Lab 1, Matlab Simulations

IE1304 Control Theory

Introduction

Matlab is installed on the lab computers. You can also install Matlab on your own computer by downloading it from KTH ProgDist, see `http://progdist.ug.kth.se/public/`. Unfortunately only *Matlab*, *Simulink* and *Symbolic Toolbox* are included. The important toolboxes *Control Systems Toolbox* and *Signal Processing Toolbox* are no longer included.

Some useful Matlab commands

- Define the vectors x and y.
  ```
  x = [0 1 2 3 4 5 6 7 8 9]
  y = [5 7 4 3 2 4 6 6 5 6]
  ```
- Draw the graph $y = y(x)$.
  ```
  plot(x,y)
  ```
- Show help for the `plot` command.
  ```
  help plot
  ```
- Define the vectors t and u containing values in the range 0 to 10 with the interval 0.1 and values in the range 0 to 20 with the interval 0.2
  ```
  t=0:0.1:10
  u=0:0.2:20
  ```
- Multiply the elements with the same indexes in t and u. Note the dot before the multiplication which instructs Matlab to do the multiplication element by element instead of performing a matrix multiplication.
  ```
  v = t.*u
  ```
- Draw the graphs $e^{-0.1} * t$ and $v(t)$ in the same figure.
  ```
  plot(t,exp(-0.1)*t,t,v)
  ```
- Create a vector containing elements 3, 4 and 5 from the vector t.
  ```
  t1 = t(3:5)
  ```

- Ending an instruction with a semicolon means that its result is not displayed.
  `t1 = t(3:5);`
- Define the matrix $\mathbf{A}$.
  `A=[1 2; 3 4]`
- Transpose the matrix $\mathbf{A}$ and store the result in $\mathbf{B}$.
  `B= A'`
- Multiply the matrices $\mathbf{A}$ and $\mathbf{B}$.
  `C = A * B`
- Invert the matrix $\mathbf{C}$.
  `D = inv(C)`
- Create a polynomial with roots in 1 and 2. The function call `poly(V)`, where V is a vector, returns a vector whose elements are the coefficients of the polynomial whose roots are the elements of V .
  `P = poly([1, 2])`
  The function returns `P = [1, -3, 2]`, which corresponds to the polynomial $P = x^2 - 3x + 2$.
- Find the roots of the polynomial $P = x^2 - 3x + 2$.
  `roots([1, -3, 2])`
- Get the fraction expansion of $\dfrac{1}{x^2 - 3x + 2}$.
  `[r,p,k] = residue([1],[1, -3, 2])`
  The `residue` function returns the residues, poles, and direct term of the fraction expansion. Here, the answer is $r = [1; -1]$, $p = [2; 1]$, $k = [\,]$, which means $\dfrac{1}{x^2 - 3x + 2} = \dfrac{1}{x - 2} + \dfrac{-1}{x - 1} + 0$
- Multiply the polynomials $x^2 - 3x + 2$ and $2x^2 + x - 5$
  `conv([1, -3, 2], [2, 1, -5])`
  The answer is `[2, -5, -4, 17, -10]` which means that $(x^2 - 3x + 2) \cdot (2x^2 + x - 5) = (2x^4 - 5x^3 - 4x^2 + 17x - 10)$

Preparation tasks for Matlab, to be solved BEFORE the lab

Task 1

Read the basic parts of the Matlab introduction. You can find it in Matlab under the menu `Help -> Product Help`, choose `MATLAB` in the tree to the left. Tutorials can be found under `Getting Started` and `MATLAB Demos`.

Also read chapter 24, *Simulering av reglersystem med Matlab och Simulink*, in the course text book.

Task 2

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Given the matrix $\mathbf{A}$ above, examine what the following Matlab commands do.

a) `sum(A)`

b) `diag(A)`

c) `A(1:2,2)`

Task 3

Use the following matrices

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

and solve the equation system $\mathbf{Ax} = \mathbf{B}$ using the operator `\`
You can get help with the Matlab command `help \`.

Task 4

Which symbol is used for imaginary numbers in Matlab?

Task 5

What is the meaning of `NaN`?

Task 6

How is π written in Matlab?

Task 7

The differential equation

$$T\frac{dx}{dt} + x = u, \quad \begin{cases} T = \text{time constant} \\ u = 0, t < 0 \\ u = u, t \geq 0 \end{cases}$$

has the solution $x(t) = x(0)e^{-\frac{t}{T}} + u(1 - e^{-\frac{t}{T}})$. Choose $T = 1s, u = 10, x(0) = 0$ and perform the following tasks in Matlab.

a) Plot $x(t)$ for the time frame $0 \leq t \leq 5s$.

b) Plot in the same figure $x(t)$ when $T = 2s$.

c) Plot in the same figure $x(t)$ when $x(0) = -2$.

d) Plot in the same figure $x(t)$ when $u = 5$.

You can get help by looking at the Matlab help text for the instructions `plot` and `hold`.

Task 8

Solve the equation $x^4 - 10x^3 + 35x^2 - 50x + 24 = 0$

Task 9

Calculate the fraction expansion of $\dfrac{2x+1}{x^4 - 10x^3 + 35x^2 - 50x + 24} \cdot \dfrac{3}{5x^3 + 2x^2 - 4}$

Preparation tasks for Simulink, to be solved BEFORE the lab

Task 1

Simulate a differential equation in Simulink using the guide below.

1. Type `simulink` at the Matlab command prompt, `Simulink Library Browser` is opened.
2. Create a new model.
3. Choose the group `Continous` in the tree to the left and drag two `Integrators` to the model, see Figure 1. To rotate a block you mark it and type `ctrl-r`.

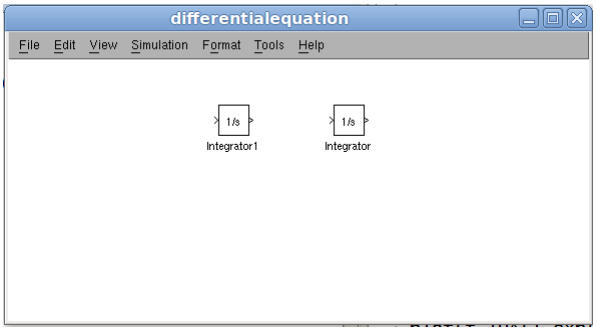

Figure 1: Simulink model with two integrators.

4. Choose the group `Commonly used blocks` in the tree to the left and drag two `Sums` to the model, see Figure 2. Double clicking on a `Sum` block opens the `Function Block Parameters` window where the `Sums` can be configured.

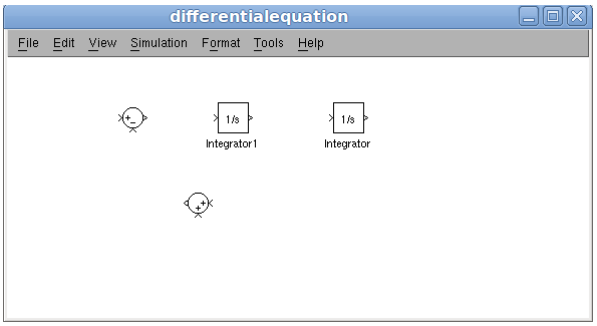

Figure 2: Simulink model with integrators and sums.

5. Next, two `Gains` are placed in the model, see Figure 3.

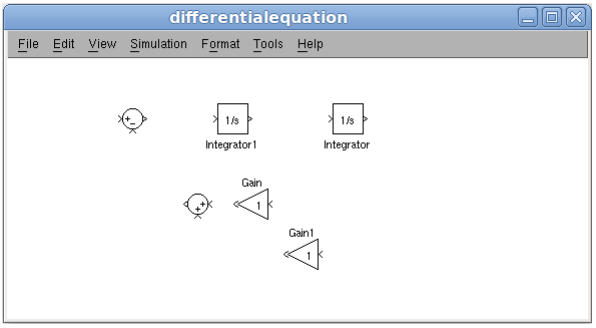

Figure 3: Simulink model with integrators, sums and gains.

6. Finally, add a `Step` from the `Sources` group and a `Scope` from the `Sinks` group and connect the blocks, see Figure 4. The `Step` will be our input signal and the `Scope` will enable us to watch the output.

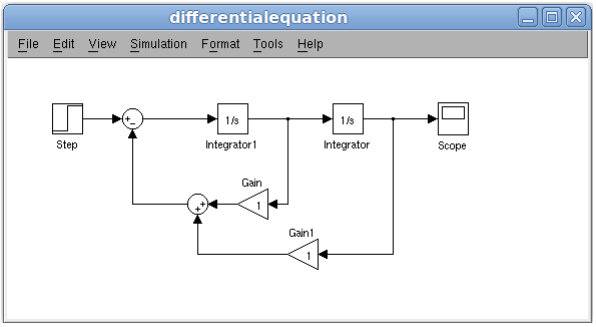

Figure 4: Complete Simulink model.

7. Now it is time to simulate. The simulation is configured in the `Configuration Parameters` window which is opened from the menu, `Simulation -> Configuration Parameters`. Here, there is no need to change any parameters.

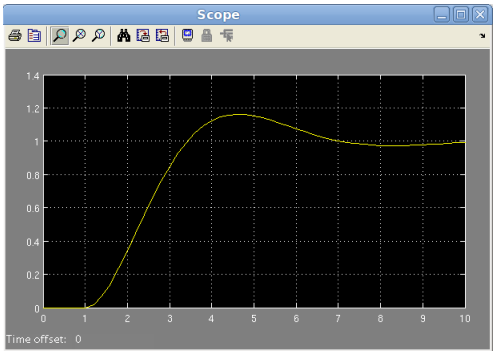

Figure 5: Simulation result.

8. Start the simulation by choosing `Simulation -> Start` in the menu and double click on the `Scope` to display the result. To zoom the curve properly, right click on it and choose `Autoscale`. The result should look like Figure 5.

Task 2

a) Which differential equation did we simulate in the previous task?

b) Verify the answer by creating the same model using a `Transfer Fcn` block (from the `Continous` group) instead of `Integrators`, `Sum`s and `Gain`s. The `Transfer Fcn` should describe the Laplace transform of the differential equation.

Lab tasks, to be solved at the lab

Task 1

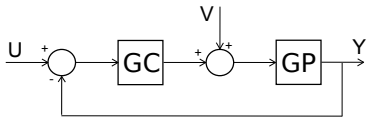

Figure 6: Feedback system.

Use Simulink to simulate the system depicted in Figure 6, let $GC = K$, $GP = \frac{1}{s^3+3s^2+3s+1}$. Decide the following by simulating in Simulink.

a) Maximum gain, K_{max}, without making the system unstable.

b) Steady-state error when $V = 0$, U is a unit step and K has the values $0, \frac{K_{max}}{4}, \frac{K_{max}}{2}, \frac{3K_{max}}{4}$.

c) Steady-state error when V is a unit step, $U = 0$ and K has the values $0, \frac{K_{max}}{4}, \frac{K_{max}}{2}, \frac{3K_{max}}{4}$.

Task 2

Do the same as in the previous task, but set $GC = \frac{K}{s}$.

Task 3

Use Matlab to draw the Bode diagram for the system in Figure 7 with $GC = 2$ and $GP = \frac{2e^{-s}}{1+2.5s}$ and decide if the system is stable. If so, decide the phase margin, Φ_m, and the amplitude margin A_m. The delay can be modeled in matlab by specifying GP as `GP=tf(2,[2.5 1],'InputDelay',1)`. The bode diagram and the margins can be displayed with the command `margin(GP)`.

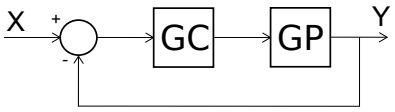

Figure 7: Feedback system.

Task 4

A heater process has the transfer function $G_P(s) = \frac{1}{(1+3s)^2}$. It is regulated by a controller with transfer function $G_C(s) = 3(1+\frac{1}{2s})$. The sensor used to feed back the output signal has the transfer function $G_S(s) = 1$.

a) Construct the system in Simulink.

b) Simulate a unit step input signal. Measure maximum overshoot, settling time and rise time. Also check control input (signal from controller to process).

c) Simulate a unit step disturbance, this can be done the same way as in task 1c above. Again measure overshoot, settling time, rise time and check control input.

d) What type of controller is used?

Task 5

Perform the same simulation as in task 4b above, but in Matlab without using Simulink. This can be done using the following commands:

1. Define the transfer function variable `s`: `s=tf('s')`
2. Define the transfer function `GP`: `GP = 1/(1 + 3*s)^2`
3. Define the transfer function `GC`: `GC = 3*(1 + 1/(2*s))`
4. Calculate the feed forward transfer function, `GCGP`: `GCGP = GC * GP`
5. Calculate the transfer function of the entire feedback loop, `GTOT`: `GTOT = feedback(GCGP, 1)`
6. Plot the step response: `step(GTOT)`
7. Calculate step response properties: `stepinfo(GTOT)`