Mandatory assignment 1 - TEK4030

Exercise 1 - Control Theory - Linear systems

In this exercise we will look closer at the control of a motor. The motor has the transfer function

$$\frac{X(s)}{U(s)} = \frac{1}{s(1 + T_M s)} \tag{1}$$

a) Analyze the stability of this system by finding the poles of the system.

b) Draw a block diagram of the system by only using basic blocks (i.e. constants and integral blocks).

We will now include the PD controller

$$U(s) = K(1 + T_D s)E(s) \tag{2}$$

where $E(s)$ is the error, defined as $E(s) = R(s) - X(s)$.

c) What is the order of the system with and without the controller?

d) Extend the block diagram with the controller. You can assume that R is constant, i.e. $Rs = 0$.

e) Find the poles and zeros of the system as an expression. Plot the poles and zeros in the complex plane for different values of K (using for instance MATLAB), assuming that $T_M = 2$ and $T_D = 1$. The resulting plot is an approximation of the root locus plot for the system. Comment the plot with regards to stability and oscillations.

Exercise 2 - Control Theory - Lyapunov direct method

Verify that the system

$$\dot{x} = -y - x^3 \tag{3}$$

$$\dot{y} = x - y^3 \tag{4}$$

is stable using the Lyapunov direct method. Use the following Lyapunov candidate function

$$V(x, y) = x^2 + y^2 \tag{5}$$

Exercise 3 - Centralized Joint Space Motion Control - Simulink

In this exercise we will use different control laws on a two-link planar arm modelled in Section 7.3.2 in the textbook (p. 2~~65~~ 82). In this exercise the model will be slightly modified to include viscous friction. The modified model of the manipulator is

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{6}$$

where

$$\boldsymbol{F}_v = \mathrm{diag}(f_{v1}, f_{v2}) = \mathrm{diag}(0.1, 0.1) \tag{7}$$

The values for the other parameters are given in Example 7.2 in the textbook (p. 2~~69~~ 87). Simulink blocks for the model is provided with this exercise. First run parameters.m to load all model parameters, then simulate your controller. First we will look at the PD control with gravity compensation.

a) Draw the full block diagram of the model in (6) and the controller (8.51) 396 in the textbook.

b) Simulate the control system in Simulink with a step response for $\boldsymbol{q}_d$ from $[0,0]^T$ to $[\frac{1}{2}\pi, \frac{1}{2}\pi]^T$ by using the provided blocks to simulate the model. The file two_link_model.slx provides the model (6). Find values for $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ by trial and error that yields a responsive system without oscillations. Report the values along with plots of $\boldsymbol{q}$ and $\boldsymbol{q}_d$.

We now turn our attention to the inverse dynamics control

c) Write the matrix expression of the inverse dynamics control law (u) for the system as a function of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\tilde{\boldsymbol{q}}$, $\dot{\tilde{\boldsymbol{q}}}$ and $\ddot{\boldsymbol{q}}_d$. Use the matrices $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{F}_v$ and $\boldsymbol{g}$ in the expression. (Tip: the control law u is found in the textbook)

d) Draw the full block diagram of the inverse dynamics control of the system.

e) Simulate the control system in Simulink using a trajectory $\boldsymbol{q}_d = [\sin 2\pi ft, \sin 2\pi ft]^T$, and gains $\boldsymbol{K}_P = \text{diag}(25, 25)$, $\boldsymbol{K}_D = \text{diag}(5, 5)$. Report the plots of $\boldsymbol{q}$ and $\boldsymbol{q}_d$ for $f = 0.5, 1, 2$. How large is the error? Show that the system is under-damped.

f) Design control gains so that the system is critically damped with $\omega_n = 5$ for both degree of freedom. Simulate the new system and report the plots of $\boldsymbol{q}$ and $\boldsymbol{q}_d$ for $f = 2$.

g) Now assume that the manipulator model has errors. Assume that $\widehat{\boldsymbol{B}} = 0.9\boldsymbol{B}$ and $\widehat{\boldsymbol{n}} = 0.95\boldsymbol{n}$. Simulate the system in Simulink using the control parameters from above. Report plots of $\boldsymbol{q}$ and $\boldsymbol{q}_d$. How and why does the performance change?

Exercise 4 - Independent Joint Control - ROS

In this exercise we will expand the nodes in to ROS intro tutorial to use a 2 DoF planar robot model. The model is based on (~~8.18~~ 527) and (~~8.11~~ 325) in the textbook, and the parameters for the model is taken from ~~Example 7.2~~ 287 and ~~Problem 5.2~~ 250 in the textbook. The control input $\boldsymbol{u} = \boldsymbol{K}_t\boldsymbol{R}_a^{-1}\boldsymbol{G}_v\boldsymbol{v}_c$.

a) Expand the ROS nodes you made in the ROS intro tutorial to use the 2 DoF robot model. The simulator should be updated to use a new model, and the controller should now control two joints instead of one. Have a look at the following site for more detailed information on how to do this.
`https://github.uio.no/TEK4030/tek4030_ros_intro/blob/master/oblig_1.md`

b) Find the complete dynamic model for the joint motors by combining (~~8.18~~ 327), (~~8.11~~ 325) and the control input u given above. Hint: have a look at (~~8.3~~ 374). The dynamic model you find is the same model that is simulated in the simulator.

c) Divide the model into one linear and decoupled part and one nonlinear coupled part.

Now we are going to look at the independent joint control using position feedback, as described on page 312 in the textbook.

327 d) Draw the full block diagram of the linear decoupled part and the controller. The coupled part should come in as a disturbance.

e) Transform the control law to the time domain, in order to be able to implement the controller in C++.

f) Implement the controller in ROS by overwriting the controller already there. Have a look at the webpage below for tips on how to implement integration into your code
`https://github.uio.no/TEK4030/tek4030_ros_intro/blob/master/oblig_1.md`

g) Simulate the system and try to find good values for K_P and T_P. Include a plot of joint angles and the values you found in your report. Set the joint set point to $(\frac{1}{4}\pi, -\frac{1}{4}\pi)$. Remember that these values must be converted to motor positions before using them. This can be done by multiplying with $\boldsymbol{K}_r$.

We now turn our attention to the independent joint controller using both position and velocity feedback, as described on page 314 in the textbook.

h) Transform the control law to the time domain, in order to be able to implement the controller in C++.

i) Implement the controller in ROS. Use ζ and ω_n as tuning parameters, as specified in (8.30) and (8.31) in the textbook. Remember that $T_V = T_m$ in order to cancel the pole in the in the system. Equation (5.12) in the textbook gives the gain and time constant of the system, where I_m is the average inertia from the $\bar{\boldsymbol{B}}$ matrix. The average inertia can be found by removing the configuration dependent terms in b_{11} and b_{22} in example 7.2 on page 267 in the textbook (alternatively by copying the terms from the planar manipulator library).

j) Simulate the controller using $\omega_n = 2$ and $\zeta = \{0.5, 1, 2\}$. Include plots of the joint angles in your report. Is the system responding as a critically damped system when $\zeta = 1$? Why or why not?

Submission

Submit a small report answering the exercises and all the questions in the exercises. Also attach your Simulink and C++ code. Submit your assignment in Canvas (`canvas.uio.no`).