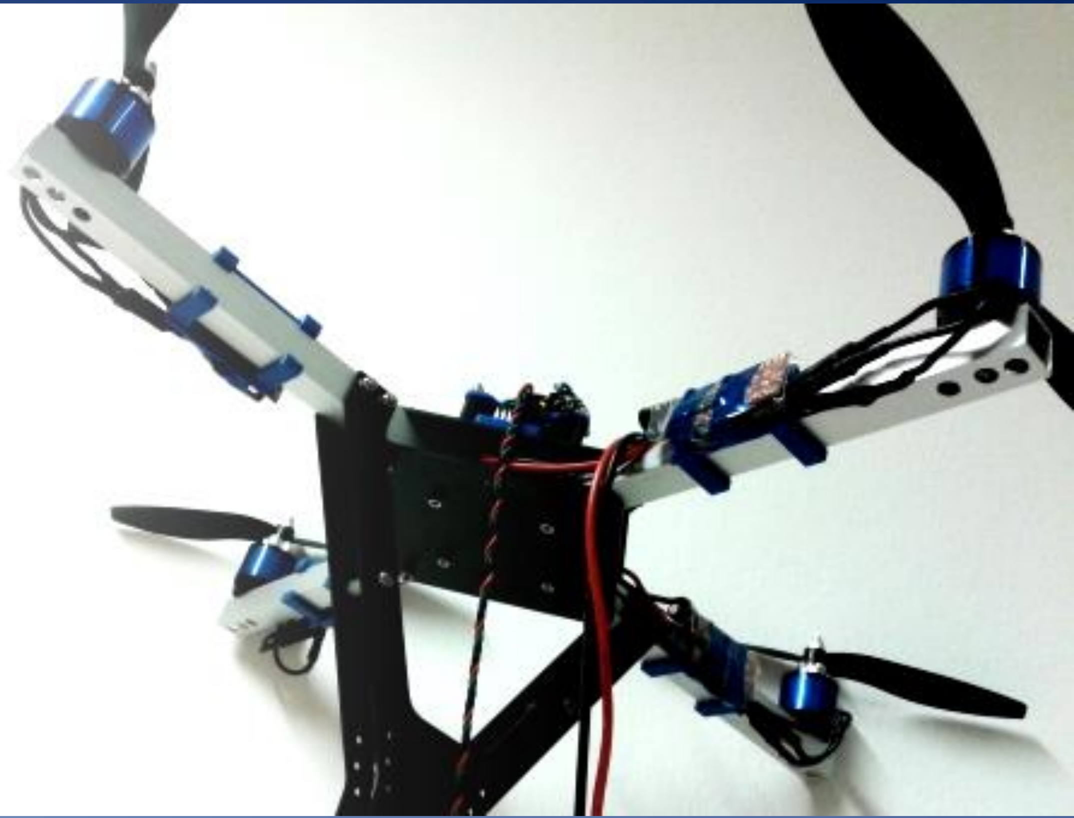
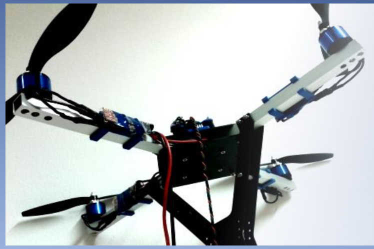
Yaw Control



Content

Time scope: ca. 1-4h

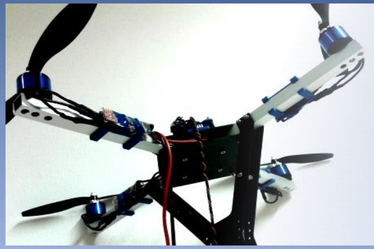
- Yaw Control
- Safety Instructions
- Control Theory
- Superposition
- Exercises & Hints



Yaw Control

<u>Definition Yaw Control:</u>
The torque of each motor makes the quadrotor turn around its z-axis. This is why two motors turn clockwise and the other two motors turn counter clockwise. To avoid interferences between the yaw control and the attitude control, the two opposing motors always turn in the same direction.

The yaw controller enables the system to perform controlled turns around the z-axis (yaw) and compensate disturbances over this degree of freedom. Compared to the attitude control the yaw control is much easier since the system does not have to work against gravity.



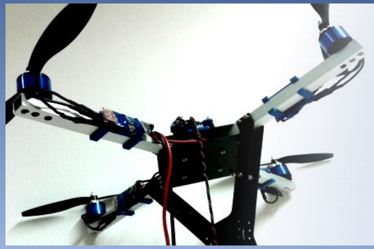
Safety instructions

Safety instructions:

- Check the system and retighten all parts (screws/propellers/motors/arms/joints/bolts/nuts) if necessary before every start!
- Only connect and start the motors under the supervision of your teacher!
- Keep a safety distance of at least 50cm to any propeller.
- Take care for your neighbor and yourself! Be ready to turn the system off immediately in case of emergency!

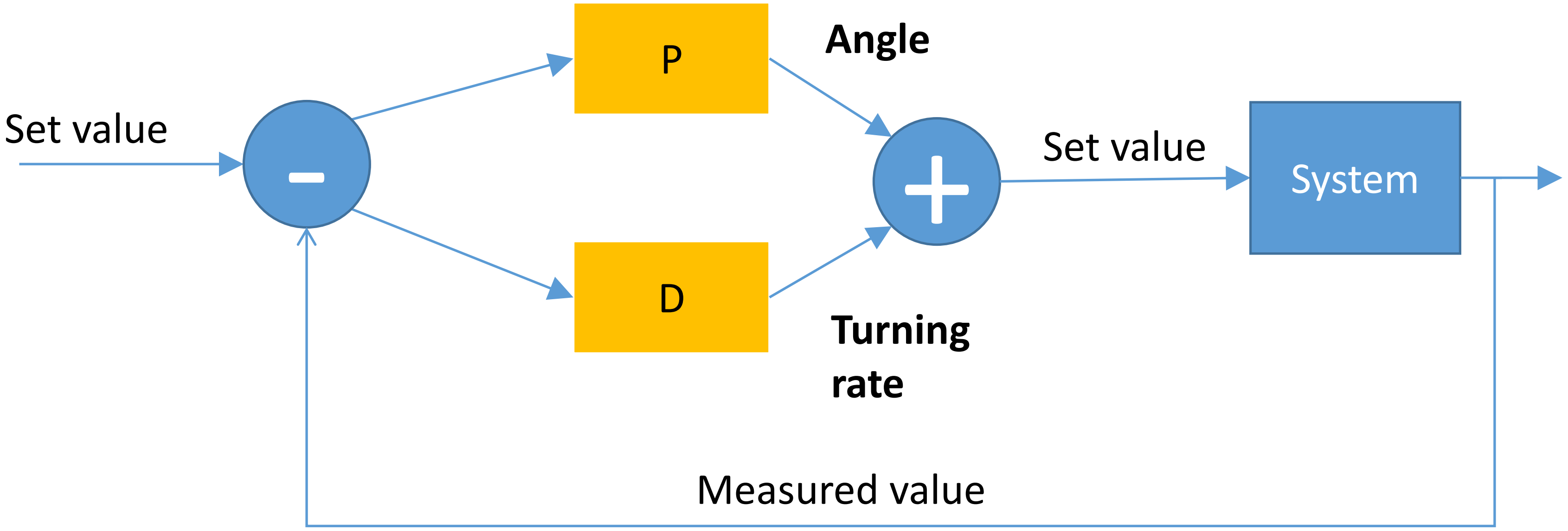
Emergency Stop:

- Unplug / Shut down Power
- Unplug TWI Cable
- Shut down MCU

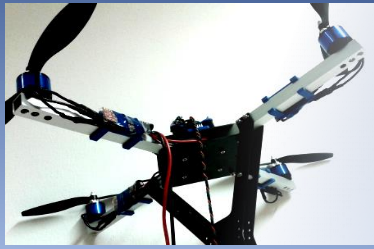


Control Theory

The system can be controlled with an PD controller using a gyroscope. The control variable is the angle of inclination.



Problem: 2 Motors / set values, 1 value of measurement (angle)
Solution: Simplification by controlling the difference of the motor set values



Superposition

Superposition:
Superposition simply means adding up the controller outputs. Prerequisite for a working superposition is that the controllers are independent of each other.

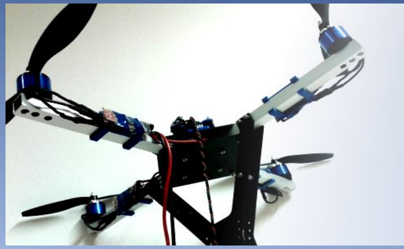
Aim of this exercise is to control the quadrotor. Each controller is developed independently. To keep the quadrotor in the air, you need to control roll, pitch and yaw. This is implemented by 2 (3) controllers:

- Attitude-Controller (x2 = Roll + Pitch)
- Yaw-Controller

After all the two (three) controllers have been set up, they are super positioned. For two dimensions (1 yaw controller, 1 attitude controller) you get:

M1 = (**unsigned char)(START_GAS + roll + yaw);**
M2 = (**unsigned char)(START_GAS - yaw);**
M3 = (**unsigned char)(START_GAS - roll + yaw);**
M4 = (**unsigned char)(START_GAS - yaw);**

Attention:
Superposition is a simplification. In reality the controllers interfere with each other. Therefore, you need to find a solution.



Exercises

Required hardware:

- EVK1100
- Micro USB cable for power and flashing
- QCS in 1DOF Yaw Control Mode (later in 2 DOF)

Required software:

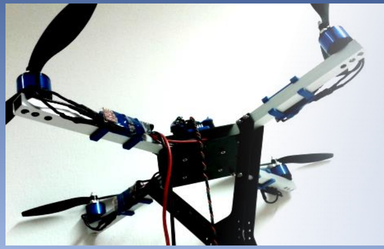
- AVR Studio 32 (with Tool Chain and FLIP Driver)
- EMQ Framework (Code)
- Documents:
    - EMQ_Framework.pdf

Exercises

Exercise 1:
Implement a PD Controller as a yaw controller which controls the quadrotors rotation around the z-axis. Find an optimized parameterization empirically by trial and error. The controller should be able to compensate small disturbances.

Hint:
Use a screw to fix the QCS in its horizontal position so that you don’t have to care about the attitude control at the beginning. Remove this screw for exercise 2!

Exercise 2:
Fusion the controllers for yaw control and attitude control. This is where you should notice, that the controllers interfere with each other. Implement some more functionality so that the overall behavior does not get worse compared to operating the QCS with one controller only. In particular strong disturbances changing the yaw position of the system should be compensated without the system tipping over. The system should still be able to control the yaw movement quickly.



Exercises

<u>Hints for control parameter dimensioning:</u>

Stay within the following parameter ranges as a guideline (without sample time or scaling):

Sample-Time:	10ms		
Bias-Samples	1000 to 2000		
P	2	| 4	
D	0.4	| 0.8	(Stops P)
I (optional)	0.01	| 0.02	(To solve stationary control deviation)

Approach for controller parameterization:

Start with D = 0 and I = 0 and increase P until the system starts to oscillate. If P is to high, the system is instable. If P is to small, the system will never reach its set point. Continue by increasing the D parameter to counteract an overshooting. D should work against the current movement of the system.

Idea:

P defines, how strong the controller reacts to errors. D slows down the system movement (Counter Controlling).