

UNIVERSITY OF JORDAN
Mechatronics Engineering Department
Measurements & Control Lab. 0908448
Experiment no.5
Twin Rotor MIMO System

Objectives

In this experiment the student is expected to cover the following objectives:

1. To take an overview about the MIMO systems and how to simplify it to a SISO systems.
2. To investigate the importance of linearization a nonlinear system in order to design a suitable controller.
3. To verify the Twin Rotor models obtained using system identification.
4. To study the Twin Rotor system in both cases: open and closed loop.
5. To study the effect of applying different types of controllers on the performance of the system.
6. Tuning PID using matlab tools: root locus and pidtrack_demo.

Pre-Lab Assignments:

1. Read this lab handout completely.
2. Read the tutorials provided in **useful links** at the end of this experiment sheet and try to design a PID controller for a certain transfer function.

Introduction

The TRMS serves as a model of a helicopter. However some significant simplifications are made. First, is the fact that TRMS is attached to a tower and second of great importance that the helicopter position and velocity is controlled through the rotor velocity variation. In the real helicopter, the rotor velocity is less constant and the propulsion is varied through the rotor blades angle modification.

Nevertheless, the most important dynamic characteristics present in a helicopter are captured in the TRMS model. Like in a real helicopter, there is a significant cross coupling between two rotors; hence, if we activate the vertical-position rotor, the helicopter will also turn in the horizontal plane.

TRMS Set Description

As shown in Figure 1, the TRMS mechanical unit consists of two rotors (Main Rotor: enables vertical positioning and Tail Rotor: enables horizontal positioning) placed on a beam together with a counterbalance. The whole unit is attached to the tower allowing for safe helicopter control experiments.



Figure 1 TRMS mechanical unit

Apart from the mechanical unit, the electrical unit (placed under the tower) plays an important role in TRMS control. It allows for measured signals transfer to the PC and control signal application via an I/O card. The mechanical and electrical units provide a complete control system setup presented in Figure 2.

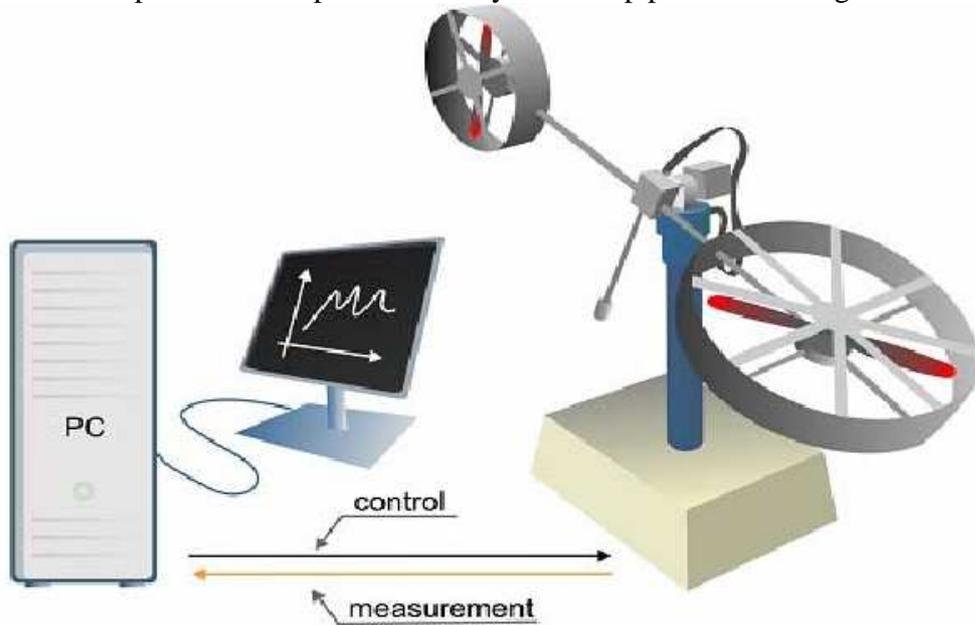


Figure 2 Control system schematic

The system consists of four main elements:

1. PC with a clocked control algorithm.
2. A/D and D/A converters – serving as an interface between the PC and external environment.
3. The controlled process.
4. Sensors.

TRMS Model

Every control project starts with plant modeling; so as much information as possible is given about the process itself. The mechanical-electrical model of the TRMS is presented in Figure 3.

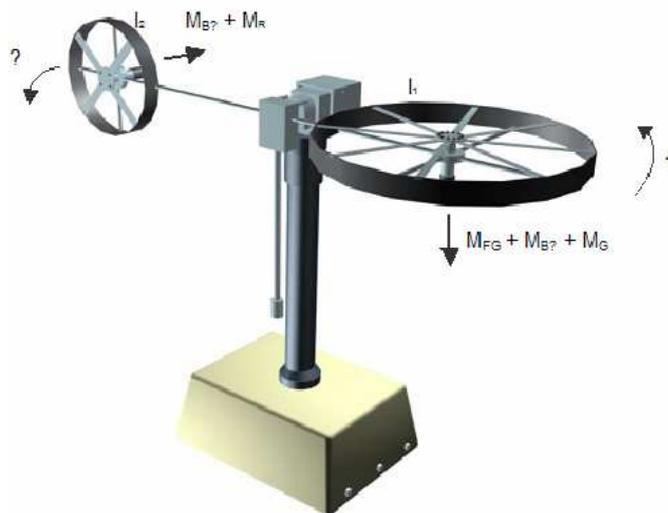


Figure 3 TRMS phenomenological model

Usually, phenomenological models are nonlinear, that means at least one of the states (i – rotor current, θ – position) is an argument of a nonlinear function. In order to present such a model as a transfer function (a form of linear plant dynamics representation used in control engineering), it has to be linearised.

The TRMS is a MIMO plant—multiple inputs multiple outputs. Figure 4 presents a simplified schematic of the TRMS.

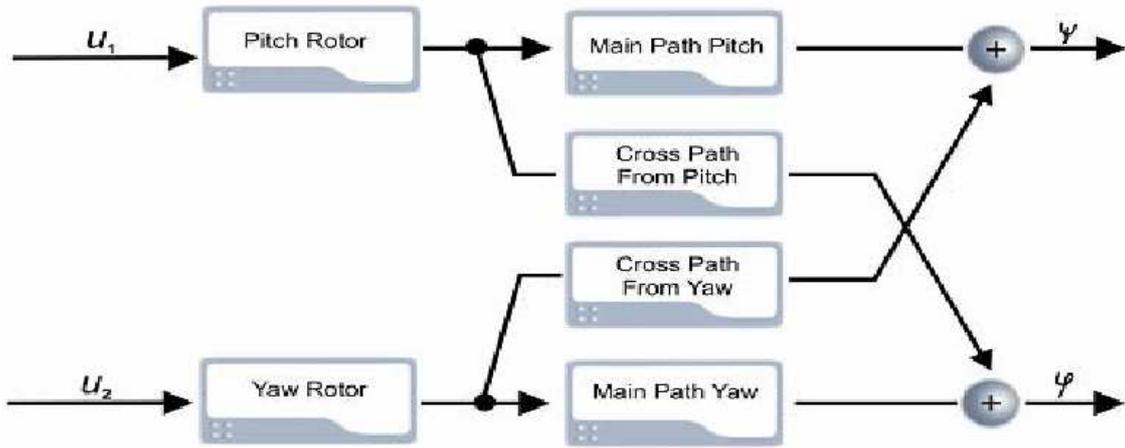


Figure 4 TRMS simplified system schematic

The TRMS is controlled with two inputs the $u1$ and $u2$. The dynamics cross couplings are one of the key features of the TRMS (Figure 4); if pitch rotor is activated, its movement has its effect on the vertical axis position. Furthermore, the yaw rotor activation will influence the position of the helicopter in the horizontal position.

As mentioned above, the TRMS is a nonlinear plant with significant cross couplings between the rotors. To keep the identification simple, the model can be treated as two linear rotor models ignoring couplings in-between as shown in Figure 5. When applying the principle of superposition, the effect of each input will be studied separately.

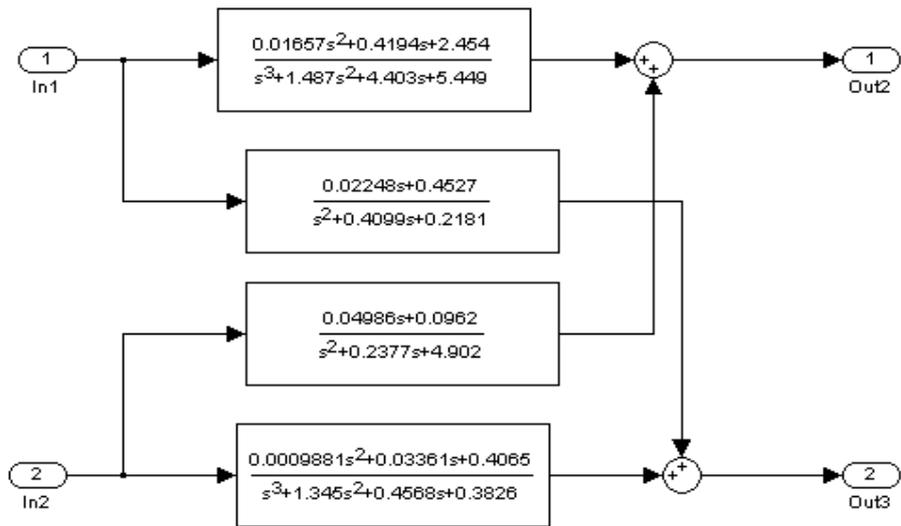


Figure 5 Identification model

Q. What is the importance of linearization a nonlinear system in control design?

Part 1: Verification of Models

In the previous section you have been provided with the TRMS models obtained by system identification; T.F1 represents pitch rotor (main) and T.F2 represents yaw rotor (tail), and your task is to test if these transfer functions are close to the actual models by examining the error that should be as small as possible.

1. Go to Start >>All Programs >>Feedback Instruments >>Twin Rotor MIMO Control System>> Simulink Models.
2. Matlab will run and the Simulink model menu will open as shown next.

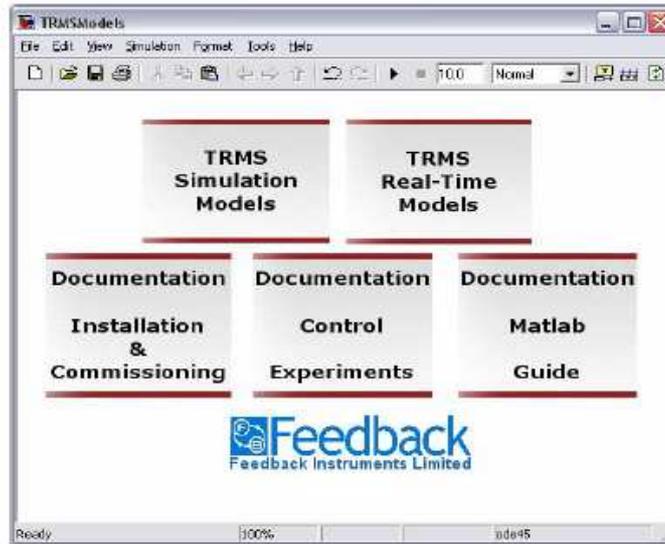


Figure 6 Simulink model menu

3. Open the simulink model file 'Model_Verification' on the desktop for validation of the Transfer Functions.

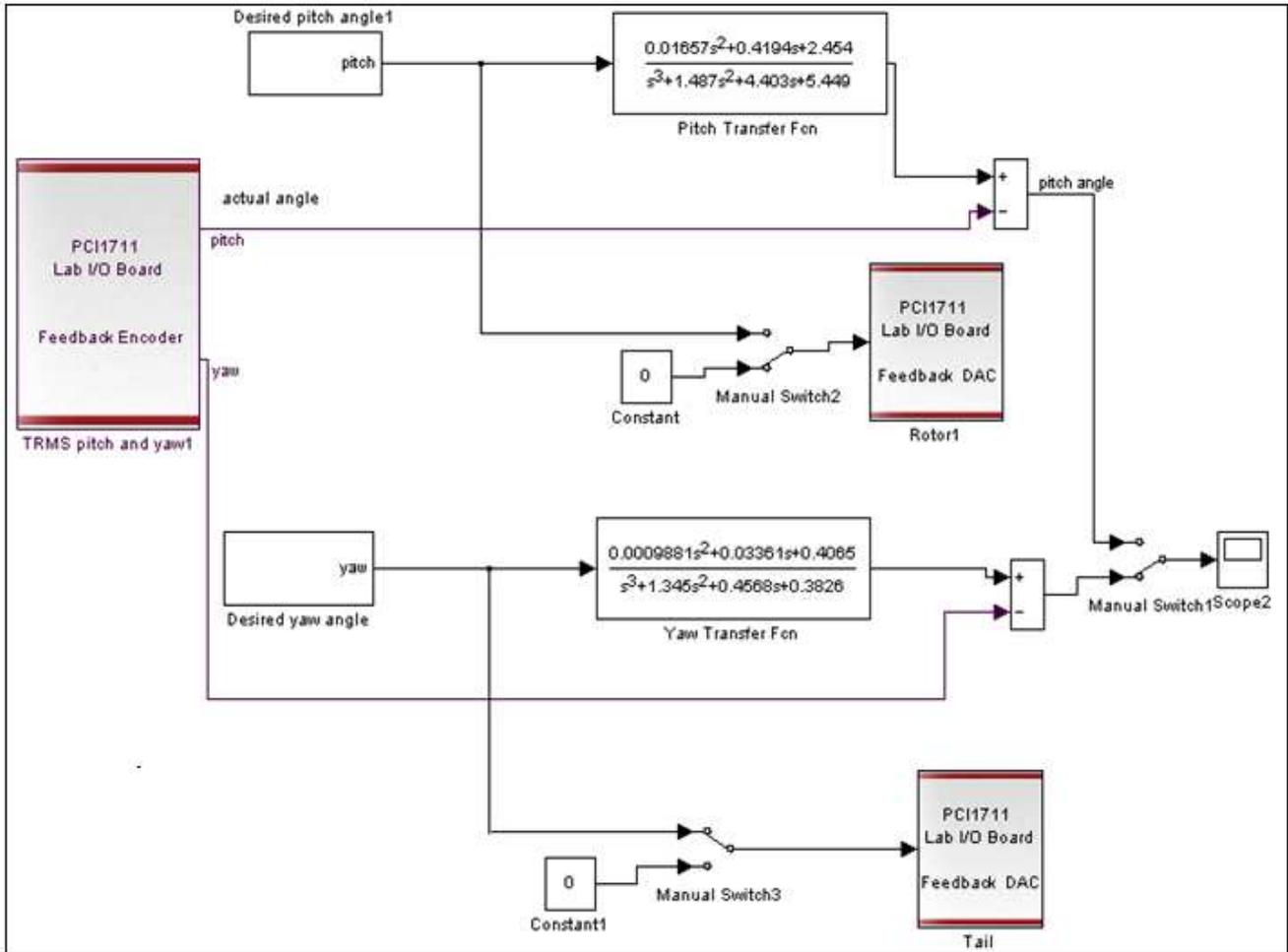


Figure 7 Model Verification

As shown in figure 7, pitch transfer function will be compared to the actual TF and the error will appear on the scope, similar test can be done for the yaw transfer function using the manual switches.

4. Check that manual switch 2 is on the pitch and manual switch 3 on 0 to obtain the response of the pitch rotor only.
5. Compile the model by pressing 'CTRL+B'. Wait for the compilation to finish. The successful completion statement should appear in the command window.
6. Press 'Connect to target' button. 
7. Turn the device power on then press the 'start simulation' button. 
8. The simulation will be carried out and the error signal will appear on the scope. Save this signal and stop simulation.
9. Justify the state of manual switches to see the response of yaw rotor and repeat the last four steps.

Q. Are the transfer functions valid or not? Why?

Q. Describe the error at the beginning of verification, can you tell why does it look the way it is?

Part 2: Open /Closed Loop Systems

In this part you will obtain the response of each rotor separately in both states; open and closed loop and notice the effect of inserting feedback on the response of the system.

1. Open the simulink model file 'PID_tuning' on the desktop. It will open the following model shown in figure 8.

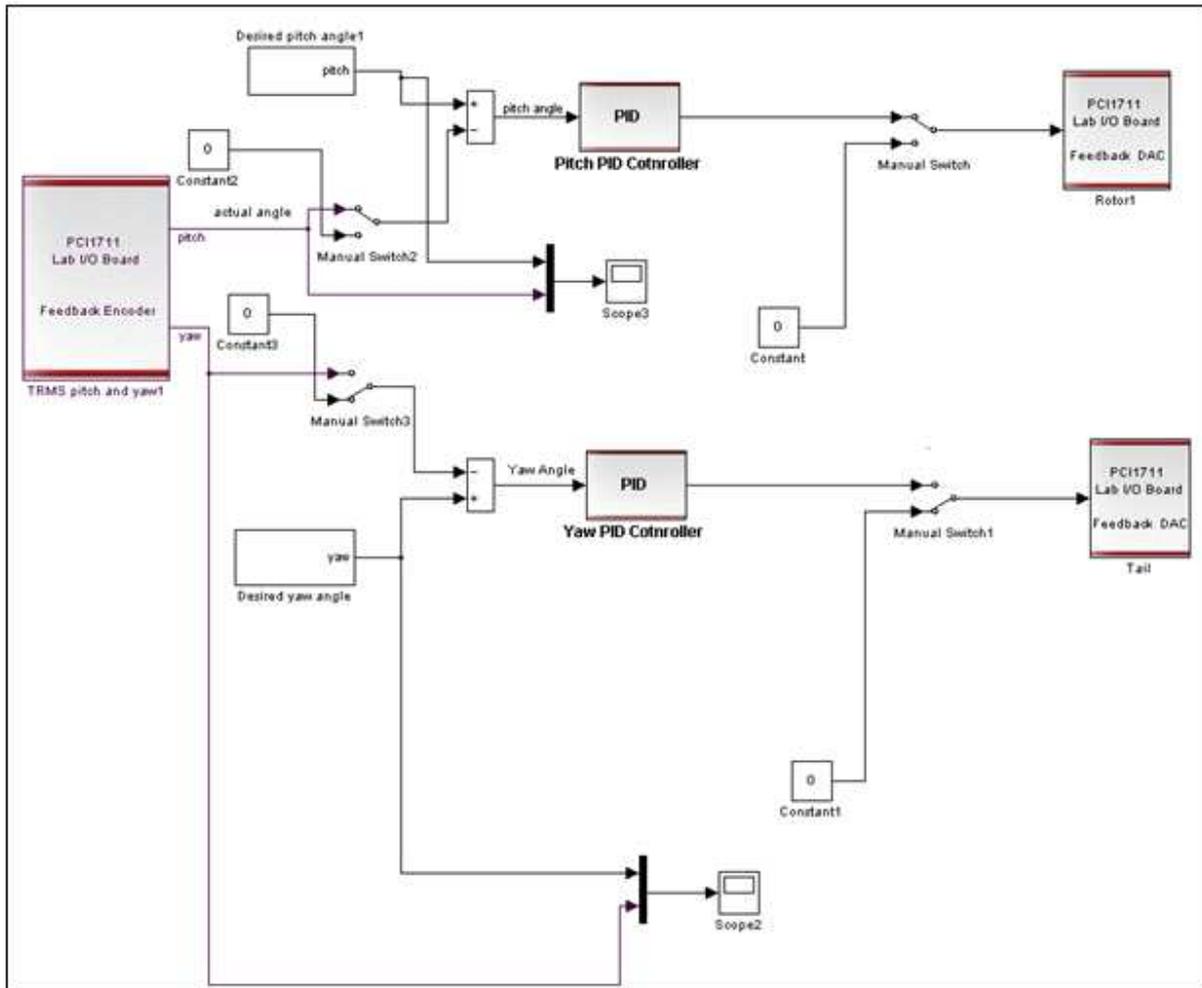


Figure 8 PID tuning simulink model

2. Check that the pitch PID controllers values are: $P=1, I=0$ and $D=0$ then set the manual switches to simulate the pitch rotor without feedback to obtain the open loop response, and check that the desired pitch angle is 1.
3. Press 'Connect to target' button. 
4. Turn the device power on then press the 'start simulation' button. 
5. The simulation will be carried out and the resulting signals will appear on the scope. Save this signal and stop simulation.
6. Adjust the manual switch to the yaw rotor and check that the desired yaw angle is 0.2 then repeat the last four steps.
7. Change the manual switch state to insert feedback and obtain the closed loop response for the pitch and yaw rotors separately using the same procedure applied in the open loop part.

Q. What is the difference between the open and closed loop responses?

Part 3: PID Controller

There are numerous control algorithms. The PID control algorithm is popular because of its simplicity. A general schematic of a simple closed loop control system is presented in Figure 9.

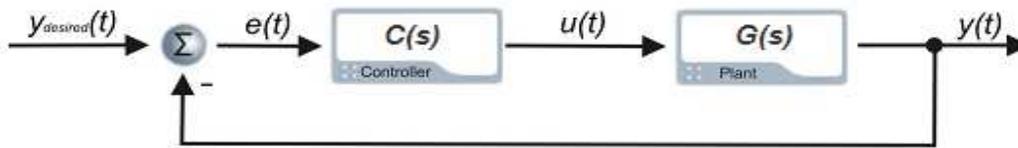


Figure 9 simple closed loop control system

The idea of control algorithms is to find a controller (transfer function, discrete transfer function, any nonlinear), which will fulfill our requirements (certain dynamic response, certain frequency damping, good response to the dynamic changes of the desired value etc.). Every controller's input is the $e(t)$ error signal. Sometimes disturbance signals are also measured. Depending on the present and past values of the error signal, the controller performs such an action (changes the $u(t)$ control signal) that forces $y(t)$ to be as close as possible to the $y_{desired}(t)$ value at all times.

PID controller consists of 3 blocks: **P**roportional, **I**ntegral and **D**erivative. The equation governing the parallel PID controller is as follows:

$$u(t) = P \cdot e(t) + I \cdot \int e(t) dt + D \cdot \frac{de(t)}{dt}$$

$$e(t) = y_{desired}(t) - y(t)$$

With the means of the Laplace transform, such a structure can be represented as a transfer function:

$$U(s) = \left(P + \frac{I}{s} + D \cdot s \right) \cdot E(s)$$

$$C(s) = \frac{U(s)}{E(s)} = \left(P + \frac{I}{s} + D \cdot s \right) = \frac{Ds^2 + Ps + I}{s}$$

Each of the PID controller blocks (P , I and D) plays an important role. However for some applications Integral or Derivative part has to be excluded to give satisfactory results. The **P**roportional block is mostly responsible for the speed of the system reaction. In some plants the oscillations may occur if the value of P is too large.

The **I**ntegral part is very important and insures 0 error value in steady state, which means that the output will be exactly what we want it to be. Nevertheless, the Integral action of the controller causes the system to respond slower to the desired value changes; so for systems where very fast reaction is very important, the integral action may be omitted. Certain nonlinearities will also cause problems for the integration action.

The **D**erivative part has been introduced to make the response faster. However it is very sensitive to noise amplitude increase and may cause the system to react very nervously. Thus very often it is omitted in the controller design. Derivative part of the output when filtered may reduce the nervous reaction but also slows the response of the controller down and sometimes undermines the sense of using the Derivative part at all. Proper filtering can help to reduce the high frequency noise without degrading the control system performance in the lower frequency band.

The task is to apply different types of controller from P to PID and change the controller values to see how the desired signal is tracked by the system output.

1. Open the simulink model file 'PID_tuning' used in part 2.
2. Set the PID values to be $P=0.5$, $I=0$ and $D=0$ to see the effect of proportional control only then set the manual switches to simulate the pitch rotor with feedback to obtain the closed loop response, and check that the desired pitch angle is 1.
3. Compile the model by pressing 'CTRL+B'. Wait for the compilation to finish. The successful completion statement should appear in the command window.
4. Press 'Connect to target' button. 
5. Turn the device power on then press the 'start simulation' button. 
6. The simulation will be carried out and the resulting signals will appear on the scope. Save this signal and stop simulation.
7. Change the manual switch to the yaw rotor and check that the desired yaw angle is 0.2 then repeat the last four steps.
8. Change the PID values to be $P=5$, $I=8$ and $D=10$ for the pitch rotor then set $P=1$, $I=2$ and $D=5$ in the state of the yaw rotor following the steps described above to obtain simulation for each rotor **separately** and note how the responses will be enhanced.
9. Save your results and discuss the effect of inserting PID controller on the performance of the system.

Part 4: PID Tuning

There are several PID tuning techniques. Very often the closed loop system roots are analyzed and set in the desired position with the means of P, I and D parts proper choice. Matlab delivers a root locus tool, which helps in such design. Another tool provided by matlab is pidtrack_demo that do automatic PID tuning.

You will use the last provided models to design a PID controller using root locus tool and pidtrack_demo with different desired pitch and yaw angles.

1. Refer to the links provided use both techniques (Root locus tool and pidtrack_demo) to obtain different values for the PID parameters changing the desired input values.
2. Check the obtained values by the instructor before applying it on the real system as done in part 3.
3. Save your results and add your comments.

Part 5: Dynamic Coupling

Check the system performance when both the main and tail rotors are active. **Do you notice anything wired?**

Useful Links

1. <http://www.mathworks.com/access/helpdesk/help/toolbox/control/getstart/f2-1043924.html>
2. http://www.mathworks.com/products/sl-design-optimization/demos.html?file=/products/demos/shipping/slido/pidtrack_demopad.html