Exercise 5

Feedforward control

5.1 Mass feedforward design

Install the 'third-degree path generator' (Ref3) on your computer:

- Download `Ref3.zip` from http://cstwiki.wtb.tue.nl/index.php?title=Ref3.
- Unzip the files into a directory, preferably in `$matlabroot\toolbox\`, which is the Matlab toolbox directory.
- Open Matlab, go to this directory and run: `install_ref3.m`.
- See `ref3_example.mdl` for an example of how to use Ref3.
- For more information on Ref3, see http://www.dct.tue.nl/home_of_ref3.htm

Make a simulink block diagram with as system

$$G(s) = \frac{1000 \cdot 3202}{s^2(s^2 + 20s + 3202)}.$$

Design a stabilizing controller with a bandwidth of 10 Hz. Use a third order setpoint and apply an acceleration feedforward to achieve the lowest tracking error possible. Can you explain this acceleration feedforward by looking at the transfer function?

5.2 Feedback control for feedforward

Open ShapeIt and choose the first example ('mass') as plant. Assume that this plant represents a rotating mass connected to a DC-motor. The transfer function is thus from a motor voltage in [mV] to a rotation in [rad]. Furthermore assume that this rotation is measured with an encoder with 8000 pulses per revolution.

- Determine the resolution of the measurements in [rad].
- Design three simple controllers (lead-filters) yielding bandwidths of 1 Hz, 10 Hz and 100 Hz, each one with the same robustness margins.
- Choose the default 3rd order setpoint in ShapeIt as the reference and compare the time responses of each of the closed loops. Which controller would you prefer for feedforward *design*? Explain why.

5.3 ShapeIt and feedforward

ShapeIt is a very useful tool to design both a suitable trajectory and an appropriate controller for eventual feedforward tuning. A good exercise is the following.
Load the 'viscous damping' example, and design a controller and a 3rd degree setpoint, such that the error profile has the desired shape (and magnitude) to accurately tune the feedforward terms k_{fa} and k_{fv} later on. Export the plant and the controller, implement them in a Simulink closed loop scheme, and tune these feedforward terms to suppress the error.
Note that you still need the Ref3-toolbox to implement your designed reference trajectory.

5.4 Feedforward design !!⋆!!

You should have the Ref3-toolbox installed for this exercise.
Download the Simulink file `feedforward.mdl` from OASE (Matlab r2006b). Students with Matlab 6.1 can download `feedforward_m6.mdl`, students with Matlab 7.0.4 can use `feedforward_m7.mdl`.
The file contains an unknown plant and a low bandwidth controller, which is already tuned correctly.

- Explain why we need a (relatively) *low bandwidth* controller for feedforward design.
- Create a challenging reference trajectory for this plant. Create it such that the three phenomena (mass acceleration, viscous damping and dry friction) are clearly visible in the error signal.
- Tune the three feedforward terms in the right order. Compare the resulting error signal with the initial error.

Exercise 6

The digital environment

6.1 Delay and sampling

Sampling a continuous time system (which is inevitable in practice) will 'make' this system discrete (i.e. your computer will 'see' a discrete system). This sampling also introduces *delay*, which is typically a non-linear phenomenon. It is known, however, that for a total delay time T_d the delay frequency response $H_d(j\omega)$ is given by

$$H_d(j\omega) = e^{-j\omega T_d}. \tag{6.1}$$

1. Derive expressions for the magnitude $|H_d(j\omega)|$ and the angle $\angle H_d(j\omega)$. What does this mean for the Bode plot? *Hint: rewrite* (6.1) *into polar form.*

2. Given is a controlled system with:

   - sampling time Ts=2 ms
   - calculational delay (in addition to the sampling interval) Tc=100 μs.

   Calculate the resulting phase delay in degrees at 10 Hz.

6.2 Digital Control Systems

Define in Matlab a lead controller with a zero at 10 Hz and a pole at 100 Hz.

- Determine 5 different discrete-time equivalents using `c2d.m`, with 'zoh', 'foh', 'tustin', 'prewarp' and 'matched', all using a sample frequency of 500 Hz. Compare their frequency responses with the continuous time one.

Add in series with this controller a notch at 200 Hz, with a depth of 20 dB, and a damping of the zeros of 0.01.

- Determine the same 5 discrete-time equivalents using `c2d.m`, all using a sample frequency of 500 Hz. Compare their frequency responses with the continuous time one.
- In a real-time experiment a controller has to be discretized before it can be used (this is either done by Simulink or explicitly by the user). Suppose the above controller was designed to stabilize a plant with a real (i.e. measured) resonance at 200Hz. Which discretization would you prefer? What happens if the sample frequency is increased?
- Explain how discretizing a continuous time controller can sometimes yield (unexpected) closed loop instability when implemented.

6.3 Discrete-time filters

One of the benefits of the discrete-time domain is the possibility to define filters with a frequency response which do not have a straightforward continuous-time equivalent. A simple example is the *Moving Average Filter*.

- Write down the difference equation of a filter whose output $y(k)$ is the average of the ten current and previous inputs $u(k-i)$, with $i = 0 \ldots 9$.
- Convert the difference equation into a transfer function by using the z-transform.
- Is this filter a FIR filter or not?
- Plot the frequency response of the moving average filter. Look at the shape of magnitude plot; which frequencies is it suppressing and explain why? Look at the phase; what's the disadvantage of using this filter in a feedback loop?
- Compute and plot the poles and zeros of the filter. Asses the stability of the filter by looking at the poles, and explain the frequency response by looking at the zeros (and the poles).