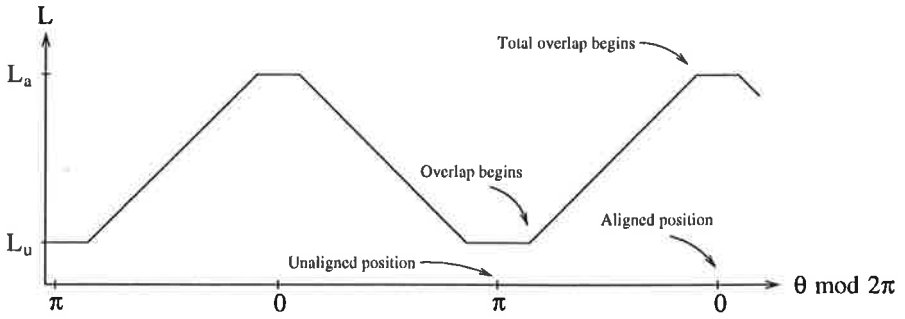


Figure 2.8 L as a function of θ in the linear case

Only one phase is plotted. The other phases look exactly the same but are displaced by $\pm\frac{2\pi}{3}$. The other phases also have their own angles defined in

the same way as the plotted phase with $\theta = \pi$ in the unaligned position. The dwells around the aligned and unaligned position come from the geometry of the poles. The rotor poles are often made a little wider than the stator poles, which explains the upper dwell. Around the aligned position the overlap is total over a range equal to the difference between the rotor pole width and stator pole width, and therefore the inductance is constant. The lower dwell exists because of the inter polar arc of the rotor is bigger than the polar arc of the stator. The overlap does not begin immediately after the unaligned position, and the inductance is assumed to be constant and very small when there is no overlap. Between the dwells the inductance is assumed to be a linear function of the angle.

The problem in the nonlinear case is that the inductance depends not only on the position, but also on the current. Fig. 2.9 shows the inductance for currents ranging from 0 to 2.2 $\times$ rated current. The graph is derived by numerical derivation in a matrix (see Section 3.1) and is therefore not very accurate. For high currents the iron gets totally saturated when the rotor is near the aligned position, and the iron behaves like air magnetically. This can be seen in the graph. We can also see that for low currents, the assumption that the inductance is linear is almost true. In this graph there is no upper dwell, and that is because the motor from which the model is taken, has equally wide stator and rotor poles. There is only total overlap in one position (the aligned).

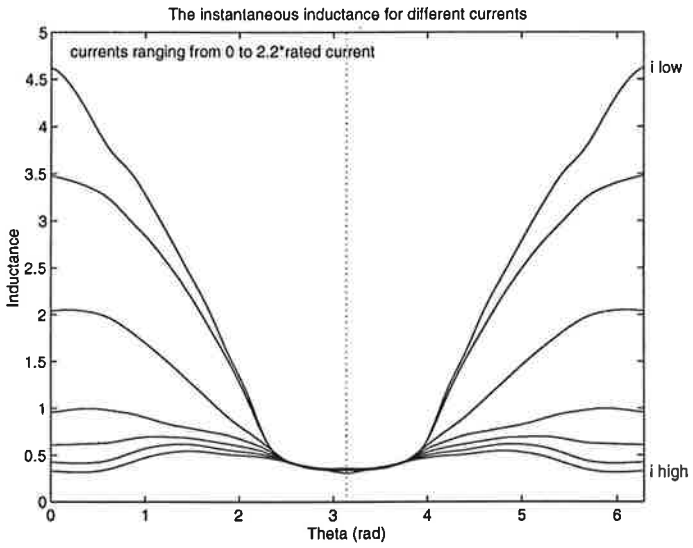


Figure 2.9 L as a function of θ in the nonlinear case.

Instantaneous Torque with Ideal Current From the previous subsection we know that in the ideal case, i.e. linear characteristics and constant current, the torque can be expressed as :

$$T = \frac{1}{2} i^2 \frac{dL}{d\theta}$$

This means that when the inductance is increasing a positive torque can be produced, and when the inductance is decreasing the motor can work as a

generator. The ideal current waveform for motoring is therefore a constant current for θ ranging from the beginning of overlap until the beginning of total overlap, and zero current for other angles. This gives constant torque. The current and inductance are shown in Fig. 2.10

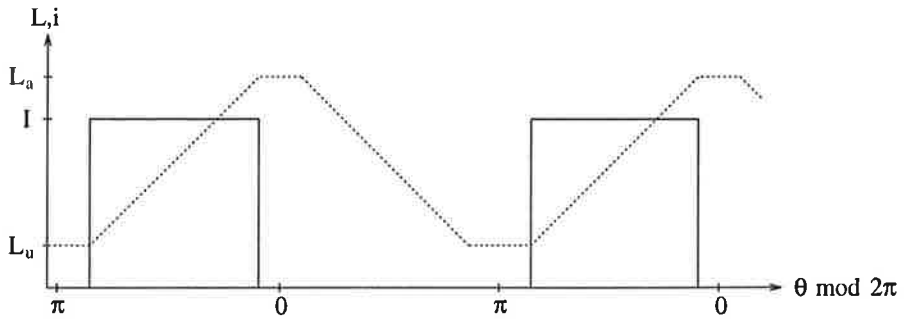


Figure 2.10 The ideal current for motoring in the ideal case.

It is essential that we have an overlap of increasing inductance between the phases. If not, the motor will not produce torque during the whole lap, and it would fail to start for some rotor angles.

If we want to calculate the torque for the nonlinear case we will have to use expression (2.4) from the previous subsection. The problem is that we need Ψ as a function of i to calculate W'. This relation is not described analytically. Therefore the torque can only be 'calculated' graphically. Fig. 2.11, illustrates the energy converted during a small change in θ. Together with (2.7) we get the instantaneous torque.

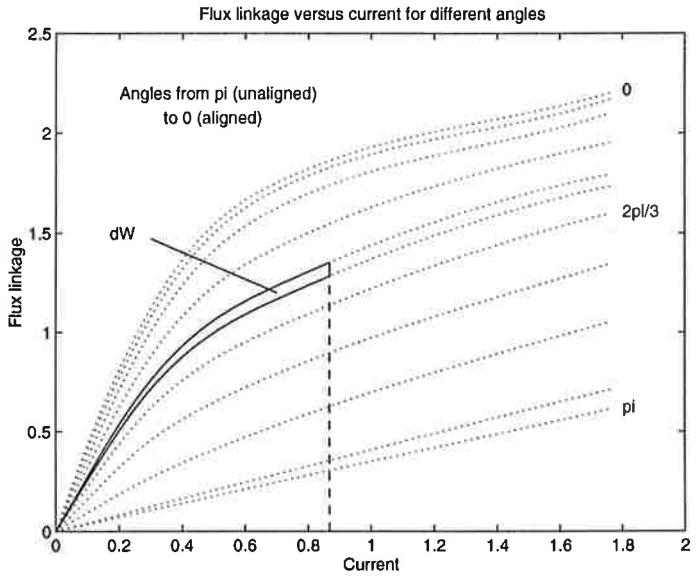


Figure 2.11 Illustration of the energy converted during a small change in θ.

$$T \approx \left.\frac{\Delta W}{\Delta \theta}\right|_{i=const} \tag{2.7}$$

where

$$\Delta\theta = \theta_2 - \theta_1$$
$$\Delta W = W'_{\theta_2} - W'_{\theta_1}$$

Note that the current must be held constant.

If we simulate the instantaneous torque production with the ideal constant current, the torque will not be linear. This can be understood when we look at the model described in Chapter 3.

Average Torque with Ideal Current Often the average torque is more interesting than the instantaneous. The average torque can be obtained in a similar way as the instantaneous. If we plot Ψ and i continuously as the rotor turns (i.e. for increasing θ), we get a loop in the $i - \Psi$ diagram. This loop is called the energy conversion loop. In Fig. 2.12 the energy conversion loop is plotted for the ideal current shown in Fig. 2.10.

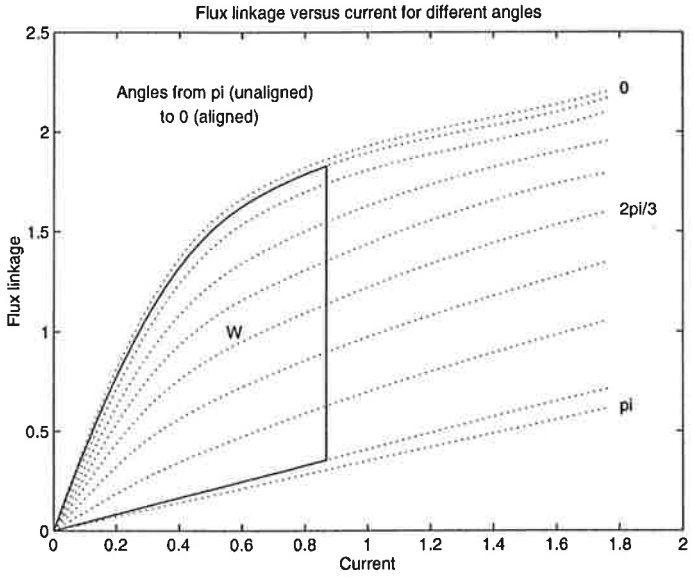


Figure 2.12 Illustration of the energy converted in one cycle with the ideal current.

The loop is plotted for one phase during one cycle. This means that θ ranges from 0 to 2π. Normally, only a part of this range is needed to get a closed loop (θ ranging from the beginning of the overlap to the beginning of total overlap). Outside this range i and Ψ should both be zero. The area inside the closed loop represents the energy that is converted. To get the torque we just divide this energy by the change in θ_{mech}. Here we must use θ_{mech} instead of θ_{el} (we want the 'mechanical' torque). We use

$$\theta_{el} = (\theta_{mech} N_r) \bmod 2\pi$$

where

$$\theta_{el} = 2\pi$$

and N_r is the number of rotor poles, to convert it. Note that this is the torque produced by one phase. To get the total torque we have to multiply this by the number of phases. This gives us:

$$\overline{T} = W\frac{q \cdot N_r}{2\pi}$$

where W is the energy and q the number of phases. This formula is also valid for other current profiles. In the case with ideal current, W can be obtained from $W'_{begin\ of\ total\ overlap} - W'_{begin\ of\ overlap}$. In Fig. 2.12, we can see that if we have current also in the dwells, we win a little extra torque. In the non ideal case the inductance is not constant in the dwells (as we have already seen).

From Voltage to Torque The ideal current is of course impossible to produce. As the voltage is limited and the circuit contains an inductor, the rise and fall time of the current is also limited. Let us instead see what we get if we hold the voltage constant in the motoring region. From Equation (2.1 we can see that the flux linkage can be derived from the voltage and current as :

$$\Psi = \int_0^t (u - Ri)\, dt \tag{2.8}$$

If we neglect the term Ri (can be done if $u \gg Ri$, which usually is the case), the flux linkage will be equal to the voltage–time surface in a $t - u$ diagram. If the speed is constant, the voltage–angle surface can be used instead. The instantaneous current can then be achieved from the magnetization curves. A problem appears when we turn the voltage off, which we do when we leave the motoring region. The flux-linkage, and the current will not return to zero. To make the current return to zero, we have to use negative voltage during a certain time. The time is chosen so that the surface with negative voltage equals the surface with positive voltage. To avoid a non zero current in the generating region (which would result in a negative torque) both surfaces should lie in the motoring region ($\pi < \theta < 2\pi$). Then the flux linkage and the current returns to zero before the generating mode is entered. The torque can be calculated as before, but when we calculate the instantaneous torque we have to choose small $\Delta\theta$ in (2.7) so that the current can be considered constant during this interval. The average torque, can be calculated just as before. In Fig. 2.13 the voltage is plotted in the $\theta - L$ diagram together with the current. The corresponding energy conversion loop is also plotted.

If we look at the current and compare it with what we got in Fig. 2.6, which is only valid for the linear case, we can see a clear difference in the end of the range with positive voltage. This is due to the saturation, which gives a higher current for a certain flux.

The instantaneous torque that corresponds to the case in Fig. 2.13 is shown in Fig. 2.14 for each phase.

We can see that the torque ripple is extremely large. The total torque is almost zero at certain times. In Section 4.3 we will see what can be done to decrease the ripple.

Speed Dependency of the Torque

In the previous section we could see the torque produced when a square wave shaped voltage was applied in the motoring region. The simulation was done with a constant speed of $0.8 \times \omega_n$ (ω_n = nominal angular velocity).

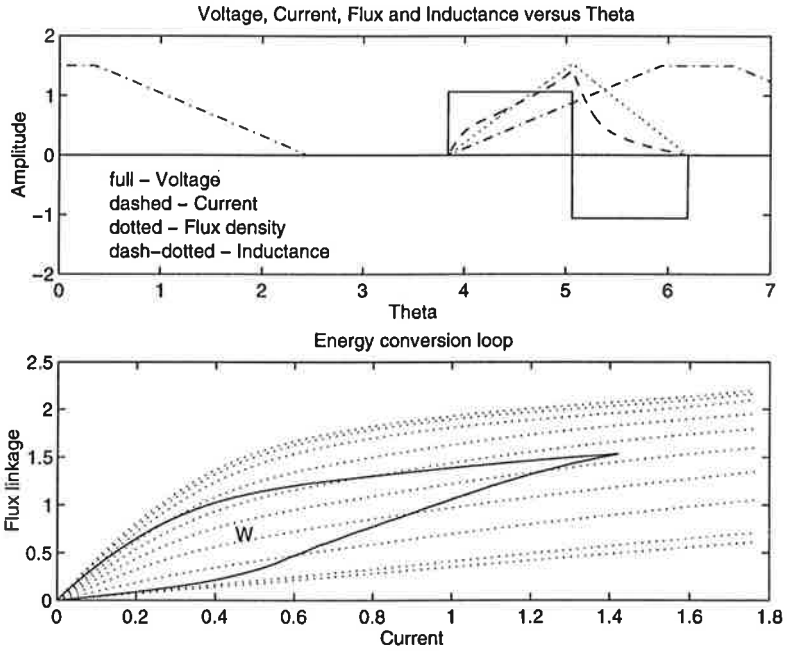


Figure 2.13 The voltage and current together with the induction, and the energy conversion loop.

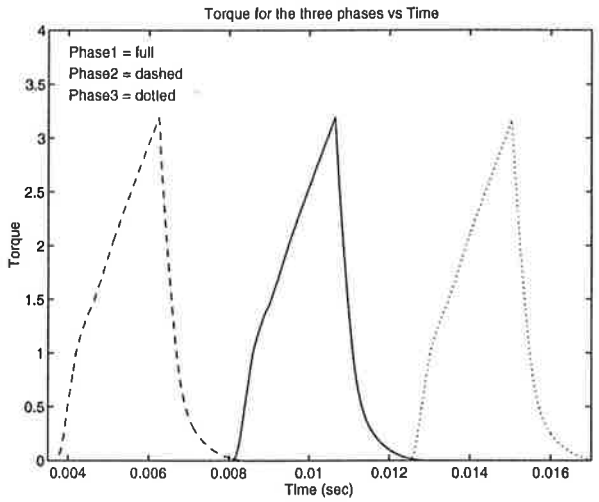


Figure 2.14 The instantaneous torque for all three phases.

Torque Production at Low Speed Let us see what happens if we decrease the speed, but keep the same voltage. Fig. 2.15 shows the voltage, the current and the flux, with the corresponding torque for one phase at two different speeds, $\omega = 0.8 \times \omega_n$ and $\omega = 0.73 \times \omega_n$.

As we can see the torque increases when the speed decreases. The reason for this is the flux, which becomes higher for a lower speed (this can be hard to see in the graph). The flux is proportional to the voltage–time surface, which (of course) becomes larger as speed decreases. When the flux increases, the current will also increase, as we know from the magnetization curves (see Fig. 2.5). As the flux and current increase, the energy conversion loop spreads

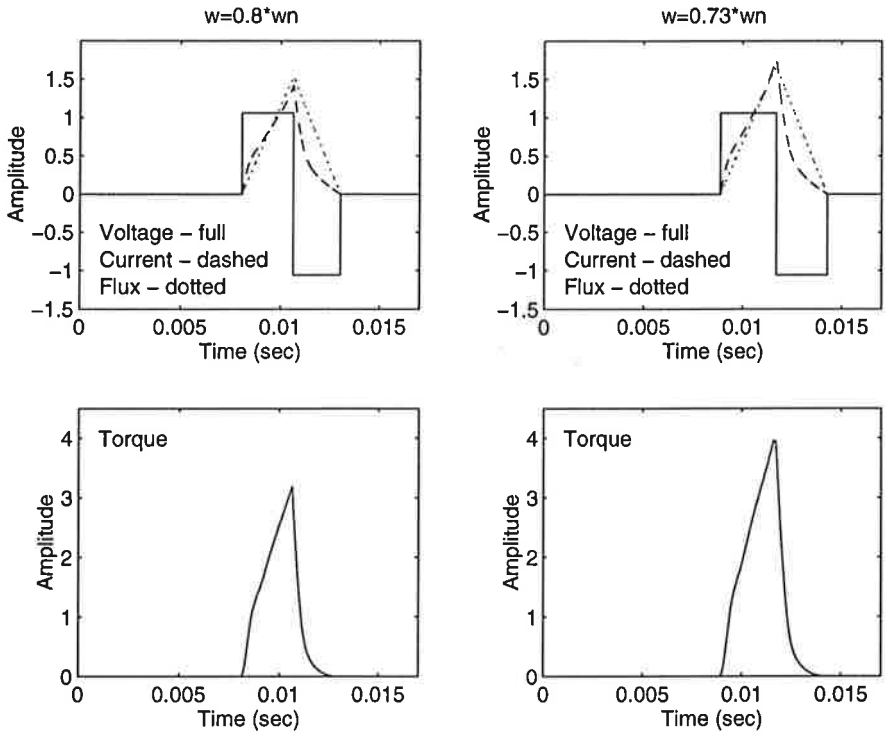


Figure 2.15 The torque produced at slightly different speeds

out, and more energy is converted, for the same change in θ. If we want the same amplitude of the torque for a lower speed we simply decrease the voltage.

So far we have only had a positive voltage during the first half of the motoring region. The reason for this is that the flux (and the current) must return to zero before the generating region begins. It is the voltage–time surface that gives a certain flux. This means that if the amplitude of the positive voltage is lower than the amplitude of the negative voltage, we can have a positive voltage during more than half of the motoring region. In all simulations this far, the maximum amplitude, which is the same for the positive and the negative voltage, has been used. The maximum voltage is denoted V_{dc}.

If we decrease the voltage to get less torque, the shape of the torque will change as well as the amplitude. In Fig. 2.16 we can see that we can only get the same amplitude of the torque if we decrease the speed (the speed in the right graph is only $0.5 \times \omega_n$). The shape of the torque then also changes. This is a problem when we want to control the motor as we will see later.

As we have positive voltage over a long time the total overlap is almost reached before the voltage is switched. When the rotor is in (or near) the region of total overlap and the current is not too low, the inductance will be small (the iron is saturated). This means that a small increase in flux results in a large increase in current. This can be seen in the upper right diagram in Fig. 2.16. This can also be understood if we look at the magnetization curves.

Torque Production at High Speed Let us increase the speed above $0.8 \times \omega_n$. Fig. 2.17 shows the voltage, current, flux and torque for one phase, with $\omega = 0.8 \times \omega_n$ and $\omega_n = 1.4 \times \omega_n$ respectively.

The shape of the current for the higher speed looks very much like in the

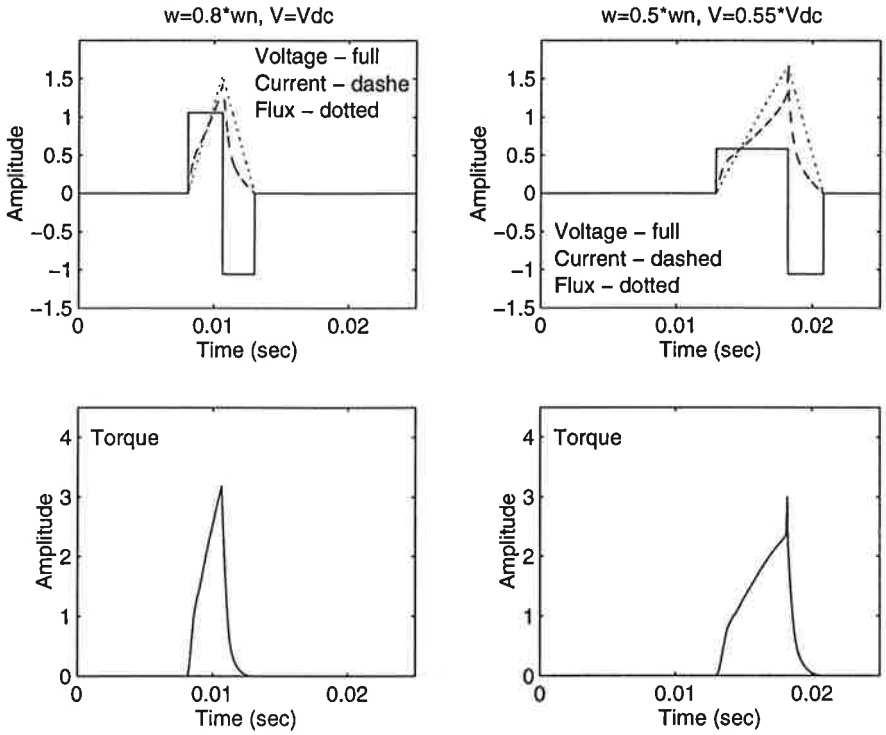


Figure 2.16 The torque produced at different speeds with different voltages

linear case (see Fig. 2.6). The reason for this is that the flux is always so low that the iron never saturates.

As we could expect the torque decreases as we increase the speed. The time with a positive voltage is shorter which means that the flux is smaller. This means that the current decreases, and the energy conversion loop shrinks. Less torque is produced.

Now we have a problem. The voltage is already maximized, and we need larger torque.

Advancing the Conduction Angle The angle for which the voltage is turned on is called the conduction angle or θ_{on}. Until now this angle has been exactly where the overlap of the poles begin. Before the overlap begins, the inductance is very low due to the large air gap between the rotor and the stator. Still (2.8) for the flux is valid :

$$\Psi = \int_0^t (u - Ri)\, dt$$

where R is the resistance of the copper windings, which is very small.

If we turn the voltage on before overlap, the current will rise very fast as the inductance is very low. The current only 'sees' the small resistance, R. Now two things can happen : if $i \cdot R$ becomes comparable with u, the increase in flux slows down, which makes the increase in current smaller. If however, the overlap is reached before the current gets large enough, the current will start to fall as the inductance becomes large. This is what happens in Fig. 2.18 where the conduction angle is set to π, and $V = V_{dc}$. The speed is again $1.4 \cdot \omega_n$. The graph also shows where the overlap begins.

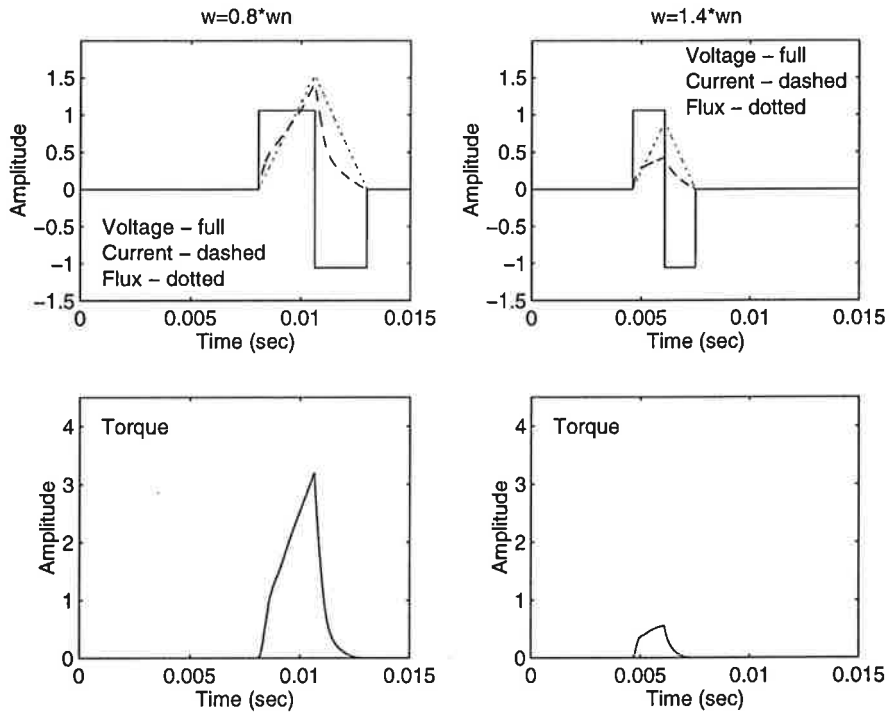


Figure 2.17 The torque produced at different speeds

For the highest value of the current in this example, the product $i \cdot R$ is less than 10% of u, which explains why the flux is almost linear.

What we can see is that the torque is much larger than it was before with the same speed. It is even larger than it was in the slower case. This is because the current is already large when the overlap begins. What we have lost is efficiency. The torque produced before the overlap is very small. And as the positive and negative voltage–time surfaces must be equal, the voltage must be switched to negative earlier than before. This means that the voltage is shifted towards smaller angles (to the left in the graph). We use current where we do not need it, and we use less current where it really produces torque. Another problem is that the shape of the torque is changed again which makes the control harder.

The maximum advance of the conduction angle, if we do not want to get negative torque, is a conduction angle of π (as in the example above). If we turn the voltage on earlier we will get negative torque. At this time the rotor pole that was the last to pass the stator pole for the phase that is about to be turned on, is still nearer the stator pole than the next rotor pole is. If the previous rotor pole has reached an angle where it does not overlap the stator pole anymore, the negative torque will be very small. If it however, still overlaps the stator pole the negative torque will be substantial. If we again look at the inductance plot, Fig. 2.19, for the linear case we can see where different torques are possible.

Note that a negative torque for one phase does not mean that the total torque is negative. For the 6-4 motor there is always at least one (sometimes two) phases that can give positive torque.

In Fig. 2.20 the simulations have been done with very high speed ($\omega = 1.76 \times \omega_n$ is the rated top speed) and low voltage. Still we get a large torque.

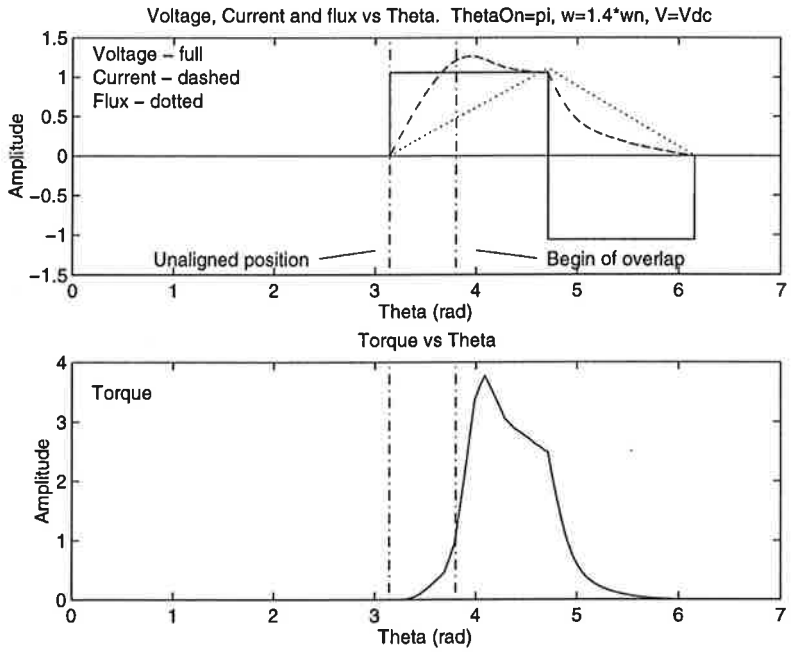


Figure 2.18 The Voltage, Current and Flux for $\theta_{on} = \pi$

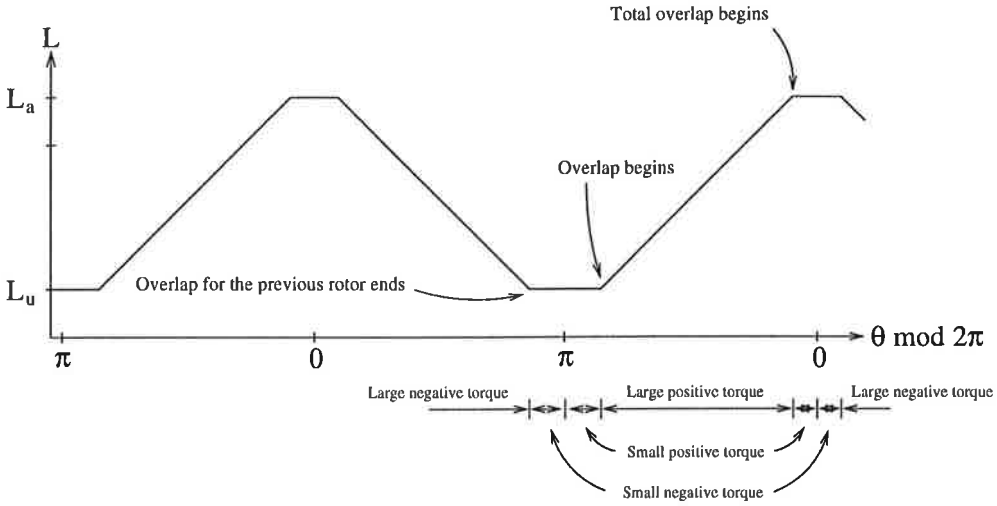


Figure 2.19 Angles where negative and positive torque production is possible.

In the case with the earliest conduction, the current does not rise as before. This is because the inductance is still high (the previous rotor pole still overlaps the stator poles) when the voltage is turned on.

Delayed Turn-off Until now we have always switched the voltage to a negative value in time so the flux and current reaches zero before the generating mode is entered. The angle where the switching takes place is called θ_{off}, and the angle where the flux and current reach zero is called θ_{end}. In Fig. 2.21 we can see how it looks when we switch later than what would be needed to avoid

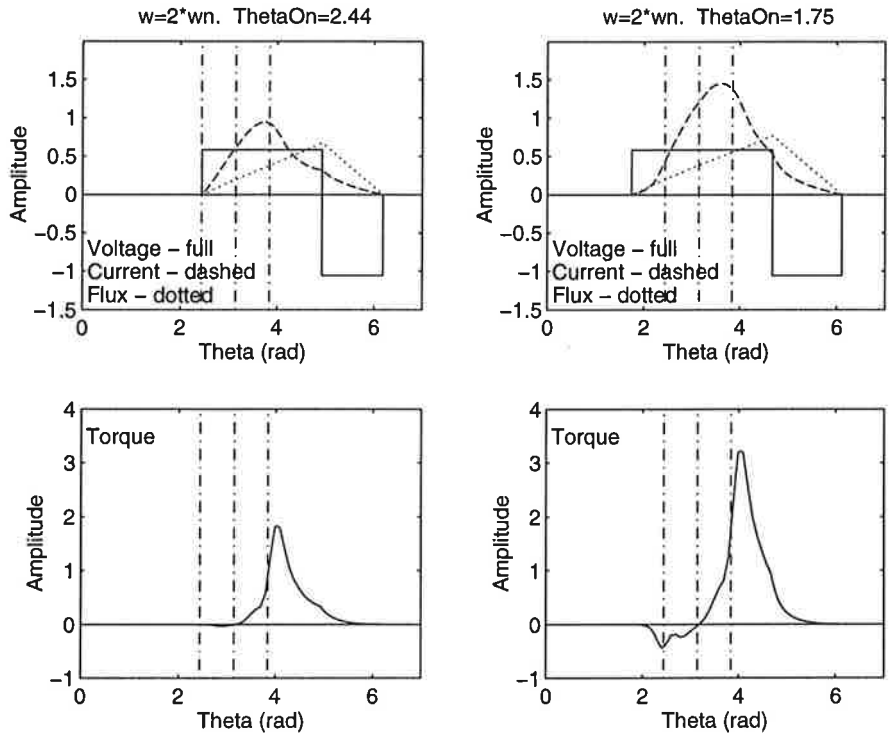


Figure 2.20 Simulation at very high speed and very early conduction. $V = 0.55 \cdot V_{dc}$. The vertical dash-dotted lines represent the end of overlap for the previous rotor pole, the unaligned position and the begin of overlap respectively.

negative torque.

The current and torque almost look like before (compared with Fig. 2.17, right), but if we zoom in when the current starts to drop we can see a clear difference. The two currents, and the corresponding torques (with and without delayed θ_{off}) are shown in Fig. 2.22.

The current in the case without delay of course starts to drop before the current in the case with delay (The drop starts when the voltage is turned negative). The shapes of the currents are also different. After the aligned position, the current in the case with delayed θ_{off} first drops very slowly, and then faster and faster until it reaches zero. This is due to the inductance. Near the aligned position it is very high. This stops the current from changing fast. As the rotor turns, the inductance decreases, allowing a faster change in current.

For the non delayed case we can see something we did not see before. The current becomes zero slightly before the aligned position. This is due to the term iR in the calculation of the flux, see (2.8). This means that the rate of change in flux is a little less than u, when u is positive, and a little greater negative when u is negative. The flux rises slower than it falls, which makes it reach zero before the aligned position.

The difference in torque is that in the delayed case it is (of course) positive longer. After the aligned position it gets negative as expected. The shape is also changed slightly. As when we advanced the conduction angle we loose efficiency also when we delay θ_{off}.

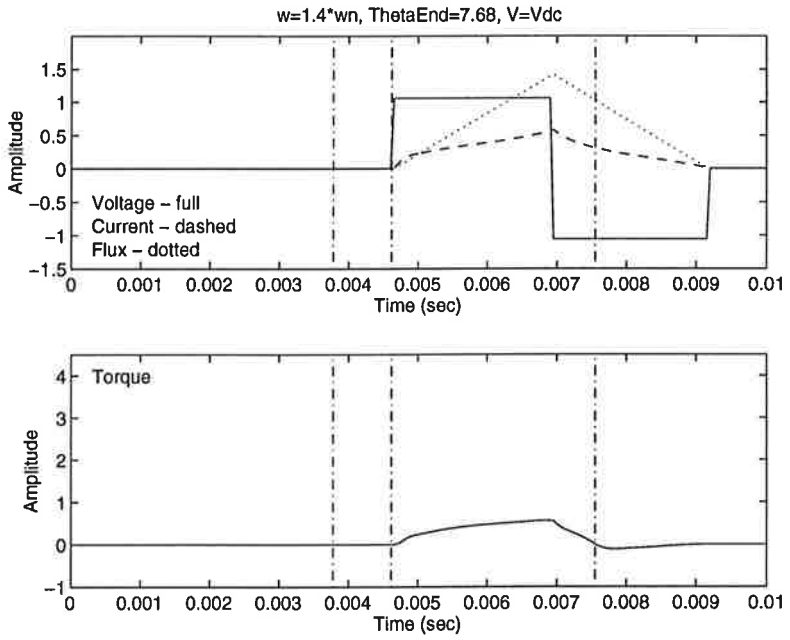


Figure 2.21 Simulation at high speed with $\theta_{end} = 7.68$, which gives $\theta_{off} = 5.76$. The vertical dash-dotted lines represent the unaligned position, the beginning of overlap, and the aligned position respectively.

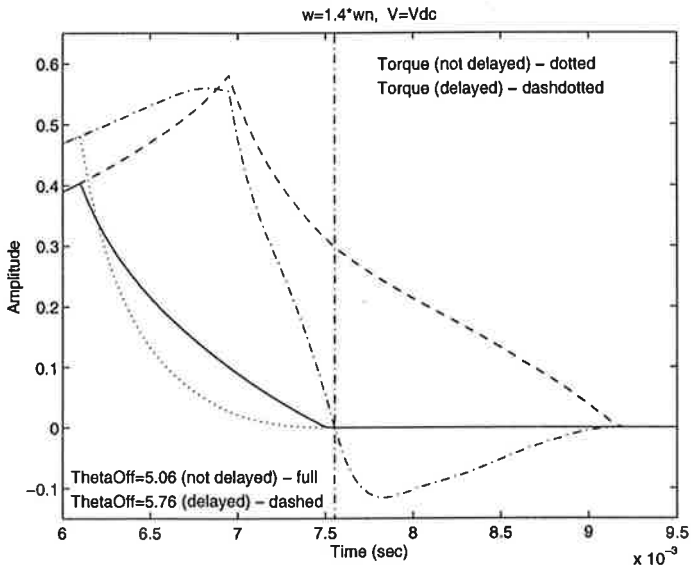


Figure 2.22 Zoom in of the current and torque from Fig. 2.17 and Fig. 2.21. The full curve is the current without delayed θ_{off}, and the dashed is the current with delayed θ_{off}. The vertical dash-dotted line represents the aligned position.

Efficiency and Effectiveness

There are two quality measurements with respect to power: efficiency and effectiveness. Efficiency is the quotient of the output mechanical power and the input electrical power. For motors (in general) this is :

$$\eta = \frac{P_{out}}{P_{in}} = \frac{T\omega}{ui}$$

The effectiveness is a measurement of how much of the input power that is converted to energy compared with the power that returns to the source after each cycle. All of this current does not result in losses. Some of it is lost as extra Ri^2-losses, but the worst thing is that the converter must be dimensioned for higher power. As we will see the effectiveness depends on whether the iron saturates or not. Fig. 2.23 tries to explain how the saturation affects the effectiveness. The curves are the magnetization curves for different cases. The surfaces in the graph represent energy. This is explained in the beginning of the section, see Fig. 2.11.

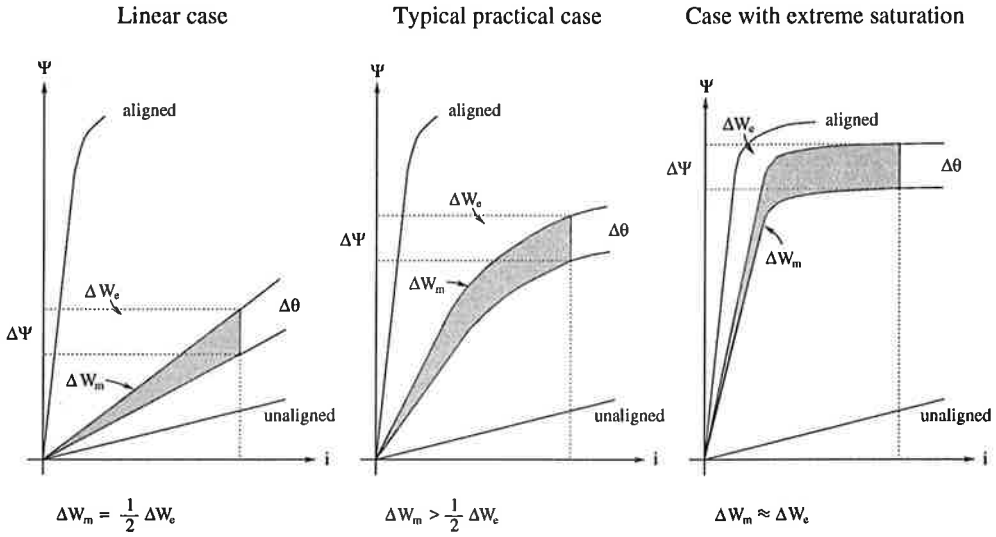


Figure 2.23 Definition of energies for a small step in θ. Left: The linear case. Middle: Typical practical case. Right: Idealized case with extreme saturation.

It is assumed that the current is constant over the step in θ. ΔW_e represents the energy that is taken from the source during the step in θ. ΔW_m (shaded) represents the energy that is converted to mechanical work during the same step. Effectiveness is defined as the quotient :

$$\frac{\Delta W_m}{\Delta W_e}$$

Note that the input energy W_e is the net energy, after the losses are subtracted (except for the extra Ri^2 losses). For the linear case, only half of the input energy is converted. The surface representing the input energy is a rectangle, and the surface representing the converted energy is a triangle with the same base and height. For the case with very high saturation (right in the graph), almost all the energy is converted to mechanical work. For the practical case the effectiveness is something in between. This means that saturation is essential to get a high effectiveness.

Losses

The losses from a complete SRM drive system consist of motor losses and converter losses. More details can be found in [7] and [2]

Motor losses The motor losses can be divided into :

- Copper losses in the stator windings.
- Iron losses (hysteresis and eddy current losses).
- Friction and windage losses.
- Stray load losses.

The copper losses, which only occur in the stator windings, are recognized as RI^2-losses. These can be seen already in (2.2) for the voltage, in the $R \cdot i$-term. After multiplying both sides with i we get the power on the left side, and the Ri^2-losses on right.

The iron losses consist of two different parts, the hysteresis losses, and the eddy current losses. The hysteresis losses come from the energy needed to remagnetize the iron. The surface encapsulated by the curves in Fig. 3.1 (see next chapter) represents the energy lost in one magnetic cycle. These losses are neglected in the model. The eddy currents are induced in the iron by the changing flux. These losses are limited by the laminations, but not completely eliminated. These losses are not included in the model either. The friction and windage losses depend highly on the application. Normally these losses are proportional to the speed, but in some cases, for example for fans, they are proportional to the speed square. In the mechanical model these are included as d in the G_{mech}-block (see Section 3.3) . Here they are assumed proportional to the speed. Stray load losses are included in the model as *Load*.

Converter losses The converter losses are negligible compared to the motor losses. In thermal design however, it is necessary to estimate them.

Control Problems

In this section we are going to have a short look on what problems come up when we want to control the speed of the motor. We especially want to obtain two objectives with the controller:

- Small torque ripple
- High efficiency

Both should be valid over a broad speed range, and for different loads. As we have seen in the previous sections there is a trade off between the two demands. As we will see in Chapter 4, most of the work done when implementing our controller is concentrated on minimizing the torque ripple. The efficiency is not taken into account.

The problem with the controller to be implemented is that the instantaneous current is not measured. The reason for this is that controlling the current demands a very fast controller, and a fast measurement device for the current. The only measured data is the speed, the position and the mean of the current. To minimize the torque ripple, a suitable shape of the voltage must be determined. With the speed sensor explained in Section 2.5 and a

normal moment of inertia coupled to the motor, there is no chance of estimating torque ripple from the speed. This means that the shape controller will work as a feed forward controller (without feedback). The appropriate shape could be determined from the speed and the mean of the current. Note that the shape of the voltage does not only mean the angles for turning it on and off, together with the amplitude. We should also be able to change the amplitude "inside" a cycle. In Chapter 4 the control strategy will be discussed in detail.

2.3. The RRA-90L

The RRA-90L is a switched reluctance motor developed by Emotron AB, in 1988. The model used for the simulations is based on this motor. It is a three phase motor, with four rotor poles. Each stator pole carries a winding of 252 turns. The windings are coupled in parallel between the poles for each phase.

In Appendix A the motor data, the normalization factors and the normalized parameters are given. The design of the motor looks like in Fig. 2.1, but the stator pole width is equal to the rotor pole width.

The Conversion of Quantities

When the rotor turns one fourth of a lap (there are four rotor poles), the change in the 'mechanical' angle is $\pi/2$, but the change in the 'electrical' angle is 2π. After this everything repeats itself. This means that the speed is different for the two definitions of a revolution. From the electrical point of view everything runs four times faster (four rotor poles). This can be seen as a gear from mechanical quantities to electrical. Fig. 2.24 shows how it can be seen. Not only the speed has two definitions. Also the torque and the moment of inertia will be different at the different sides of the gear. The power and energy must be the same on both sides. This means that :

$$P = T_{mech}\omega_{mech} = T_{el}\omega_{el}$$
$$E = \frac{1}{2}J_{mech}{\omega_{mech}}^2 = \frac{1}{2}J_{el}{\omega_{el}}^2$$

where P is the power and E is the energy. We already know that ω_{el} is four times ω_{mech}. This leads to :

$$T_{el} = \frac{1}{4}T_{mech}$$
$$J_{el} = \frac{1}{16}J_{mech}$$

Now all quantities are defined from the electrical point of view.

Normalization

In the model (see Chapter 3) all quantities are normalized so they get unit free. The new 'unit' we call PU (Per Unit). The advantage of this is that if we use the same motor, but change one parameter, for example the number of

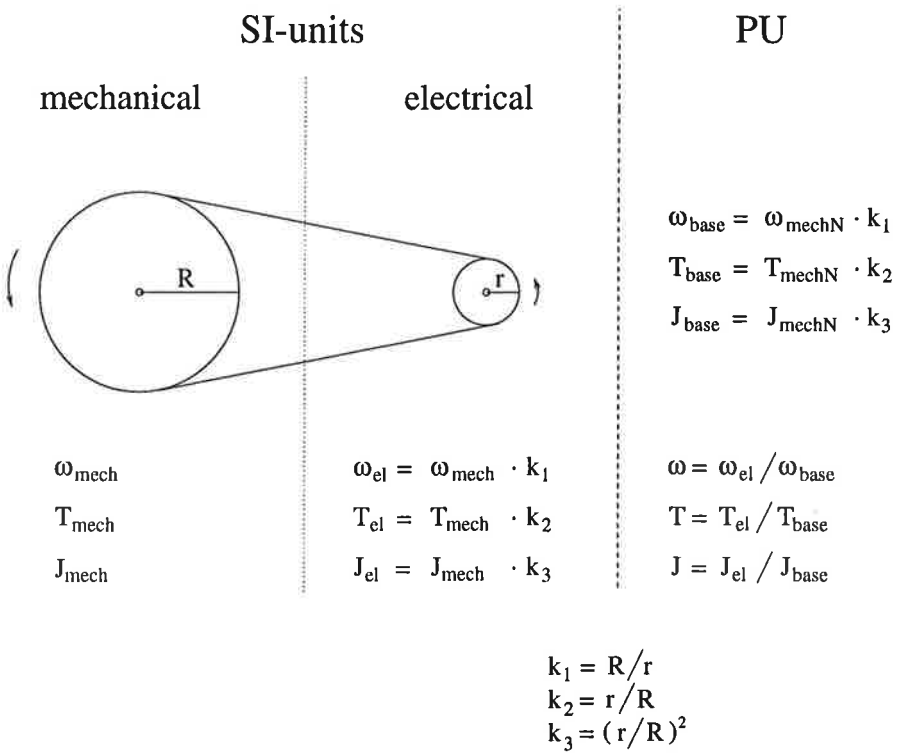


Figure 2.24 The conversion between electrical and mechanical quantities can be seen as a gear. The suffix N denotes nominal.

turns/winding, we can still use the same model. We only have to change the normalization factors. The factors are chosen so that the nominal (electrical) speed becomes 1 PU. The factors are given in Appendix A.

In Fig. 2.24 the conversion between SI-units and PU are shown. If we for example want to convert a mechanical torque of 10 Nm to PU we do the following :

$$T_{PU} = \frac{T_{el}}{T_{base}} = \frac{\frac{1}{4}T_{mech}}{T_{base}} = \frac{\frac{1}{4} \cdot 10}{5.6} \text{ PU} = 0.45 \text{ PU}$$

When we go between units on the mechanical side to PU we must go via the electrical side.

Every time an integration takes place in the model, the quantity (which is in PU) to be integrated is multiplied with a time (in seconds). The result then comes out with the unit seconds. To go back to PU, we have to multiply the quantity with a normalization factor with the unit s^{-1}. ω_{base} is used for this. Integration takes place on three places in the model (see Section 3.3). After all these integrations, the result is multiplied with ω_{base}. The block doing this is called PUConv.

From now on all quantities are in PU unless noted.

2.4. The Converter

Between the controller and the motor we need a power circuit. The controller asks for a voltage between $-V_{dc}$ and $+V_{dc}$, and as the power source only

supplies the constant voltage $+V_{dc}$, we need a converter. The converter for one of the phases (the others look exactly the same) is illustrated in Fig. 2.25.

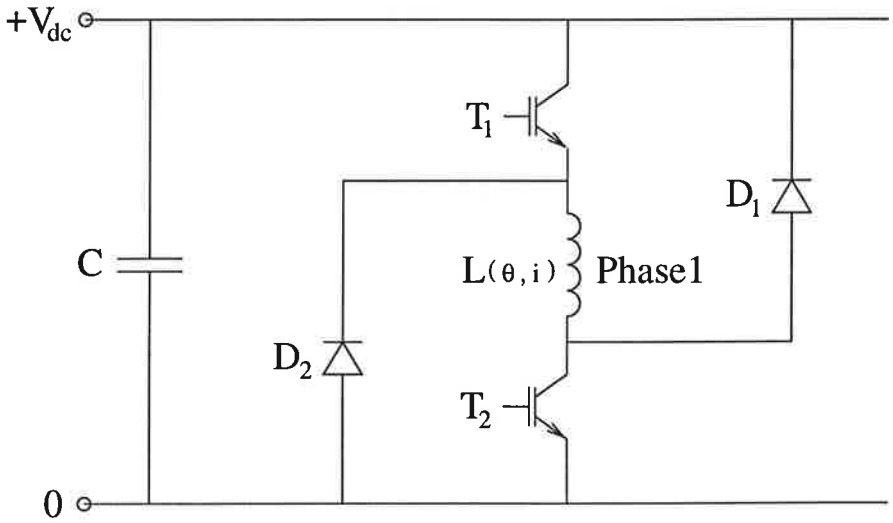


Figure 2.25 The converter circuit for one phase. L represents the inductance for phase 1.

The voltage V_{dc} is easy to obtain. We simply open the two transistors T_1 and T_2.

If we want a voltage between 0 and V_{dc}, we have to chop it. This is done by keeping one of the transistors open and switching the other one on and off at a certain frequency, f_s. The duty cycle, d, is the time the switched transistor is open divided by the switching time, $\frac{1}{f_s}$. d is then chosen so that

$$d = \frac{V_{desired}}{V_{dc}}$$

Negative voltage can only be achieved when a current flows through the windings. By closing both transistors the current continues to flow back to the source through D_1 and D_2. The current can never be negative (i.e. flow backwards through the windings) because of the diodes. Normally (in motoring applications) a voltage between $-V_{dc}$ and 0 is not interesting, because the reason for having a negative voltage is only to remove the current. In most cases this should be done as fast as possible. As we will see later however, there will be cases where it could be useful with a voltage between $-V_{dc}$ and 0. This can be achieved by leaving one transistor open, and switching the other one on and off.

The chopping method where all three values $+V_{dc}$, 0 and $-V_{dc}$ are used as above, is called soft chopping. Another chopping method that only uses $+V_{dc}$ and $-V_{dc}$ can also be used. This is called hard chopping. Here however, only soft chopping will be used.

The capacitor, C, is used for stabilizing V_{dc}.

2.5. The Speed Sensor

The speed sensor is based on the Hall-effect. The stator is equipped with three Hall-elements, each between two stator poles as shown in Fig. 2.26.

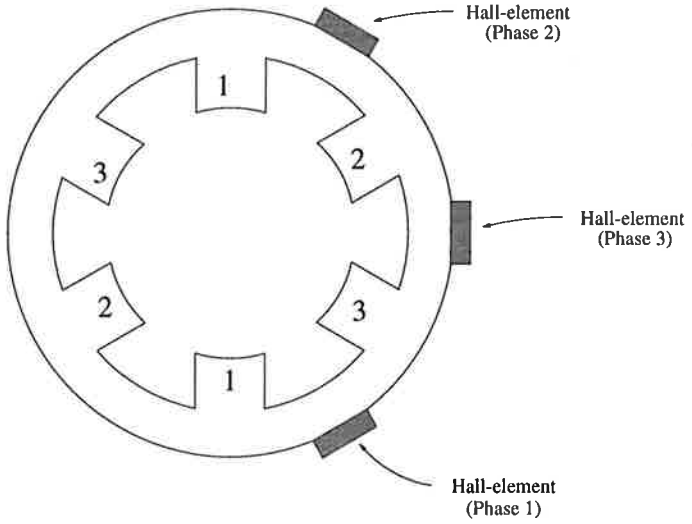
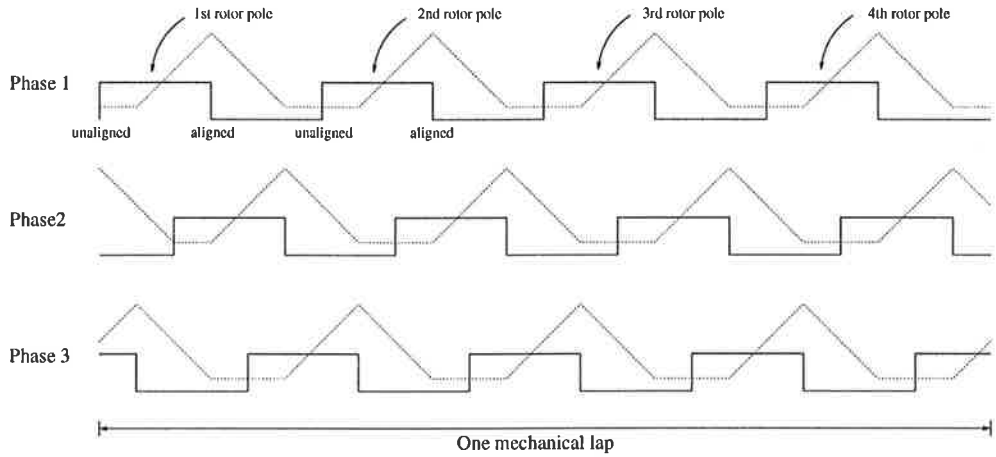


Figure 2.26 Placement of the hall-elements.

Each element gives a logic high when a rotor pole passes, which is at the unaligned position for each phase. This is reset to low when rotor pole reaches the aligned position. The signal from the three elements at constant speed is shown in Fig. 2.27.



Inductance - dotted
Hall-element signal - full

Figure 2.27 The signal from each of the three hall-elements together with the inductance for the corresponding phase. The rotor rotates at constant speed.

As we see each phase generates four pulses during one mechanical lap. When we sum up the pulses from the elements we get a pulse train as shown

in Fig. 2.28.

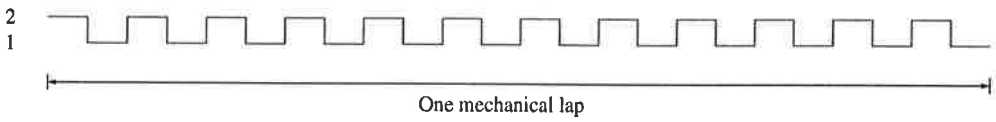


Figure 2.28 The pulse train when the pulses from all elements are summed up.

From the frequency of these pulses, we get the speed. From the speed, together with the position for certain angles (at the unaligned, and the aligned position) the position (at any time) can be derived.

Included in the sensor is also a filter after the speed signal.

3. The Model

In this chapter the model used in the simulations will be presented. The measurements and calculations to obtain the model were made on the RRA-90L by the Department of Industrial Electrical Engineering and Automation, (IEA), at Lund Institute of Technology.

3.1. The IEA Model

The model was acquired in the following way. The rotor was fixed in a certain position (angle, θ). An alternate voltage of 50 Hz was applied over the windings for one phase. The amplitude of the voltage was chosen so that the current peaks almost reached 20 A which is the maximal current allowed. When the voltage and current were stabilized they were measured 600 times during 3 periods. That is, with an interval of $\tau_s = 0.1$ ms. The flux was then calculated using (2.8) in the following way:

$$\begin{aligned} &\Psi_k = \Psi_{k-1} + (u_k - Ri_k)\tau_s && k = 1, \ldots, 600. \\ &\Psi_0 \text{ chosen arbitrary} \end{aligned}$$

where R is the resistance of the windings.

When all calculations were made the mean of Ψ was subtracted from all Ψ-values. The magnetization curves are shown in Fig. 3.1.

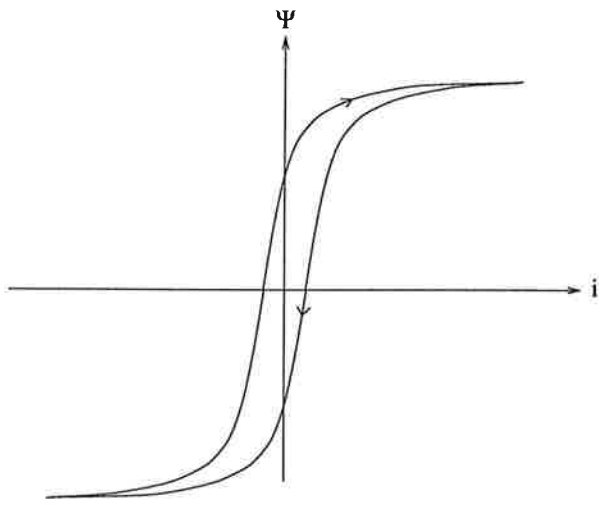


Figure 3.1 The magnetization curve for the motor at a certain angle.

The lap was run through three times. For negative values of the flux, the curve was first folded in the i axis, and then in the Ψ axis. To eliminate the hysteresis, the mean of the 12 curves was calculated.

This was repeated for 30 angles ranging from unaligned to aligned (i.e. from π to 2π). The curves were then folded in $\theta = \pi$ to cover the whole electrical

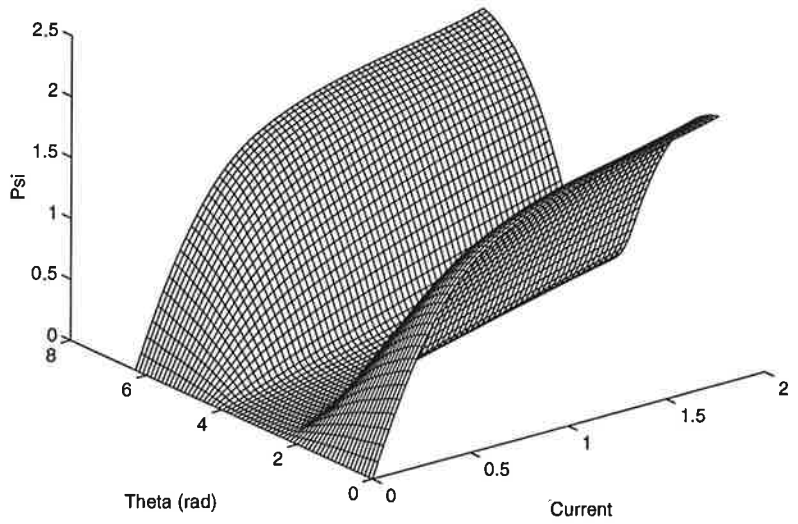


Figure 3.2 $\Psi = f(i, \theta)$.

revolution. All measurements and calculations were made in SI-units, which afterwards were converted to PU. Fig. 3.2 shows Ψ as a function of i and θ.

The magnetization curves from Fig. 2.5 are recognized here.

In the MatLab Simulink-Model the current as a function of flux and angle is used instead (as we will see later). This function is plotted in Fig. 3.3.

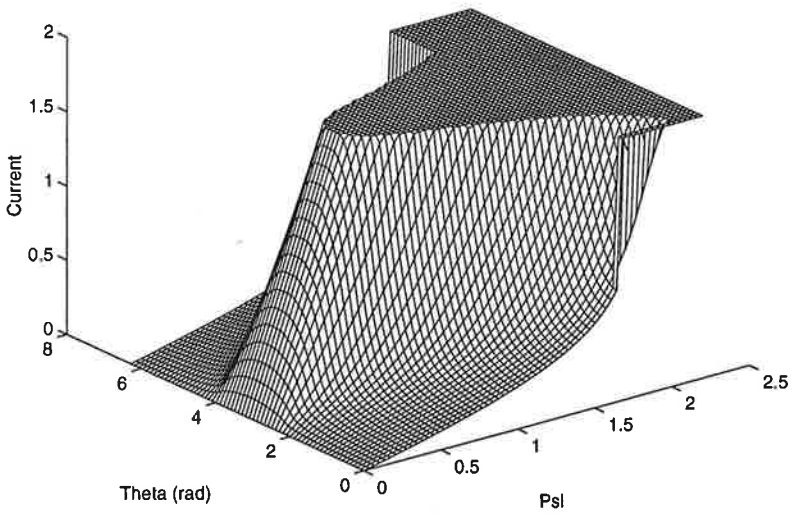


Figure 3.3 $i = f(\Psi, \theta)$.

As we have seen before the magnetization curves can also be used to calculate the instantaneous torque. This is however, very complicated and the precision could be lost when derivating and integrating in the matrices. The torque was instead computed as a function of the current and the angle using a program called ACE, developed by ABB. This program uses Maxwell's

equations and the finite element method to calculate the torque from the magnetization curves. This calculation was also done by IEA. The function is shown in Fig. 3.4. In Fig. 3.5 the same function is shown in 2-D for $\pi \leq \theta < 2\pi$ for six different currents.

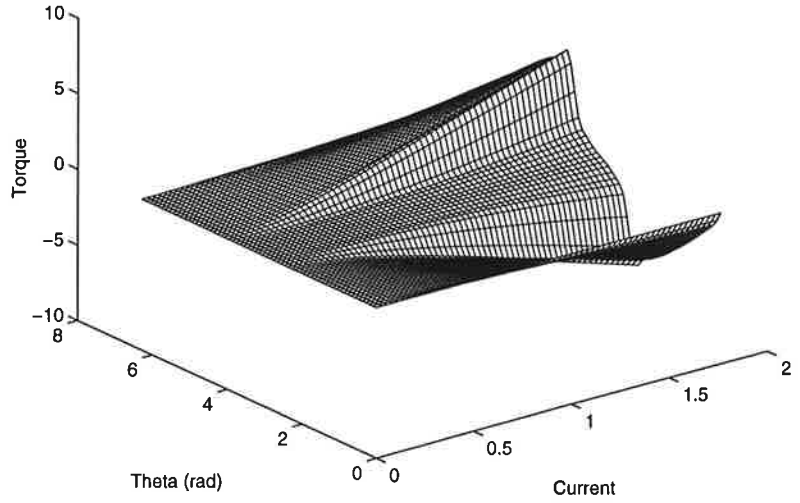


Figure 3.4 $T = f(i, \theta)$.

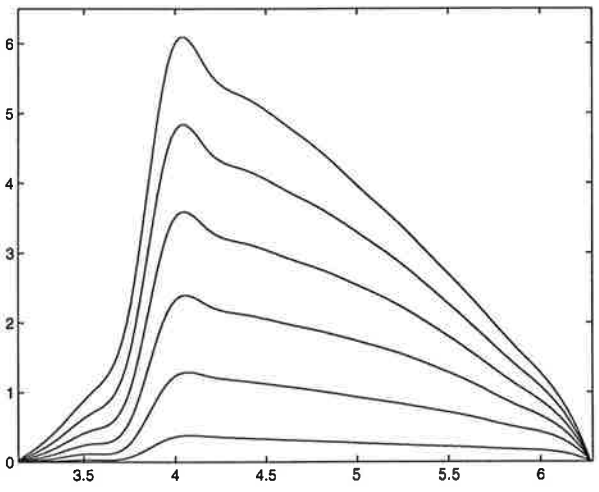


Figure 3.5 $T = f(\theta)$ for six different constant currents.

The graphs show us that even if we could achieve the ideal current described in Section 2.2, Fig. 2.10 the torque would not be linear.

Now we have three matrices (256 × 256), from which $T = f(i, \theta)$ and $i = f(\Psi, \theta)$ will be used in the MatLab Simulink-Model.

3.2. The Control Loop

Before we look at the Simulink implementation we will have a look at the control loop.

Fig. 3.6 shows the whole control loop. The blocks $i(\Psi, \theta)$ and $T(i, \theta)$ are the nonlinear functions described as matrices in the previous section.

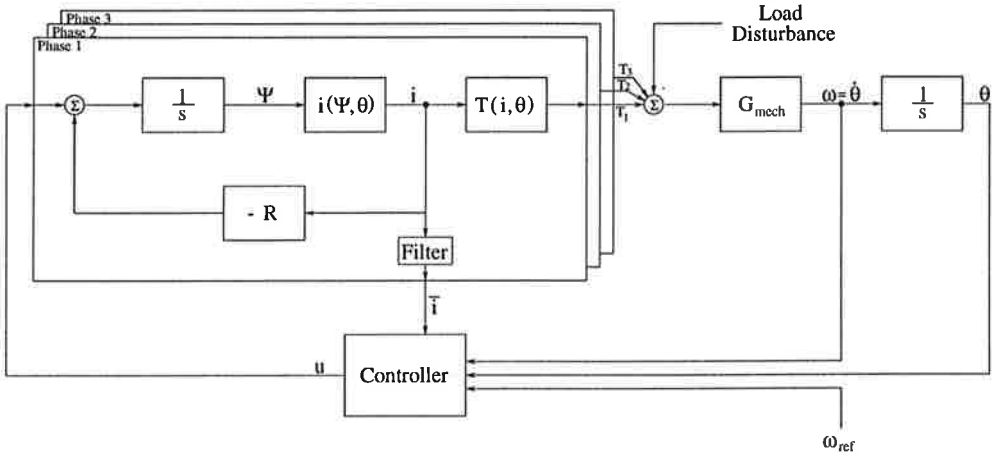


Figure 3.6 The control loop

G_{mech} is the mechanical transfer function described by (see (2.6)):

$$G_{mech} = \frac{1}{Js + d}$$

where J is the moment of inertia and d is a load proportional to the speed, i.e. viscous damping (friction). Load disturbances are added to the total torque directly.

The inner loop represents the equation for the flux, see (2.8). The current times the resistance, R, is subtracted from the voltage and integrated. From the integrator comes the flux which gives the current through the $i(\Psi, \theta)$–function. The torque is then obtained through the $T(i, \theta)$–function. To get the total torque, the torques from the different phases are summed up.

The current is filtered through a low pass filter and connected to the controller (the instantaneous current is assumed not to be measured). The filtered current can be used as a measurement of the load.

3.3. The Implementation in MatLab Simulink

This chapter gives a description of the whole system implemented in MatLab Simulink. For details about MatLab Simulink see [3].

Fig. B.1 shows the block diagram of the system with the process and the controller. The speed sensor and the converter circuit are considered ideal and are therefore not included in the system. The controller will not be described here, but in Section 4.4.

The controller uses the speed, ω, the desired speed, ω_{ref}, the position, θ, and the mean of the current, $\bar{i}$, as shown in the control loop. It also gets the instantaneous current which was supposed not to be measured. The reason for this will become clear later. The controller delivers a voltage for each phase to the process. Here the three inner loops are hidden in the 'Process' block. From this block comes the total torque, the filtered current and instantaneous current for each phase. The currents are directly fed to the controller, and the torque is connected to the mechanical transfer function, G_{mech}, after the load disturbance is subtracted. In G_{mech} the speed is calculated, which is fed to the controller, and also used in the computation of the position. The position is fed to the controller and the process.

Here follows a brief description of the blocks.

The Process Block

The process block is shown in Fig. B.2. This block contains the three inner loops (one for each phase), and the computation of the torques. The inputs are the voltage and the position angle, θ. The voltage, which contains the voltage for all three phases are connected to their respective phase. θ is re-calculated for each phase and also connected. Each 'Phase' block delivers the instantaneous torque, which is added to the total and fed to the out-port. The 'Phase' blocks also deliver the instantaneous currents which are summed up, filtered and sent out. The instantaneous currents are also fed to the out ports.

The 'Phase' Block The 'Phase' block for one phase is shown in Fig. B.3. As in the control loop, the inner loop represents the equation for the flux, see (2.8). The current times the resistance, R, is subtracted from the voltage and integrated. From the integrator comes the flux which immediately is converted to PU with the 'PUConv' block (as explained in Section 2.3). To get the current we use the matrix, that was presented in the previous section, together with the flux and the angle. First however, we must scale both of them, to know which column and row to look in. The flux ranges from 0 to something greater than zero in PU. The angle ranges from 0 to 2π. They are both scaled to range from 1 to 256 (the matrix dimension is 256×256). The 'Look up table'-block automatically interpolates linearly for non integer values. The current is used both to calculate the flux for the next sample, and to calculate the torque. The torque is computed from the scaled current and the angle in the same way as the current above.

The Scaling Blocks The 'Theta Scaling' block is shown in Fig. B.4. This block is used to scale the real value, θ, to a row or a column in a matrix. The real value is first scaled to a value between 0 and 255. Then 1 is added to make the value fit as a row or column in the 256×256 MatLab matrix. The 'Psi Scaling' and 'Curr Scaling' blocks look the same.

The Mechanical Transfer Function Block

The 'G_{mech}' block is shown in Fig. B.5. This block represents the transfer function described above. It computes the speed from the net torque. After the integrator we must go back to PU (see Section 2.3). This is done with the 'PUConv' block.

The 'Speed to Position' Block

The 'Speed to Position' block is shown in Fig. B.6. This block integrates the speed to get the position. After the integrator we must again go back to PU (=radians) using the block 'PUConv' (see Section 2.3). The position must be made periodic ranging from 0 to 2π. This is done by the 'Theta Periodic' block.

The Controller Block

The controller block takes the speed, the current (instantaneous and filtered) and the position, and delivers the voltage for each phase. This block will be explained in Section 4.4.

3.4. Approximations in the Model

Then model is based on measurements on the motor, and calculations using the commercial program ACE, developed by ABB. When generating the $\Psi - i$-curves the hysteresis effects are eliminated. The copper losses are included with the inner loop. The effects of the delay of the current one sample can be neglected. The matrices are generated by interpolation, using the 5th order spline algorithm. The two major approximations are that the converter, and especially the speed sensor are considered ideal.

The time constant of the inner loop is :

$$\tau = \frac{L}{R}$$

where the inductance depends on the position of the rotor and the current. For the worst case (the smallest time constant), we should use the inductance for the unaligned position, L_u. The time constant becomes :

$$\tau = \frac{L_u}{R} = \frac{0.15 \cdot 10^{-3}}{1.84} \text{ s} = 8.15 \cdot 10^{-5} \text{ s}$$

The sample time, τ_s should be about a tenth of this. A sample time of about 10^{-5} has proved to be good. The integrations done in the simulations use a fifth order Runge-Kutta algorithm. The inner loop (the current feedback) is made discrete, and the outer ($T \rightarrow \omega$, $\omega \rightarrow \theta$ and the controller) is made continuous. In reality it is the other way around. The inner loop is made discrete to have control of the sample time. Even if it had been made continuous in the model, it would have been sampled in the simulations. The controller is in reality event based. It should be sampled every time a new cycle begins. This means that the sample time would change with the speed. MatLab Simulink does not provide such blocks, therefore a continuous block is used instead.

4. The Control Strategy

4.1. Motor Control in General

In this section we will discuss the usual way to control the speed of a motor. Fig. 4.1 shows the control loop.

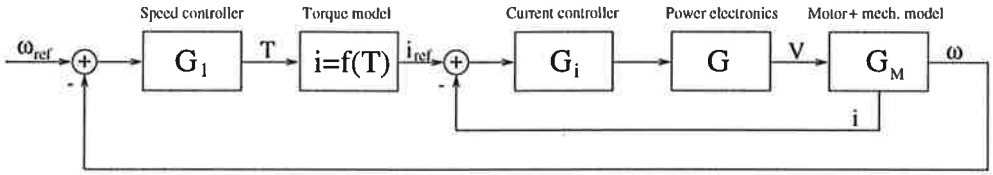


Figure 4.1 The normal way to control the speed of a motor.

The first controller computes the torque needed to get the appropriate speed. A model is then used to calculate the current that corresponds to the torque. For the DC machine this is just a constant, but for the SRM it is a nonlinear function, that depends also on the position. The calculated reference current is then compared with the measured value, and the difference is fed to the current controller. This controller computes a reference value to the power electronics which produces the appropriate voltage.

This type of controller is used for all kinds of electrical motors. For the SRM however, there is a problem. The model used to determine the torque from the current is nonlinear and depends on the position of the rotor. In our case the current is supposed not to be measured, which means that the inner current loop is lost. For more details see [4]

4.2. The Controller Used Today

There are two different controllers that are used to control the SRM, one with current feedback, and one without.

The controller with current feedback works as an on/off-controller. When the current exceeds an upper level the voltage is turned negative, and when it drops below a lower level the voltage is turned positive. The levels change with the position so that the optimal current is used for a specific task (for example minimal torque ripple). Two controllers, one using hard chopping (as above), and one using soft chopping (see Section 2.4) are implemented. These controllers works good, but requires that the current is measured. As the current changes fast, the controller must work with a high sampling frequency.

The controller without current feedback consists of two separate controllers, one for the level of the voltage, and one for the turn on angle (θ_{on}). The turn off angle is computed so that the current reaches zero at the aligned position. The level controller is a PI-controller that uses the measured speed, and the reference speed, to calculate the level. To avoid wind-up, the PI controller

is equipped with anti-reset wind-up. The θ_{on}-controller is a P-controller that only uses the speed. The faster the rotor turns the earlier the voltage should be turned on. The torque ripple for this kind of controller is very high.

In the following sections we are going to try to make the controller without current feedback better (minimize the torque ripple).

4.3. Shaping the Voltage

In this section we are going to find out how to obtain a smooth torque, without using the current. In the first section we will see which current shape gives the optimal torque. After that we will try to shape the voltage to get the specific current shape.

The Optimal Current

In the simulation model, we can of course use the current we wish. Let us see which current gives a torque, without ripple. This current can be derived from the torque matrix (see Section 3.1). This is a 256×256 matrix which contains the values for the torque for a specific rotor position and a specific current. This could be transformed to contain the current as a function of rotor angle and torque. Fig. 4.2 shows this function.

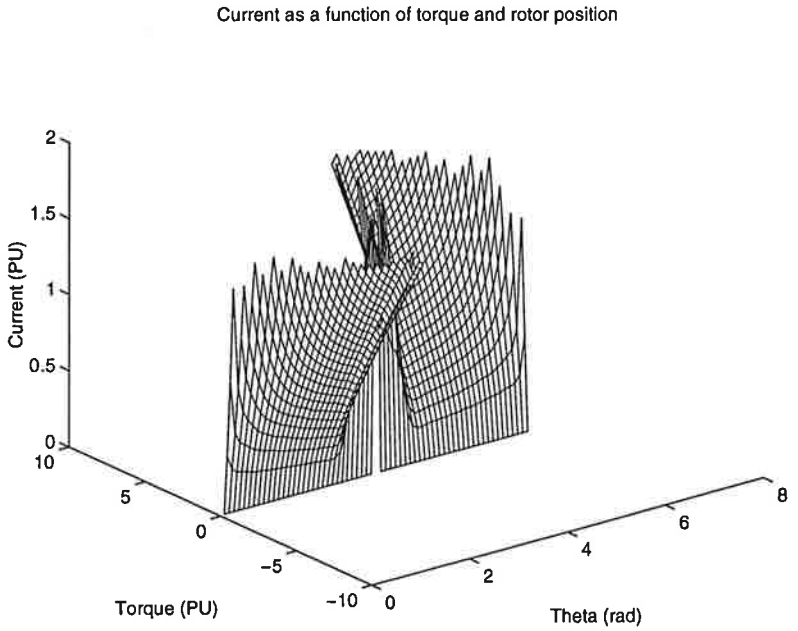


Figure 4.2 The current as a function of torque and rotor position.

As we could expect, several setups of angle and torque have no finite corresponding current.

Let us see what current we need if we choose a certain constant level of the torque. Fig. 4.3 shows the current needed for five different values of constant torque.

As we can see only a part of the motoring region ($\pi \leq \theta < 2\pi$) can be used to produce torque with finite current. The part needed to obtain torque the whole lap is $\frac{2\pi}{3}$rad (because of the three phases). Instead of making an abrupt

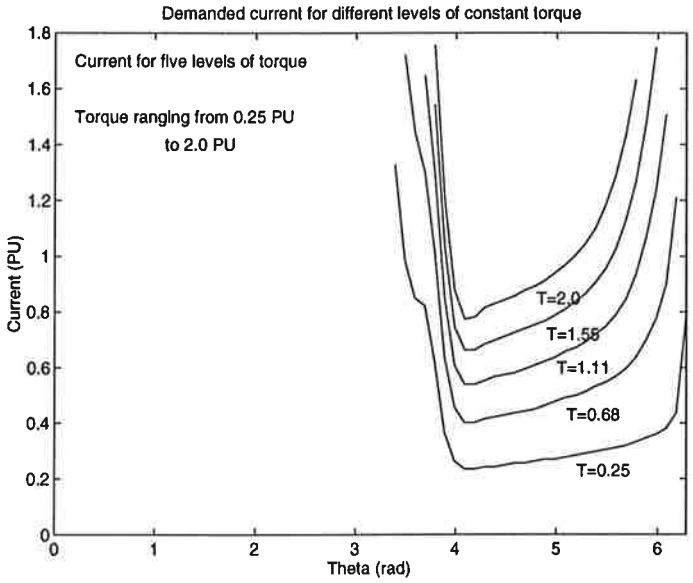


Figure 4.3 The current needed for one phase to produce five different values of constant torque. When the curves end, there is no finite current that can produce the desired torque.

transition from one phase producing torque to the next, the transition should allow two phases to produce torque at the same time. The current needed for such a transition is easier to achieve, and a small displacement of the torque for one phase does not result in a too large drop (or rise) of torque. Fig. 4.4 shows a possible way to change from one phase producing torque to the next.

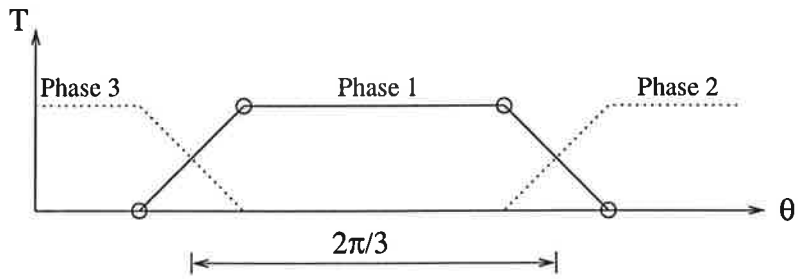


Figure 4.4 A soft transition.

The torque envelop should of course be placed so it can be obtained from a finite current. In Fig. 4.5 it is placed between 3.84 rad and 6.24 rad (overlap begins at about 3.84). The two other 'rings' are chosen at 4.14 rad and 5.94 rad. The corresponding current is shown in the upper graph. The lower graph shows a simulation with the calculated current. As we see the torque becomes quite smooth.

As we can not control the current (no current feedback), the voltage profile that gives the desired current shape, must be computed. This will be done in the next section.

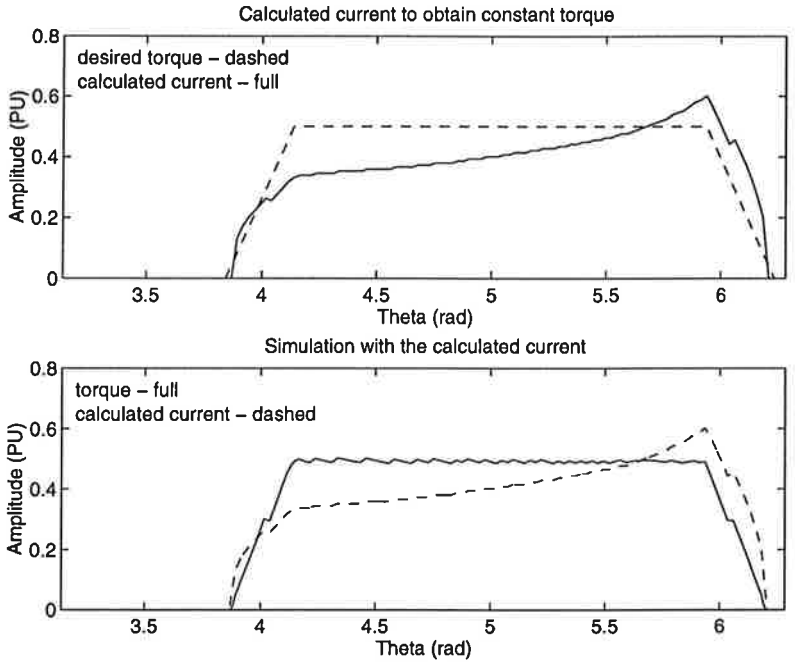


Figure 4.5 Desired torque with calculated corresponding current (upper). Simulation with the pre-calculated current (lower).

The Optimal Voltage

Now that we have the current that gives a smooth torque, we have to find the corresponding voltage. Given the current, the voltage can be derived from (2.8). The voltage becomes.

$$u(t) = R \cdot i_0(t) + \frac{d\Psi(t, i_0(t))}{dt}$$

where i_0 is the pre-computed current for a certain desired torque. It is hence a given function that only depends on θ. If we assume steady-state (ω is constant), t can be replaced by $\frac{\theta}{\omega}$. Now we can express u, i and Ψ as functions of θ (we use the same notation for the new functions).

$$\begin{aligned} u(\theta) &= R \cdot i_0(\theta) + \frac{d\Psi(\theta, i_0(\theta))}{d\theta}\frac{d\theta}{dt} \\ &= R \cdot i_0(\theta) + \omega\frac{d\Psi(\theta, i_0(\theta))}{d\theta} \\ &= R \cdot i_0(\theta) + \omega\left(\frac{\partial\Psi(\theta, i_0(\theta))}{\partial i_0} \cdot \frac{di_0(\theta)}{d\theta} + \frac{\partial\Psi(\theta, i_0(\theta))}{\partial\theta}\right) \end{aligned}$$

A program in MatLab was written to calculate the voltage from the above equation. The derivatives are calculated by taking differences in the matrices. The results are therefore not very accurate. The first term, $R \cdot i$ is easy to calculate when we know the current. The terms $\frac{\partial\Psi}{\partial i}$ and $\frac{\partial\Psi}{\partial\theta}$ are derived by taking differences in the $\Psi - i$–matrix (See Section 3.1).

Fig. 4.6 shows the voltage calculated from the current in Fig. 4.5. The equation for the voltage contains the speed, ω, which must be given. Here we have chosen $\omega = 0.5 \cdot \omega_n$.

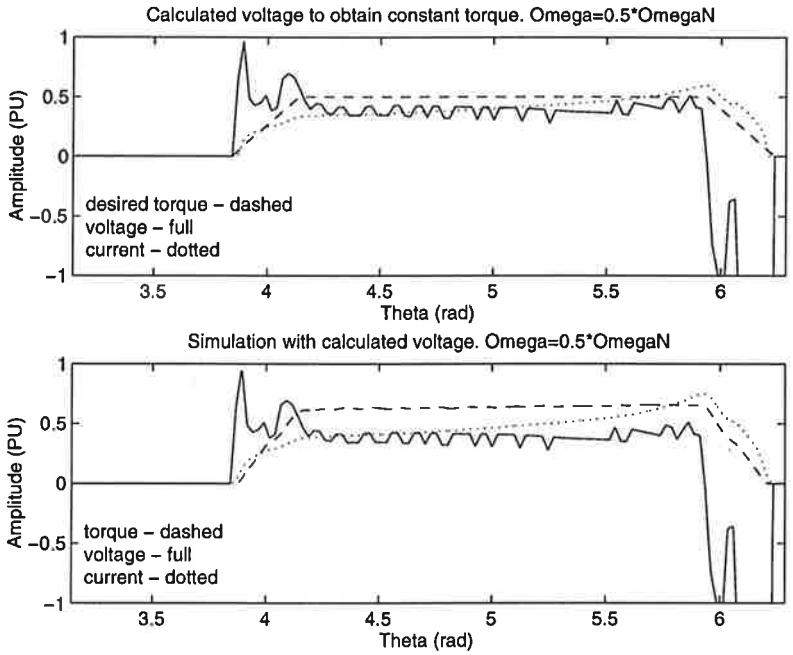


Figure 4.6 Desired torque (=0.5 PU) with corresponding current and voltage (upper). Simulation with the pre-calculated voltage (lower). $\omega = 0.5 \cdot \omega_n$.

As we see the torque is slightly larger than it should be. This is due to the errors we get from derivating in the matrices. Especially the high peaks in the beginning of the cycle causes errors for the whole cycle, because of the integration. The errors from the numerical differentiation also explains why it has a jagged form.

The voltage can also be calculated directly from the torque, using only the matrix shown graphically in Fig. 4.2. The results from those calculations are not better than the results obtained above, but they differ slightly. This indicates that we have numerical problems.

Optimal Voltage at a Low Speed Fig. 4.7 shows the same as Fig. 4.6 but at a lower speed, $\omega = 0.1 \cdot \omega_n$.

There is one problem so far. The maximum amount of voltage allowed (positive or negative) is $V_{dc} = 1.05$ PU. The negative voltage in the end of the cycle exceeds this by far. The negative voltage is about -10 PU in both cases above (this is not shown in the graph). This means that the voltage must be turned negative earlier, or we have to allow a non-zero current in the generating region.

If we turn the voltage negative earlier, the region of torque production will shrink, and the total torque will drop between the phases. To avoid this we must advance the conduction angle, θ_{on}. If we turn the voltage on before the beginning of the overlap the torque will not behave like before. As the inductance is small and almost constant in this region, the current will rise fast, and the torque production will be very low. This changes the shape of the torque, which has to be compensated.

If we instead allow a non-zero current in the generating region, the torque will be negative after the aligned position. This must be compensated by a larger torque at the beginning of the cycle for the following phase.

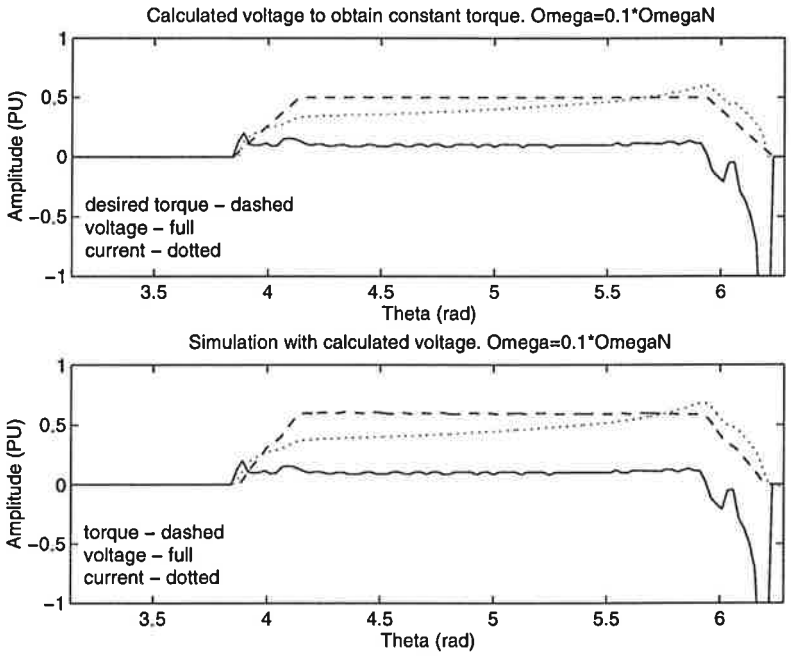


Figure 4.7 Desired torque with corresponding current and voltage (upper). Simulation with the pre-calculated voltage (lower). $\omega = 0.1 \cdot \omega_n$.

Using one of these methods means that the shape of the voltage has to be changed. For small loads (desired torque) at low speed the changes are small, whichever method we use.

Optimal Voltage at a High Speed Let us consider what happens if we increase the speed. Fig. 4.8 shows a simulation (as above) with a speed of $1.5 \cdot \omega_n$.

Now we have another problem. The positive voltage also exceeds V_{dc}. To avoid this we have to turn the voltage on earlier so that the current can start to rise before the overlap begins. This means that the current is already high when the torque production begins (at begin of overlap). Unfortunately we get a small torque before the beginning of the overlap, which has to be compensated by a lower torque in the end of the cycle. Hence, when we want to run the motor at a high speed we can not use the approach to analytically calculate the optimal voltage profile.

Optimal Voltage at a Large Load If we increase the desired torque, we get the same problem. Fig. 4.9 shows the same simulation with a higher desired torque.

In this case there is also another problem. The iron saturates totally at $\theta \approx 5.5$ rad, and the current reaches its maximum ($i_{max} = 1.76$ PU). This is due to the flux which is built up continuously during the cycle. At a certain level, and for a certain position, the iron gets totally saturated, and the current rises very fast. This is only a problem when the load/speed ratio is high. For high speeds the time to build up the flux is smaller, and therefore this does not occur.

For small loads at a low speed, the analytical approach seems to be successful. As we will see later, speeds above 0.5 PU, and loads greater than 0.5 PU

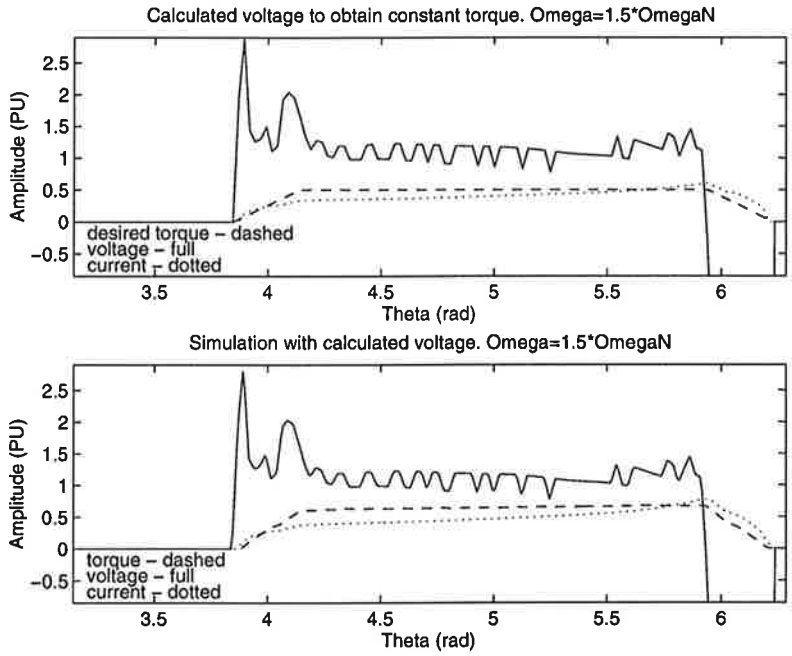


Figure 4.8 Desired torque with corresponding current and voltage (upper). Simulation with the pre-calculated voltage (lower). $\omega = \omega_n$.

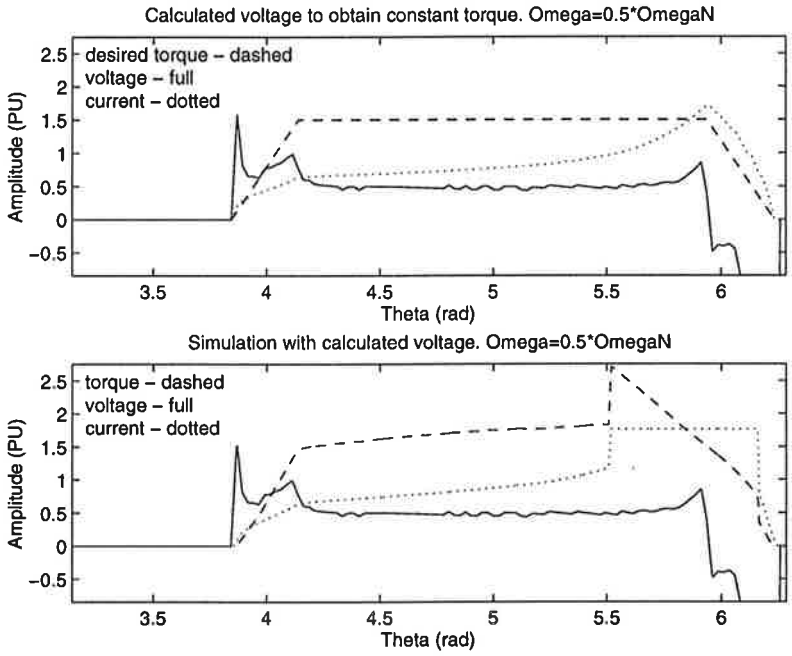


Figure 4.9 Desired torque with corresponding current and voltage (upper). Simulation with the pre-calculated voltage (lower). $\omega = 0.5 \cdot \omega_n$. $T_{desired} = 1.5$ PU

demand another approach. In the simulations above, the generated torques look good for higher speeds and greater loads, but problems arise when we limit the voltage. The analytical approach also gives a very jagged voltage due to numerical errors in the derivations of entries in the numerical matrices. Such voltage profiles would be hard to generate. We will derive suitable

approximative voltage profiles in the next section.

4.4. The Controller

The controller consists of two parts, the level controller, and the shape controller. The level controller takes the reference value for the speed and the measured speed, and computes at which level the voltage should be. The shape controller takes the speed and the torque, and computes the shape of the voltage. Since the torque is not measured this can not be used. Instead we use the filtered current as a measurement of the average torque production.

The Shape Controller

Defining Operating Points As we could see above the shape of the voltage should change as we change the speed and the torque demand, in order to obtain a smooth torque. To make control possible we can not have an optimal shape of the voltage for every setup of load and speed. We have to define operating points, at which certain voltage shapes can be used. Between the operating points the shapes should be derived in a way that gives smooth torque also here. The operating points depend on the application the motor should be used in. Emotron has suggested that the 12 operating points presented in Table 4.1 should be used :

Speed (PU)

Torque (PU)	0.10	0.25	0.50	1.00
0.25	OP 1	OP 2	OP 3	OP 4
0.50	OP 5	OP 6	OP 7	OP 8
0.75	OP 9	OP 10	OP 11	OP 12

OP = Operating Point

Table 4.1 The operating points.

Dividing the Cycle We have to define a voltage shape that can be defined by a finite number of parameters. This can be done by dividing the cycle in regions where the level of the voltage is constant. By varying the regions (angles that define where a region begins and ends) and the level in each region, a shape that looks like the optimal can be obtained. The number of parameters depend on the number of regions. Six regions including the negative in the end of the cycle has proven to be a good choice. Fig. 4.10 shows how the cycle is divided.

Each part has a level-parameter which determines the level of the voltage (in relation to each other). These are called, *level*1–*level*4 and *NegLevel*.

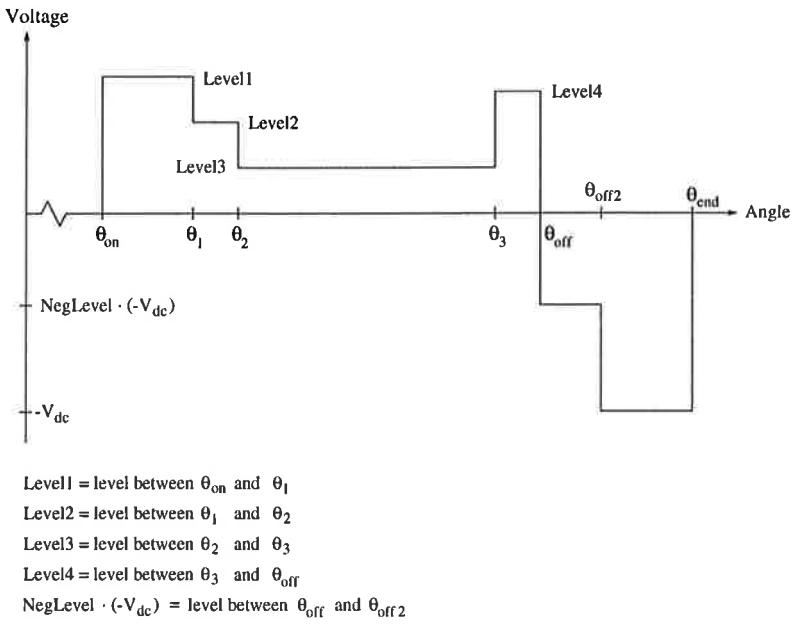


Figure 4.10 The cycle divided into six regions.

The angles where the parts start (and end) are θ_{on}, θ_1–θ_3, θ_{off}, θ_{off2} and θ_{end}. As the surfaces above and below the horizontal-axis must be the same (see Section 2.2), all parameters can not be chosen arbitrarily. Normally all parameters are chosen except for θ_{end}. When the current returns to zero, the voltage is automatically switched off, which means that θ_{end} does not have two be pre-calculated. The level for the last part with negative voltage (between θ_{off2} and θ_{end}), is fixed to $-V_{dc}$. The number of shape parameters are thus twelve, from which eleven are chosen.

Before the level of the voltage for a certain position is sent to the motor, the level is multiplied with the output value of the level controller. This is not the case with $NegLevel$. $NegLevel$ is given as parts of $-V_{dc}$, for example $NegLevel = 0.5$ means that the actual level in the region is $-V_{dc}/2$.

Defining Voltage Shapes for the Operation Points In the simulations used to find a good voltage profile for a certain operating point, the model in Fig. 4.11 is used.

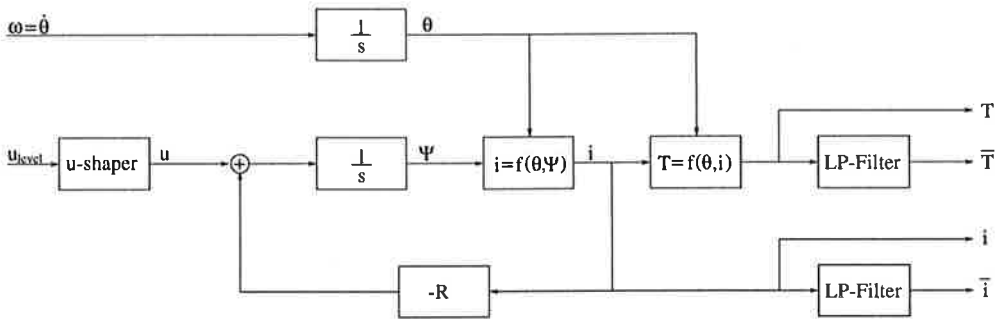


Figure 4.11 One phase of the model used to find an optimal voltage shape.

Note that only one phase is shown in the figure. The other two phases look exactly the same. The model is fed with a constant speed and a voltage level, u_{level}. This voltage level comes from the level controller in the normal case. This is the level that the voltage is shaped around. The real voltage at a certain angle (position) becomes u_{level} times the level for the region corresponding to the angle. The torque and current is measured, both the instantaneous values and low pass filtered. The voltage level is set so that the filtered torque becomes what it should be for a specific operating point.

First the optimal voltage shape is computed using the approach described above (Section 4.3). Then the eleven shape parameters are set so that the voltage is close to the optimal. A simulation is run, and if the torque ripple is too high, the parameters are adjusted, and a simulation is run again. This is done for all operating points. Fig. 4.12 shows the results for OP 2.

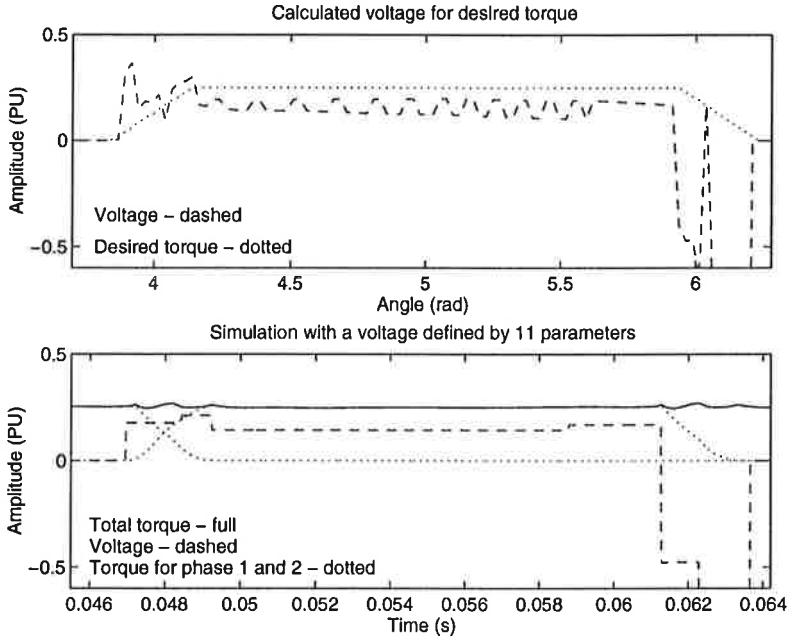


Figure 4.12 The results for OP 2 ($T = 0.25$ PU, $\omega = 0.25$ PU). The upper graph shows the calculated voltage. The lower shows a simulation using a voltage defined by eleven parameters.

As we see the torque is quite smooth except during the transitions between the phases. Complete elimination of the ripple requires more parameters.

This approach works good for the operating points : 1,2,3,5 and 6. For OP 5 and 6 we have to allow a small current in the generating region, but we do not need to advance the conduction angle. For these OPs we can therefore start with the calculated optimal voltage profiles, and improve them by making small adjustments. If we advance the conduction angle, we get a totally new torque profile. This is the case for OP 4,7,8,9,10,11 and 12. Here we have to use another voltage profile with advanced conduction angle. For all cases the conduction angle, θ_{on} is chosen to 3.22, and $level1$ is increased with speed, and torque demand. The reason for the need of advancing the conduction angle, is that the current can not rise infinitely fast. The voltage-time surface before the overlap is practically only used to build up a sufficient current before the overlap begins. The torque produced before this is very small. Fig. 4.13 shows a simulation at OP 8.

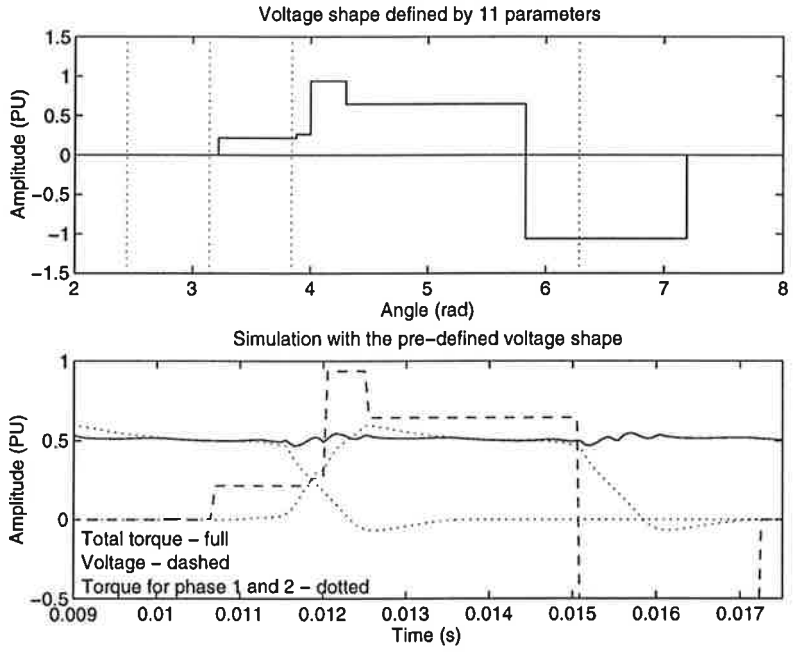


Figure 4.13 The results for OP 8 ($T = 0.5$ PU, $\omega = 1.0$ PU). The vertical lines in the upper graph represents (from left) : the end of the overlap for the previous rotor pole, the unaligned position, the beginning of the overlap and the aligned position.

The torque is also here quite smooth. The operating points for which it is difficult to get a smooth torque are those with low speed, and high torque demand. Fig. 4.14 shows a simulation at OP 10.

We can see that the torque ripple is more evident than for OP 2 and OP 8.

Scheduling the Shape Parameters If we want to change the speed or the load during runtime, we need to switch between the voltage profiles to get a smooth torque. The first thing we have to do is to find out in which operating point we are. This is determined by the torque, and the speed. The problem is that we do not measure the torque. The average (e.g. filtered) current, $\bar{i}$, however, can be used as a measurement of the torque. As we also measured the filtered current when we searched the optimal voltage profile (see Fig. 4.11), we can define the operating points from this instead of the torque. Fig. 4.15 shows how the operating points can be defined from the average current instead of the torque. We can see that $\bar{i}$ depends not only on T, but also on ω.

The switching can be done in many ways. The easiest way of doing it is simply to switch when we are midway between two operating points (the dotted lines in the figure). A better way could be to interpolate the parameters between the operating points. As we have eleven parameters and twelve operating points, this would be hard. One way of getting around this, is to keep some of the parameters constant for all OPs. Which these should be must be decided before the parameters are determined. The torque ripple at the OPs would probably be larger with this approach.

If we allow switching during step responses the speed will get jagged. Another problem is when the reference speed is close to a switching border. This could make the system oscillate. A solution to both problems is to determine the shape according to the desired (reference) speed, instead of the actual

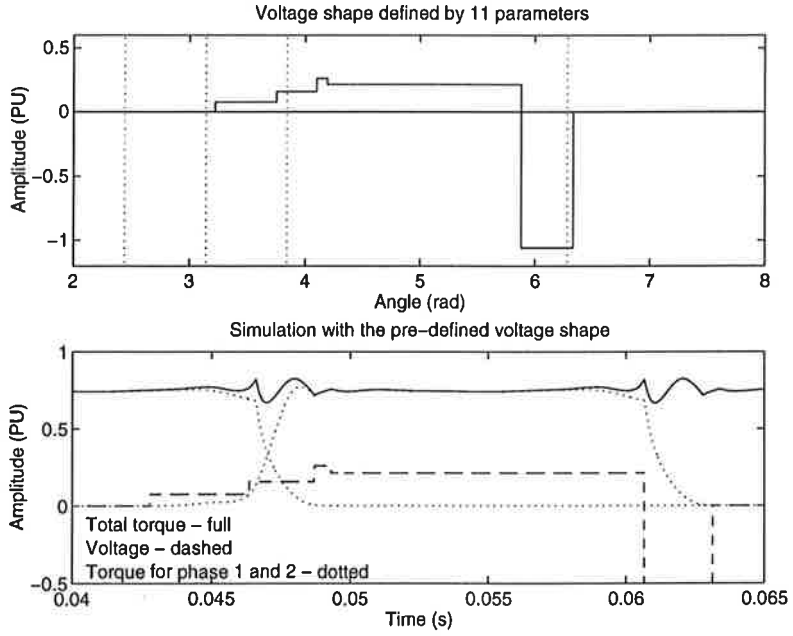


Figure 4.14 The results for OP 10 ($T = 0.75$ PU, $\omega = 0.25$ PU). The vertical lines in the upper graph represents (from left) : the end of the overlap for the previous rotor pole, the unaligned position, the beginning of the overlap and the aligned position.

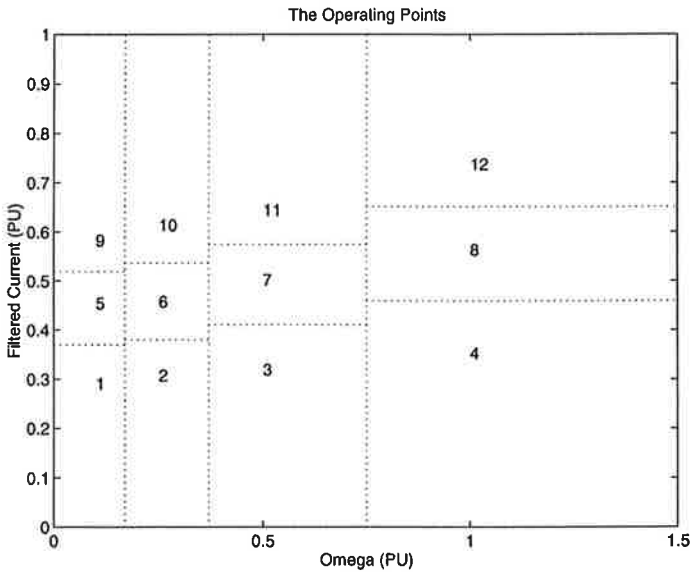


Figure 4.15 The operating points defined from the speed and the filtered current.

measured speed. If the torque load is close to a border, this will not help. A solution to this is to introduce hysteresis between the operating points. This means that we will have different borders depending on if we are inside or outside an OP. In this thesis the easiest way of switching is used, without hysteresis, but with the reference speed instead of the actual speed. The scheduling of the shape parameters is thus based on the average current and the reference value of the speed.

The Level Controller

The level controller should solve two different tasks. It should control the speed at, and close to, each operating point, and it should also control the speed during a transition from one operating point to another.

Let us first concentrate on controlling the speed around an operating point. We will use the same operating points as for the shape controller. We linearize the system around each operating point, and then design a PI-controller by the pole-placement method.

Linearization Around the Operating Points In Fig. 4.16 the inner loop, together with the voltage shaper and the mechanical transfer function, is shown.

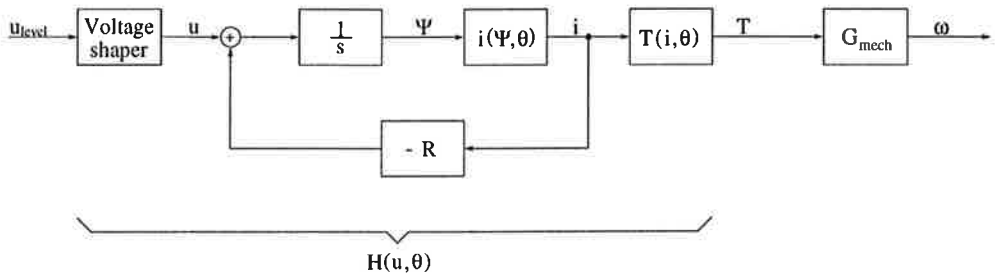


Figure 4.16 The inner loop together with the mechanical transfer function and the voltage shaper.

What we need to do is to approximate the nonlinear system marked as H in the figure. This system depends on both the voltage (shape, and level) and the position, θ. The dependency of θ is periodic, so by taking the average over one period, this dependency is eliminated. As the averaging and linearization is done around each operating point, the shape parameters are kept constant. This means that we do not have to take the dependency of the shape into account. This is also the reason for the name voltage shaper instead of shape controller in Fig. 4.16. The linearization is done by simulating the u_{level} to $\overline{T}$ dependency using the model in Fig. 4.11, for constant shape and speed. u_{level} is increased and decreased up to 10% from the level that would give the demanded torque for the operating point. By measuring the average (e.g. filtered) torque, the θ dependency is eliminated. Fig. 4.17 shows the results for OP 8.

The slope of the curve is in this case 2.45. This means that the system H can be replaced by the constant 2.45, when we are in the neighborhood of operating point 8. The same simulations are done for all operating points, and K varies from 1.29 to 21.9. What we have done here is not only a linearization, but a simplification of the inner loop, together with the voltage shaper.

Implementation of a PI-Controller After the simplification above, the control loop with a PI-controller looks like in Fig. 4.18.

With a PI Controller the closed loop transfer function becomes :

$$G_{cl} = \frac{G_c K G_{mech}}{1 + G_c K G_{mech}} = \frac{K_c K T_i s + K_c K}{J T_i s^2 + (T_i d + K_c K T_i)s + K_c K}$$

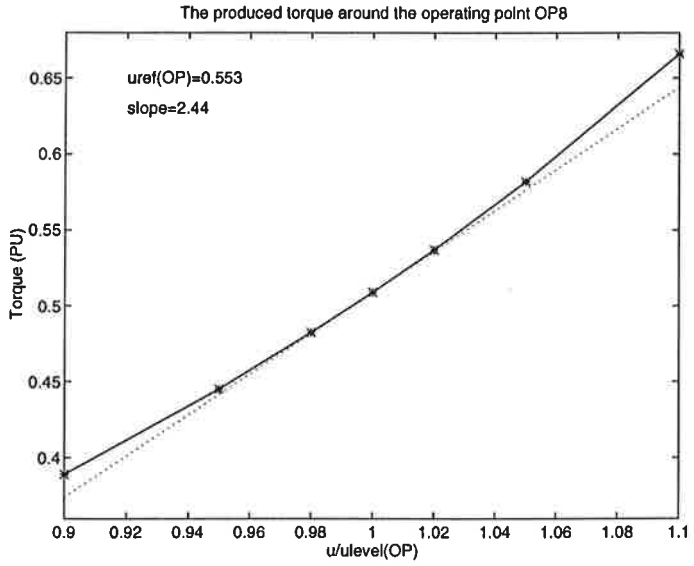


Figure 4.17 $\overline{T}$ as a function of u_{level} for OP 8 ($T = 0.5$ PU, $\omega = 1.0$ PU).

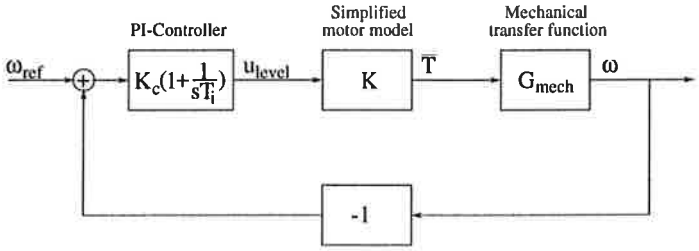


Figure 4.18 The simplified control loop with a PI-controller.

where G_c is the controller transfer function, K is the Δu_{level}-$\Delta\overline{T}$ gain around the operating point and G_{mech} is the mechanical transfer function.

This gives us the denominator polynomial :

$$s^2 + \frac{d + K_c K}{J} s + \frac{K_c K}{J T_i}$$

which should be compared with the desired polynomial :

$$s^2 + 2\zeta\omega_0 s + \omega_0^2$$

This gives us :

$$K_c = \frac{2J\zeta\omega_0 - d}{K}$$

$$T_i = \frac{K_c K}{J\omega_0^2}$$

If we assume that we have no load disturbance (i.e. all the torque losses are proportional to the speed), d can be derived from

$$d = \frac{T_{load}}{\omega}$$

where ω is the speed. The moment of inertia, J, can be chosen three times the moment of inertia of the axis. This is a realistic value for axis and load. To get a well damped system ζ is chosen to 1.0. To determine ω_0 we want to know the location of the open loop pole, s_0. This is given by :

$$s_0 = \frac{d}{J}$$

As d varies for different operating points, s_0 will also vary. To speed up the system, ω_0 should be chosen larger than the value of s_0. In order to obtain uniform dynamic response over the whole operation range, ω_0 should not vary too much between the operating points. In Table 4.2, s_0, and appropriate choices of ω_0 are presented together with d, K, and the calculated values of K_c and T_i. ω_0 is given in rad/s.

T (PU) \ ω (PU)	0.10	0.25	0.50	1.00
0.25	OP1 K = 7.33 d = 2.5 s_0 = 142 ω_0 = 150 K_c = 0.377 T_i = 0.007	OP2 K = 3.64 d = 1.0 s_0 = 57 ω_0 = 100 K_c = 0.690 T_i = 0.0143	OP3 K = 1.97 d = 0.5 s_0 = 28 ω_0 = 75 K_c = 1.083 T_i = 0.0216	OP4 K = 1.29 d = 0.25 s_0 = 14 ω_0 = 50 K_c = 1.167 T_i = 0.0343
0.50	OP5 K = 12.0 d = 5.0 s_0 = 284 ω_0 = 250 K_c = 0.316 T_i = 0.0034	OP6 K = 6.39 d = 2.0 s_0 = 114 ω_0 = 150 K_c = 0.237 T_i = 0.0086	OP7 K = 4.64 d = 1.0 s_0 = 57 ω_0 = 100 K_c = 0.541 T_i = 0.0143	OP8 K = 2.45 d = 0.5 s_0 = 28 ω_0 = 75 K_c = 0.871 T_i = 0.0216
0.75	OP9 K = 21.9 d = 7.5 s_0 = 426 ω_0 = 400 K_c = 0.299 T_i = 0.0023	OP10 K = 12.5 d = 3.0 s_0 = 170 ω_0 = 250 K_c = 0.332 T_i = 0.0057	OP11 K = 6.59 d = 1.5 s_0 = 85 ω_0 = 150 K_c = 0.572 T_i = 0.0095	OP12 K = 2.43 d = 0.75 s_0 = 43 ω_0 = 100 K_c = 1.136 T_i = 0.0157

Table 4.2 K, d, s_0, ω_0, K_c and T_i for the different operating points.

Fig. 4.19 shows the response to reference changes around OP 2, together with the response for the linear system shown in Fig. 4.18 with $F = K = 2.45$. All parameters are chosen according to Table 4.2.

Gain Scheduling When the motor states enter a new operating point the parameters should be changed. As for the shape controller the switching could be done in many ways. As it is only two parameters that changes, K_c and T_i, a linear interpolation is used. The average current, and the speed could be used to determine in which operating point we are. In order to decrease

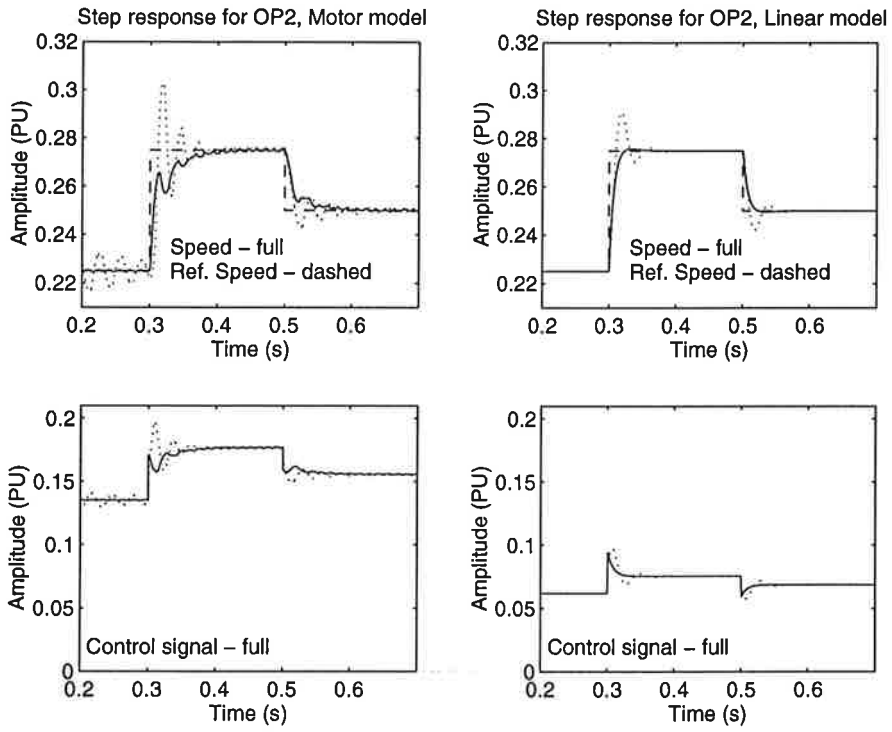


Figure 4.19 Response to reference changes in the motor model (left), and in the linear system (right) around OP 2. The lower graphs show the corresponding control signals, u_{level}. The dotted curves are the results when wrong parameters are used, in this case $K_c = 0.299$ and $T_i = 0.0023$. These are the parameters for OP 9.

the sensitivity to the measurement noise in the speed, the speed should also be filtered. The same filter as for the current is used. This means that the gain scheduling is based on $\bar{i}$ and $\bar{\omega}$. To make interpolation easier it would be better to use the average torque instead of the average current. This is because there are fewer levels of torque (3) defining the operating points than the levels of average current (9). The average torque is easy to determine if we have the average speed and the average current. When we have the average torque we can use this together with the average speed to determine K_c and T_i using interpolation between the OPs. Fig. 4.20 shows how the scheduling works.

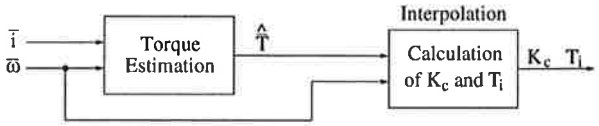


Figure 4.20 The scheduling of the parameters based on $\bar{i}$ and $\bar{\omega}$.

For more detailed information of gain scheduling in general see [1].

Anti-Reset Windup As the voltage is limited to $\pm V_{dc}$, anti-reset windup is implemented to avoid windup in the integral part of the controller. The difference between limited voltage and the voltage from the controller ('u generator' in the model) is fed back to the controller. If it is negative (i.e. the

voltage is cut), no updating of the integral takes place. The level from the controller should never be negative. Negative voltage is only used to eliminate the flux at the end of each cycle, and this is not handled by the PI-controller. Therefore the signal immediately after the controller must be limited and sent back as above. Normally more than one phase is active (has non zero voltage) at a time, and if the voltage for any phase is cut, the integral is not be updated.

The Complete Controller

Fig. 4.21 shows the complete controller in block diagram form.

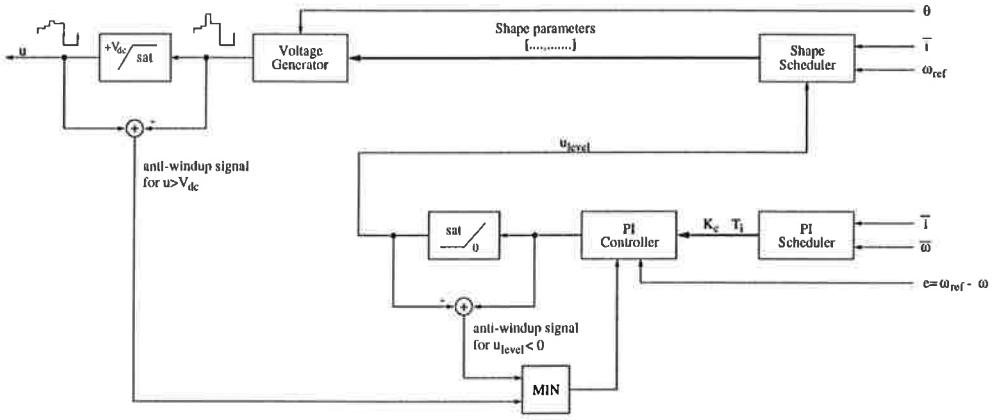


Figure 4.21 The complete controller.

A close–up of the PI-Controller is shown in Fig.4.22. First K_c and T_i are scheduled based on the filtered current, and the filtered speed.

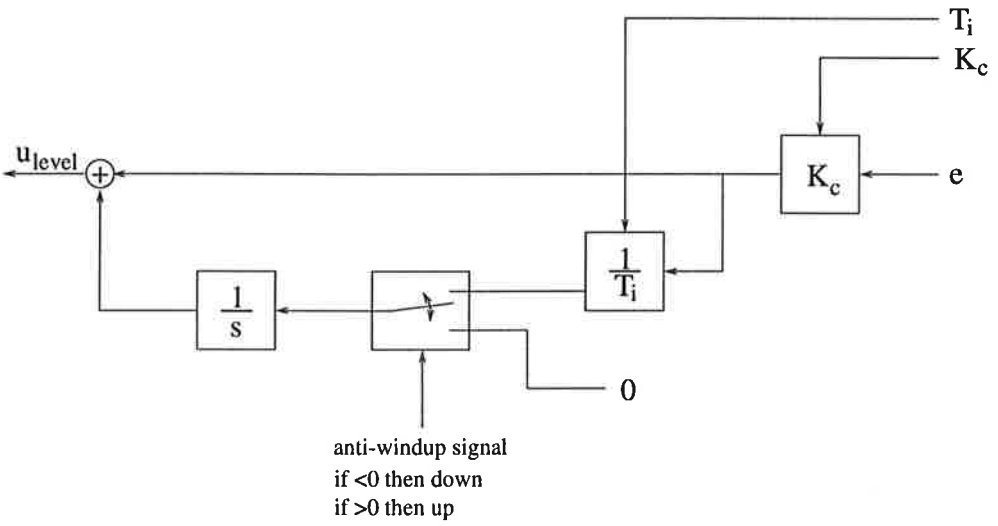


Figure 4.22 The PI-Controller block.

Then the speed error is calculated from ω_{ref} and ω, and is immediately multiplied with K_c. This signal is then sent to the integral part of the controller.

If the voltage is between its limits (greater than zero from the controller, and less than V_{dc} from the u generator), the integrator is updated. The output from the integrator is then added to the error times K_c and sent to the limiter. This only checks if the voltage is greater than zero.

The level of the voltage is fed to the shape controller, but is not really used here. For conveniency, the level is added to the shape parameters. The shape parameters are determined when the OP is known. This is derived from ω_{ref} and $\bar{i}$. The shape parameters are sent to the voltage generator together with the position. There is one voltage generator for each phase (only one is plotted in the graph). The voltage generator compares the position (angle), with the angles in the shape parameters (θ_{on}–θ_{off2}), and then delivers the right voltage.

Implementation in MatLab Simulink

In Section 3.3 the implementation of the motor in Simulink was explained. Now we are going to see how the controller is implemented. The controller block is shown in Fig. B.7. We can see that it consists of three parts, the shape controller, the PI-controller, and the u generator.

The Level Controller Block The task of the level controller is to put the voltage on the appropriate level according to the speed error. The PI-controller described above is used for this. Fig. B.8 shows how this is implemented. It is easier to make interpolation if we use the average torque instead of the average current. This is because there are fewer levels of torque (3) than of current (9). To calculate K_c and T_i, we first calculate the average torque from the filtered current and the filtered speed. This is done in a look-up table, which uses interpolation. Then this is used together with the filtered speed to calculate K_c and T_i. This is done using the same type of look-up table. The values of K_c and T_i are limited before they are used. The rest of the flow of information is described in above.

The Shape Controller Block The shape controller, explained in the beginning of this section, is used to give the voltage a certain shape. Fig. B.9 shows how this looks in Simulink. The block 'OPGen' determines the operation point, and the block 'ShapeGen' delivers the shape parameters. Both blocks are S-functions (see [3]), which contains if-then statements.

The 'u generator' Block The 'u generator' block gets the shape parameters, the position and the instantaneous current. A close-up of this block is shown in Fig. B.10. Here we can see that the block is divided in three parts, one for each phase. The difference between this block and the shape controller is that the shape controller determines an appropriate amplitude of the voltage for each position (independent of which phase), while the 'u generator' block feeds out the instantaneous amplitude of the voltage for each phase continuously, taking the shape parameters and the position into account. Here we must also recalculate the position for each phase.

Let us go even deeper and see what is under the 'u generator' for a certain phase. Fig. B.11 shows a picture of this. Again we have three parts, the 'Inner u-shaper', the 'Outer u-shaper' and the 'Current check'.

The 'Inner u-shaper' sets the voltage amplitude for angles between θ_{on} and θ_{off2}. Fig. B.12 shows the details of this block. It simply compares the position with the positions determined by the shape controller (θ_1–θ_{off}) and sets the voltage to levels computed by the shape controller and the PI-controller.

The 'Outer u-shaper' sets the amplitude for angles smaller than θ_{on} and greater than θ_{off2}. This means that it delivers the voltage computed by the 'Inner u-shaper', V_{dc} or 0, depending on the position. As we can see the instantaneous current is used here. This is because we need to know when to turn the negative voltage off, which should be done when the flux and current returns to zero. In the real motor this is done by itself. The current does not have to be measured. When the current becomes zero the voltage can not be negative anymore (see Section 2.4). Fig. B.13 shows a close-up of this block.

The third block, the 'Current Check', is a safety device which also uses the current. Fig. B.14 shows a close-up of this block. It should save the motor from overload. If the current exceeds a certain amount, the voltage should be switched off. When the current then returns to a lower amount, the voltage should be switched on again. The actual switching takes place outside the block, in the 'Safety Switch'. This is also implemented in the real motor.

Before the voltage is sent to the process it is limited between $-V_{dc}$ and $+V_{dc}$. The converter can not produce a voltage outside this range. The difference is sent back to the PI-controller which uses the information to determine whether the integral part should be updated or not.

5. Simulation Results

In this chapter some simulation results will be presented. All simulation are done on the model described in Chapter 3.

In the first simulation, the torque ripple for the new controller is compared with what we should get if we used the voltage controller used today. The comparison is made in OP 1. Fig. 5.1 shows the result. We can see that even as the voltage profiles are almost the same, there is a big difference in the torque ripple. This means that the torque is very sensitive to changes in the voltage profile, and robustness could be a problem. The small negative voltage in the "optimal" voltage profile is very important. The result for OP 8 shows an even greater difference, see Fig. 5.2. This is due to the higher speed. We can see that the difference is now very big also between the two voltage profiles. The level of the voltage before the beginning of the overlap is very important. It is also of great importance, that we allow a non-negative current in the generating region.

The next simulation shows how the controller works inside an operating point. In Fig. 5.3 a step response inside OP 4 is shown. We can see that a change in the reference speed is no problem for the PI-controller. We can also see that the torque ripple is smallest when we are in the middle of the operating point. The same simulation on the linear system (see Section 4.4) Fig. 4.19 would look almost the same.

In Fig. 5.4 we can see how the torque looks when we make a step in the reference speed far away from the center of an operating point. We can see that the torque ripple is very bad. This is because we are midway between two operating points, as we can see in the trajectory plot. This is one of the worst cases we can get if we are inside the working area (i.e. $\omega \leq 1.0$ PU and $T_{load} \leq 0.75$ PU). The vertical movement in the trajectory plot is because when we increase the speed, the load increases as d is constant. Fig. 5.5 shows how the torque looks when we make a step in the torque load far away from the center of an operating point. The torque is better than in the previous case. This is because we are closer to the center of the OPs, and OP 3 and OP 6 are not as sensitive as OP 8 and OP 4.

The results for these simulations are not perfect, but the results with the old controller would be even worse.

In the next simulation, the process is run in 1.05 seconds. Steps in the reference speed and in the viscous damping are done, to cover as much of the operating area as possible. Fig. 5.6 shows the result. We can see that we get a lot of torque ripple in the beginning. Here we are in OP 9, which is the OP that gives the most ripple. It is hard to find a voltage shape for low speeds when the load is high. In the other operating points, the torque ripple is acceptable.

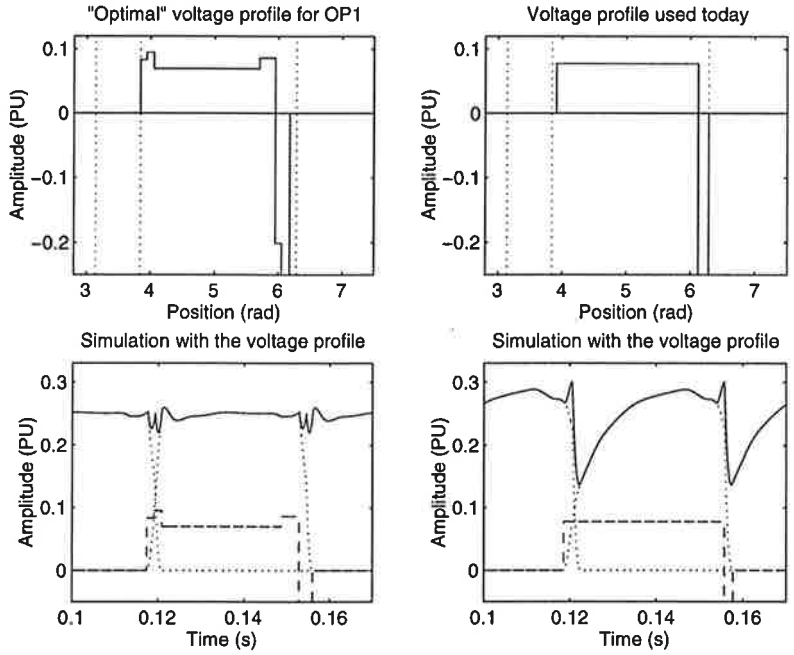


Figure 5.1 Top left : The voltage shape we should get with the new controller. Bottom left : The total torque (full) and the torque for each phase (dotted), together with the voltage (dashed). Top right : The voltage profile that would be used with the old controller. Bottom right : The total torque (full) and the torque for each phase (dotted), together with the voltage (dashed).

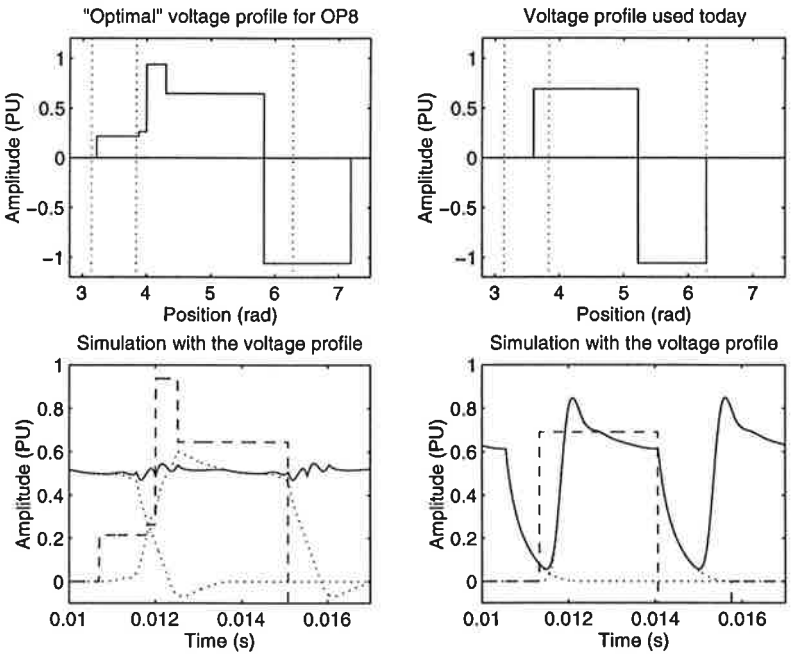


Figure 5.2 Top left : The voltage shape we should get with the new controller. Bottom left : The total torque (full) and the torque for each phase (dotted), together with the voltage (dashed). Top right : The voltage profile that would be used with the old controller. Bottom right : The total torque (full) and the torque for each phase (dotted), together with the voltage (dashed).

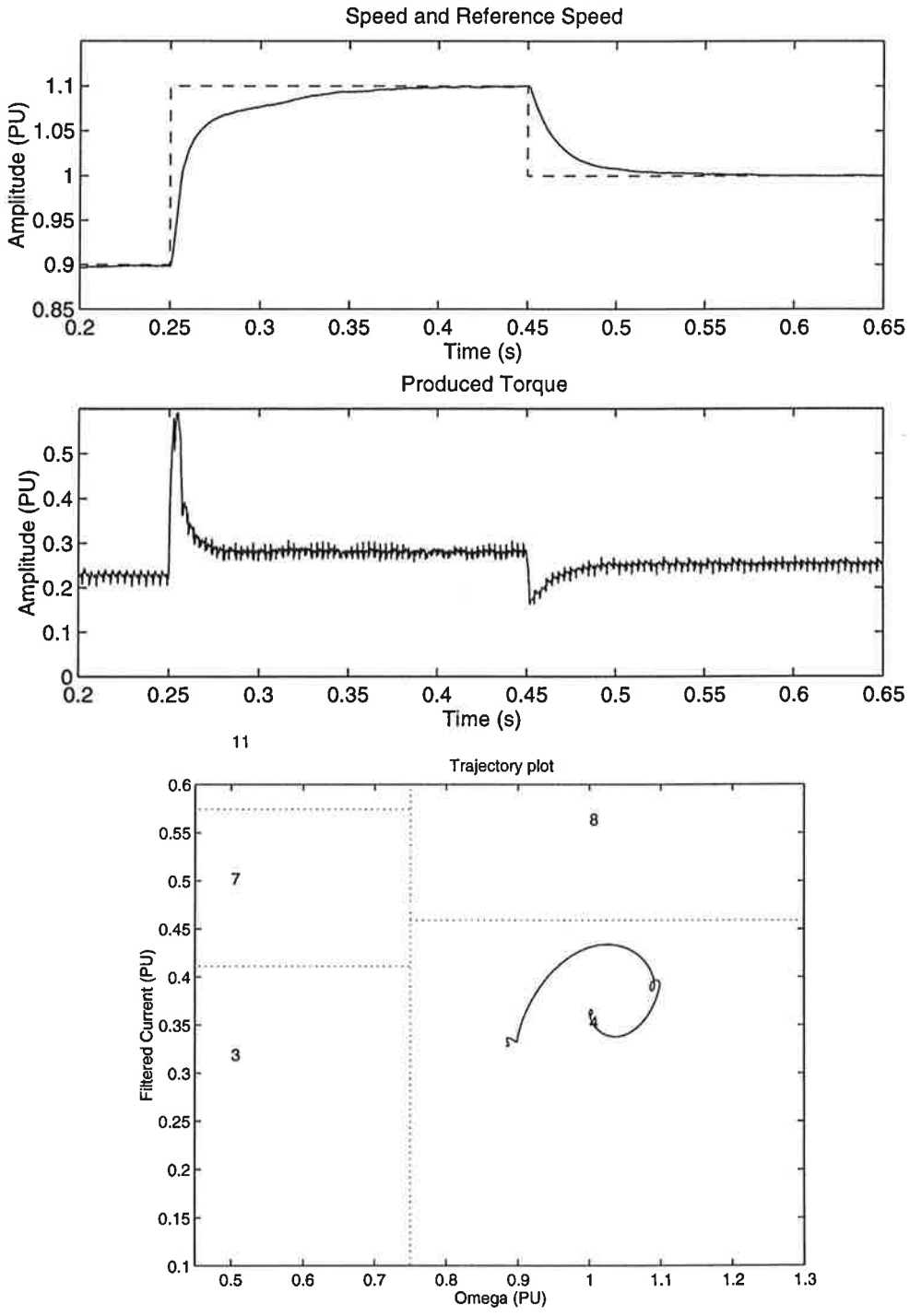


Figure 5.3 Upper : The speed (full) and the reference speed (dashed). Middle : The produced torque. Lower : The trajectory path.

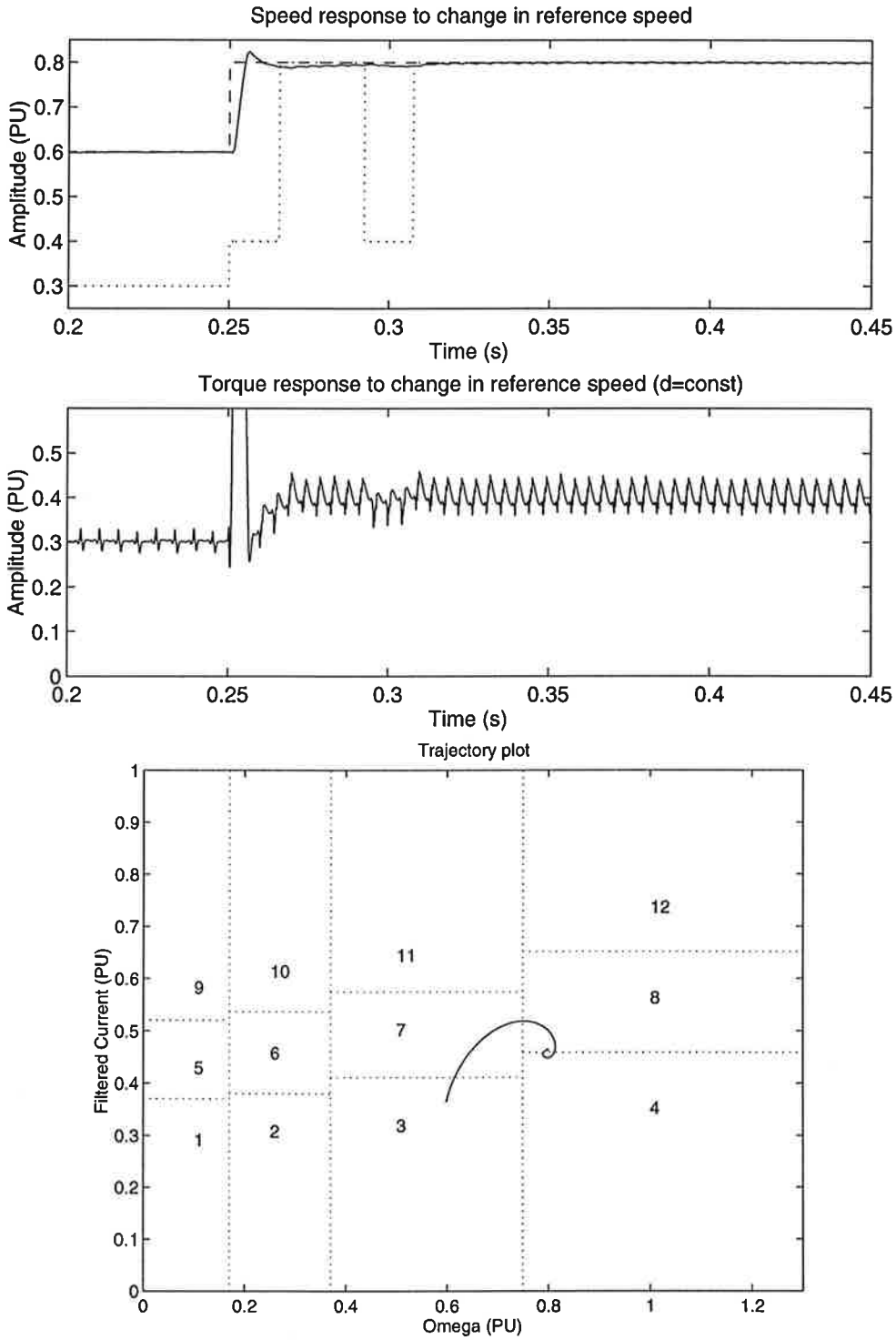


Figure 5.4 Upper : The speed (full) and the reference speed (dashed). Middle : The produced torque. Lower : The trajectory path.

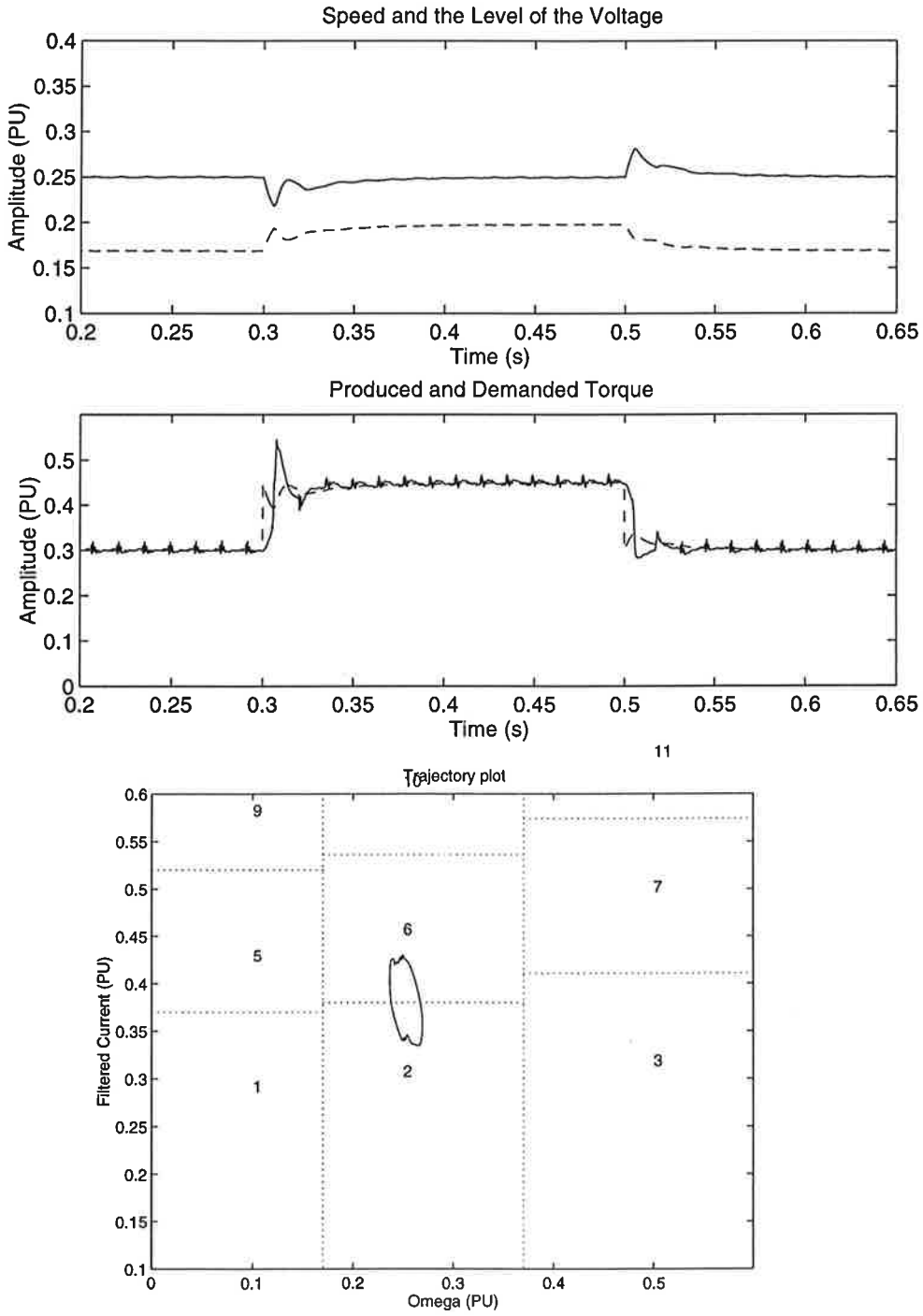


Figure 5.5 Upper : The speed (full) and the control signal u_{level} (dashed). Middle : The produced torque (full) and the torque load (dashed). Lower : The trajectory path.

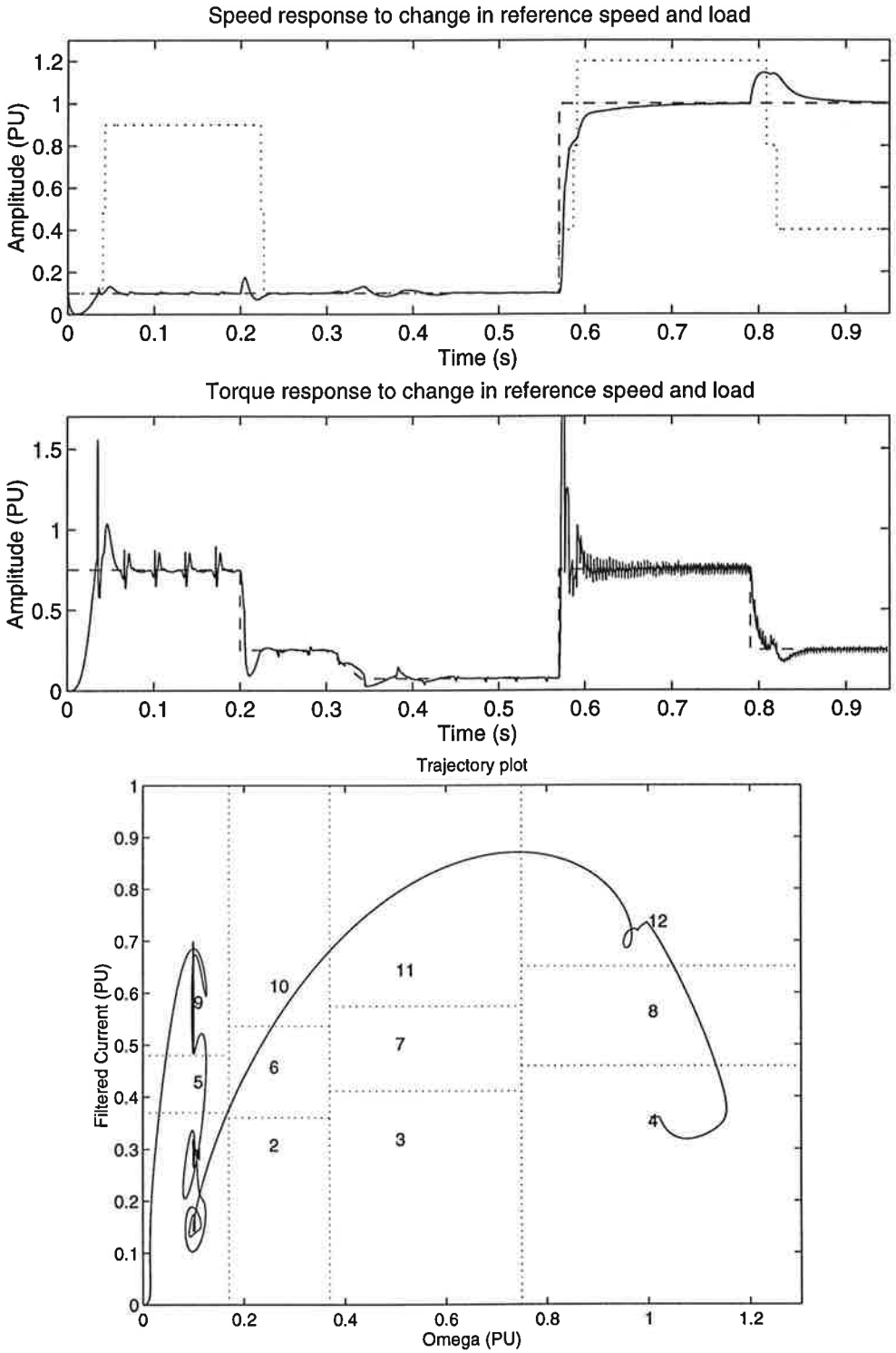


Figure 5.6 Upper : The speed (full) and the reference speed (dashed). Middle : The produced torque (full) and the torque load we would have if the speed was equal to the reference speed (dashed). Lower : The trajectory path.

6. Conclusions and Suggested Improvements

Here follows conclusions of what has been presented in the thesis. A discussion of what can be improved is also given here.

This thesis starts with a short introduction to the switched reluctance motor, and several typical characteristics are verified by short simulations done on the model of the motor. This model is presented in Chapter 3.

Then we derive a control strategy which is based on changing the shape of the voltage according to the speed and the torque in order to get a smooth torque. Operating points are defined from these variables. As the torque is not measured, we use the filtered current instead. A parameterization of the torque is presented, that uses piecewise constant levels to define the shape. The shape controller is a feed forward controller, which does not take the speed error into account. A PI-controller is implemented that uses the speed error to determine the level the voltage should be shaped around. To obtain good control characteristics for all operating points, gain scheduling is used. The PI-controller is also equipped with anti-reset windup.

The simulation experiments in Chapter 5 shows that the control strategy works well. There are however, some things that could be improved :

- The switching of voltage shapes between the operating points is made as easy as possible. Switching takes place when we are midway between two operating points. A lot can be done to improve the switching.
    1. We can uses hysteresis to avoid oscillations when we are close to a switching point.
    2. Another thing is to use interpolation when we are between the operating points. This switching could be made linear or of a higher order. One way of implementing this is to use fuzzy logic.
    3. We could also use another parameterization of the shape. If we want to use interpolation we should try two minimize the number of parameters. A completely other parameterization, where we do not use piecewise constant levels to define the shape, is also possible.
- One way to determine the shape is to use neural networks. The network could take the position, the speed and the filtered current, and compute the level. The "optimal" profiles presented in Section 4.3 could be used as training data.
- The operating points cover most of the possible working area for the motor. In certain applications it could be possible to limit this area, or to concentrate on a smaller region. The operating points should then be concentrated to this region in order to increase the quality of the torque.
- Efficiency is not discussed at all in this thesis. Of course the efficiency is affected by the shape of the voltage. This should be investigated before a possible implementation.
- One big problem with the shape controller is that it is a feed forward controller. This causes robustness problems. A robustness analysis of the controller is essential if it should be implemented.

- An analytical approach is not discussed in this thesis. The magnetization curves can easily be approximated with analytical functions. If this is done, the optimization problems can be solved analytically.

If these statements are taken into account, implementation of the controller for the motor could probably be done. Of course a lot of new problems will come up if this is done. This thesis should be seen as an introduction of how a controller without current feedback could be implemented.

7. Bibliography

[1] K.J. Åström and B. Wittenmark. *Adaptive Control*, pages 390–418. Addison-Wesley, 1995.

[2] Jan Boivie. Losses in a small switched reluctance machine, 1995, Department of Electrical Engineering and Information Technology, Royal Institute of Technology.

[3] Cheryl Chekoway et.al. *Simulink – a Program for Simulating Dynamic Systems*, 1992, Mathworks.

[4] Gustaf Olsson et.al. *Elmaskinsystem*, pages 293–305. KF–Sigma, 1994.

[5] T.J.E. Miller. Introduction – switched reluctance drives. In *Switched Reluctance Motor Drives – a reference book of collected papers*, October 1988.

[6] T.J.E. Miller. *Switched Reluctance Motor and Their Control*, pages 1–51. Magna Physics Publishing and Clarendon Press, 1993.

[7] Lars Sjöberg. The Switched Reluctance Motor – A Conference Paper and Journal Article Review. Technical Report TEIE–7081/1–20, Department of Industrial Electrical Engineering and Automation, Lund Institute of Technology, 1995.

A. The Data of the RRA-90L

The Motor Parameters in SI-units

Voltage (V_{dc})	= 311 V
Rated current (I_n)	= 6.55 A
Maximum current (I_{max})	= 20 A
Impedance (R)	= 1.84 Ω
Nominal speed (n_n)	= 1420 rpm
Nominal angular velocity (ω_n)	= 148 rad/s (594.8 rad/s electrical)
Maximum speed (n_{max})	= 2500 rpm
Rated Torque (T_n)	= 11.2 Nm
Moment of inertia (J)	= $8.816 \cdot 10^{-4}$ kgm^2

The Normalization Factors

$\Psi_{base} = \frac{B_{max} S N q}{2\pi}$ Vs = 0.4937 Vs

$\omega_{base} = \frac{1420 \cdot 2\pi \cdot N_p}{60}$ s^{-1} = 594.8 s^{-1}

$V_{base} = \omega_{base} \Psi_{base}$ = 293.7 V

$I_{base} = I_n \sqrt{3}$ = 11.34 A

$Z_{base} = \frac{V_{base}}{I_{base}}$ = 25.90 Ω

$P_{base} = V_{base} I_{base}$ = 3331 W

$T_{base} = \frac{P_{base}}{\omega_{base}}$ = 5.6 Nm

$J_{base} = \frac{T_{base}}{\omega_{base}^2}$ = $1.583 \cdot 10^{-5}$ kgm^2

where

B_{max}	=	2.0 T is the maximum flux density
S	=	$2.0515 \cdot 10^{-3}$ m^2 is the surface of a stator pole seen from the direction of the flux
N	=	252 is the number of turns/phase of the winding
q	=	3 is the number of phases
I_n	=	6.55 A is the rated current

The Motor Parameters in PU

V_{dc}	= 1.06
I_n	= 0.58
I_{max}	= 1.76
R	= 0.071
ω_n	= 1
T_n	= 0.5
J	= 3.48

B. The MatLab Simulink Model

Here follows pictures of all the Simulink blocks described in Section 3.3 and 4.4.

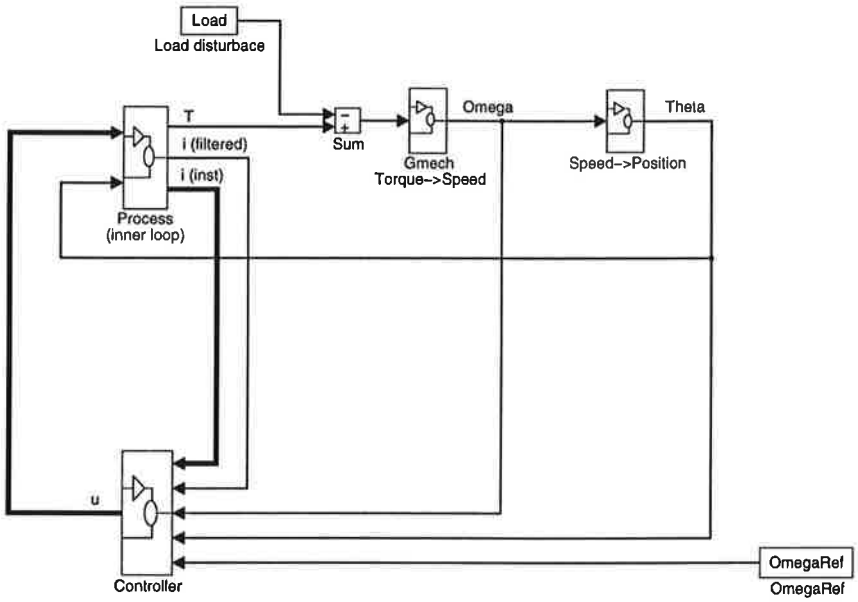


Figure B.1 The system implemented in MatLab Simulink.

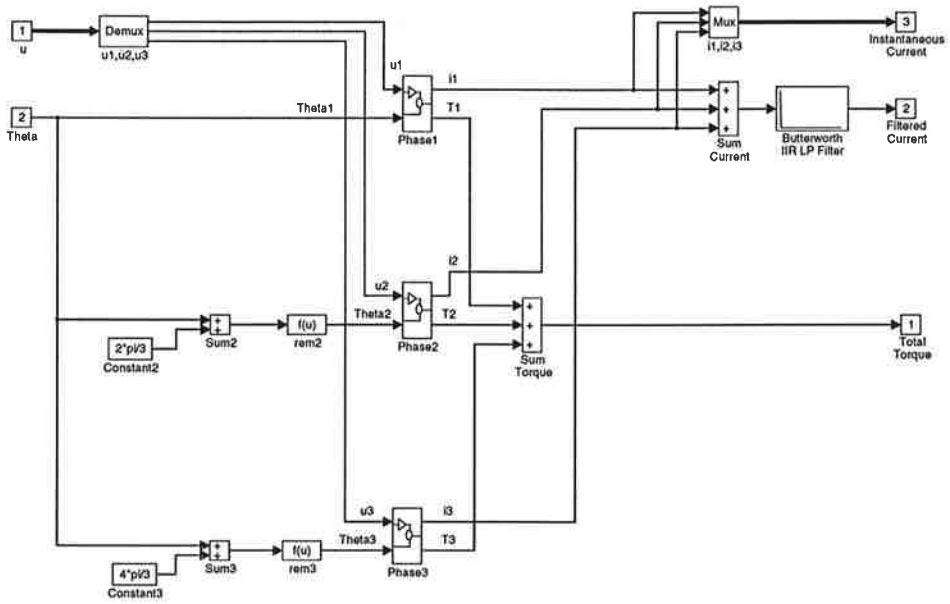


Figure B.2 The process.

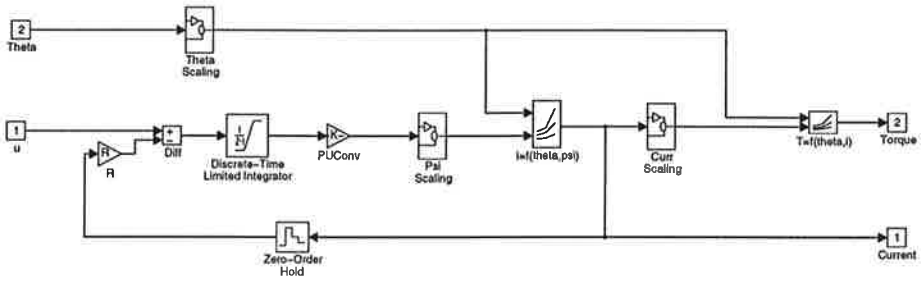


Figure B.3 The inner loop for one phase.

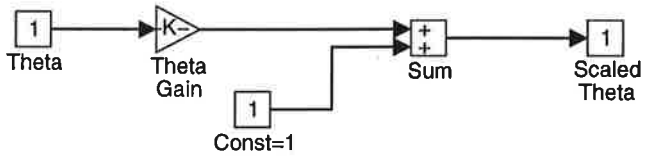


Figure B.4 The 'Theta Scaling' block.

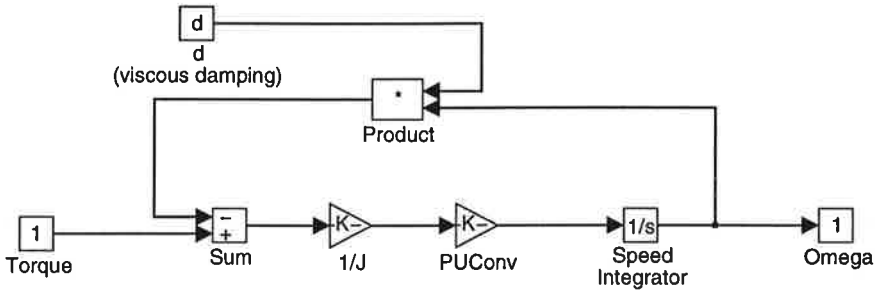


Figure B.5 The G_{mech} block.

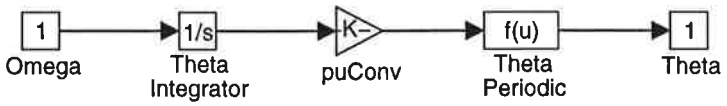


Figure B.6 The 'Speed to Position' block.

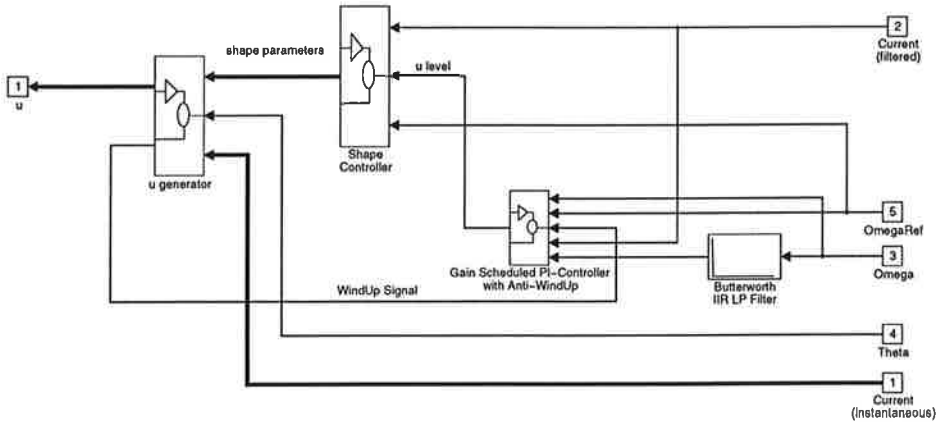


Figure B.7 The control block.

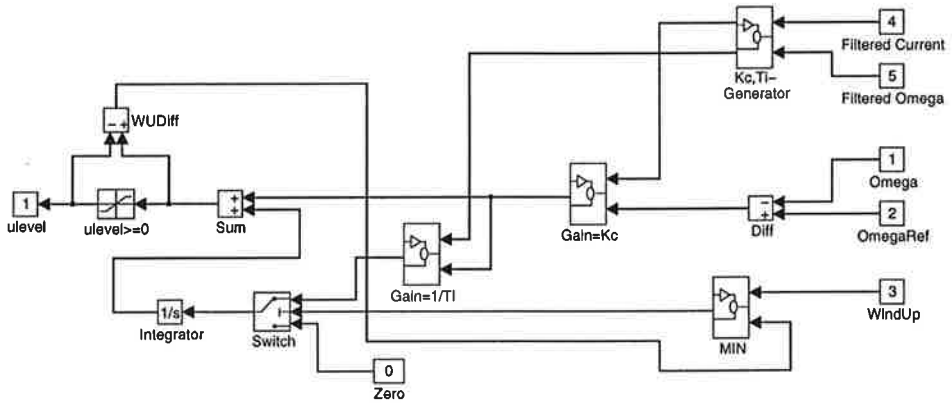


Figure B.8 The PI-controller.

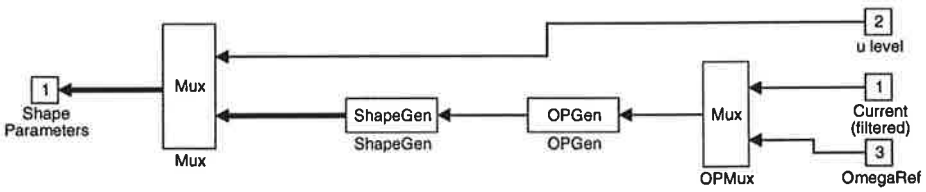


Figure B.9 The shape controller.

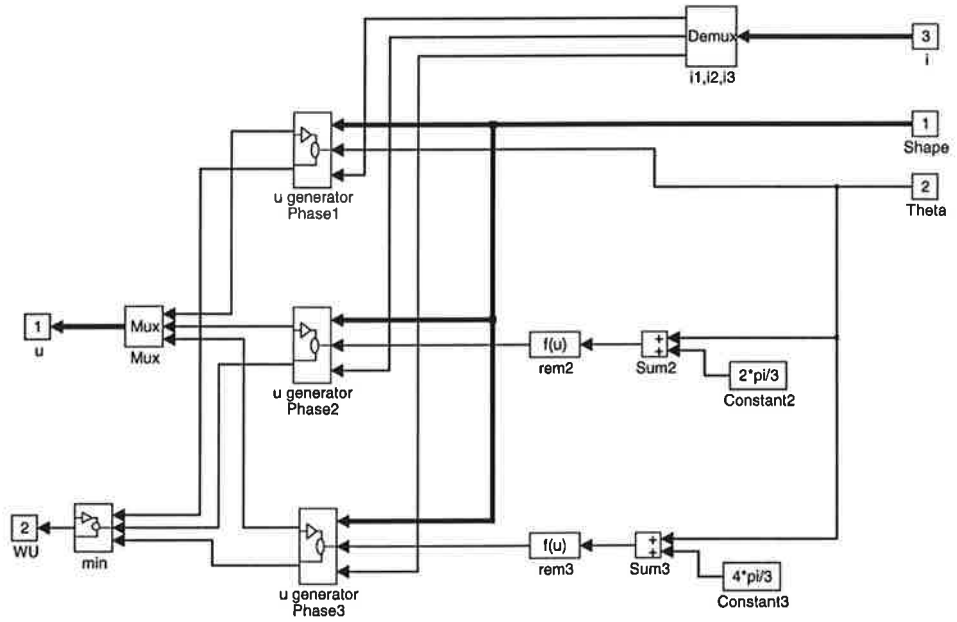


Figure B.10 The 'u generator' block.

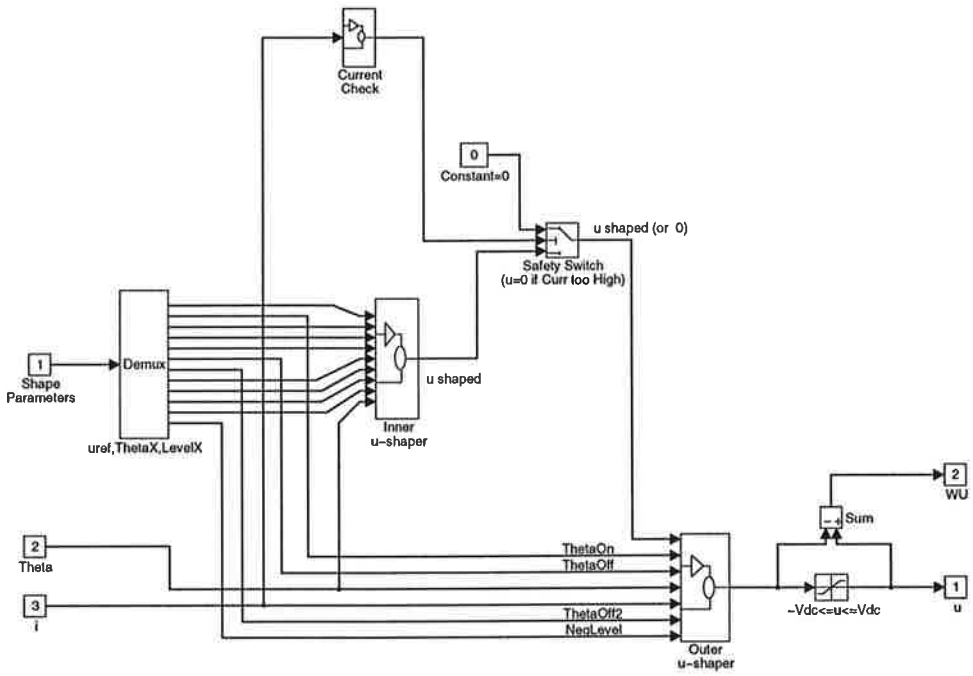


Figure B.11 The 'u generator' phase1 block.

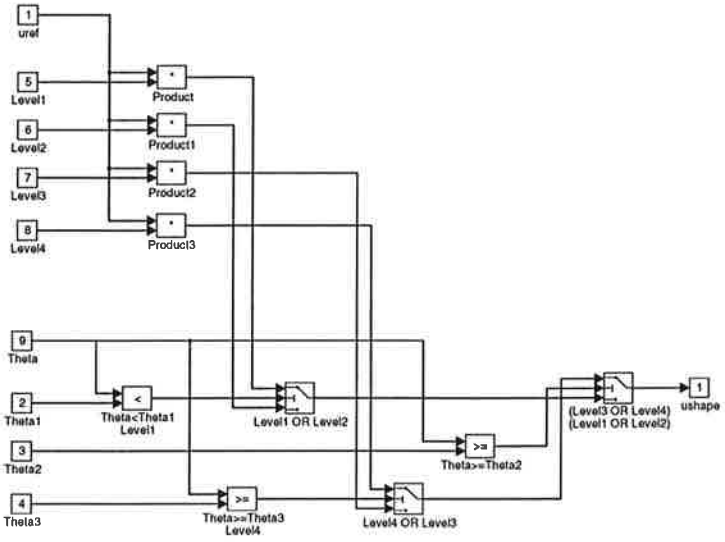


Figure B.12 The 'Inner u-shaper' block.

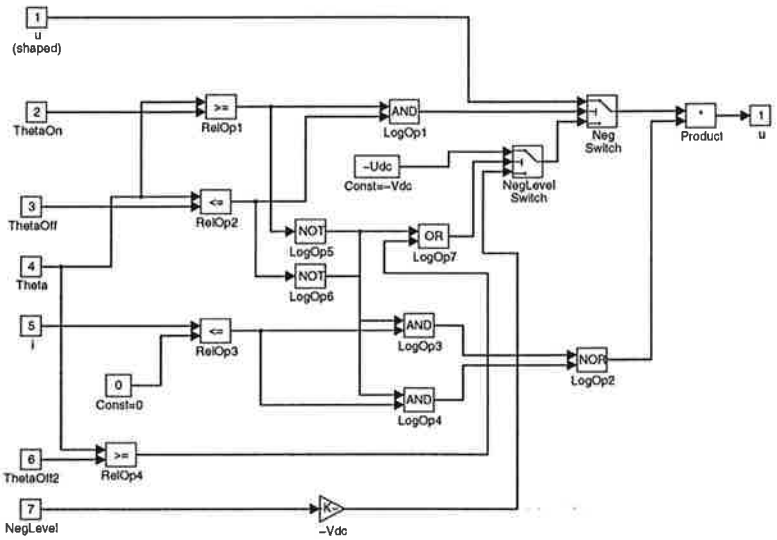


Figure B.13 The 'Outer u-shaper' block.

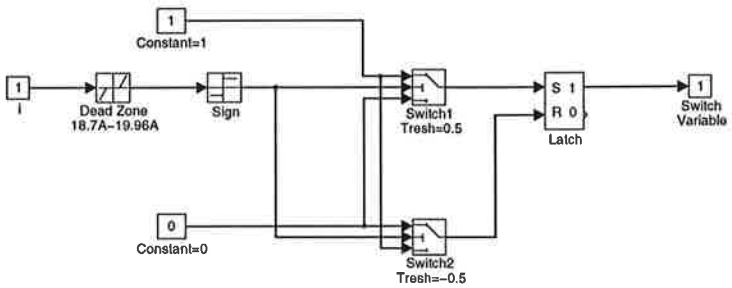


Figure B.14 The 'Current Check' block.

Notation

J	Moment of inertia
L	Instantaneous inductance
$\overline{L}$	Global inductance
L_a	Inductance for the aligned position
L_u	Inductance for the unaligned position
N_r	Number of rotor poles
N_s	Number of stator poles
q	Number of phases
T	Torque
V_{dc}	Maximum amount of voltage from the converter
W'	Coenergy. Surface under a magnetization curve
W_f	Field energy. Surface above a magnetization curve
ω	Angular velocity of the rotor (electrical or mechanical)
ω_n	Nominal angular velocity (electrical or mechanical)
Ψ	Flux linkage
θ	Angle of the rotor position (electrical or mechanical)
θ_{end}	Angle where the flux and current should be zero
θ_{off}	Angle where the voltage should be switched negative
θ_{on}	Conduction angle. Angle where the voltage should be turned on