MOBILE ROBOTS

Laboratory 3: Quadrotor attitude control

Assistants: Maja Varga and Giovanni Iacca

14.10.2014

1 INTRODUCTION

In this exercise you will become familiar with the mathematical model the quadrotor rotational motion and you will develop an attitude controller of the quadrotor. Attitude controller plays an important role in the control of flying platforms and it is a core of all flying control systems. We split a design process in multiple stages. We begin with simulation in Matlab and end with the testing of the controller on the real system. You will develop a simplified mathematical model of the quadrotor [1] and analyse the performance of the quadrotor's controllers in simulation. To allow autonomous operation you will develop attitude flight controller that controls propellers of the quadrotor based on the IMU sensor measurements. Every controller design process starts with the modelling of the system. This allows us to better understand the dynamical properties of the system and to properly choose a good controller that will stabilise the system. We will develop a simplified linear model of the quadrotor and we will design linear controllers to control quadrotors attitude. A quadrotor is highly complex and nonlinear dynamical system so we will neglect some physical effects acting on the platform to ease the mathematical analysis.

In this analysis, quadrotor is considered as a rigid body having 6 degrees of freedom with regarding to the earth fixed frame. The motion of the quadrotor is presented in two coordinate systems, earth fixed frame and body fixed frame. Orientation of the axis in both frames is shown in the Figure 1. We assume that quadrotor platform is rigid and symmetrical and centre of gravity (CoG) coincides with the origin of the body fixed frame. Motion of the quadrotor platform is controlled by four actuators, brushless motors that rotate propellers and create a lift force and rotational moments. Main physical effects that are acting on the quadrotor are aerodynamical effects created by propellers, friction, gyroscopic effects, gravity and inertial counter torques. We use Newton-Euler formalism to describe the rotational motion of

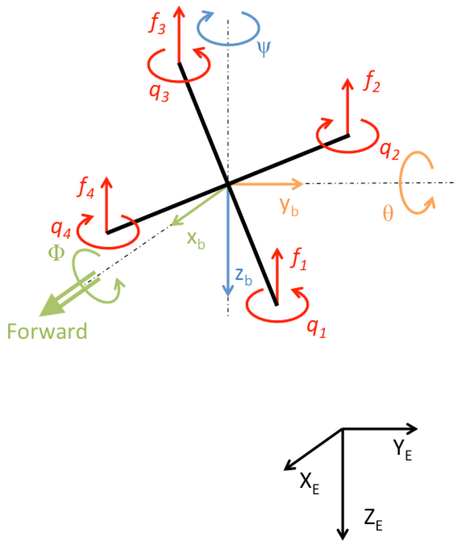


Figure 1.1: LE Quad, earth and body fixed frames, forces acting on the quadrotor.

quadrotor. To develop a rotational model we need to define which moments are acting on the rotation around x,y and z axis in body fixed frame. According to the law of conservation of the momentum we get:

$$\begin{aligned} I_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}(I_{yy}-I_{zz})+\tau_x \\ I_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_{zz}-I_{xx})+\tau_y \\ I_{zz}\ddot{\psi} &= \dot{\theta}\dot{\phi}(I_{xx}-I_{yy})+\tau_z \end{aligned} \tag{1.1}$$

where I_{xx}, I_{yy} and I_{zz} are moments of inertia around axis of the body fixed frame, ϕ, θ and ψ are Euler rotation angles, $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are roll, pitch and yaw rates and τ_x, τ_y and τ_z are torques around respected axes of the body fixed frame generated by the rotation of the propellers. The force created by propellers is proportional to the square of the rotational speed and is given by:

$$T_i = b\Omega_i^2 \tag{1.2}$$

where b is a thrust factor and Ω_i is an angular speed of the i-th propeller. Drag on the propellers is creating the moment around z-axis of the body frame and is defined as:

$$D_i = d\Omega_i^2 \tag{1.3}$$

where d is a drag factor.

Having a cross configuration of the quadrotor, torques around x,y and z body axis are defined as:

I_{xx}	I_{yy}	I_{zz}	d	b
0.0075	0.0075	0.013	3.13e-5	7.5e-7

Table 1.1: Parameters of the rotational model.

$$\begin{aligned} \tau_x &= l(-T_2 + T_4) \\ \tau_y &= l(T_1 - T_3) \\ \tau_z &= -D_1 + D_2 - D_3 + D_4 \end{aligned} \tag{1.4}$$

2 Exercise : [Analysis of the quadrotor dynamics]

1. From the rotational model 1.1 the dynamics of each orientation angle is determined by a second order differential equation. If roll, pitch, yaw rates and angles are states of the rotational dynamics $[\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T = [x_1, x_2, x_3, x_4, x_5, x_6]^T = \mathbf{x}$ and $\mathbf{u} = [\tau_x, \tau_y, \tau_z]^T$, rewrite the rotational dynamics in the state space in the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ (note that since the system is nonlinear, it can't be written in the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u}$).

   Answer: ____

2. Download the *matlab_files.zip* file from Moodle that contains the Matlab model. Uncompress it and open the model file *quadrotor_model.mdl.* Open the subsystem block *model_coupling.* Implement the rotational dynamics model developed in the previous task (2.1). Use the Simulink blocks in the bottom left corner of the schematics. If you need additional blocks, you can simply add them from Simulink library or duplicate already existing blocks in the model file.

3. From equation 1.1 you can observe that the rotation of the robot around body x axis is coupled with the rotation around y and z axis. This also holds for rotations around quadrotor's y and z body axis (the rotations are coupled with x,z and x,y rotations respectively). If we assume that motion of the quadrotor is near a hover flight, rotation speed around x,y and z axis are low and the cross coupling between states can be neglected. Rewrite the state equations taking into account this simplification.

   Answer: ____

4. Implement the model without coupling in Matlab in the subsystem *model_nocoupling*

5. Transfer simplified state equations derived in the previous task from time domain to the frequency (Laplace) domain using Laplace transformation. Determine transfer functions $\frac{\Omega_x(s)}{\tau_x(s)}$, $\frac{\Omega_y(s)}{\tau_y(s)}$ and $\frac{\Omega_z(s)}{\tau_z(s)}$ ($\Omega_x(s)$, $\Omega_y(s)$ and $\Omega_z(s)$ are roll pitch and yaw rate).

   Answer: ____

3 Exercise : [PID controller design and simulation in Matlab]

To control the attitude of the quadrotor we will use a linear PID controller. This method of control can be used for nonlinear systems, such as the quadrotor, under the assumption that the quadrotor is operating near the hovering state. For more aggressive manoeuvres, advanced control methods should be used.
PID controller consists of three terms, the proportional term, the integral term and the derivative term. Each of these terms has its own specific effects on the system we are trying to control. The proportional term increases the speed of response and it can reduce the error response to disturbances but it will still allow a non-zero steady state error. The integral term eliminates a steady state error but it deteriorates the dynamic response. The derivative term can damp the dynamic response. PID controller transfer function is given by:

$$D(s) = \frac{U(s)}{E(s)} = K_p(1 + T_I\frac{1}{s} + T_D s) \tag{3.1}$$

We are implementing PID controllers in a cascade structure shown in the Figure 3.1.

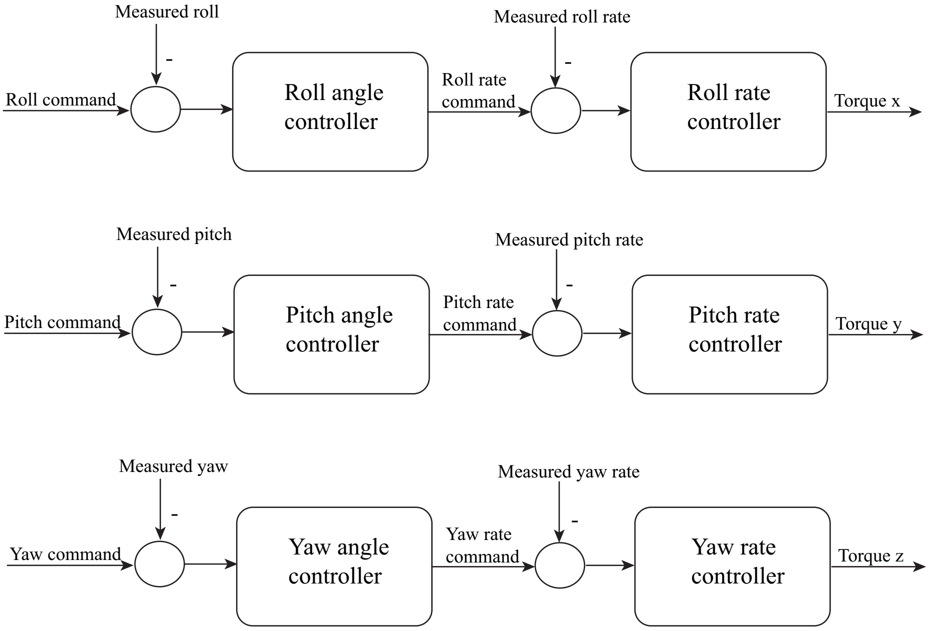


Figure 3.1: The structure of the cascade attitude controllers

An error between a desired and measured roll, pitch and yaw angle is an input to the angle controllers that are on the outer level of the cascade controller. The output of the angle controller, a desired roll, pitch and angle rate is subtracted from measured roll, pitch and yaw rate and fed to the inner level of the cascade controller. The output of the inner level controller, the torques around x,y and z body axis are sent to the quadrotor actuators.

1. The first controller you will implement in Simulink is a rate controller. Based on the transfer functions obtained in Exercise 1, decide which type of linear controller you

could use to obtain zero steady state error (for example, if the transfer function has an integrator, there's no need to have an integrator also in the controller). Do you need proportional, integral or derivative term? Do you need a combination of proportional, integral and derivative terms? Try to use the simplest control structure and design roll, pitch and yaw rate controller.

Answer: ____

2. Open the model file *quadrotor_model.mdl* and open controller subsystem. Here you will see already prepared subsystem blocks for your controllers (roll, pitch and yaw rate controller subsystems). When you open each rate controller, you will see in the bottom left corner already prepared Simulink library blocks needed to implement the controller. If you need an additional blocks, you can drag and drop them directly from Simulink library.

3. After you implemented rate controllers in Simulink, it is time to develop the controller to stabilise roll, pitch and yaw angle. Follow the same instructions as for the rate controllers.

4. Experimentally tune the gains of the controllers. Start with the gains equal to 1 and increase/decrease gains to obtain a stable response.

5. Now when you developed rate and angle controllers, it is time to run the simulation to see how the controllers perform and to examine whether the proposed controller structure will stabilise the quadrotor. The commanded roll, pitch and yaw can be modified by editing the Matlab function *msf_command.m*

6. In Matlab command window you can run script *run_test.m* by typing the script name in the command prompt. This will run both the simulation and visualisation script. If you open subsystem Visualize you will find *Scope* Simulink blocks. If you double-click on each block, you can observe the plots of each quadrotor state. Change the command to higher and lower roll, pitch and yaw angles and observe how much time does it take for the quadrotor to reach the commanded values. Change the gains of the controllers and observe how your response changes.

4 Exercise : [Quadrotor tests]

After you developed and tested your attitude controllers in Matlab, now it is time to examine the structure of the controller implemented on the real quadrotor and test these controllers on the real setup. Before you proceed with the testing, **READ THE SAFETY PROCEDURE**.

- Before operating the quad, you must wear protective gloves and goggles.
- Make sure that the propellers of the quad are protected with the protective frame made with foam.
- Make sure that the quad is firmly connected to the gimbal joint.

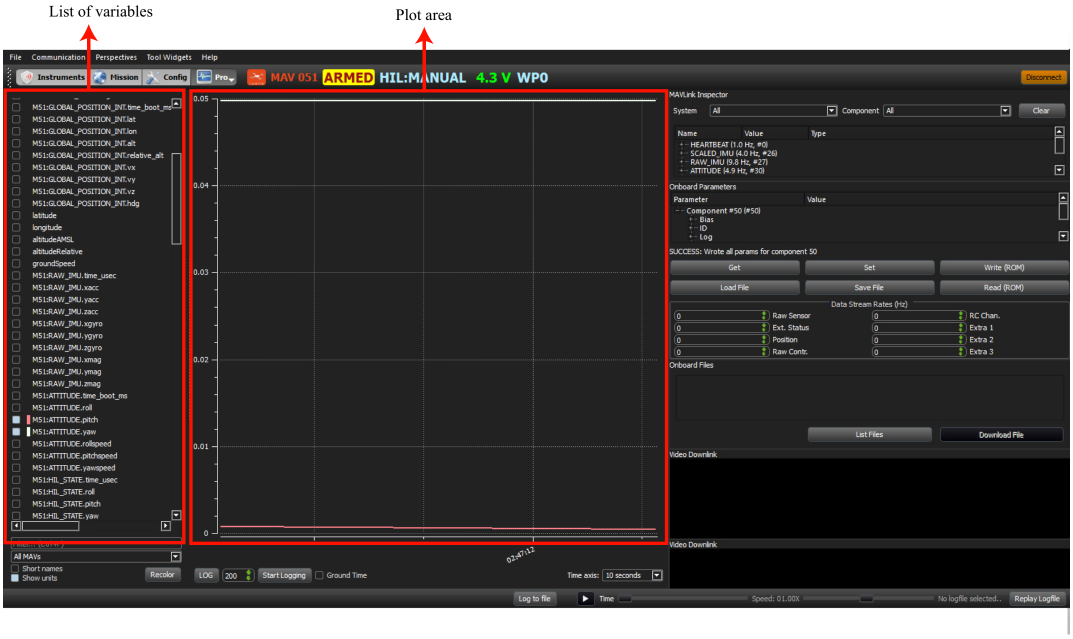
- Check whether gimbal joint is firmly connected to the metal pole.
- Check whether the metal pole is firmly connected to the wooden panel.
- Check the orientation of the propellers. Propellers with the mark 7x6R should be mounted on the front left and back right motor shaft. Propellers with the mark 7x6 should be mounted on the front right and back left motor shaft.
- Make sure that every propeller is mounted on the motor shaft and attached with two rubber bands.
- Only students wearing protective gloves can plug in the battery.
- When quad is in the armed state, students should keep the distance of at least 1m from the quad.
- Thrust value should never be set above 25% of the maximum thrust.
- Only one student at the time can operate the joystick.
- Input only the PID parameters provided in the documentation.
- In case of any problem, switch off the motors immediately by reducing the thrust on the joystick.
- For any doubt or issue ask an assistant.

1. Download the quadrotor firmware code from Moodle (*quadrotor_control.zip*). Uncompress it and open controllers source files *pid_control.c* and *stabilisation_copter.c*. Which type of PID controllers are used for rate and angle controllers?
2. Switch on the quadrotor by plugging in the battery at the bottom of the platform. Make sure that the platform is correctly attached to the metal pole and wooden plank. Observe that you can only change pitch and yaw angle of the quadrotor while roll angle is blocked.
3. Plug in the Xbee antenna in your laptop control station. Plug in the joystick. Open qgroundcontrol and connect the control station and the quadrotor. Now you can monitor the output from the quadrotor sensors and the operating state of the quadrotor.
4. Make sure that the quadrotor is in DISARMED MANUAL state. You can check this at the top of the qgroundcontrol screen. Go to the Instruments mode and check whether the quadrotor is properly leveled.
5. Open File-Joystick configuration and set up the joystick. Tick the Enable joystick and set joystick axes to 0 - roll, 1-pitch, 2-throttle (tick inverted) and 3-yaw. Move a bit a throttle and set it to -100. Press ok.

6. Set the thrust of the quadrotor to the minimal value. Go to Instruments tab. Activate control widget in the qgroundcontrol (go to Tool widgets - Control). You will see the button that can be used to put the quadrotor in armed state. Press this button and check if the quadrotor changed the state to ARMED MANUAL at the top of the qgroundcontrol screen. Be careful, propeller will start to rotate.

7. Leave the thrust on the low value (20%), send a maximum pitch command to the quadrotor and observe what is happening with the platform. Set the pitch command to zero.

8. Open the display to plot the data. Go to Pro - Plot bar. On the left side of the screen you will see a list of all variables that can be monitored in qgroundcontrol. To test whether the controllers are tuned properly we should monitor inputs and outputs to all rate and angle controllers.



□ M51:GLOBAL_POSITION_INT.time_boot_ms
□ M51:GLOBAL_POSITION_INT.lat
□ M51:GLOBAL_POSITION_INT.lon
□ M51:GLOBAL_POSITION_INT.alt
□ M51:GLOBAL_POSITION_INT.relative_alt
□ M51:GLOBAL_POSITION_INT.vx
□ M51:GLOBAL_POSITION_INT.vy
□ M51:GLOBAL_POSITION_INT.vz
□ M51:GLOBAL_POSITION_INT.hdg
□ latitude
□ longitude
□ altitudeAMSL
□ altitudeRelative
□ groundSpeed
□ M51:RAW_IMU.time_usec
□ M51:RAW_IMU.xacc
□ M51:RAW_IMU.yacc
□ M51:RAW_IMU.zacc
□ M51:RAW_IMU.xgyro
□ M51:RAW_IMU.ygyro
□ M51:RAW_IMU.zgyro
□ M51:RAW_IMU.xmag
□ M51:RAW_IMU.ymag
□ M51:RAW_IMU.zmag
□ M51:ATTITUDE.time_boot_ms
□ M51:ATTITUDE.roll
☑ M51:ATTITUDE.pitch
☑ M51:ATTITUDE.yaw
□ M51:ATTITUDE.rollspeed
□ M51:ATTITUDE.pitchspeed
□ M51:ATTITUDE.yawspeed
□ M51:HIL_STATE.time_usec
□ M51:HIL_STATE.roll
□ M51:HIL_STATE.pitch
□ M51:HIL_STATE.yaw

HEARTBEAT (1.0 Hz, #0)
SCALED_IMU (4.0 Hz, #26)
RAW_IMU (9.8 Hz, #27)
ATTITUDE (4.9 Hz, #30)

Component #50 (#50)
Bias
ID
Log

SUCCESS: Wrote all params for component 50

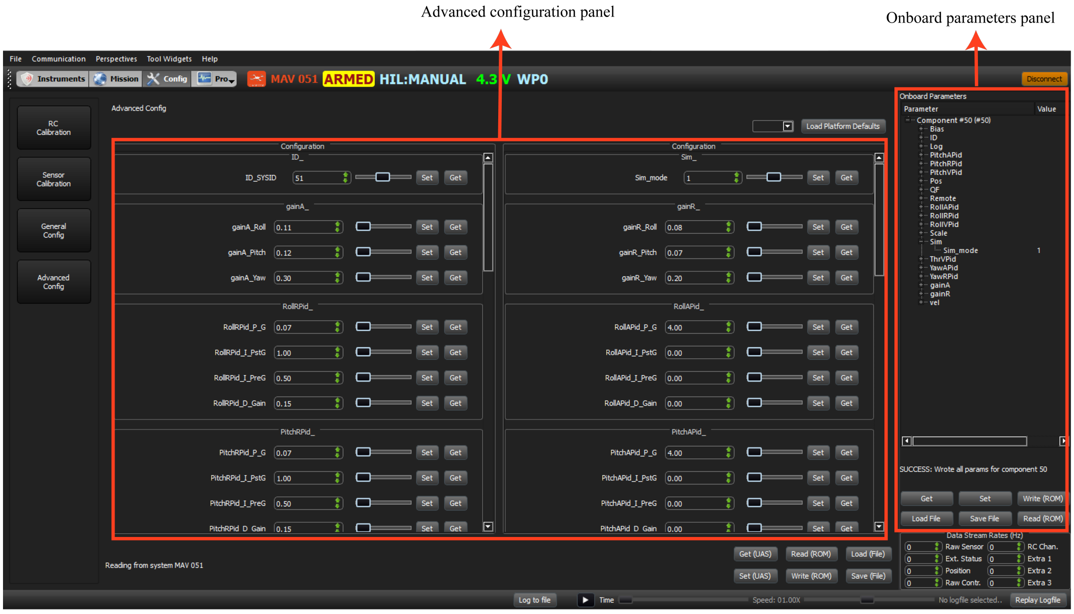
Figure 4.1: The structure of the cascade attitude controllers

9. To observe the transient response of the pitch controller, set the commanded pitch to the maximum value and hold it for 10 seconds. Observe the pitch plot on the screen. Set the pitch command to zero. How much time does it take for the pitch controller to stabilise for both cases?

10. Increase the thrust value to approximately 50% of the maximum thrust. Repeat the previous experiment by setting pitch to the maximum value and to the zero value. How much time does it take for the pitch controller to stabilise when you increased the thrust value?

11. Set the thrust command on the low value. To observe the transient response of the yaw controller, set the yaw rate command to the maximum value and hold it for 10 seconds. Observe the yaw rate and yaw angle plot on the plot screen. Set the yaw command to zero. How much time does it take for the yaw rate controller to stabilise for both cases?

12. Disarm the quadrotor. Now you will change the parameters of the pitch rate and angle controller and observe the effects of these changes. Open Config tab. You will see two panels, Advanced configuration panel and Onboard parameters panel (Figure 4.2). You can change the value of the parameters on both panels. Change *RollRPID_P_G* to 0.05 and *RollAPID_P_G* to 3.0. Press button Set to write the send the modified values to the quadrotor controller. Arm the quadrotor and repeat the procedure from exercise 4.1 to test the response of the pitch attitude and rate controllers.



Advanced configuration panel

Onboard parameters panel

Advanced Config

Configuration

ID_

ID_SYSID 51

gainA_

gainA_Roll 0.11

gainA_Pitch 0.12

gainA_Yaw 0.30

RollRPid_

RollRPid_P_G 0.07

RollRPid_I_PstG 1.00

RollRPid_I_PreG 0.50

RollRPid_D_Gain 0.15

PitchRPid_

PitchRPid_P_G 0.07

PitchRPid_I_PstG 1.00

PitchRPid_I_PreG 0.50

PitchRPid_D_Gain 0.15

Configuration

Sim_

Sim_mode 1

gainR_

gainR_Roll 0.08

gainR_Pitch 0.07

gainR_Yaw 0.20

RollAPid_

RollAPid_P_G 4.00

RollAPid_I_PstG 0.00

RollAPid_I_PreG 0.00

RollAPid_D_Gain 0.00

PitchAPid_

PitchAPid_P_G 4.00

PitchAPid_I_PstG 0.00

PitchAPid_I_PreG 0.00

PitchAPid_D_Gain 0.00

Reading from system MAV 051

Onboard Parameters

Parameter	Value
Component #50 (#50)	
Bias	
ID	
Log	
PitchAPid	
PitchRPid	
PitchVPid	
Pos	
QF	
Remote	
RollAPid	
RollRPid	
RollVPid	
Scale	
Sim	
Sim_mode	1
ThrVPid	
YawAPid	
YawRPid	
gainA	
gainR	
vel	

SUCCESS: Wrote all params for component 50

Data Stream Rates (Hz)

0	Raw Sensor	0	RC Chan.
0	Ext. Status	0	Extra 1
0	Position	0	Extra 2
0	Raw Contr.	0	Extra 3

Speed: 01.00X

No logfile selected..

Figure 4.2: The structure of the cascade attitude controllers

13. Disarm the quadrotor. Set *RollRPID_P_G* to 0.08 and *RollAPID_P_G* to 5.0. Repeat the procedure from exercise 4.8 to test the response of the pitch attitude and rate controllers.

14. Disarm the quadrotor.Set *RollRPID_P_G* to 0.07, *RollAPID_P_G* to 4.0, *YawRPID_P_G* to 0.05, and *YawAPID_P_G* to 3.0. Repeat the procedure from exercise 4.10 to test the response of the yaw attitude and rate controllers.

REFERENCES

[1] Samir Bouabdallah, *Design and control of quadrotors with application to autonomous flying*. PhD Thesis, École Polytechnique federale de Lausanne, 2007.