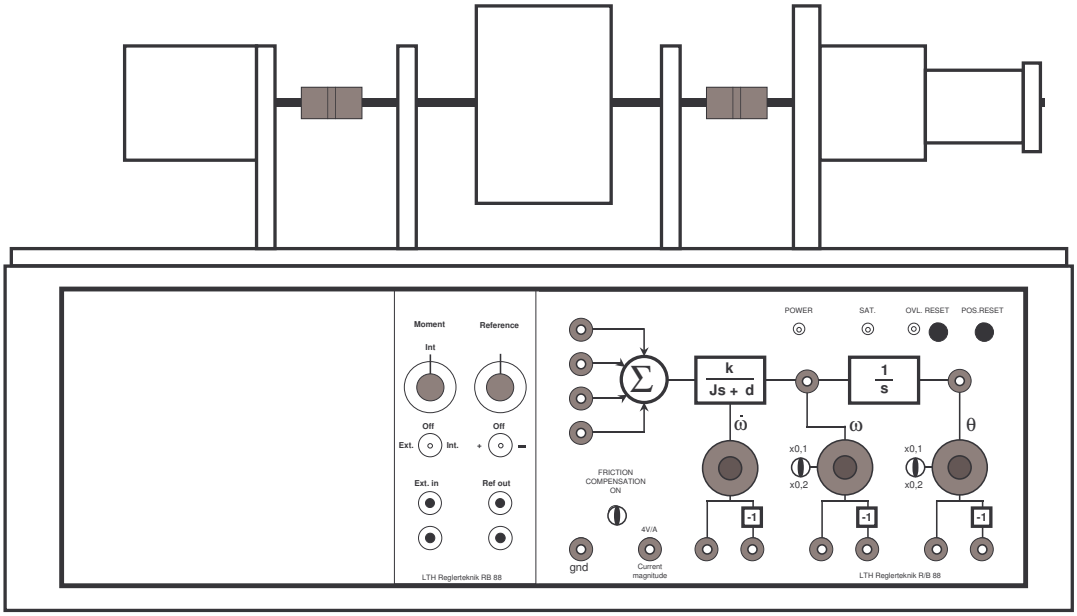
Laboratory Exercise 1

DC servo

Per-Olof Källén



Last update Feb 2013

1. Introduction

1.1 Goal

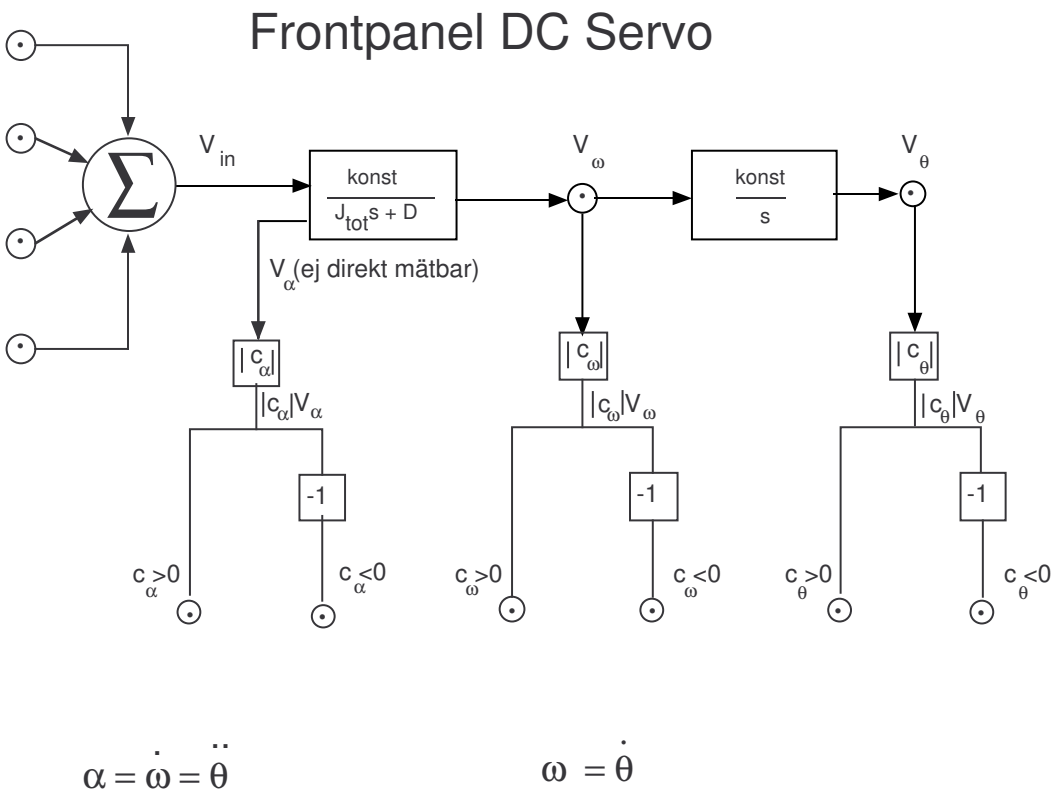
The purpose of the lab is to provide empirical experience of how a simple servo system behaves and study what can be achieved with feedback.

1.2 Set-up

The assignment is performed on a servo motor which can be feedback coupled either directly on the front panel or, connected to a computer with AD/DA-card which implements the controller. In this exercise the computer controller is implemented in Matlab/Simulink using a real-time extension.

2. Influence of simple feedback

This first section covers how to manipulate the servo behavior by connecting feedback signals proportional to the rotor shaft angle position θ, to the angular velocity ω ($\equiv \dot{\theta}$) and to the angular acceleration α ($\equiv \ddot{\theta}$), respectivly, back to the input voltage of the motor. These signals are available on the servo front panel, see Figure 1.



Figur 1: Picture of the front panel of the DC-servo, with indication of what signals you can measure and use for feedback, either directly on the fron panel or via a controller implemented in a computer.

2.1 Rotating Mechanical Systems

A rotating mechanical system with spring action, can be described by the linear second order differential equation

$$J \cdot \ddot{\theta} + D \cdot \dot{\theta} + K \cdot \theta = M \tag{1}$$

where J is the moment of inertia of the rotor, D is the viscous damping coefficent, K is the system's spring constant and M the sum of the torques acting on the system. For a mechanical system without feedback we have that J, D, and K are all positive or zero.

2.2 A simple process model of the servo

To describe what is happening at interconnction we need to have a rough mathematical model of the servo system. In principle, this is described by the equations for the motor and for the interconnected flywheel

$$J_1 \frac{d^2\theta}{dt^2} + D_1 \frac{d\theta}{dt} = M_e + M_{ext} \tag{2}$$

where J_1 is the inertia of the flywheel, D_1 is the viscous damping factor, M_e the torque from the electric motor and M_{ext} represents any external torque acting on the servo (e.g., disturbances like friction etc).

2.3 Feedback

Through various interconnections directly on the front panel of the servo you can obtain the following motor torque

$$\begin{aligned} M_e &= k \cdot (\theta - \theta_0) && \text{position feedback} \\ M_e &= k \cdot \frac{d\theta}{dt} && \text{velocity feedback} \\ M_e &= k \cdot \frac{d^2\theta}{dt^2} && \text{acceleration feedback} \end{aligned}$$

or a linear combination of these. The magnitude of a feedback signal may be varied with the associated potentiometer and its sign may be changed by changing the corresponding connection point (see Figure1).

Example The influence of velocity feedback can be seen by inserting $M_e = k \cdot \dot{\theta}$ in Eq. (2). One then gets the resulting system

$$J_1 \frac{d^2\theta}{dt^2} + (D_1 - k)\frac{d\theta}{dt} = M_{ext}$$

At a comparison with (2) it can be seen that the resulting viscous damping is changed from D_1 to $D_{res} = D_1 - k$. Negative feedback $(k < 0)$ provides increased viscous damping $(D_{res} > D_1)$ whereas positive feedback $(k > 0)$ gives reduced viscous damping. The feedback system is unstable when D_{res} is negative, i.e., for sufficiently large positive values of k. In generally, the feedback system will be unstable if any of the parameters J_{res}, D_{res} or K_{res} is negative.

2.4 Front panel buttons

θ is the angular position, i.e., the rotation around the axis with respect to some initial value θ_0. One may reset θ_0 to the desired (present) shaft position by pressing the "POS. RESET"-button on the panel.

The toggle switch labeled "FRICTION Compensator" should in the ideal case compensate for nonlinear friction when activated (but does so only partially).

NOTE! In case of overloading of circuits and the servo motor an "auto-fuse" is triggered and the amplifier is shut down, which is indicated by the LED labeled "OVERLOAD" and the servo stops operating. The fuse is reactivated by pressing the button marked "RESET".

In some of the following tasks, you should investigate the servo behavior at different simple feedback connections. This can be done by manually pushing the flywheel in/into motion, and then study its transients. When position feedback is investigated an alternative way to study load disturbances is to add an extra disturbance voltage from a variabel external voltage source (also available at the front panel of the servo).

Write down what you know about how the parameters J, D, and K affect the behavior of (1).

Examine the open loop system to see how it behaves. Then introduce velocity feedback. Vary the gain of the velocity feedback and the sign of it. Record both qualitative and quantitative observations.

Can you explain your observations using equation (1) - (2) and the type of feedback you use (as in the example above)?

Introduce position feedback instead. Vary the gain and the sign of the position feedback. Change the position of equilibrium θ_0 and see how this affects. Record your observations as in the previous exercise.

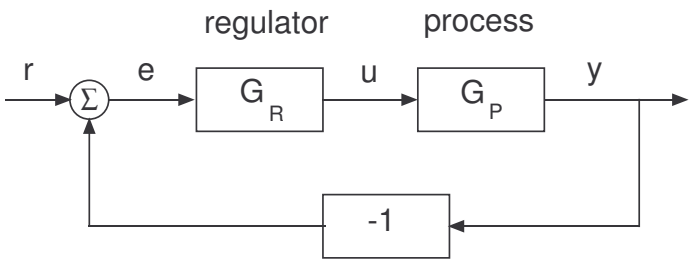
Can you explain your observations using Equations (1) – (2) and the kind of feedback you are using?

Make a feedback connection from the angular acceleration and note its effets.

Now use all the different kinds of feedback from prevouis experiments together. Choose by the conclusions you have drawn in the previous tasks appropriate signs of the different feedback signals you use and change their magnitudes so that you will receive a non-oscillatory system whose axis angle quickly settles into a equilibrium position when you disturb the system by an external torque (represented e.g., by an external voltage added to the servo input). Take notes of your found feedback settings (signs and potentionmeter settings/magnitudes).

2.5 Observations on simple feedback

You have now seen that by appropriate feedback one can get the servo to behave more or less as desired. However, due to that the feedback signals are limited in magnitude, one can not change the behavior beyond a certain degree. What you can do is e.g., to get the servo to behave as if it had larger, or smaller, flywheel than it really has. One can also obtain a strongly damped behavior to prevent undesirable oscillations from occurring. You can also receive spring action which tends to return the servo shaft angle towards a predetermined position. By combining different types of feedbacks one can therefore obtain a servo with the desired behavior.



Figur 2: Closed loop system

2.6 Controller instead of simple feedback

Instead of using feedback on the front panel of the servo we will have a look at a computer implemented controller to accomplish the same, or in some cases better behavior. The difference in the various ways to perform the feedback is e.g., in the way in which the measured signals are filtered/processed. In many cases the feedback only consists of a single signal (e.g., the angle measurement) and one then needs to process this signal in such a way that the feedback will be equivalent to several feedbacks on front panel. That feedback is based on only one signal can often be that it is only possible to measure a certain signal or because it is very expensive with equipment capable of measuring other interesting quantities. Regardless of whether the controller measures one or more signals the reason for introducing different types of controllers is that there is a broad mathematical foundation to use solve the different control problems.

Control

In the first section, you looked at how simple feedbacks can affect a system's behavior. You should now proceed to study the different types of controllers for the servo (considering both the servo and the regulation problem, e.g., considering both reference following and disturbance attenuation). The simple feedback on the fron panel is then exchanged when to feedback with different types of controllers in Figure 2.6.

You will connect the servo to a PC and implement a PID-controller.

A realizable PID-controller based on continuous time representation can by the Laplace transform be descried as

$$U(s) = K \cdot \left(E(s) + \frac{1}{sT_I} E(s) - \frac{sT_D}{1 + sT_D/N} Y(s) \right)$$

$$E(s) = R(s) - Y(s)$$

where the control signal $U(s)$ is the output from the controller, $R(s)$ is the reference value (setpoint), $Y(s)$ is the process output (measured value), and $E(s)$ is the control error, i.e., the difference between desired and actual process output. The derivative term appears here only acting on the output y, and is limited through the low pass filtered factor $1/(1+sT_D/N)$. The factor N is typically set to $N = 10$ in the controller. The controller parameters are the proportional gain K, the integral time T_I and derivative time T_D (but also N may be changed).

The time representation of the controller is obtained by the inverse Laplace transform

$$u(t) = K \cdot \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau - \frac{T_D}{1 + \frac{T_D}{N}(\frac{d}{dt})} \cdot \frac{dy}{dt} \right) \tag{3}$$

In the expression above, the reason for the notion of integral time and derivative time could be seen, but also other parametrizations exist where e.g., $K_I = K \cdot \frac{1}{T_I}$ may be used as a lumped parameter for the integral gain.

2.7 Controller Parameters

Sometimes not all parts of the complete PID-controller needs to be used, but e.g., leaving out either the derivative action, the integral action or both. The derivative action is eliminated by putting $T_D = 0$. The integral part can theoretically be eliminated by allowing the integral time to go to infinity because T_I is in the denominator of the regulator expression or by setting $K_I = 0$ in case of this representation. The former version has some practical drawbacks and in general there is a activation switch for the integral part.

The proportional gain K in Eq. (3) can not be eliminated because then the control signal will be identically zero, but there is no problem of alternative implementations allowing e.g., only integral action.

2.8 Process Model

Although in the sequel we will perform different types of control of the same DC-servo, the servo process model will vary. This is because we control various quantities of the servo (the angular position or angular velocity) and consequently also the process model will vary. Therefore, also the controller parameters, K, T_I, and T_D, will have different values for good control of the angular velocity and for good control of the angular position, respectively.

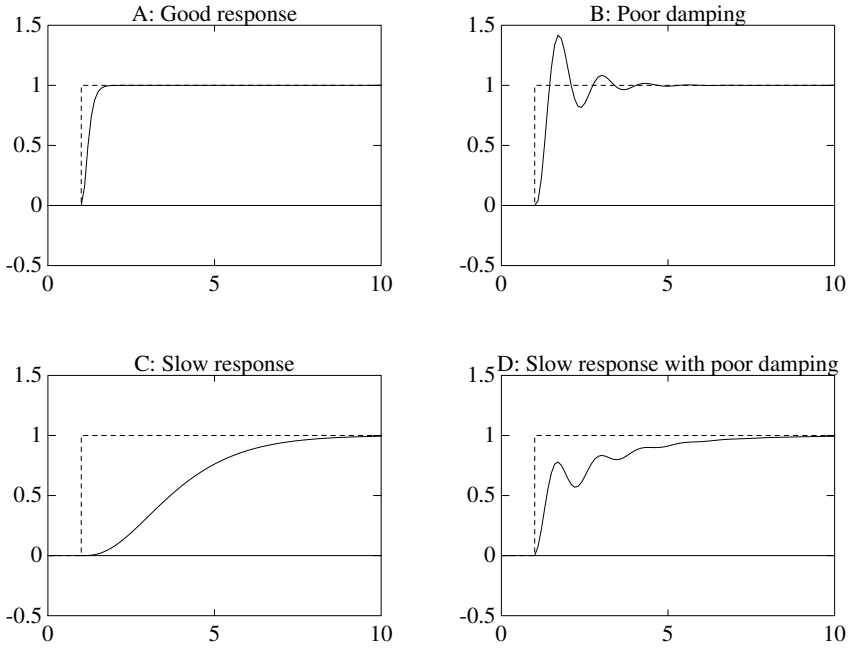
The goal with the control of the angular velocity is to achieve a it desired rotational speed while for the control of angle position is to achieve a it desired angular position, so the model orders in the two cases will also change.

2.9 Inputs and outputs

If you look at the controlled system as a block you see immediately that the reference signal is the input of the system. The closed loop system output and the process output, are the same signals as long as no measurement/output disturbances affect the control system.

2.10 Good control

Ideally, the output should follow the setpoint without deviations. This is not possible for a dynamic system, as the control system has generally the property that it takes



Figur 3: Step responses of varying quality

time for an input to completely propagate through the system. What you can do depends on what kind of process you control. Good control can be said to be when the output as quickly as possible and without unnecessary oscillations or overshooting approaches the setpoint. In particular wen the setpoint is constant you should not have have any stationary control errors, but for robot systems the tracking of reference trajectories is also very important and will be considered in the sequel.

In Figure 2.10 a number of different step responses of the closed system are shown for different settings of the controller parameters. In the different sub-figures also the ideal step response, which never will be quite achieved, is plotted. Step response A has a fast response without both overshoot and undershoots and it also lacks steady state error. This type of transient response is what you strive for. Step response B oscillates (poorly damped), but is yet relatively fast. In C one has a slow system, where it takes long time for the control system to settle after changes in the reference signal. Finally, in D there is a system which is both poorly damped and slow. Note that all the four step responses correspond to stable closed-loop systems, i.e., system where the output does not grow unbounded for a limited input.

The first problem we are goint to have a look at - velocity control - is one of the most common applications of control engineering and essential for servo control.

Velocity control

Draw a block diagram of the closed system obtained by PI-control of the servo's angular velocity. Highlight in particular the reference value, load disturbance, measurement signal and the control signal and clarify by labeling the signals.

The Simulink implementation of the controller may look very similar to your block diagram.

First, use a simple proportional controller, i.e., a controller of the form

$$G_R(s) = K$$

Examine the control for different values of the gain K. Start with a low gain and increase gain gradually until a degenerated behavior is obtained. What is happening? What is it that limits the practically usable gain?

In order not to unnecessarily complicate the solution of a control problems one may in the modelling neglect/ignore effects that are difficult to describe mathematically. In the previous task you saw that, these effects may show up e.g., when trying to push the performance of the controller and that the assumptions/simplifications may then not be valid any longer.

2.11 Disturbance torque

On the experimental set-up there is an additional motor which can be used to give a disturbance torque. This can also be operated from the front panel of the servo and by applying a voltage connected to this one obtains a braking or driving torque.

Investigate for different controller gains of the P-controller how the stationary error depends on the disturbance torque. (The control error $e = r - y$).

Now switch to PI-control. Try to find appropriate values of the controller parameters of the PI-controller

$$G_R(s) = K(1 + \frac{1}{sT_I})$$

Watch the settling phase (the step response) when making incremental changes in the setpoint and varying the load (disturbance torque). Make a note of the parameter values for a good controller.

What control error do you get in stationarity using a PI-controller for a constant load disturbance?

Can you explain why?

Start with the best values of K and T_I which you found in the previous task. Studying in turn how the increase or decrease of K and T_I, respectively, affect the closed-loop behavior. Make a table where you either in words describe the impact of changing parameters or by sketching the step response (a rough picture). This scheme can be used to determine how the parameters of a poorly tuned PI controller should be changed to improve the control. The conclusions, however, are only valid for a control of a processes of a certain type (i.e., the DC-servo in this case).

Position control

In the previous section the angular velocity of the servo was controlled. Here we will look at a somewhat more difficult control problems, namely to control the servo angle to a desired position.

Draw a block diagram of the closed system when this is directly controlled by a PID-controller. Mark setpoint, load disturbance, control signal and measurement signal.

Try to find a PI-controller for controlling the angular position by trial and error with different values of K and T_I. Is a PI-controller sufficient for controlling the angular position?

Now try with a PID-control. Vary as before setpoint and watch the settling. Determine appropriate parameter values K, T_I, and T_D. Make a note of a set of good parameter values.

Hint: To simultaneously determine three different parameters can be cumbersome and time consuming. A simpler method to find a PID-controller is to first tune a PI- or PD-controller. In this case the PD-controler is preferable to start with. Then the third parameter (T_I or T_D) is introduced, so that its effect can be clearly visible in the behavior. From this mode, you can usually find a good set of parameters by moderate variations in K, T_I, and T_D.
Starting now from this or these sets of controller parameters and description of each parameter as variations of it has the impact on the closed system. Make up a table where you either in words describing or by sketching step response, a picture of the impact parameters.

3. Cascade control

Now make a cascade controller according to Figure 4, where you re-use you velocity controller from a previous task and add an outer position controller.

You may start with the outer position controller as a P-controller. Make some step responses to determine the gain and evaluate it against load disturbances.

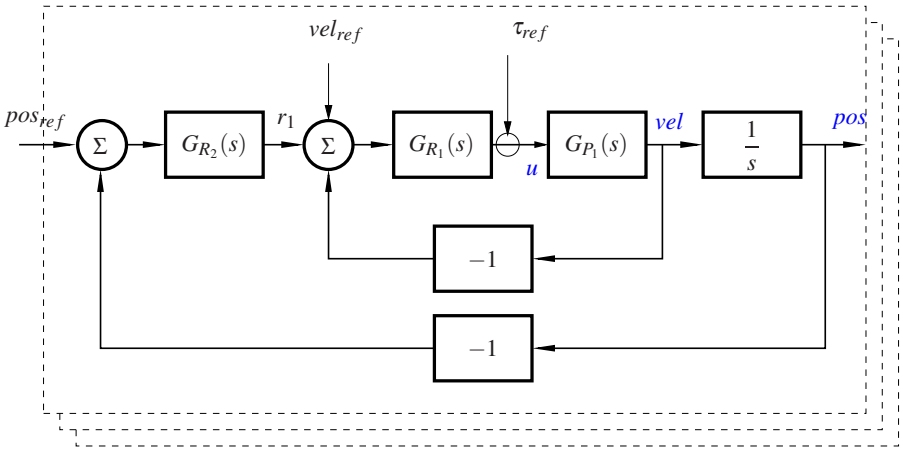
Remark! In robot applications it is important that the step response of the servo does not have any overshoot in position/angle!

Note: In contrast to many process control applications where cascade control also is used, for the robot servo you will in general not have any load disturbances acting between “velocity” and “position”. What does this imply for where you need to have integral action in your controller?

The cascade structure allows for feedforward of reference signals. Generate a smooth angular position reference (e.g., a sinusoid of suitable amplitude and period time) and use that as reference signal for the angular position for your cascaded controller. Increase the frequency of the reference until there is a noticeable lag in the servo response. Make a feedforward of the corresponding velocity reference (e.g., cosine as a “derivative of the sinusoid reference”) and feed this also to your cascaded controller according to Figure 4. Do you notice any difference?

Extra: How would you generate the torque feedforward?

Extra: Try any other design method of you choice for the servo (pole placement/ Kalman filter+state feedback design/LQG).



Figur 4: Cascade controller for a servo / robot joint