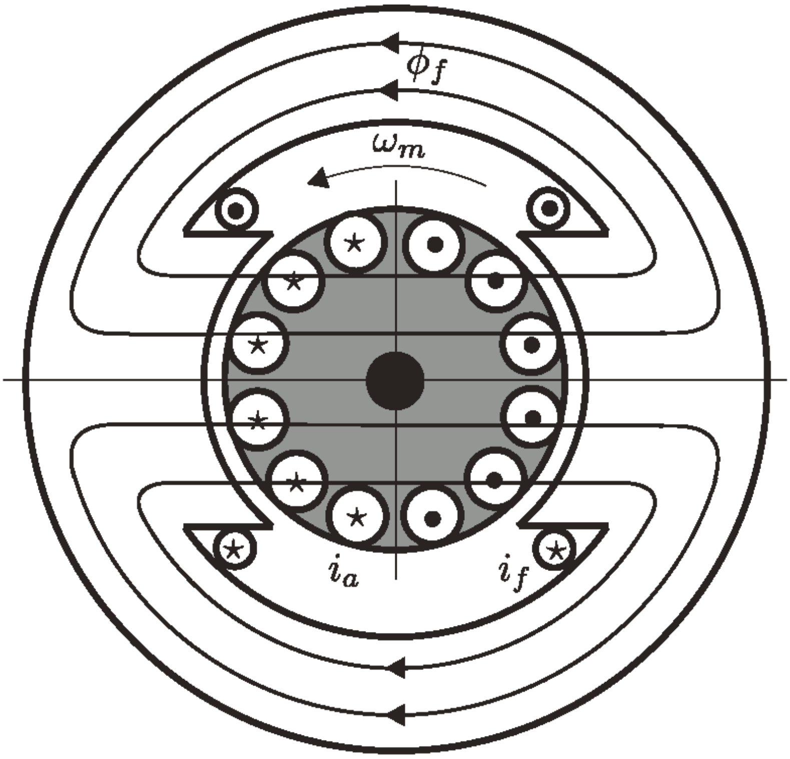
EIE 041 Control of Electrical Drives

Laboratory exercise 1

Torque and Speed Control of a DC Machine



1. Introduction

In this lab a current controller for a DC machine and a speed control for the electric drive system will be studied. The control system is a Simulink block diagram-based model that will be implemented in the dSpace real-time hardware. The Real-Time Interface (RTI) compiles, links, downloads, and configures the Real-Time Scheduler and the I/O and runs the Simulink code in a real time application in the same way as if it was a simulation. This way of using Simulink is useful for evaluation of the code but does also have a strong didactic value.

2. Laboratory Set-Up

The laboratory set-up can represent as a combination of three separate blocks. These blocks are shown shaded in figure 2-1 below.

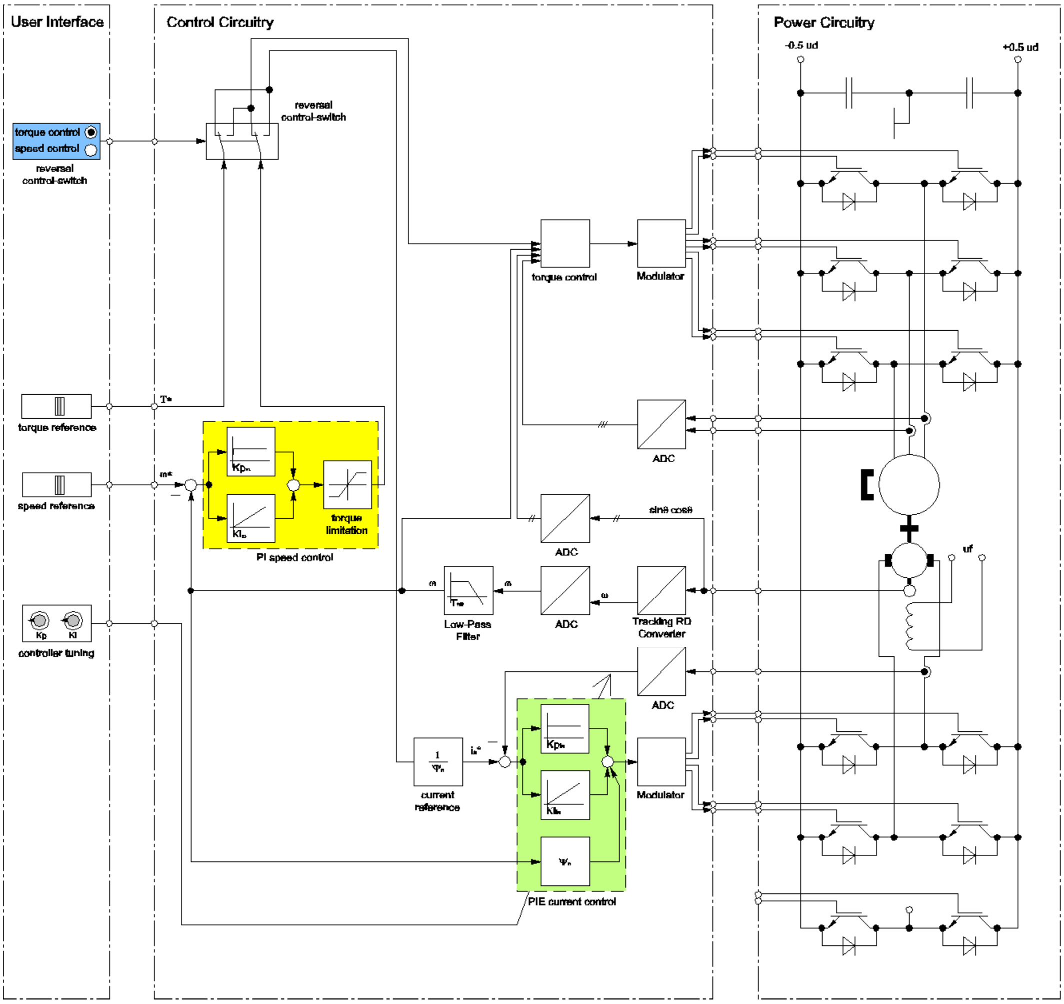


Figure 2-1 *Principal diagram of the lab system*

The block on the left is the user interface from where one can set the reference values, tune the controller parameters and make diagnostics of the control of the electric drive system. The control circuitry for the drive system is placed in the middle block, where the connections from the user interface to the power circuitry are shown. If you look carefully, you will recognise the structure from the simulation program you have built in the home assignment. The power electronic circuit together with the DC (Direct Current) machine and the PMSM (Permanent Magnet Synchronous Machine) are shown on the right.

Note in particular how the speed and torque reference given in the user interface are delivered to the two electrical machine controllers. One of them is only torque controlled, and the other is torque controlled via a speed controller. By flipping the switch *reversal control-switch*, either of the machines can play either role.

The following sections describe briefly the contents of the blocks visible in figure 2-1.

2.1. User Interface

The experiment layout and graphic programming in dSpace has been used to build the blocks of the user interface. The user interface helps to set the torque or the speed references, to switch the control of the DC machine from a torque controlled system to a speed controlled system, and finally to tune the parameters of the specified controller. Figure 2-2 shows the user interface.

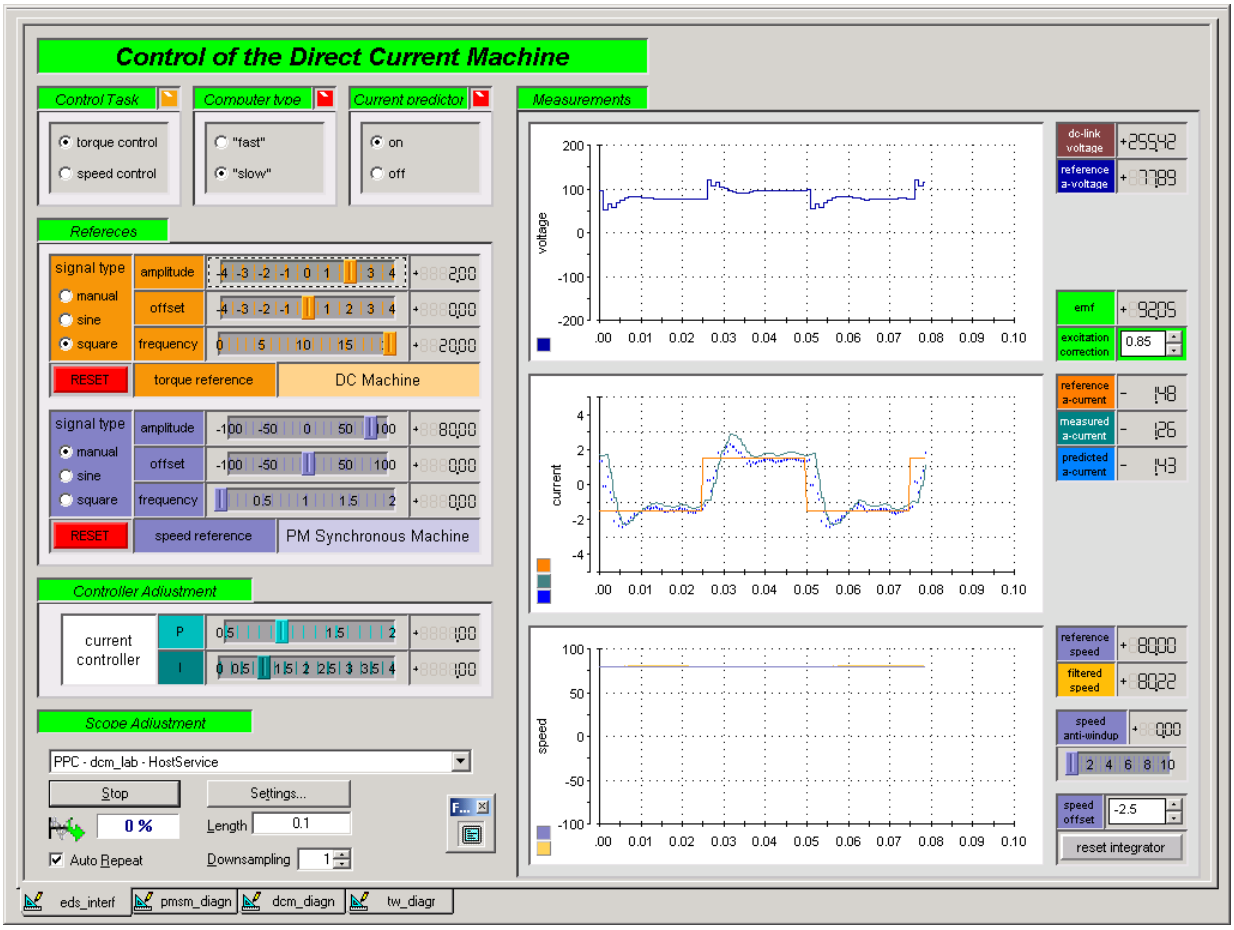


Figure 2-2 User interface

Torque vs. Speed Control

Note that the DC machine and the PMSM are coupled to the same axis. Both have the same reference direction for torque, i.e. a positive torque on either of the machines accelerates the shaft in the same direction. Stationary operation thus requires that the torque contributions from the two machines are of equal size, but with opposite signs.

The reversal control-switch specifies the control task and links the torque controller of one machine directly to the torque reference, and the torque controller of the other machine to the output of the speed controller.

Controller Adjustment

The calculated controller parameters can be further adjusted at the controller adjustment block. The adjustment is connected to the controller specified by the reversal control-switch. If the DC machine is only torque controlled, then the controller adjustments are made on the current controller parameters. If the DC machine is connected to the speed controller, then the controller adjustments are made on the speed controller parameters.

Diagnostics

Apart from the references, a scope with control and feedback signals is shown in the user interface.

2.2. Control System

The control system is based on a Simulink model that is downloaded to the dSpace hardware. Even though the control system includes the PMSM control system (Vector Control of torque), the torque control of the DC machine and the speed control for the whole electric drive system will be the main point of interest.

Despite the torque of the DC machine has to be controlled, the torque reference is recalculated into a current reference and the armature current is the one that is controlled. The voltage references for the power amplification are calculated in the current controller, and the current references are calculated with a PI speed controller. Consequently the control system consists of a cascade control of two loops where the inner loop is the current control and the outer is for the speed control.

PWM Modulation

Power amplification, with the help of two converters, is used to control and to provide the specified power input to the machines. The switching states in the converters and the voltage input to the machines are determined from the PWM modulators. In order to form the control pulses, a triangular carrier wave is modulated with symmetrized references.

Transducers

Current sensors for the armature current of the DC machine and two phase currents of the PMSM, a DC link voltage sensor and the instantaneous angular position sensor (Resolver, Rotasyn) are used to get the feedback from the electric drive system. The angular speed is calculated from the resolver signals and used as an additional feedback to the control system.

Signal Conditioning

Signal conditioning (scaling) to the PWM modulators and from the Analog-Digital converters has been defined in such a way that the control system has the same voltage, current etc. rates as the controlled (real) system.

2.3. Power Circuitry

One PWM controlled 4-quadrant converter and one three-phase converter are coupled in parallel to the DC-link voltage (U_d), providing the specified voltage to the DC machine and the PM synchronous machine. The DC machine has a separate excitation (u_f).

3. Content of the Laboratory Exercises

The content of the simulation and the laboratory exercises are divided to two parts:

1 Design of a current controller for the DC machine and

2 Design of a speed controller for the electric drive system.

3.1. Implementation of the current control loop

At first, the DC machine is torque controlled and the speed controller is connected to the torque controller of the PMSM, exactly like it is shown in figure 2-1. Your task is to study the torque control of the DC machine and the principle of the PIE current controller both as a “fast” and a “slow” computer.

Current control of a generic single-phase load with a fast and slow computer

The main goal here is to study the principle of the PIE controller, the controller parameters and the influence of small changes of the controller parameters. (Sections 4.4 and 4.5 from the course book)

Sampling Time

The speed of the control system and the parameters of the controller are dependent on the sampling time. The selection of the sampling time is a compromise between the requirements of the drive system dynamics, measurements and noise, the available computer capacity and the process hardware.

Current predictor

The time delay introduced by a slow computer creates a need for predicting the current one sampling interval “ahead”. (Section 4.5 from the course book). The control system used in this lab contains such a predictor, but can also act as a “fast computer”. You may wonder how that is possible, it is after all a real computer we are running and it is not infinitely fast. The answer is that *we are running at a rather low sampling speed (the computer is able to operate much faster) and that leaves us room to emulate a slow or a fast computer* by deliberately introducing or not introducing a time delay.

3.2. Implementation of the speed control loop

Problems related to speed measurements and filtering together with stability criteria appear when the speed control loop is to be designed.

Symmetric optimum

The concept of symmetric optimum will be used to design the speed controller. The practical hints relating to the speed measurements and filtering are helpful to follow. (Section 2.4 from the course book)

Anti-windup

The limited range (non-linearities due to the constrained output) of a controller cause unnecessary growth of the integrator part in a PI (PIE) controller – called integral windup that needs to be prevented. The aim of the anti-windup is to stop the integration in the controller when the controller output signal saturates. Correctly done, this improves the dynamics of the controller significantly.

3.3. System Data

The data of the DC machine, the drive system and the settings are presented below in table 3-1. Use the data to design the torque control for the DC machine and the speed control for the whole drive system.

Table 3-1 *DC machine data*

Measure	***Symbol***	***Value***	***Unit***
Nominal power of dc machine	P_n	2000	W
Nominal voltage of dc machine	U_n	220	V
Nominal current of dc machine	I_n	12	A
Nominal speed of dc machine	n_n	1400	rpm
Nominal torque of dc machine	T_n	16	Nm
Nominal flux linkage of dc machine	Ψ_n	1.35	Vs
The estimated inertia of the shaft	J	0.034	kgm^2
Armature inductance of dc machine	L_a	0.02	H
Armature resistance of dc machine	R_a	2.5	Ω
DC link voltage	U_d	250	V
Sample time	T_s	1e-3	s
Low-pass filter time constant	T_f	5e-2	s

4. Working with the Real-Time Interface

In order to open an existing model or to make further changes in the control model the useful steps are presented here.

4.1. Opening an existing model

The control system for the DC machine and the PM synchronous machine together with the signal conditioning is already constructed. The following steps are useful to get started with the laboratory exercises, to open an existing model and the corresponding user interface, and to download the model to dSpace:

1. In order to open an existing control model in Simulink:

   a) Double click in the Matlab 6.1 icon to open the Matlab program;

   b) From the “current directory” window choose the working path – ***C:/SED_Lab/dcm/***

   c) Execute the Matlab script ***dcm_setup.m***, which includes the parameters for the model and further opens the control model ***dcm_lab.mdl***;

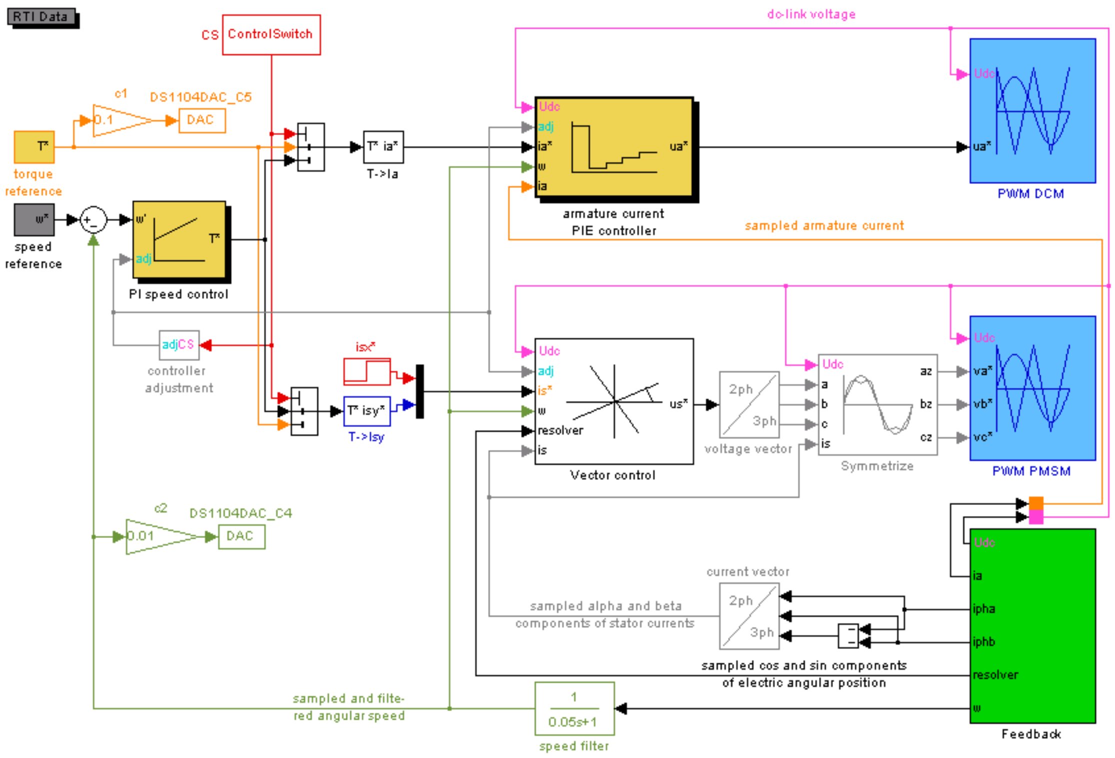


***Figure 4-1** Real-Time Simulink model*

The filled blocks in the RT simulink model are connected to the user interface and to the PWM modulators and transducers, which correspond to the cyan blocks on the left and the light blue and light green blocks on the right respectively.

2. To open the dSpace experiment:

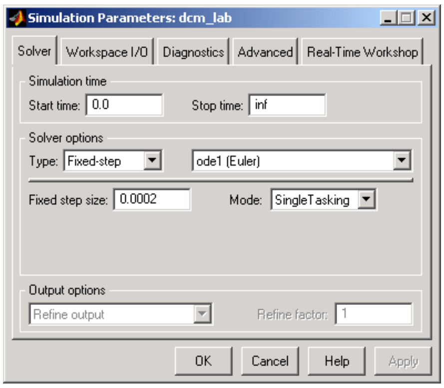
   a) Double click on the dSpace ControlDesk icon to open the dSpace software;

   b) From the menu *"File"*, choose *"open experiment"* to open the corresponding experiment called ***dcm_exp.cdx***;

   c) The experiment definition opens two layouts ***eds_interf.lay*** and ***dcm_diagn.lay***, which are used respectively as a user interface and for system diagnostics:

3. To build the real-time simulation model in Matlab:

   a) Ensure that the simulation parameters fulfil the demands for the real-time simulation environment as shown below:



Simulation time

Start time: 0.0

Stop time: inf

Solver options

Type: Fixed-step

ode1 (Euler)

Fixed step size: 0.0002

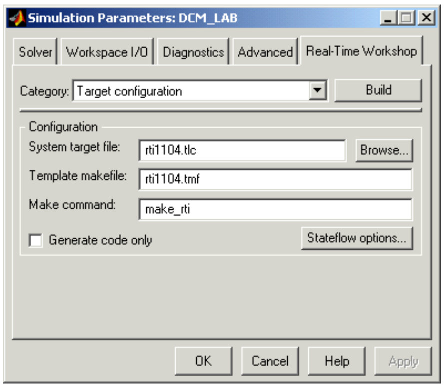
Mode: SingleTasking

Output options

Refine output

Refine factor: 1

b) With the Matlab / simulation model, choose *Tools → Real-Time Workshop → Build model* from the menu or click *“**Ctrl+B**”* to build the real-time model for dSpace



Category: Target configuration

System target file: rti1104.tlc

Template makefile: rti1104.tmf

Make command: make_rti

☐ Generate code only

(***Note:*** *every time after you change your simulation model, you have to re-build the model so that the new model can run on the dSpace hardware*)

The Step Size for the Solver and the Sample Time for the Control Model

The step size for the solver is selected to be Tc=0.001 sec as a compromise. The sample time ***Ts*** for the control model is selected to be equal the solver step size and it is already defined in the parameter file ***dcm_setup.m***.

4. The work with the Control Desk and the User Interface built:

   a) Once you build the model, the RTI building procedure will compile, link and load the application to the real time processor;

   b) In order to see the connected instrumentations on the user interface layout switch the *Instrumentation* from the *edit mode* (**Shift+F5**) to the *animation mode* (**F5**) and make the panel large by choosing *View* and *Full Screen*

   c) *Esc* brings you back to the *Control Desk;*

4.2. Building and downloading a new model

The following steps are useful to follow for the further changes and improvements in the control model and on the control desk.

5. To run and stop the simulation experiment in dSpace:

   a) Before you are able to stop the model switch *instrumentation* back to *edit mode;*

   b) In order to control the current state of the loaded application you can either use the icons to stop or reload it, or find the commands from the menu: *Platform → application → stop/reload*; 🟥 ▶

Note: *If you change the simulink model in Matlab and rebuild it while the dSpace simulation is running, and the block or subsystem variables are connected to the instruments, the ControlDesk automatically changes from Animation mode to the Test mode.*

6. To make changes in the simulink model:

   a) stop the application;

   b) make the changes in the model;

c) Build the real-time model and let RTI download it to the real-time processor;

4.3. System Input/Output

Table 4.1 System Input/Output

Input	**dSpace Interface**	**symbol**	**source/measurement**
dc-link voltage	ADC1	Udc	C&V measurement card
Mechanical speed	ADC2	w	Resolver card
Cosine of position	ADC3	cos	Sin&Cos card
Sine of position	ADC4	sin	Sin&Cos card
Armature current	ADC5	ia	
Phase a current	ADC7	ipha	C&V measurement card
Phase b current	ADC8	iphb	C&V measurement card
Output	**dSpace Interface**	**symbol**	**source/measurement**
2φ PWM*	PWM1 PWM2	va vb	DS1104SL_DSP_PWM3
Sampled armature current	DAC1	ia	RT Simulink model
Predicted armature current	DAC2	ia^	RT Simulink model
Reference voltage	DAC3	ua*	RT Simulink model
Filtered speed	DAC4	wref	RT Simulink model
Torque reference	DAC5	Tref	RT Simulink model

***PWM signal for the “first” converter phase is available on the blue connection pin situating on the right hand side on the dSpace interface panel.**

4.4. Security Requirements

Make sure that the DC-link voltage is zero before you start or stop the control system in dSpace!

5. Lab Assignments

First of all keep the safety requirements in mind when you are working with dSpace and the other lab equipments. Have a look to section 4.4!

The preparatory part in the lab is to open the existing control system and the user interface that belongs to it. Follow the steps 1 – 4 written in the section 4.1.

The control system refinement is necessary and therefore some changes have to be made in the control blocks. The practical hints in section 4.2 are useful in order to make further changes in the control model.

5.1. Torque Control

In order to control the torque of the dc-machine, the current has been controlled. Current control without and with current prediction will be investigated. Formulate PI controller parameters for a given sampling time Ts and calculate the values for the Pand the I parts. (compare the initial parameters in dcm_setup.m Kpi_LM and Kii_LM)

. . . .

. . . .

Table 5.1 A dead-beat PIE current controller parameters

Controller parameters	*Theoretical*	*Correction*	*Actual*
Proportional gain Kp_i “fast” “slow”			
Integral gain Ki_i “fast” “slow”			

First, connect both the machines to the output of the power converter by turning on the main circuit brake. Set the external excitation of the dc-machine by the help of the upper autotransformer to the nominal value of 0.8 Ampere. The other autotransformer that is on the table is used to regulate the magnitude of dc-link voltage. Increase the dc-link voltage about 250V. In the beginning the armature current controller for the dc-machine operates as a “fast computer” without current prediction.

1. Increase the speed reference until the induced voltage of the dc machine is about 100V and study the case. The DC machine is now running at no load.
   - What can you say about the phase potential ***va*** of the DC machine power converter (the upper one) compared to the voltage ***ua*** over the dc-machine? The switching frequency is . . . . Hz and the output voltage pulse frequency is . . . . Hz.
   - The induced electromotive force ***ea*** should be about 100 V**.** Can you calculate the armature inductance **La** by using the current and the applied voltage ***ua***, knowing the induced voltage and disregarding the resistive voltage drop? . . . .
     . . . .
     . . . .

2. Create a square-wave torque reference ***T**** with an amplitude of ±2Nm and 20Hz frequency;
3. Study the step response of the armature current controller. With the knobs on the desktop you can adjust the proportional and the integral gain. Vary the parameter until the step-response looks the best for you;
4. Sketch the current step response (in diagram 5.1) a few pulses before and after the positive current step.

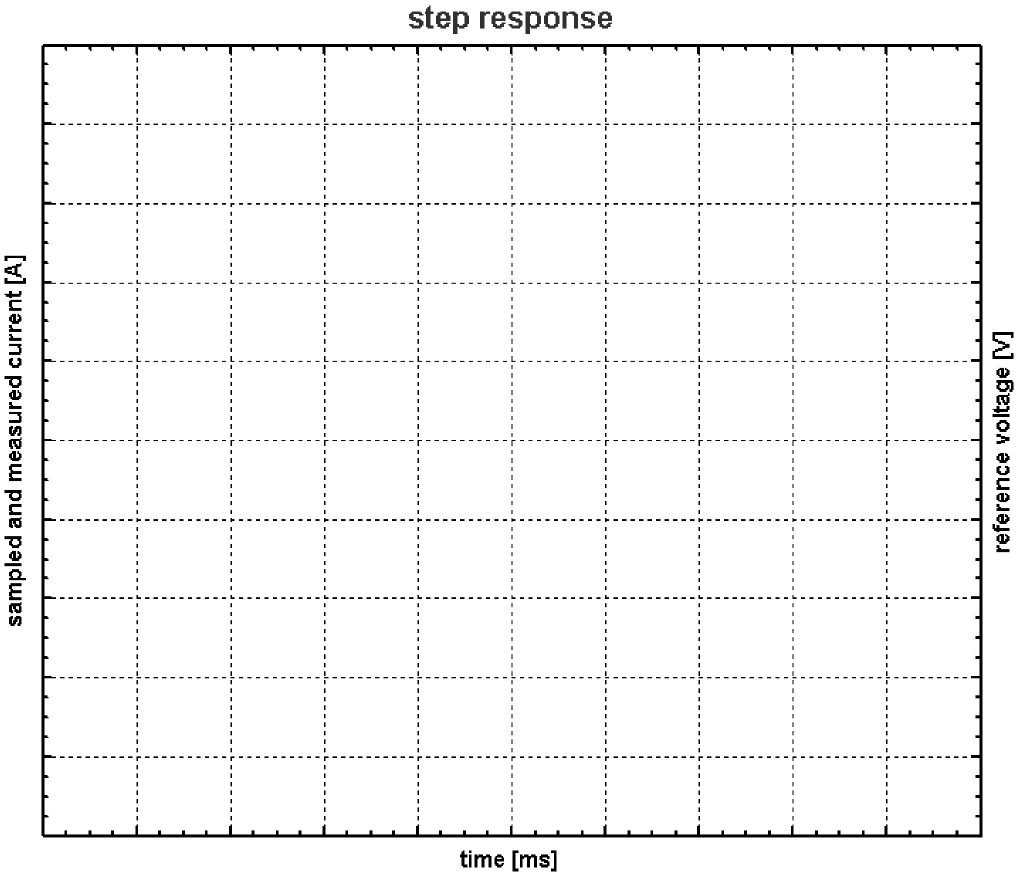


Figure 5.1 The current reference and the response for slow and fast computer

5. Update table 5-1 with the correction factors and the recalculated controller parameters;
6. Run the steps above both with “fast” and “slow” computer, with and without “prediction”. When you run the slow computer without prediction the current controller may be unstable. Why?

. . . .

. . . .

. . . .

. . . .

5.2. Output Voltage

Connect a differential high voltage probe to the output of the 4Q-converter. Use the same current reference as in the previous exercise. Try to catch the output voltage while triggering the oscilloscope on the torque reference (look at table 4.1). Run the machine at a low speed backward (***ea<0***) and study the pulse width of the first few samples after the positive current reference step.

7. Draw the output voltage response due to the step in the torque reference and try to find the explanation to the following question;

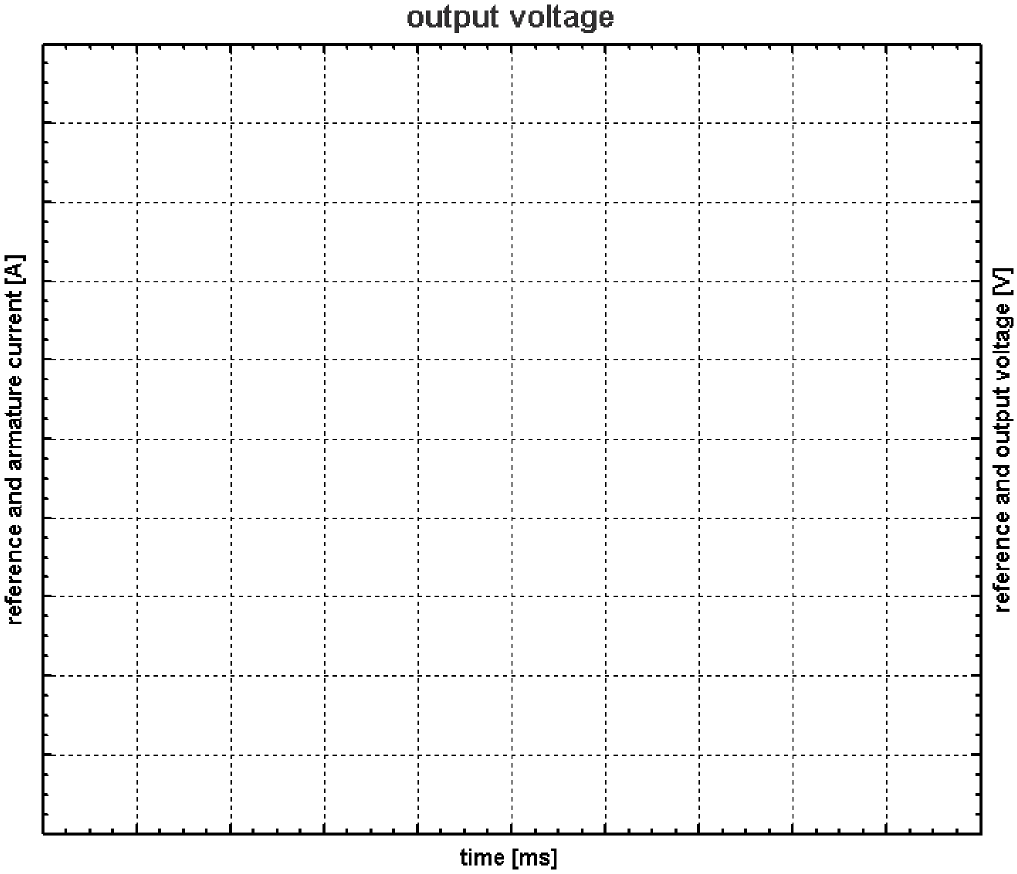


Figure 5.2 The output voltage

8. What happens with the pulse width if you reduce the dc link voltage and why?

. . . .

. . . .

5.3. Speed Control

Use the reversal control-switch to switch the output of the speed controller to the input of the torque (current) controller of the DC machine. Now the DC machine is speed controlled and the PM synchronous machine is torque controlled (you will study the PMSM torque controller later in the course). Make sure that you are using the right reference with the right machine! The speed is measured and then filtered with a time constant of 50ms.

What is a suitable gain and integral time constant for a PI speed controller? . . . .
. . . .

What is the torque limitation for a maximum armature current about ±10A; . . . .
. . . .

Write the theoretical values of the PI speed controller into table 5-2.

9. Set the torque reference to “manual” and parameters to zero;
10. Create a square-wave speed reference with an amplitude of ±10rad/s and a frequency of 0.5 Hz;
11. Study the step response of the armature current controller. With the knobs on the desktop you can adjust the proportional and the integrator gain. Vary the parameter until the step-response looks the best for you;
12. Update table 5-2 with the correction factors and the recalculated controller parameters;

Table 5-2 A PI speed controller parameters

Controller parameters	*Theoretical*	*Correction*	*Actual*
Proportional gain Kp_{ω}			
Integral gain Ki_{ω}			

13. Increase the speed reference to ω*=±50rad/s 0.5Hz square, study the speed controller response without and with anti-windup, draw speed response.

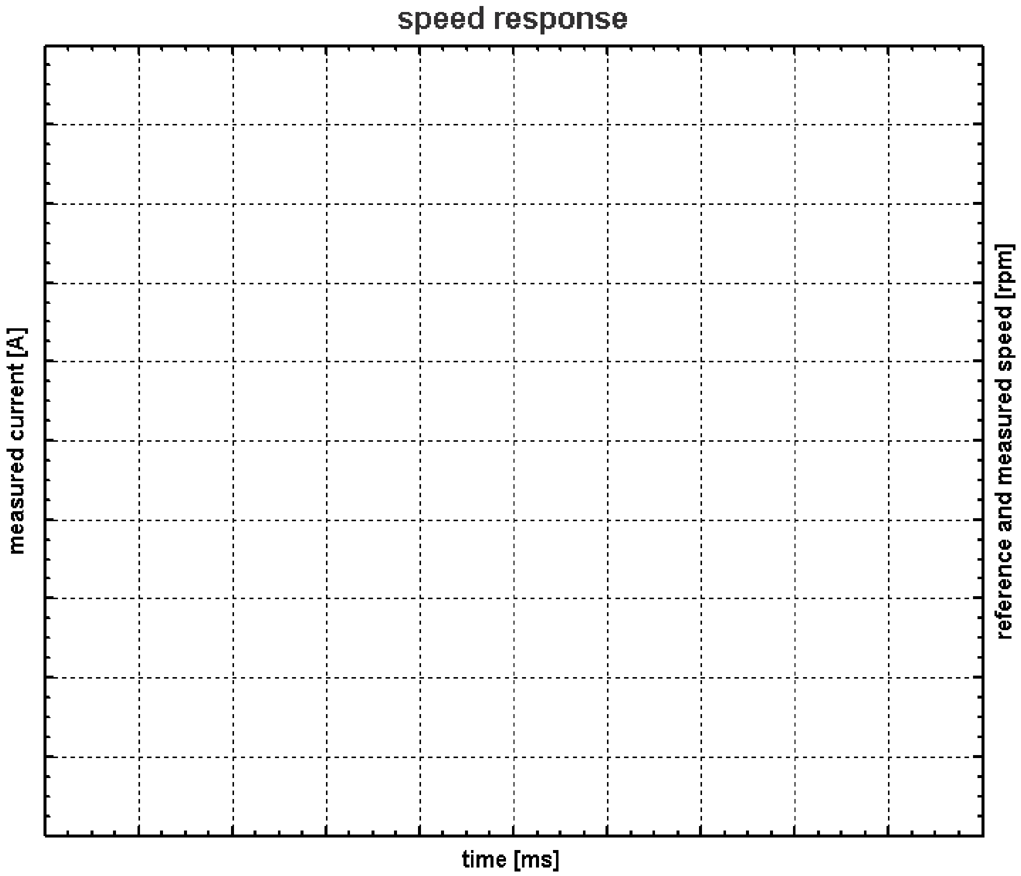


Figure 5.3 Speed response

14. Analyse the speed step response. Can you reach oscillatory poles by changing the speed controller parameters, and in that case:- In what direction do you change the parameters?

. . . .

. . . .

15. Try to estimate the inertia of the rotating part.

. . . .

. . . .