Exercise 1

Simulation of 2 DOF freedom helicopter model in Simulink

Aim:

a) To be familiar with the use of Simulink for making a simulator to simulate a dynamic system whose nonlinear mathematical model is provided.
b) To simulate the linear model and to compare the results with between the linear and nonlinear model. This simulator can be later on used for the project part of the course (so keep your files safe).

1. Preliminary requirements

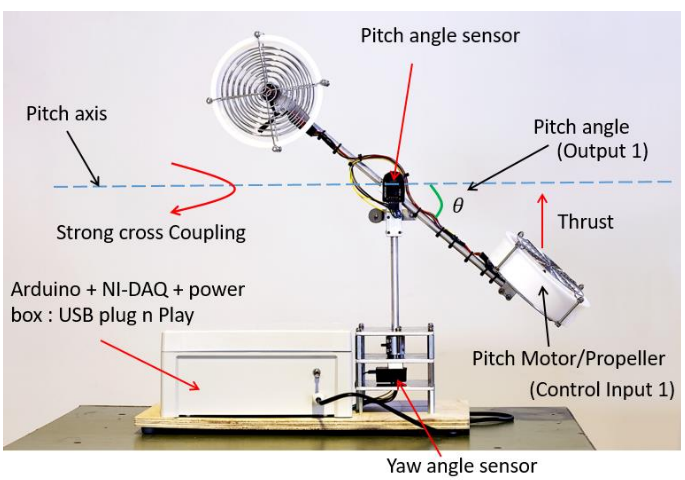
You should read the description of the 2 DOF helicopter model and be familiar with the mathematical model of the system. Please read the document
https://web01.usn.no/~roshans/mpc/downloads/description_helicopter.pdf

You should also have already seen the following two videos (examples of simulating an inverted pendulum):

https://web01.usn.no/~roshans/mpc/videos/lecture1/openloop-simulation-pendulum.mp4

https://web01.usn.no/~roshans/mpc/videos/lecture1/realtime-simulation-pendulum.mp4

2. Mathematical model of the helicopter unit



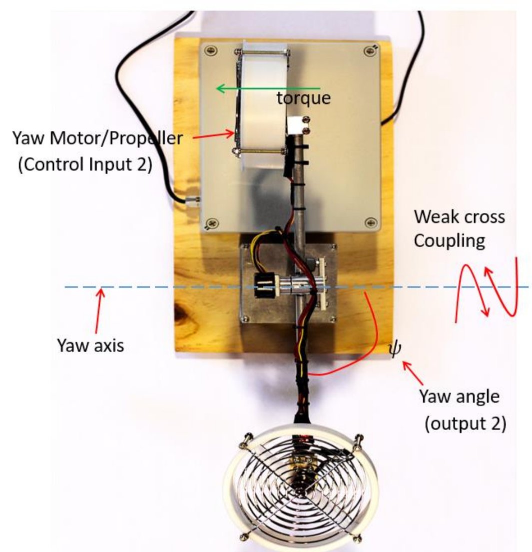


Figure 1: 2-DOF helicopter unit at USN (side view and top view)

Let us define the following:

<u>Control Inputs:</u>

V_{mp} := voltage applied to the picth motor, V_{my} := voltage applied to the yaw motor

Measured Outputs:

θ := pitch angle, ψ := yaw angle

The system can be described with four states:

θ := pitch angle ψ := yaw angle

ω_θ := pitch angular velocity ω_ψ := yaw angular velocity

The block diagram of the system showing the inputs, outputs and the states is shown in Figure 2.

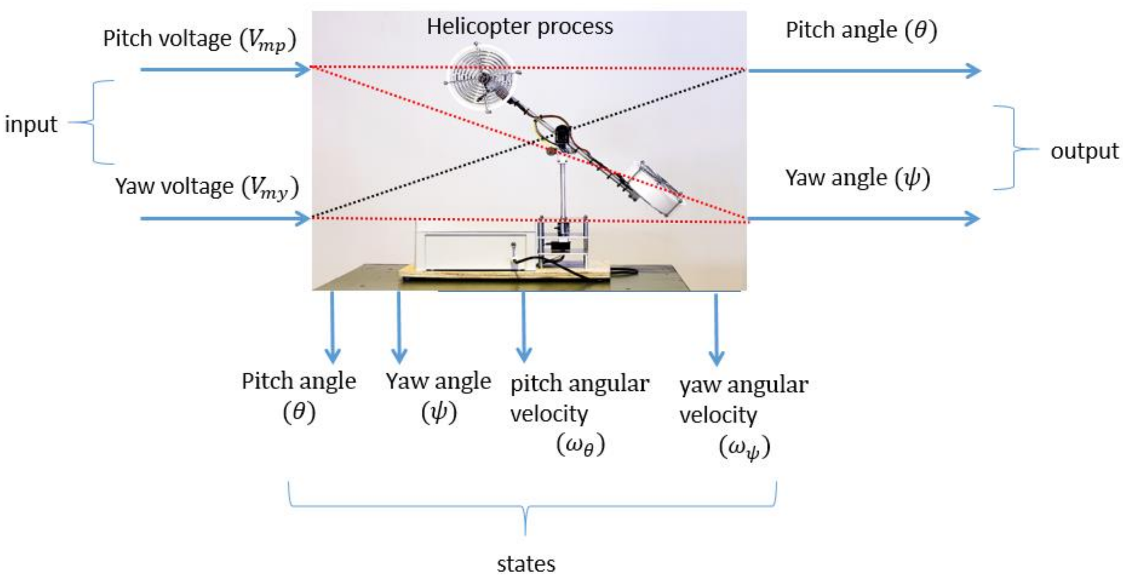


Figure 2: Block diagram of the system showing inputs, outputs and states

The nonlinear model is described by a set of four ordinary differential equations (ODEs).

$$\frac{d\theta}{dt} = \omega_\theta \tag{1}$$

$$\frac{d\psi}{dt} = \omega_\psi \tag{2}$$

$$\frac{d\omega_\theta}{dt} = \frac{K_{pp}V_{mp}}{J_{eq,p} + m_{heli}\, l_{cm}^2} - \frac{K_{py}V_{my}}{J_{eq,p} + m_{heli}\, l_{cm}^2} - \frac{B_p\omega_\theta + m_{heli}\, \omega_\psi^2\, \sin(\theta)\, l_{cm}^2 \cos(\theta) + m_{heli}\, g \cos(\theta)\, l_{cm}}{J_{eq,p} + m_{heli}\, l_{cm}^2} \tag{3}$$

$$\frac{d\omega_\psi}{dt} = \frac{K_{yp}V_{mp}}{J_{eq,y} + m_{heli}\cos^2(\theta)\, l_{cm}^2} - \frac{K_{yy}V_{my}}{J_{eq,y} + m_{heli}\cos^2(\theta)\, l_{cm}^2} + \frac{2m_{heli}\, \omega_\psi \sin(\theta)\, l_{cm}^2 \cos(\theta)\, w_\theta}{J_{eq,y} + m_{heli}\cos^2(\theta)\, l_{cm}^2} - \frac{B_y\omega_\psi}{J_{eq,y} + m_{heli}\cos^2(\theta)\, l_{cm}^2} \tag{4}$$

Table 1: Parameters of the system: part 1

Parameter	Description	Value
l_{cm}	Distance between the pivot point and the center of mass of helicopter	0.015 [m]
m_{heli}	Total moving mass of the helicopter	0.479 [kg]
$J_{eq,p}$	Moment of inertia about the pitch axis	0.0172 [kg-m^2]
$J_{eq,y}$	Moment of inertia about the yaw axis	0.0210 [kg-m^2]
g	Acceleration due to gravity on planet earth	9.81 [m-s^{-2}]

Table 2. Parameters of the system: part2

Parameter	Description	Value
K_{pp}	Torque constant on pitch axis from pitch motor/propeller	0.0556 [Nm/V]
K_{yy}	Torque constant on yaw axis from yaw motor/propeller	0.21084 [Nm/V]
K_{py}	Torque constant on pitch axis from yaw motor/propeller	0.005 [Nm/V]
K_{yp}	Torque constant on yaw axis from pitch motor/propeller	0.15 [Nm/V]
B_p	Damping friction factor about pitch axis	0.01 [N/V]
B_y	Damping friction factor about yaw axis	0.08 [N/V]

3. Tasks

The following two tasks should be completed.

Task 1:

Openloop simulation of the helicopter nonlinear model in Simulink

By following the example videos, please make a simulator for simulating the nonlinear mathematical model (equations 1 -4 and the parameters in Tables 1 & 2) of the helicopter unit in Simulink. Use real time pacer (or built in simulation pacing for newer versions of matlab/Simulink) for real time simulation of the model.

Some useful information for simulator development:

a) The units of the pitch and the yaw angles are in radians and all calculations in Simulink should be performed in radians. However, it is easier to view the plots in degrees. So, when plotting θ and ψ, please use degrees i.e. convert radians to degrees only for plotting.
b) Set proper simulation setup. Sampling time should be 0.1 seconds. Use fixed time step and use Runge-Kutta 4 as solver. Set simulation stop time to be “inf” so that you can keep on running the simulator until you press stop yourself. This way you can change voltages and see how it affects the system while the simulation is still running in real time.
c) Initial values of the states can be $[-\frac{10\pi}{180}, \frac{\pi}{2}, 0, 0]$ for $\theta, \psi, \omega_\theta$ and ω_ψ respectively. Please initialize your Simulink integrator blocks appropriately.
d) Limit integrator output in integrator block in Simulink. The pitch angle cannot be less than -45 degrees and cannot be more than +45 degrees i.e. $-45^0 \leq \theta \leq 45^0$. Similarly, the yaw angle cannot be less than 0 degrees and cannot be more than 180 degrees i.e. $0^0 \leq \psi \leq 180^0$. Please limit the integrator

values by double clicking each integrator block for θ and ψ and putting the upper and lower values for them appropriately. But remember that integrator blocks in Simulink need units to be in radians.

e) Always start your simulator at the operating point/steady state. For this, you can set $V_{mp} = 1.333787314407593$ and $V_{my} = 0.948909586231924$. You can easily put these values in the constant block that are connected to your input voltage sliders/knobs. When you run the simulator with these values of voltages, you should be getting flat lines with θ at 10 degrees, ψ at 90 degrees and both angular velocities at 0. If you don't see this then something is wrong with your Simulink implementation.

f) After you have started your simulator at the steady state (operating point) then you can start playing around with different values for V_{mp} and V_{my} and see how it affects $\theta, \psi, \omega_\theta$ and ω_ψ. Just remember that you should change the voltages very slowly by small steps. Don't give sudden big changes to voltages. The helicopter is very sensitive to changes in voltages.
Important note: For normal operation (when you play around with the input voltages manually), the input voltage for the pitch motor does not need to be very high. For example it can be between 1 [V] to 1.5 [volts]. The normal operating input voltage for the yaw motor can be between 0.4 [V] to 1.3 [V]. It will be easier for you to see the actual dynamics of the system if you change the voltages gradually. Never use high voltage that can take your pitch angle to greater than 10 or 15 degrees in pitch.

Individual submission: Submit a lab journal with screenshots of your implementation and various plots. You should submit your lab journal in Canvas. Make sure to show all the details of your implementation (i.e. show what is inside your subsystem blocks).

Please look at the homepage under *Exercise 1* for links to see the implementation screenshots and plot/graph screenshots. These screenshots will give you an idea of how the implementation should look like, and how the results should look like.

Task 2:

Simulation of linear model of the lab helicopter unit.

The linear model of the lab helicopter can be obtained by linearizing the nonlinear model of Equations 1- 4. To linearize a model, we should at first select the point of linearization (also known as the operating points). For the helicopter unit, you can choose the operating points for the states to be as follows:

$$\theta_{op} = -\frac{10\pi}{180}, \qquad \psi_{op} = \frac{\pi}{2}, \qquad \omega_{\theta,op} = 0 \quad \text{and} \qquad \omega_{\psi,op} = 0$$

Here the subscript *'op'* means operating point. You cannot choose the operating points for the control inputs (voltages) yourself arbitrarily. They should be calculated based on the chosen operating points for the states. The corresponding values of the operating points for the control inputs can be calculated as,

$$V_{mp,op} = \frac{K_{yy} m_{heli}\, g \cos(\theta_{op})\, l_{cm}}{(K_{yy}K_{pp} - K_{yp}K_{py})}$$

$$V_{my,op} = \frac{K_{yp}V_{mp,op}}{K_{yy}}$$

Tips: You can set the state operating points, and then calculate control inputs operating points in a matlab initialization file which you can run once and load these variables in MATLAB workspace. These can then used/accessed from Simulink later on.

Now let us collect all the four states of the system into a vector and let us denote this vector as x.

$$x = \begin{bmatrix} \theta \\ \psi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} \quad \text{i.e. the state vector}$$

In the same way, let us collect all the four operating points of the states in a vector and denote it by x_{op}

$$x_{op} = \begin{bmatrix} \theta_{op} \\ \psi_{op} \\ \omega_{\theta,op} \\ \omega_{\psi,op} \end{bmatrix} \quad \text{i.e. the operating points of the states}$$

We can also collect the two corresponding operating points for the control inputs (two voltages) in a vector and denote it by u_{op}

$$u_{op} = \begin{bmatrix} V_{mp,op} \\ V_{my,op} \end{bmatrix}$$

The state deviation vector denoted by δx denotes the deviation of the actual states from its operating points. It is written by definition as,

$$\delta x = x - x_{op}$$

If we expand the state deviation vector it can be written as,

$$\delta x = \begin{bmatrix} \theta - \theta_{op} \\ \psi - \psi_{op} \\ \omega_\theta - \omega_{\theta,op} \\ \omega_\psi - \omega_{\psi,op} \end{bmatrix} \quad \text{i.e. the deviation of the states from its operating points}$$

Similarly, the control input deviation vector denoted by δu denotes the deviation of the actual control inputs (voltages) from its operation point. It can written by definition as,

$$\delta u = u - u_{op}$$

If we expand the control inputs deviation vector we get,

$$\delta u = \begin{bmatrix} V_{mp} - V_{mp,op} \\ V_{my} - V_{my,op} \end{bmatrix} \quad \text{i.e. the deviation of the control inputs from its operating point voltage}$$

The linear model of the helicopter is then given by the following state space model in the deviation form:

$$\boldsymbol{\frac{d}{dt}(\delta x) = A_c \delta x + B_c \delta u} \tag{5}$$

Equation (5) is a differential equation but it is in a vector form. We have the linear model in the deviation form because our chosen operating points were non-zero numbers (for two of the states).

The system matrices are,

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dfrac{m_{heli}\, g\, l_{cm} \sin(\theta_{op})}{J_{eq,p} + m_{heli} l_{cm}^2} & 0 & \dfrac{-B_p}{J_{eq,p} + m_{heli} l_{cm}^2} & 0 \\ 0 & 0 & 0 & \dfrac{-B_y}{\left(J_{eq,y} + m_{heli} \cos^2(\theta_{op})\, l_{cm}^2\right)} \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{K_{pp}}{J_{eq,p} + m_{heli} l_{cm}^2} & \dfrac{-K_{py}}{J_{eq,p} + m_{heli} l_{cm}^2} \\ \dfrac{K_{yp}}{J_{eq,y} + m_{heli} \cos^2(\theta_{op})\, l_{cm}^2} & \dfrac{-K_{yy}}{J_{eq,y} + m_{heli} \cos^2(\theta_{op})\, l_{cm}^2} \end{bmatrix}$$

Tips: You can calculate these system matrices in the initialization file, run it once at the beginning and load A_c and B_c it in the workspace to be used/accessed from simulink later on.

Individual submission:

Now, your task is to simulate the linear model together with the existing nonlinear model using Simulink. You should plot the results of nonlinear model and linear model in the common plot for same variable type and compare them. Comment on the response that you get from the linearized model and the nonlinear model.

Also try to go far away from the operating point by changing the voltages and observe the dynamics from the linearized model and from the nonlinear model. Are they different or same? Comment on it.

Some notes:

Very special care should be taken while simulating the linear state space model.

a) For simulating the linear model in Simulink we need the initial values of the states in the deviation form. This is because the linear model is in the deviation form. You can take the initial values of the deviation states to be zero for each of the states. These initial values can be set by double clicking the integrator block for the linear model and putting [0,0, 0, 0] in appropriate place.

b) If you look at the differential equation of the linear model of the system ($\frac{d\delta x}{dt} = \delta\dot{x} = A_c\delta x + B_c\delta u$), you can see that it contains δu, but not u on the right hand side. So the linear model should be excited with the deviation of the control inputs i.e. you should apply ($V_{mp} - V_{mp,op}$) and ($V_{my} - V_{my,op}$) as the two deviation of control inputs to your linear model. See the homepage under *LabWork 1* for a link to a picture which shows you how the implementation should look like. All you have to do is to subtract the operating point voltages from actual voltages.
But remember that the nonlinear model should still be excited with V_{mp} and V_{my}.

c) Although, the linear model (equation 5) has only single equation, you should remember that it is in compact form and it already contains all the four states of the system in the deviation form. You can use MATLAB function block in Simulink to write down the right hand side of linear equation 5. From the MATLAB function block you can return back (or pass out) the left hand side of equation 5 (e.g. as *d_delta_x*). After the integrator, you can then split up *delta_x* signal into four parts using demuxer. You will then get ($\delta\theta, \delta\psi, \delta\omega_\theta$ and $\delta\omega_\psi$), but not $\theta, \psi, \omega_\theta$ and ω_ψ as the solution of the differential equation of the linear model. However, if you want to calculate actual values of the states ($\theta, \psi, \omega_\theta$ and ω_ψ), then you should add the operating point values of the states to $\delta\theta, \delta\psi, \delta\omega_\theta$ and $\delta\omega_\psi$ i.e.

$$\theta = \delta\theta + \theta_{op}, \qquad \psi = \delta\psi + \psi_{op}, \qquad \omega_\theta = \delta\omega_\theta + \omega_{\theta,op}, \;\; \omega_\psi = \delta\omega_\psi + \omega_{\psi,op}$$

Please have a look at the homepage for a link which you can click to see a picture of how the implementation should look like. All you have to do is add the operating point values.

d) Don't forget to limit the output of the integrator for the linear model as well. For upper limit you can put $[\frac{45\pi}{180} - \theta_{op}\ ,\ \pi - \psi_{op}\ , inf, inf]$. Similarly for the lower limit you can put $[\frac{-45\pi}{180} - \theta_{op}\ ,\ 0 - \psi_{op}\ , -inf, -inf]$. These values for the upper and lower limits can be put by double clicking the integrator block for linear model and putting them at appropriate places.

Good luck!