University of Stuttgart
Institute of Flight Mechanics and Controls

Altitude Controller Design for a Quadcopter

using PD Control

R. Erdem Uysal

Flight Robotics and Perception Group

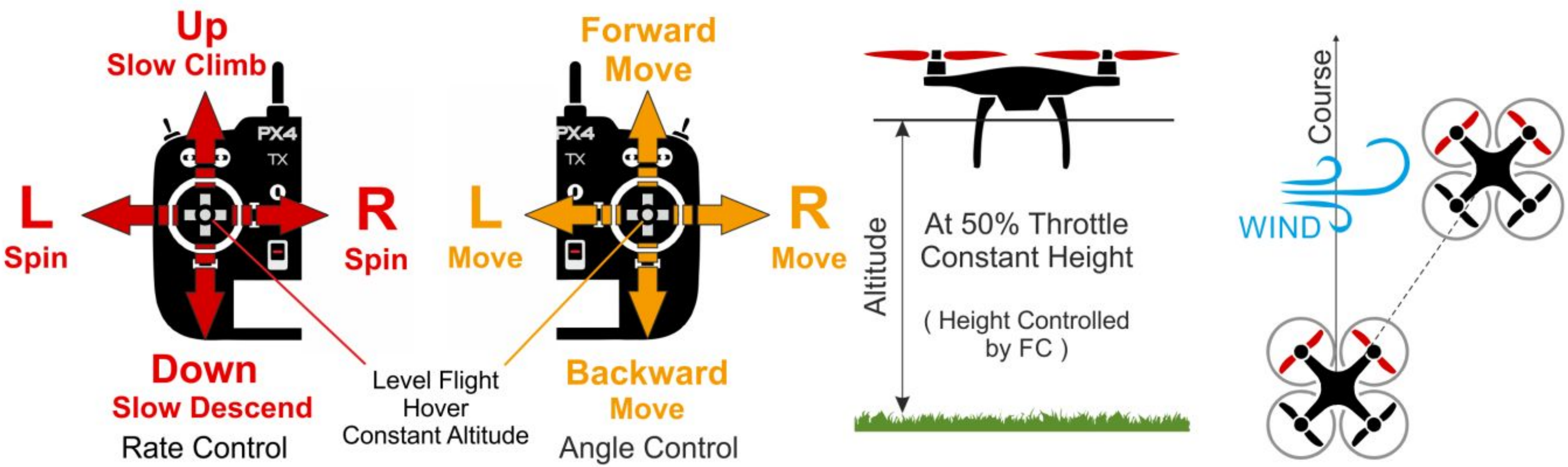
PERCEIVING SYSTEMS
MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

Agenda

- Tasks of an altitude controller
- Possible sensors to measure height
- Short explanation of the purpose of the Laplace transformation
- Controller structure chosen for this task
- General stability analysis
- Implementation in the Simulink model
- Presentation of results

Task of an Altitude Controller

The task of an altitude controller is to regulate quadcopter's vertical movement automatically based on sensor inputs available on the flight controller board.



Task of an Altitude Controller

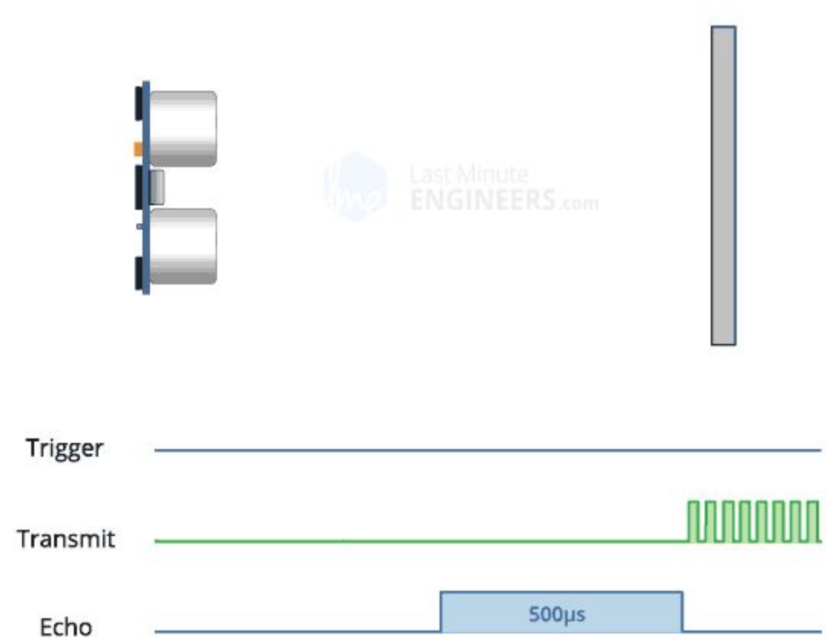
Example of Hovering

Possible Sensors to Measure the Height

Ultrasonic Sensors & Sonars

- Vertical distance calculation made based on transmitted and received sound pulse's travelling time on air.

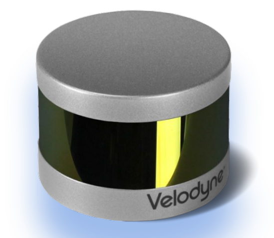
HC-SR04 Ultrasonic Sensor



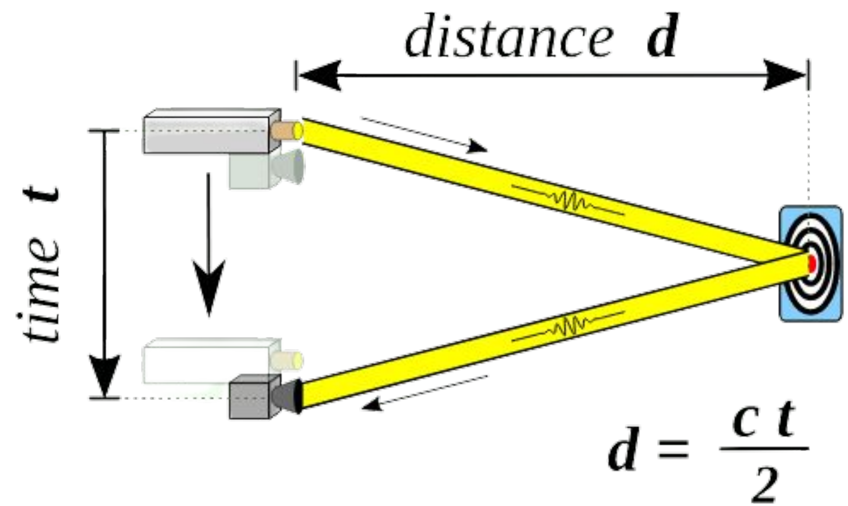
Possible Sensors to Measure the Height

Lasers, LiDARs & Time of Flight Sensors

- Vertical distance calculation made based on emitted light waves' travelling time on air.



Velodyne 360° Lidar

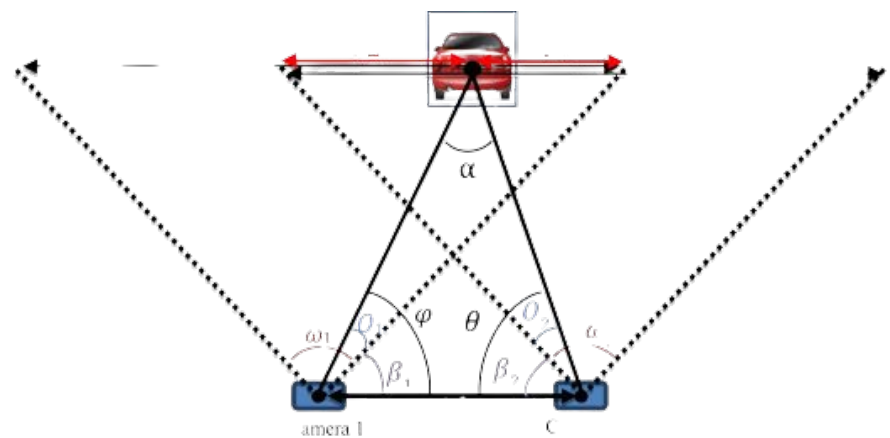


Possible Sensors to Measure the Height

Stereo/Depth Cameras

- Vertical distance calculation made based on disparity of the objects between the camera lenses.

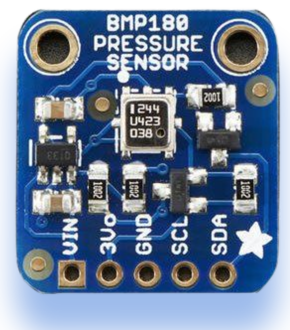
Intel d435i Stereo Camera



Possible Sensors to Measure the Height

Barometric Sensors

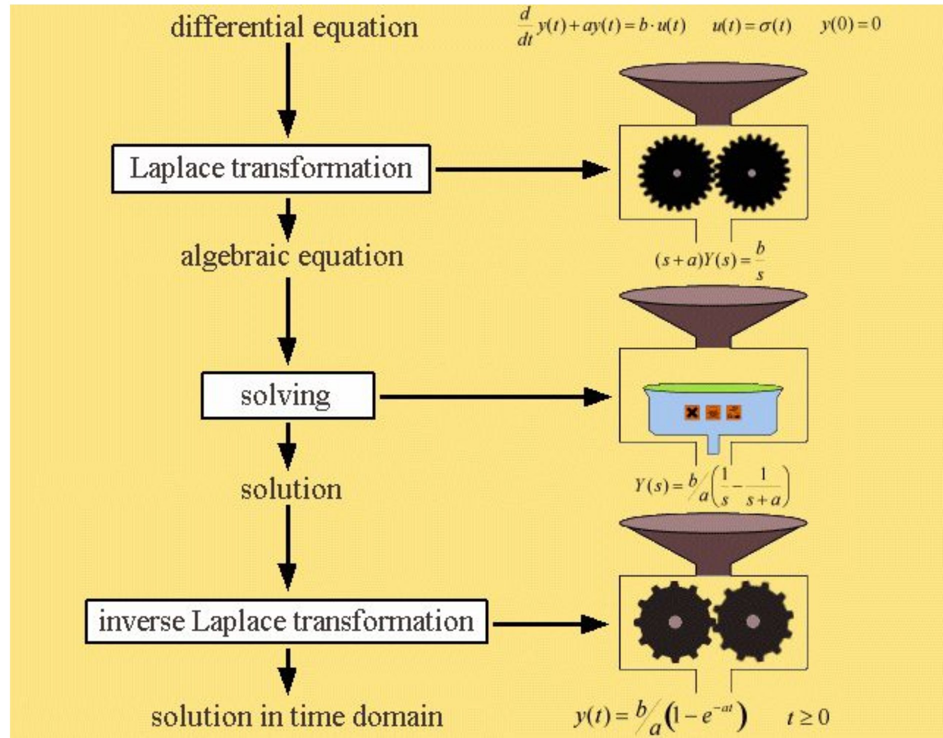
- Vertical distance calculations made based on change in measured barometric value.



Adafruit BMP180 Pressure Sensor

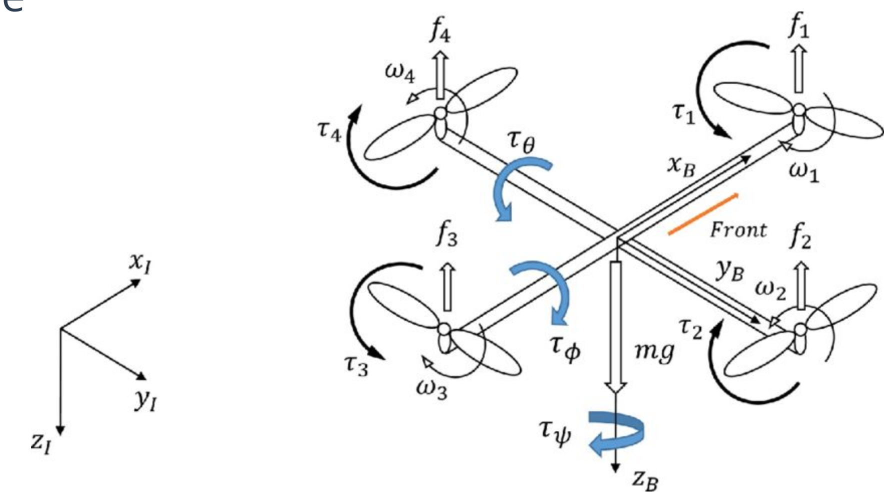
Laplace Transform
Purpose of the Laplace Transform

- The purpose of the Laplace Transform is **to transform ordinary differential equations (ODEs) into algebraic equations**, which makes it easier to solve ODEs.



Overview of the System

- Plant model → $\mathbf{Z = m \cdot a_Z}$
- Measured data to reform input of the controller → h, height
- Trim condition → Hovering condition, $\mathbf{v_Z=0}$ **m/s**, $\mathbf{a_Z=0}$ **m/s²**
- Height to vertical speed via derivative



Structure of Controller

PD Controller

P Controller:

- \+ Easy to implement
- \+ Minimize fluctuating
- \- Long settling time
- \- Steady state error

PD Controller:

- \+ Easy to stabilize
- \+ Faster response than just P controller
- \- Can amplify high frequency noise

$$err = h_{ref} - h$$

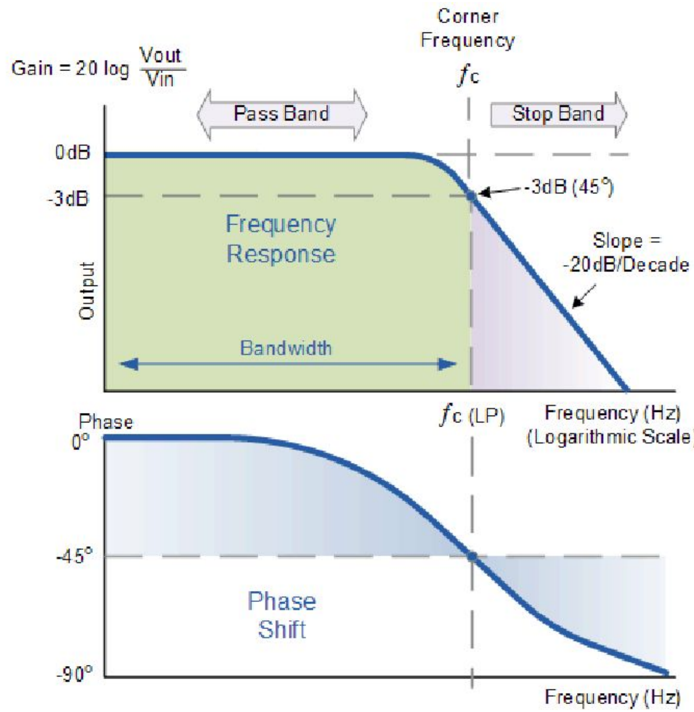
$$C_p = K_p \cdot err$$

$$C_d = K_d \cdot \frac{\mathrm{d}err(t)}{\mathrm{d}t}$$

Structure of Controller

Low Pass Filter

A low-pass filter **(LPF)** is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.

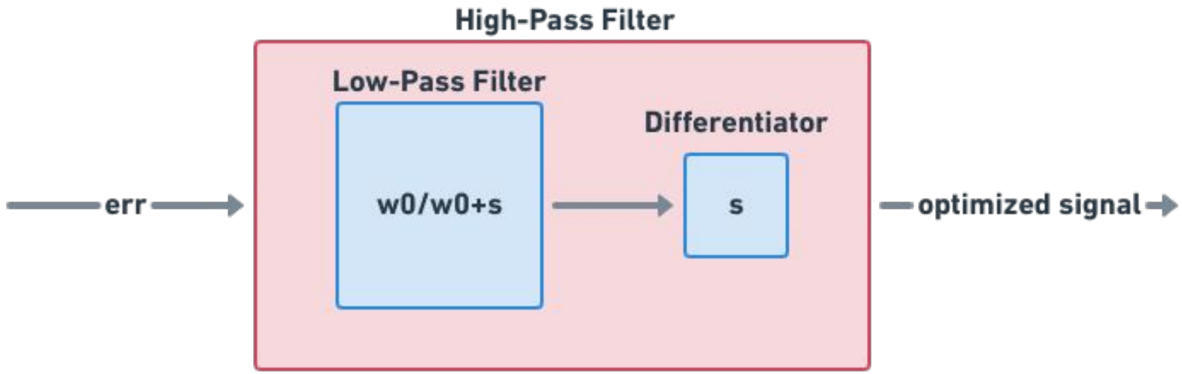


Structure of Controller

High Pass Filter

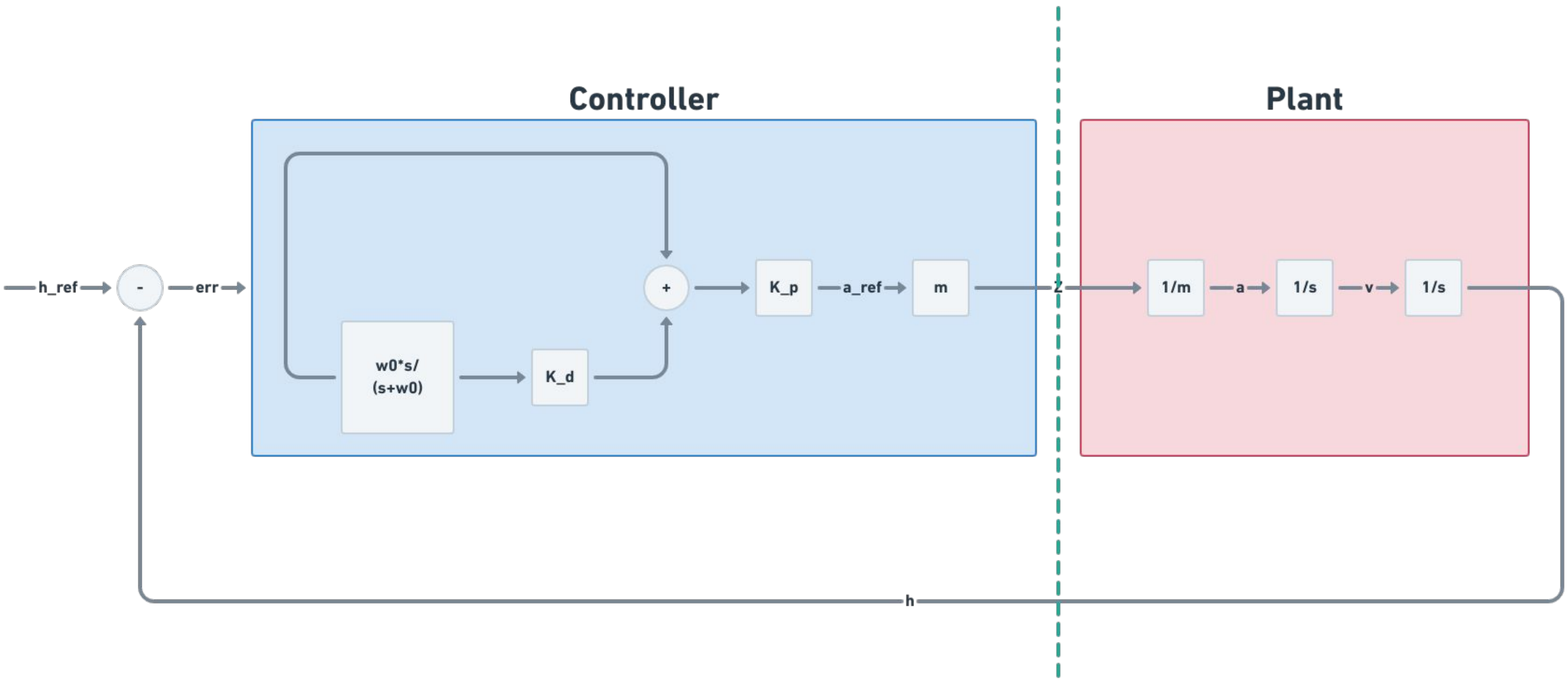
$$T(s)_{LPF} = \frac{w_c}{w_c + s}$$

$$T(s)_{HPF} = \frac{w_c \cdot s}{w_c + s}$$



Structure of Controller

Controller & Plant



Transfer Functions

Block Diagram Reduction Technique

$$G(s) = \frac{Y(s)}{U(s)}$$

Transfer function of the system→G(s)

Output of the system→Y(s)

Input of the system→U(s)

- Transfer function of a system, is a mathematical function which theoretically models the system's output for each possible input.

- For the successful implementation of this technique, some rules for block diagram reduction to be followed:

  Parallel Blocks ⇒ Summation

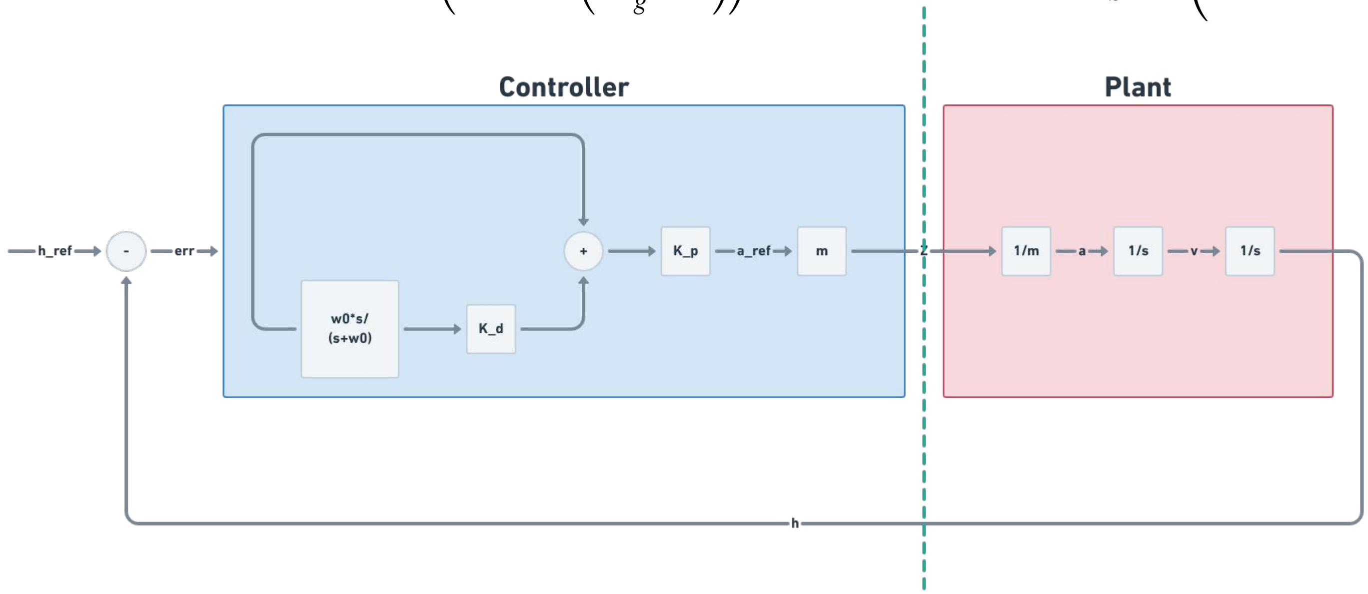
  Sequential Blocks ⇒ Multiplication

Transfer Functions

Transfer Function of the Open Control Loop

Applying block diagram reduction technique to find transfer function of the open control loop

$\rightarrow$ $$G_o(s) = \frac{1}{s}\cdot\frac{1}{s}\cdot\frac{1}{m}\cdot m\cdot K_p\cdot\left(1+K_d\cdot\left(\frac{\omega_g\cdot s}{\omega_g+s}\right)\right) \qquad G_o(s) = \frac{K_p}{s^2}\cdot\left(1+\frac{K_d\cdot s\cdot\omega_g}{s+\omega_g}\right)$$



Transfer Functions

Transfer Function of the Closed Control Loop

Deriving closed loop control transfer function from open loop control transfer function→

$$T(s) = \frac{G_o(s)}{1 + G_o(s)}$$

Solving above equation by putting **G_o(s)** back to its place→

$$T(s) = \frac{K_p + s \cdot K_p \cdot K_d}{s^2 + s \cdot K_p \cdot K_d + K_p}$$

Transfer Functions

PT2 Element

$$a_2 \cdot \frac{d^2 y(t)}{dt^2} + a_1 \cdot \frac{dy(t)}{dt} + a_0 \cdot y(t) = b_0 \cdot u(t)$$

$$\xrightarrow{\text{Laplace Transform}}$$

$$a_2 \cdot s^2 \cdot Y(s) + a_1 \cdot s \cdot Y(s) + a_0 \cdot Y(s) = b_0 \cdot U(s)$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{a_2 \cdot s^2 + a_1 \cdot s + a_0}$$

$$\xrightarrow{\text{After simplifications}}$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_0^2}{s^2 + 2 \cdot \zeta \cdot \omega_0 + \omega_0^2}$$

$$0 < \zeta < 1\ ,\ 0 < \omega_0 < \frac{4}{3}\pi$$

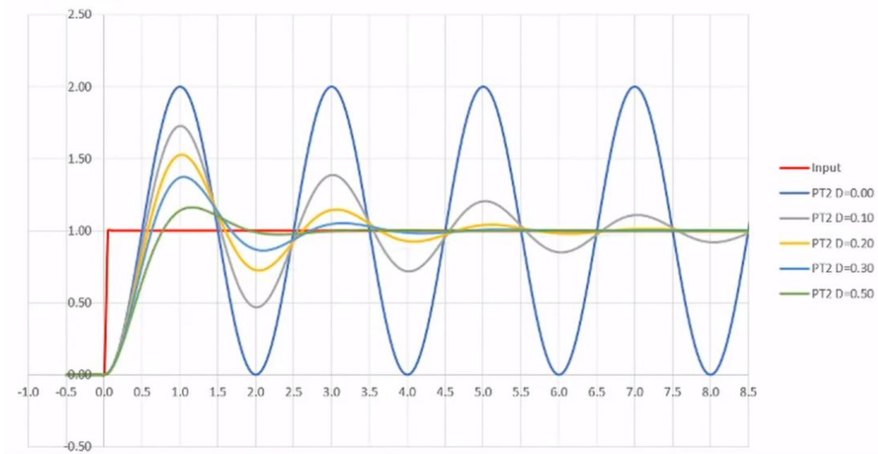
ζ: damping factor, ω_0: natural angular frequency

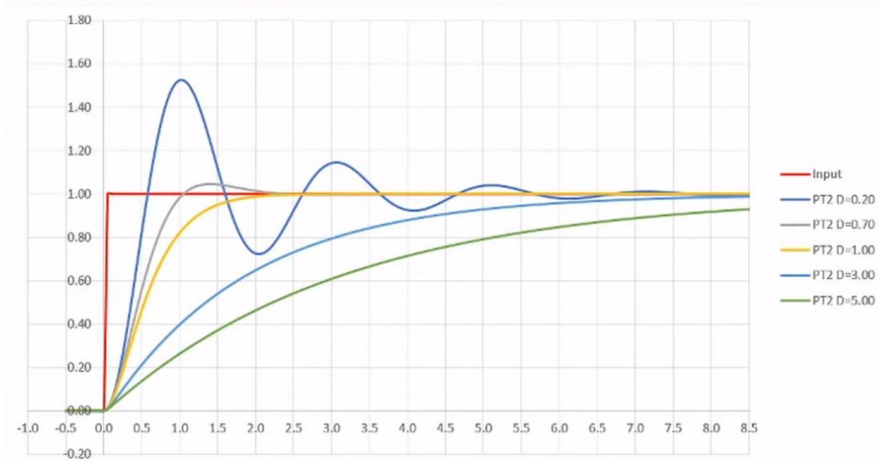
$$T(s) = \frac{K_p + s \cdot K_p \cdot K_d}{s^2 + s \cdot K_p \cdot K_d + K_p}$$

$$K_p = \omega_0^2,\ K_d = \frac{2 \cdot \zeta}{\omega_0}$$

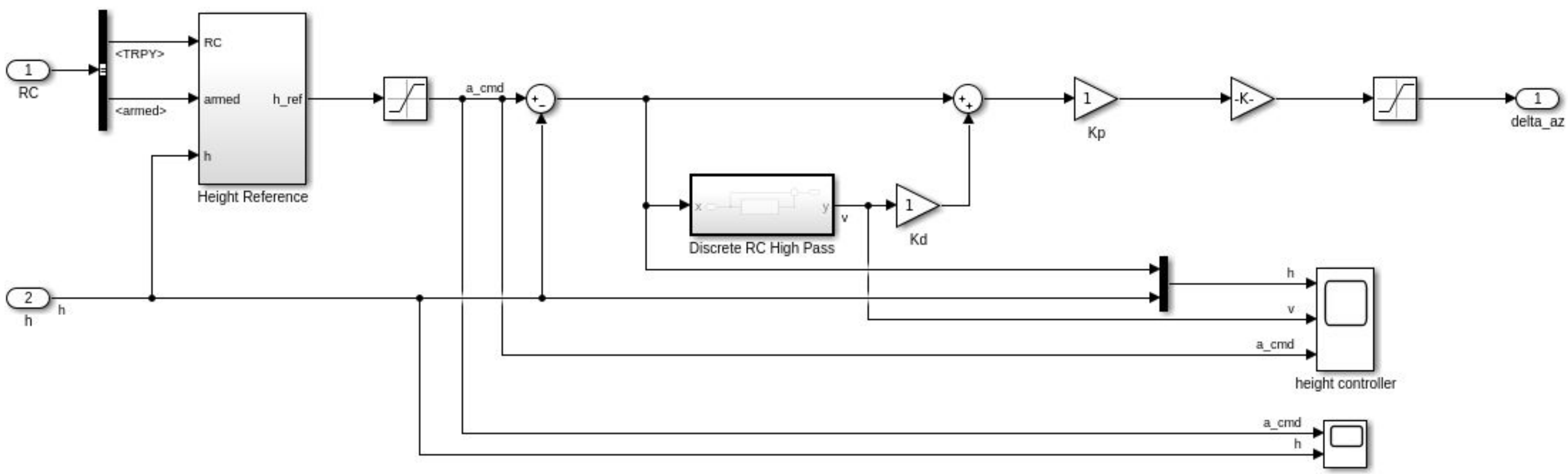
PT2 Element

System Response



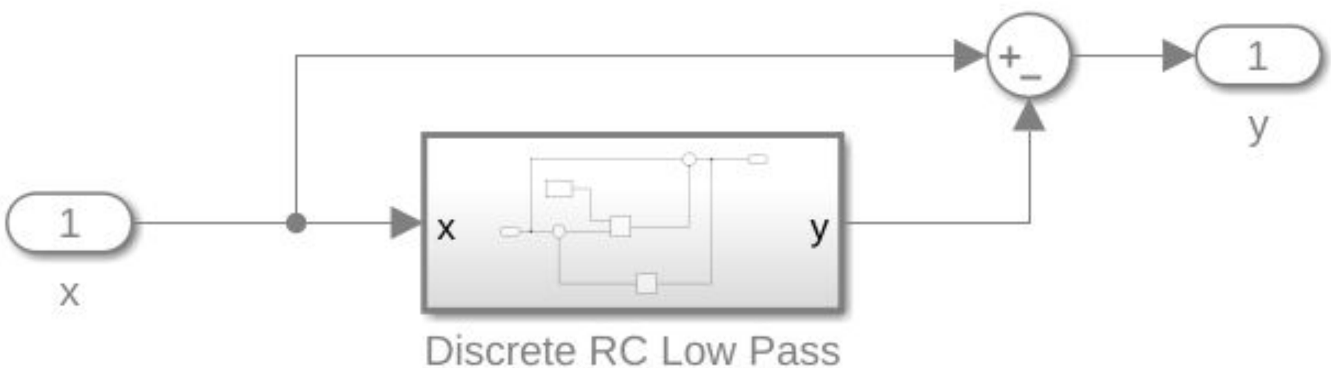


Simulink Models
Altitude Controller



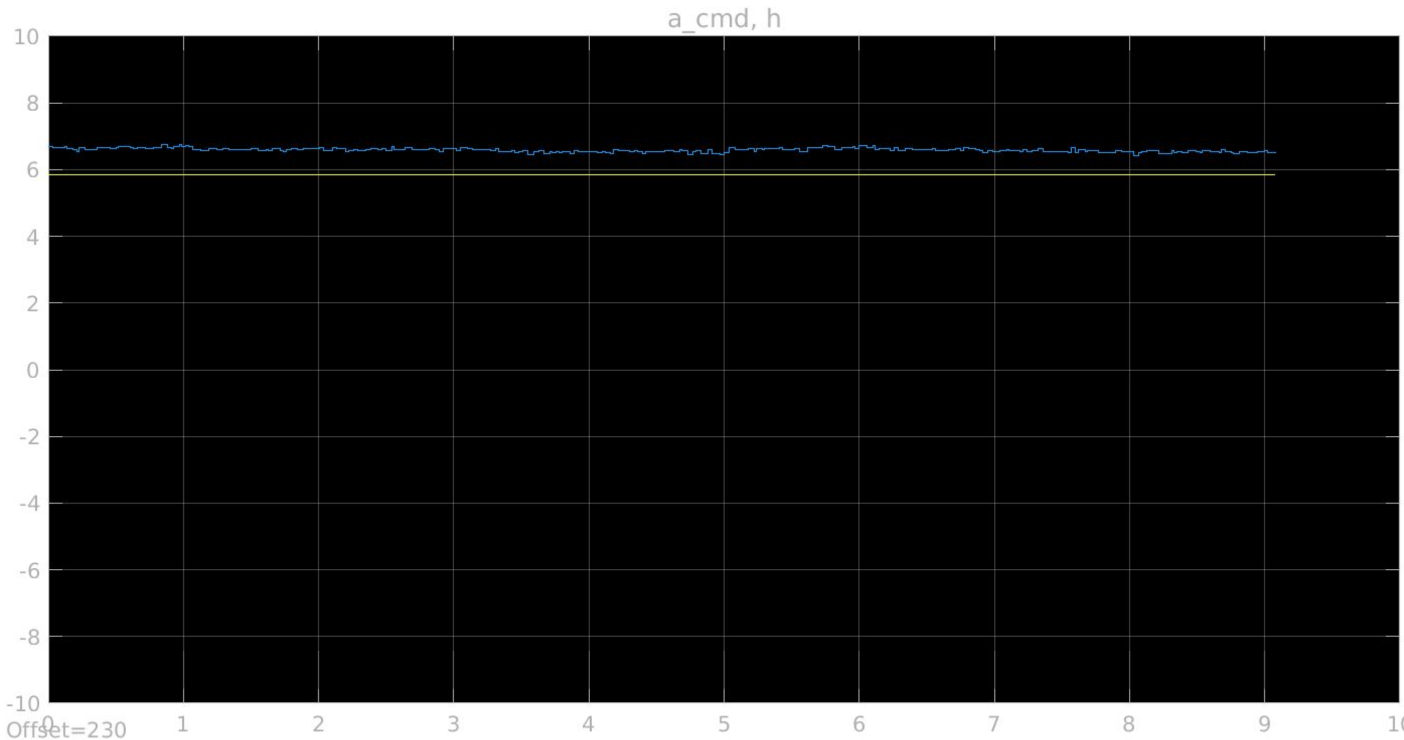
Simulink Models

High-Pass Filter



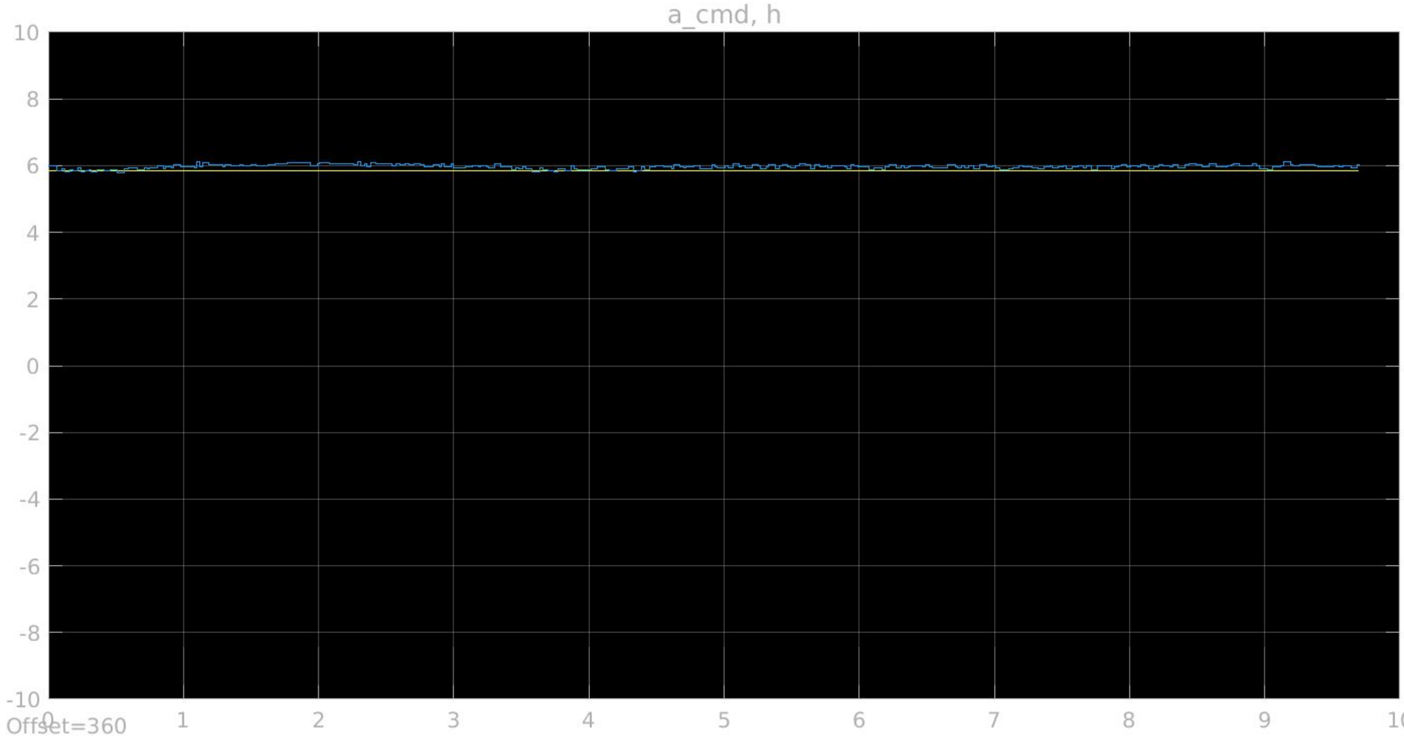
Simulink Model Outputs

ς=0.5, ω=2^1/2,K_p=2, K_d=0.7

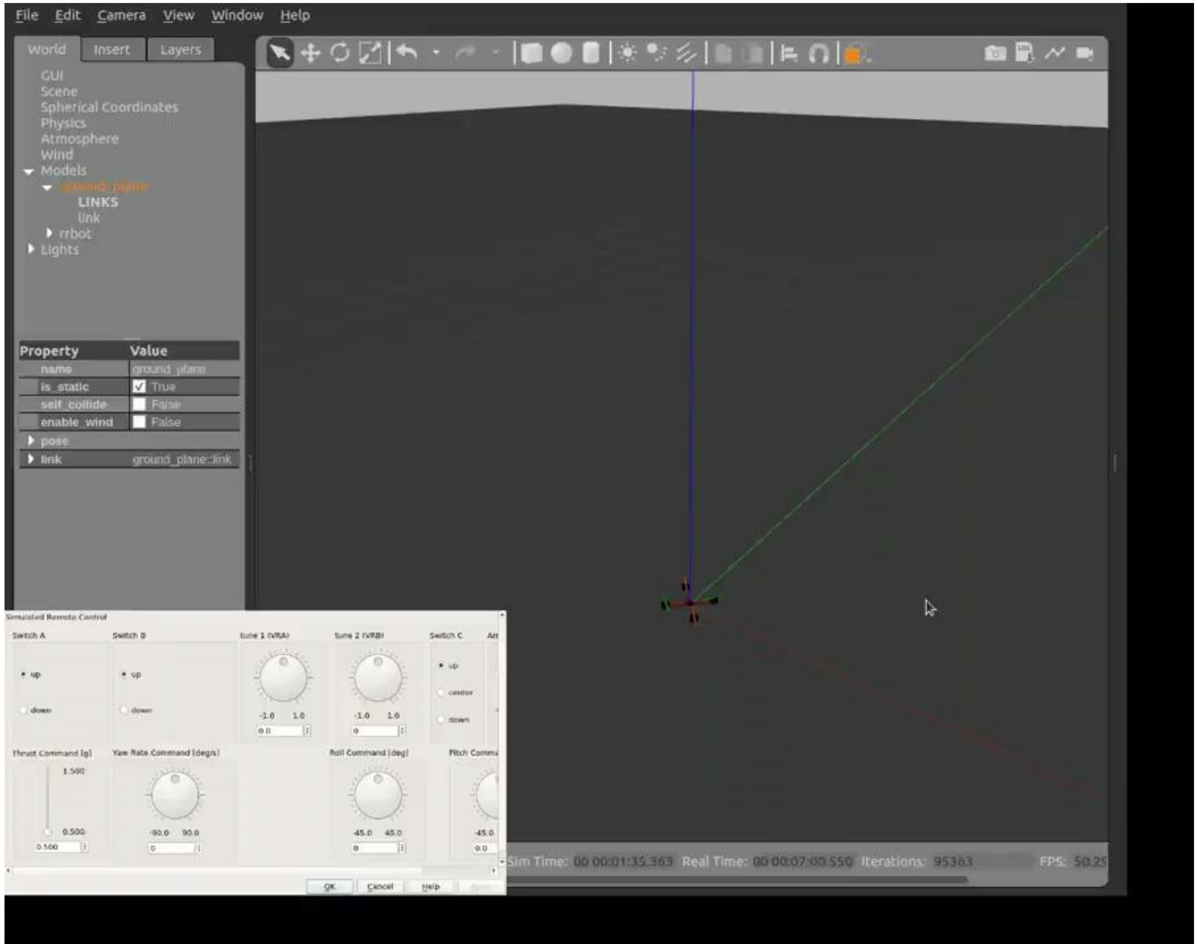


Simulink Model Outputs

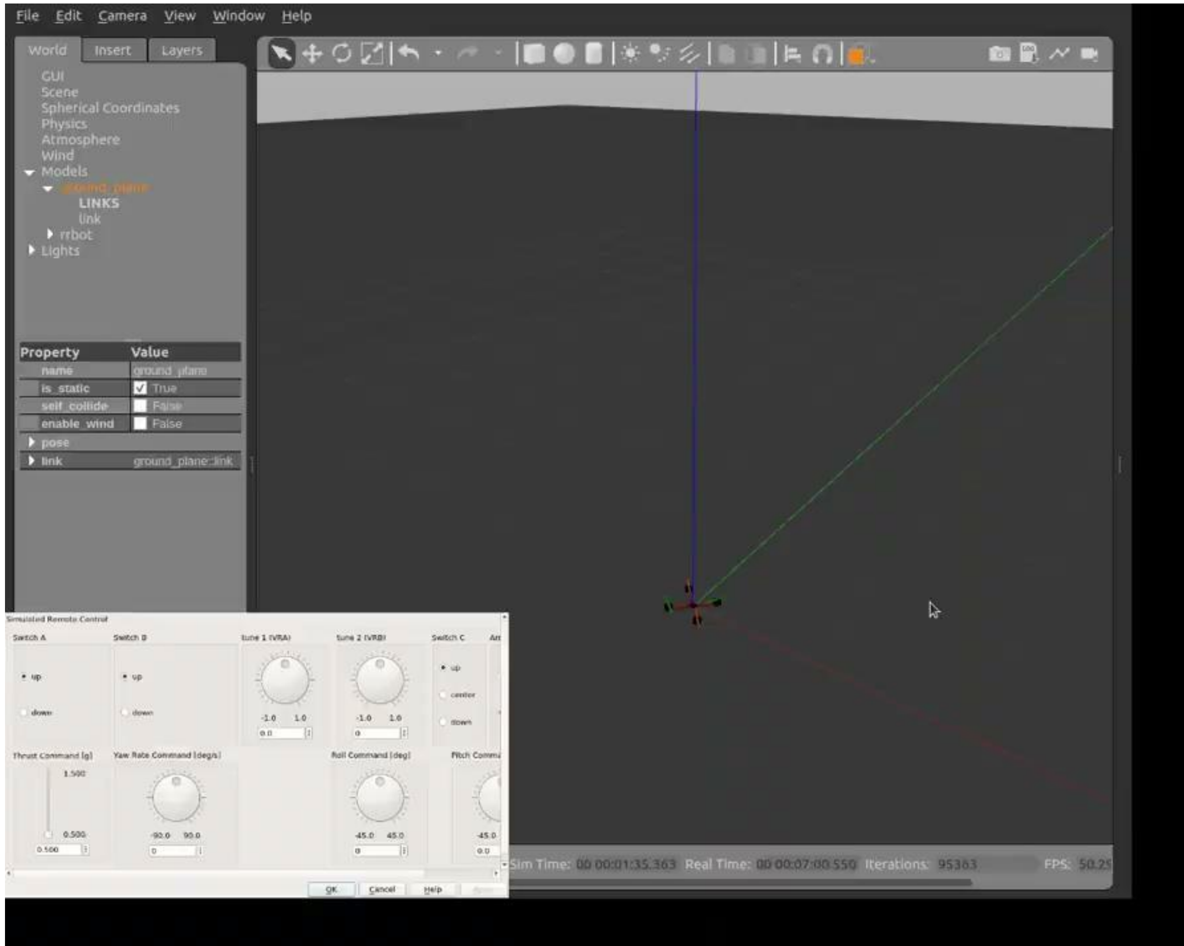
ς=0.9, ω=4,K_p=16, K_d=0.5



Video Demonstration



Video Demonstration



References

- http://wescottdesign.com/articles/pid/pidWithoutAPhd.pdf
- https://www.electrical4u.com/block-diagrams-of-control-system/
- https://www.allaboutcircuits.com/technical-articles/understanding-the-first-order-high-pass-filter-transfer-function/
- https://www.allaboutcircuits.com/technical-articles/understanding-transfer-functions-for-low-pass-filters/
- https://second.wiki/wiki/pt2-glied

Thank you!

R. Erdem Uysal

e-mail
phone +49 (0) 711 685-
fax +49 (0) 711 685-

University of Stuttgart
Flight Mechanics and Controls Lab
Pfaffenwaldring 27, 70569 Stuttgart