Modelling and simulation of a measurement robot

Denna version: 4 oktober 2017

---

Name: __________________________

Personnr: ______________________

Datum: ________________________

Godkänd: ______________________

---
1 Introduction

The lab consists of two four hour sessions and the goal is to create a model of the robot in figure 1 and verify the model using simulations. The robot is used to perform accurate measurements. There is a sensor on the robot arm which is activated when it comes in touch with something, and by letting the robot move between two surfaces, the distance between these can be computed.

There are preparatory exercises for both sessions, see chapter 4.1 for session 1 and chapter 5.1 for session 2.

Since there is a limited amount of time available at the session, the exercises have to be well prepared, and they will be checked before the session starts.

In the first session, a model of the robot motor is developed and simulated using SIMULINK and MathModelica. Block-oriented modeling and object-oriented modeling is compared.

In the second session, a more complete model of the robot is built and simulated using MathModelica. This will entail more complex modeling, using hierarchical models and sub-components.

Figur 1: The robot, with and without cover.
1.1 The Tools

The two tools used are Simulink and MathModelica. The goal with the is understand and appreciate different ways to approach modeling. Specific instructions to Simulink and MathModelica and examples to get started will be available on the course homepage.

Simulink:

Simulink is a simulation environment running in MATLAB, based on block-oriented models. The purpose of Simulink is to allow users to easily describe and simulate systems. To simplify modeling, a graphical signal-based user interface is used.

MathModelica:

MathModelica uses the open object-oriented equation based modeling language Modelica. There are many components available in the standard library for Modelica, and the parts that will be used here are

- Blocks.Sources
- Electrical.Analog
- Mechanical.Rotational
- Mechanical.Translational

See the course homepage for links to the component libraries. You should be able to complete the labs with these components only, but you are allowed to use additional ones if you want.

2 Description of the system

A sketch of the system is seen in figure 2.

The system consists of five parts connected according to figure 3. A short description of every sub-system follows below.
2.1 Current controller (Strömregulatorn)

Inputs to the current controller is a voltage which describes a reference signal for the motor current, and the motor current. The output is the voltage driving the motor, and the current drawn from the reference source.

For simplicity on lab 2 we will use a current source instead of a current controller, which means we assume we have an ideal controller where the motor current always is the desired. Note though that if we use an ideal current source, we cannot have any inductance in the circuit. (why?)
2.2 Servo motor

The servo motor is a simple DC-motor mechanically connected to a tachometer. The current controller is electrically connected to the servo motor according to figure 4. The tachometer allows for yet another feedback loop from motor angular velocity which is proportional to tachometer voltage. This feedback is not used in our model of the robot, so the modeling of the tachometer can be seen as an extra exercise. Another addition would be to design a controller which controls the motor angular velocity by feedback from the tachometer.

![Electrical components sketch.](image)

**Figure 4:** Electrical components sketch.

Data for the motor and the tachometer can be found in figure 9 in appendix A Tables. The used motor has model number M-586-0585. The plot shows the relation between revolutions per minute (rpm) and torque. The upper curve is limiting the working area of the motor. From the curve we can see 3 different limitations; the motor has an upper rpm limit 6000 rpm, a maximum torque 1.05 Nm, and a maximum power. The lower curve shows the relation between torque and angular velocity on motor axle when the current to the motor is kept constant. The current can be read as *Continuous stall current*, i.e., 3.9 A.

Under *Winding specifications* in figure 9 there are two resistors and one inductor. The reason for this is that the coil is not an ideal inductor, but is both inductive and resistive. With the notation in figure 5, *Armature resistance* is $R_I$, *Terminal resistance* is given by $R_e + R_I$ and *Armature inductance* is $I_I$. 
2.3 Belt transmission (remtransmission)

The servo is connected to a toothed cylinder shaped belt disc with outer diameter 20 mm and thickness 10 mm. A rubber belt drives another belt disc with outer diameter 80 mm and thickness 15 mm. The length of the belt is 750 mm and by elasticity it elongates 0.4% of its full length at max load 200 N. In the Physics Handbook you can find how to compute inertia for the discs. The discs are made out of aluminum (density $2.7 \cdot 10^3 \, \text{kg/m}^3$). Energy losses happens at both discs; the smaller disc has a friction $2 \cdot 10^{-5} \, \text{Nms/rad}$ while it is $5 \cdot 10^{-5} \, \text{Nms/rad}$ for the larger one.

2.4 Screw transmission (skruvtransmissionen)

The larger disc on the belt transmission drives a screw with pitch 1 revolution/inch. The screw drives the vertically working robot arm. Data for the screw transmission is found in figure 7 in appendix A. The screw in the robot is B-8000. The screw is connected to the robot arm through a spring. The spring constant is 75 kN/m. The length of the screw is 1 m. Friction in the screw is assumed negligible.

2.5 Robot arm

The robot arm is driven by the screw via the spring and moves vertically. The mass of the arm is 5.5 kg and the friction is 25 Ns/m. Note that we want

---

1 $16 \, \text{oz} = 0.45359 \, \text{kp}$, $1 \, \text{kp} = 9.81 \, \text{N}$ and $1 \, \text{in} = 25.4 \, \text{mm}$ according to the Table

2 Note that the inertia is given per unit length in the table.
to control the velocity of the arm, but for practical reasons we feedback the velocity of the motor which is connected to the tachometer.

3 Some modeling tips

To simplify the two sessions, here are some general tips.

Sub-models: To begin with, divide the large model into smaller models, and test every part by itself before you connect everything. This will also give you a better feeling for how the different parts work.

Start with a simple model: Start with the most basic properties. As an example, if you model two cog wheels with different radii, the most important property is that they scale torque and angular velocity. Later properties that might be important are things like friction and possible elasticity in the cogs.

Slow dynamics first: Extend the simple model with slow dynamics first. If a cog wheel is attached to a long aluminium axle, the axle flexibility probably has slower dynamics than the wheel flexibility dynamics. In other words, the weakness of the axle is more important to model than weakness of the cogs.

Changing standard blocks in MathModelica: To change the equations in a standard block in MathModelica it is easiest to copy the block and save in another name. You can then edit the copy.
4 Session 1

During session 1 two models of the robot motor are built and verified by simulations.

- A block-oriented model in SIMULINK.
- An object-oriented model in MathModelica.

In this part, we assume the motor is driven by a voltage source.

4.1 Preparations

1. Draw a bond-graph of the motor and mark causality. Is the graph conflict free? Which variables are suitable states?

2. Create a model of the motor using SIMULINK-blocks.

3. Think through how an object-oriented model should look like in MathModelica and which standard blocks you need. Available standard components can be found in the links given in the course home-page.

4. Find all numerical constants required for the motor model. Be careful with the units! For the mechanical friction coefficient, use the diagram in figure 9 in appendix A in particular the slope on the straight line.

5. In exercise 5 on session 1 you write MODELICA-code for a nonlinear inductance. Think through what the code should look like. Use MathModelica possibilities for object-oriented modeling. In [1, par. 9.3] you find some guidelines. The constants $k_1$, $k_2$ och $k_3$ in the relation (1) on the next page can be declared as parameter Real. (Which units should they really have?)

Tips on the Preparations.

- For the bond graph of the servo motor, don’t forget that you should have 2 effort sources $S_e$. One is of course the input voltage $u_r$ and the other one is the load on the rotation side (a torque).
For the simulink model: to see how the signals are connected and to easily answer question 1 during session 1, we recommend blocks such as Integrator, Gain, Sum in the model. Assume that the model is \( \dot{x} = -\frac{2}{3} \cdot x + \frac{1}{3} \cdot u \) and we want to simulate the model and plot the state \( x \) when the input \( u \) is a step. In figure 6 is one way to solve it using simple blocks. For students not familiar with Simulink, a small guide is available in the course web page.

Note that 60 RPM = 1 Hz = \( 2\pi \) rad/s.

### 4.2 Exercises

1. The electrical time-constant for the motor is defined as the time-constant for the current when a voltage step is made with the axle held still. The mechanical time-constant is defined as the time-constant for angular velocity during a voltage step with no load.

   The two time-constants are given in figure 9 appendix A. These can be computed using the constants you have derived (create differential equations, see [2]), or by simulation. Check that the values are consistent with the table. What could the reason be for one of the time-constants to deviate more than the other?

2. Do some comparisons between the two modeling programs. Same results? Why/why not?

3. Now study the MathModelica-model closer. One way to figure out which variables have been selected as state variables for simulation is

![Simulink Model Diagram]
by looking at the box available for setting initial conditions. Compare with the choice in the bond-graph. Are they the same?

4. Now assume that the motor instead of an inductance contained two inductors in series with the same total inductance as earlier. Modify the bond-graph and the MATHMODELICA-model. Is the bond-graph conflict free? If not, how can you solve this? Can the new MATHMODELICA-model be simulated? How has MATHMODELICA selected state variables? Explain why!

5. If you have larger currents in an inductor you might want to use a nonlinear model to better describe the physics. A model suitable for some coils is

\[
\begin{align*}
\Phi(t) &= k_1 \arctan(k_2 i_L(t)) \\
\dot{\Phi}(t) &= k_3 u_L(t)
\end{align*}
\]

where \( i_L(t) \) is the current, \( u_L(t) \) is the voltage, \( \Phi_L(t) \) is the magnetic flow through the coil. Create such a MATHMODELICA-model and use instead of the earlier inductance in the motor model. Let \( k_1 = 3.39 \cdot 10^{-3} \), \( k_2 = 1 \) and \( k_3 = 1 \). Try simulating the model for different input voltages. How large inputs are needed before you can see that the model is nonlinear?

6. Make sure the MATHMODELICA-model of the motor can be re-used in lab 2. A good solution is to make the motor a block, and thus allow hierarchical modeling. This can be done by giving the model in- and outputs using the Flange-blocks. An icon can be drawn for the motor too.
5 Session 2

Now we model the whole robot in MathModelica. For simplicity we use a current source instead of a current controller.

5.1 Preparations

1. Using an ideal current with an inductance can cause problems — how? In this lab we avoid the problem by not including the inductance in the model.

2. Find all numerical constants needed for the different sub-models. Be careful with units!

3. Plan for how the motor model can be extended gradually to a complete model. Think about how all sub-components should be simulated. The models should be constructed with the object-oriented approach used by MathModelica (hierarchical modeling) and extensions should be done according to the hints in Chapter 3.

5.2 Exercises

1. Extend to a model of the whole robot

2. Comment on all assumptions made. Which simplifications have been made? (E.g.: we assume all voltages in the controller are within limits)

3. When a system is extended with more dynamics simulations will take longer. Sometimes the difference in time can be larger than what can be explained from having a larger system. What other explanations could there be?

4. The answer on the previous question and the hints in Chapter 3 gives another rule of thumb for which dynamics to include in a model. Formulate this rule-of-thumb!

5. (a) The real system is very oscillatory. Can this be seen in the simulations? What causes the oscillations? This question is the reason the control group has studied this robot in reality.
(b) Play around by increasing and decreasing parameters to see if you can reduce the oscillations. Why is it not suitable to do these changes on the real robot?

6. Assume the spring between the screw and the arm is nonlinear

\[ F = \frac{0.0012K}{\pi} \tan \left( \frac{\pi}{0.0012} (s - s_0) \right) \]  

(2)

where \( K = 75 \text{kN/m} \) is the spring constant, \( s \) is the length of the spring, and \( s_0 \) is the length of the spring at rest.

(a) compare the linear and the nonlinear spring force by using MATLAB to draw spring force against spring length. (Hint: From (2) you can see the period length of the force, i.e., how large values on \( s \) which are needed.) Describe similarities and differences between the linear and nonlinear spring.

(b) How is the robot behavior changed in simulation by changing the spring?

Referenser


### MECHANICAL PROPERTIES

<table>
<thead>
<tr>
<th>Screw/Nut Series</th>
<th>Static Frictional Drag Torque oz.-in. (NM)</th>
<th>Screw Inertia oz.-in.-sec.²/in.</th>
<th>Anti-Backlash Life</th>
<th>Anti-Backlash Life w/TFE Coating</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 4000</td>
<td></td>
<td>.3 · 10⁻⁵</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>B 6000</td>
<td></td>
<td>1.5 · 10⁻⁵</td>
<td>Typical Backlash</td>
<td>Typical Backlash</td>
</tr>
<tr>
<td>B 7000</td>
<td></td>
<td>3.5 · 10⁻⁵</td>
<td>.003”-.010” (.076-.25mm)</td>
<td>.003”-.010” (.076-.25mm)</td>
</tr>
<tr>
<td>B 8000</td>
<td>Free</td>
<td>5.2 · 10⁻⁵</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B 10000</td>
<td>Wheeling</td>
<td>14.2 · 10⁻⁵</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B 12000</td>
<td></td>
<td>30.5 · 10⁻⁵</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B 14000</td>
<td></td>
<td>58.0 · 10⁻⁵</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7:** Mechanical data for the screw transmission.
La serie 500 DT appartiene ad una più vasta famiglia di servomotori a corrente continua con eccitazione a magneti permanenti, particolarmente studiata per soddisfare le esigenze in un ampio campo di applicazioni industriali e professionali, quando siano richieste alte prestazioni di precisione, di velocità e/o posizionamento.

The series 500 DT belongs to a large family of permanent magnet DC servomotors, they were studied to satisfy the demands of a broad range of industrial and professional applications, where highly precise speed and/or positioning performances are required.

**FORMA** (secondo IEC 34-7) IM55 (fissaggio con flangia)  
**PROTEZIONE** (secondo IEC 34-5) IP 44  
**EQUILIBRATURA** (secondo DIN 46666) classe N  
**POLE number 2**  
**MATERIALI ISOLANTI in classe F ed H**  
**CAMPO DI FUNZIONAMENTO** alitudine < 1000 m s.l.m.  
**TEMPERATURA AMBIENTE MIN 0° C**  
**CONSTRUCTION** (according to IEC 34-7) IM55 (flange mounting)  
**PROTECTION CLASS** (according to IEC 34-5) IP 44  
**BALANCING** (according to DIN 46666) class N  
**POLE NUMBER 2**  
**INSULATION CLASS F AND H**  
**OPERATING RANGE** below 1000 m above sea level  
**MINIMUM OPERATING AMBIENT TEMPERATURE 0° C**

**Figgur 8: Servo motor.**
### SPECIFICATIONS (1)

<table>
<thead>
<tr>
<th>Operating Specifications</th>
<th>M 586 0585</th>
<th>M 588 1100</th>
<th>M 589 1270</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous stall torque</td>
<td>Nm</td>
<td>0.2</td>
<td>0.35</td>
</tr>
<tr>
<td>Peak Stall torque</td>
<td>Nm</td>
<td>1.05</td>
<td>1.50</td>
</tr>
<tr>
<td>Continuous stall current</td>
<td>A</td>
<td>3.90</td>
<td>3.30</td>
</tr>
<tr>
<td>Maximum pulse current</td>
<td>A</td>
<td>18.7</td>
<td>14.2</td>
</tr>
<tr>
<td>Maximum terminal voltage</td>
<td>V</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>RPM</td>
<td>6000</td>
<td>5200</td>
</tr>
</tbody>
</table>

#### Mechanical data

| Rotor moment of inertia (including tachometer) | kg m² | 3.88 · 10⁻⁵ | 5.5 · 10⁻⁵ | 6.8 · 10⁻⁵ |
| Mechanical time constant                  | ms    | 10.2        | 10          | 8          |
| Motor mass (including tachometer)           | kg    | 1.3         | 1.7         | 1.9         |

#### Thermal data

| Thermal resistance (armature to ambient)(2) | ºC/W | 5 | 4.2 | 4 |
| Maximal armature temperature               | ºC   | 155 | 155 | 155 |

#### Winding specifications

| Torque constant (3) Kₜ | Nm/A | 0.056 | 0.105 | 0.12 |
| Voltage constant (back emf)(3) | V/kRPM | 5.8 | 11 | 12.7 |
| Armature resistance (4) | Ω    | 0.8 | 1.6 | 1.8 |
| Terminal resistance (4) | Ω    | 1.15 | 2 | 2.2 |
| Armature inductance | mH    | 3.39 | 5.2 | 6.4 |
| Electrical time constant | ms    | 2.95 | 2.6 | 2.9 |

#### Tachometer data

| Linearity (maximum deviation) | %   | 0.2 |
| Ripple (maximum peak to peak) | %   | 5.0 |
| Ripple frequency | cycles/rev | 11.0 |
| Temperature coefficient | %/ºC | -0.05 |
| Output voltage gradient | V/kRMP | 14±10 % |

(1) Ambient temperature (if not otherwise specified): 40 ºC.
(2) Test conducted with unit heatsink mounted on a 254x254x6 mm.
(3) Tolerance ±10%
(4) At 25ºC.

**Figur 9:** Data for servo motor.