Dynamics and motion control

Workshop A, Parameter identification and control of a DC-motor.

1 Introduction

Written report

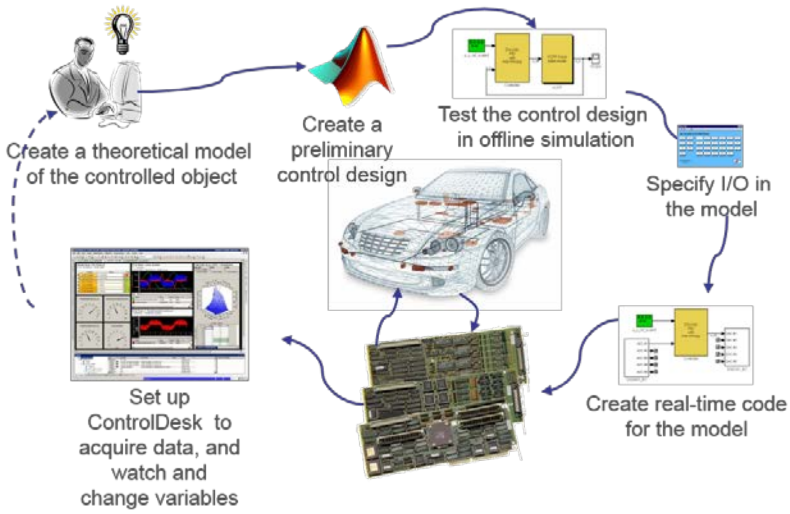
The report should not be thick it should also not be too thin. A good report includes exactly the information which is needed for the reader to understand the results and nothing more.

You don’t have to include all the steps in the report when you derive controllers etc. The important part are the results in terms of figures with plots of *appropriate* signals for each design phase, experiment etc. Appropriate signals can be all or some of the signals, position, velocity, voltage and command signals. If necessary you should also include frequency, pole zero etc. plots. A short discussion on the results shown in the plots should also be included.

There are two levels of the problems below, if you only do the level 1 can you not get the highest grade, i.e. an A on the workshop. Solving both level 1 and level 2 problems is however not equal with getting an A. The quality has to be high throughout the whole report.

dSPACE tools

The dSPACE tools consists of both hardware and software. A combined processor and I/O board is mounted on an PCI slot in the PC. The board has a Motorola power PC, MPC8240 which is dedicated for the real-time simulation. The board also has various I/O functions, e.g., AD, DA, digital I/O and incremental encoder interfaces. A C-code generator in Simulink, (RTW) generates the C-code of a Simulink file, a software module RTI (Real Time Interface) and a compiler that compiles and download this code on to the MPC8240 processor. To interact with the real-time simulation, e.g., changing parameters and plotting outputs, there is an experiment control software called ControlDesk. The work flow is depicted in the figure below.



2. Learning how to work with the dSPACE tools

Work through the example in the document *dSPACE Tutorial*.

3. Parameter identification

Not the best, but the simplest way to identify the model parameters is to make a step response of the model and the real motor and compare the results. A step excites all frequencies of the process. There are two main behaviours that can be used to find the parameters. They are,

- Steady state velocity gain
- Time constant

The electric parameters can be taken from the data sheet, they should be correct. The unknown model parameters are the *inertia* and the *friction*.
Copy the Simulink blocks of your dc-motor model into the Simulink file that you already have used in section 2 and rebuild the application from the Simulink menu. You have to reload the model.sdf file in the project manager in ControlDesk. Right click the sdf file in the experiment and select, *reload*.

Create instruments in ControlDesk and connect them to the parameters of the Simulink model that should be identified. If the same parameter exists in more than one Simulink gain block can one ControlDesk instrument be connected to more than one Simulink block. See the section *Connecting more than one parameter to one instrument* in the dSPACE manual.

Work with the parameters until a good fit between model velocity and the velocity that is calculated from the encoder is achieved. Save the experiment and the identified parameters.

Observe that the polarity of the motors can vary. If the rotation direction directions of a motor is wrong you can fix it by inverting the polarity in the simulink model, i.e., a gain of -1 on the voltage signal.

Level 1.)

Identify the parameters, linear friction and inertia.

Show the velocity signals from the model and the real motor in the same plot for a step response and any other input signal that make sense to show.

Level 2.)

Identify the two unknown parameters of the nonlinear friction model, $M_f = d\dot{\varphi} + F_c \operatorname{sgn}(\dot{\varphi})$
Show the velocity signal from both model and motor from steps with at least two different amplitudes and from a sinewave input with low amplitude. Discuss any improvment from the linear model in level 1.

4. Controlling the motor

Velocity control

Design and implement a controller that controls the velocity of the motor. Note that you need the good parameters, linear friction coefficient and inertia from the identification part first. The controller must be implemented as a discrete time controller. You may select any sampling period. In the report you must show that the specifications are fulfilled. Give the control law in numeric format, with closed loop poles and sampling period.

Level 1.)

Velocity control system specifications for three different cases.

1.)

- Step response to 50 *rad/s* with as fast as possible response time.
- No overshot
- Maximum steady state error 0.5 *rad/s*.

2.)

- Sine wave reference signal $\dot{\varphi}_{ref} = 10\sin(2\pi 0.2t)$

3.)

- Same sine wave reference signal $\dot{\varphi}_{ref} = 10\sin(2\pi 0.2t)$ and at the same time braking the motor slightly by applying a force with your finger on the flywheel. Apply a force so that the voltage almost saturates.

All three cases must use the same controller with the same parameters.

Do not forget to plot the voltage for all three cases!

Position control

Design a controller that controls the position of the motor The implementation of the controller should be in discrete time. In the report you should give the control law and show that the specifications are fulfilled. Depending on the design may it be necessary to include an anti-windup in the controller. Give the control law and the closed loop poles numerically. Don't forget the voltage plot!

Level 1.)

Position control system specifications

- Step response from 0 to 4 *rad/s* with rise time $t_r = 0.3\,s,\ \pm 0.05\,s$. Rise time is here defined as 95% of the reference step.
- Maximum overshot 2% of the step amplitude.
- Maximum 0.5% steady state error
- Any sampling period can be used.

Level 2.)

With the same specifications as in level 1. Design and implement the controller with as long as possible sampling period. Compare results when the controller is approximated to discrete time from a continuous time design with those when the controller is designed directly in discrete time. Include a discusion of what limits the sampling period