RPO project report

Artyom Voronin

July 8, 2020

Contents

1 Review odborného članku

Název článku: Aircraft Control System Using LQG and LQR Controller with Optimal Estimation-Kalman Filter Design
Autoři článku: Labane Chrif, Zemalache Meguenni Kadda
Odkaz na článek: www.sciencedirect.com
Seznam Příloh: aircraft.pdf

1.1 Review

Tato práce se zabývá implementaci řízení letadla pomocí metod **LQG** (Linear–Quadratic–Gaussian control) a **LQR** (Linear–Quadratic regulator) pomocí nástrojů Matlab/Simulink. Kombinace metody řízení LQR a **Kálmánová filtra** umožnuje lepší odhad parametrů, získaných ze sensorů a následovné řízení letadla.

V oblastech letectvy nároky na systémy řízení jsou násobně větší vzhledem k účinnosti a spolehlivostí systémů. S rostoucí požadavky na řízení a autonomnost letadel jedním řešením je zvětšení čísla senzoru a akčních členu. Což má za následek zvětšení ceny výrobku. Nicméně moderny přístupy k řízení systému máji byt schopny pracovat s mnoha vstupní a výstupní parametry. Jedním z řešení je LQG, který je vhodný pro použiti v praktických úlohách, kde systém ovlivněn rušením a šumem měřeni.

Pro řízeni letadla jsou 3 dostupné rotace, které umožnují změnit směr letu letadla. To jsou **Pitch**, **Roll** a **Yaw**. Řízení dále se dělí na podélný směr a boční směr.

Dynamika v podélným směru (longituadinal dynamic) ve stavové representace vypadá nesledující:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \mathbf{A} \begin{bmatrix} w \\ q \\ \theta \end{bmatrix} + \mathbf{B}[\delta_e] \tag{1}$$

$$y = \mathbf{C} \begin{bmatrix} w \\ q \\ \theta \end{bmatrix} + \mathbf{D} \tag{2}$$

Matice **A**, **B**, **C**, **D** lze dohledat v članku.

Dynamika v bočním směru (lateral dynamic) ve stavové representace má tvar:

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \mathbf{B} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{3}$$

$$y = \mathbf{C} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \mathbf{D} \tag{4}$$

Matice **A**, **B**, **C**, **D** lze dohledat v članku.

LQG regulátor se skládá z filtru Kalmana a LQR regulátora zařazeních za sebou, jak znázorňuje následující diagram 1.

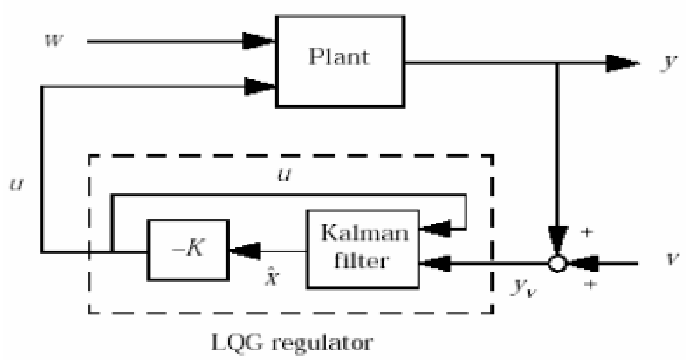

Figure 1: Plant model in Simulink.

Pro přenos regulátora platí:

$$\frac{d}{dt}\hat{x} = [A - LC - (B - LD)K]\hat{x} + Ly_v \tag{5}$$

$$u = -Kx \tag{6}$$

Soustava obsahující rušeni je doplněna o w a v, které představuji bily sum:

$$\dot{x} = Ax + Bu + Gw \tag{7}$$
$$y_v = Cx + Du + Hw + v \tag{8}$$

Při návrhu LQR regulátoru uživatel na základě zkušeností odhaduje matice Q, N, a R které představuje rychlost, se kterou požadovaný parametr se přibližuje k žádané poloze a také úsilí, které regulátor musí vykonat pro dosazeni této polohy. LQR regulátor pracuje se znalostí celého stavového vektoru. Ale ve většině případu nejsme schopny měřit cely stavový vektor x. Proto je nutné použití Kálmánův filtr, který, pokud systém pozorovatelný a řiditelný, je schopný tento stavový vektor odhadnout.

Při návrhu Kálmánova filtru počítáme že systém obsahuje určitou míru rušení (v případě letadla to může způsobit vítr, nebo změny v hustotě vzduchu, které vyvolá vibrace) a také měření ze sensorů obsahuje šum. Hlavním cílem je navrhnou Kálmánův filtr tak aby odhad parametrů byl co nejpřesnější.

Výsledky simulace znázorněny v článku.

1.2 Krirické zhodnocení

Z uvedených výsledku lze konstatovat že řízení LQG je schopné řídit pitch angle, roll angle a sideslip angle letadla. Přítomnost Kálmánova filtru zaručuje optimální odhad a rekonstrukce parametru stavového vektoru pří výskytů bílého sumu.

Článek vyžaduje určitou míru znalosti v problematice dynamiky letadla. Většina koeficientu v kapitolách 2. a 3. vyžaduje studium dalších zdrojů. Při čtení mě přispěla presentace adl.stanford.edu/ tykající základní informace o dynamice letadla. Nicméně následující kapitoly dostatečně přehledné a informativní. Implementace diskrétního Kalmana vyžaduje od čtenáře znalost použiti přechodů v Simulinku mezi systémem se spojitým časem a diskrétním. Toto lze zrealizovat například blokem Zero-Order-Hold.

2 State space control (LQG) of stable aircraft

2.1 Creation and description of the model

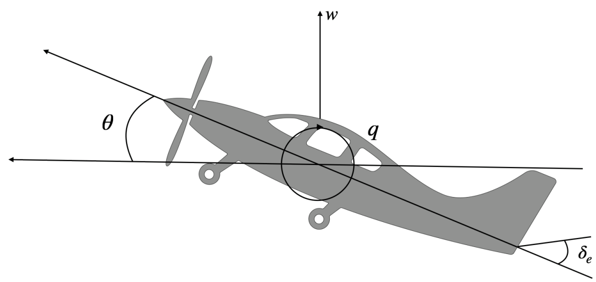

Figure 2: Aircraft longitudinal motion.

β	Sideslip angle
p	Roll rate
r	Yaw rate
Φ	Roll angle
w	Vertical velocity
q	Pitch rate
θ	Pitch angle
δ_e	Elevator deflection

TODO: More about the system model.

The dynamic model of the aircraft system 2 is taken from the article [1]. For longitudinal motion:

$$\beta = p = r = \Phi = 0 \tag{9}$$

The transfer function is represented in state-space form by matrices:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{10}$$

$$A = \begin{bmatrix} -0.3149 & 235.8928 & 0 \\ -0.0034 & -0.4282 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -5.5079 \\ 0.0021 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$D = 0$$

$x^T = \begin{bmatrix} w & q & \theta \end{bmatrix}$

$u = \delta_e$
The implemented Model in Simulink is presented in the following diagram 3.

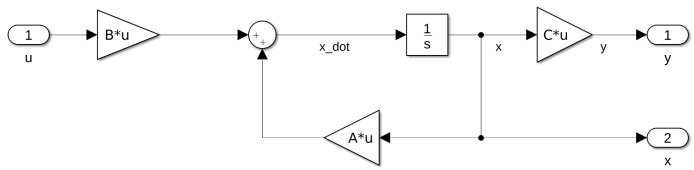

Figure 3: Plant model in Simulink.

In the Matlab system can be represented by the state space function:

```
>>sys = ss(A,B,C,D)
```

2.2 Open loop response

The system is stable only if all the real parts of the eigenvalues are negative.
The dynamics of the model is determined by the following parameters:
Pols:

$$\begin{array}{r} 0.0000 + 0.0000i \\ -0.3715 + 0.8938i \\ -0.3715 - 0.8938i \end{array}$$

The real parts of eigenvalues are negative. The system is **stable**.
Controllability matrix rank: 3
If the rank of Controllability matrix $Co = [B\ AB\ A^2B\ \ldots\ A^{n-1}B]$ is equal to n (number of states), the system is **controllable**.
Observability matrix rank: 3
If the rank of Observability matrix $O = [C\ CA\ CA^2\ \ldots\ CA^{n-1}]$ is equal to n, the system is **observable**.

The whole system is described by 10 equations. Open-loop impulse response is presented on figure 4. The result corresponds to our expectations.

Open-loop step response is presented on figure 5. The result corresponds to our expectations. Aircraft started to increase pitch angle.

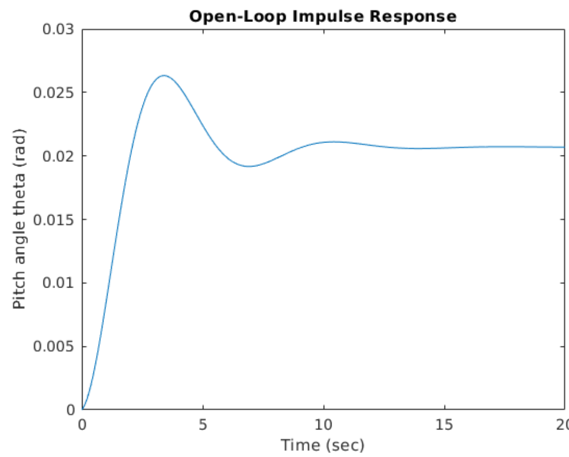

Figure 4: Open-loop impulse response.

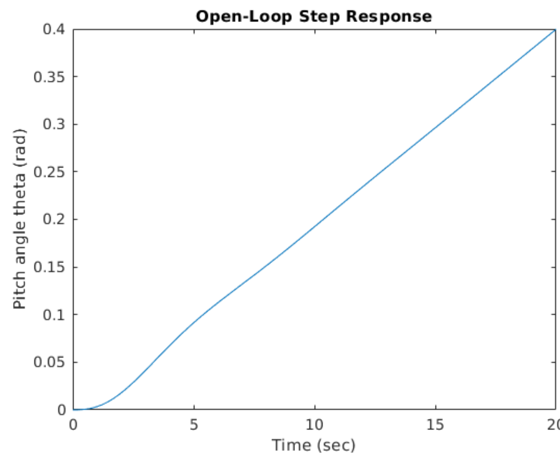

Figure 5: Open-loop step response.

2.3 LQR design - zero control

In modeling zero control response, the input of the system will be $u = -Kx$, $(r = 0)$. Substituting into equation 10, the result of Close-loop system will correspond to equation 11.

$$\dot{x} = (A - BK)x \tag{11}$$

For calculation K gain were used LQR method. This method based on quadratic cost function J. Where Q matrix represent how "fast" controller will approximate the following state. R matrix represent how much we depend on energy that we add to system to control.

$$J = \int (x^T Q x + u^T R u) dt \tag{12}$$

In Matlab we define Q and R matrix and use a **lqr** command.

```
Q = diag([0, 0, 500]);
R = .1;
K = lqr(A,B,Q,R);
```

Simulink model of Close-loop system is presented in the following diagram 6.

2.4 LQR design - control to the point - rscale

N gain is used to scalar our input for a full-state feedback system to eliminate the steady-state error.

$$u = -Kx + rN \tag{13}$$

where r is final point. Substituting equation 13 into equation 10, the result of Close-loop system controlled to r will correspond to equation 14.

$$\dot{x} = (A - BK)x + BNr \tag{14}$$

For calculation **N** gain in matlab were used rscale function from [2].

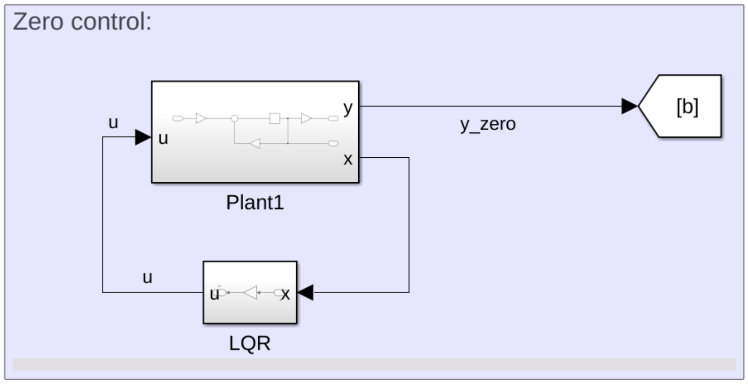

Figure 6: Zero control model in Simulink.

```
N = rscale(A,B,C,D,K);
```

Simulink model of Close-loop system controled to point is presented in the following diagram 7.

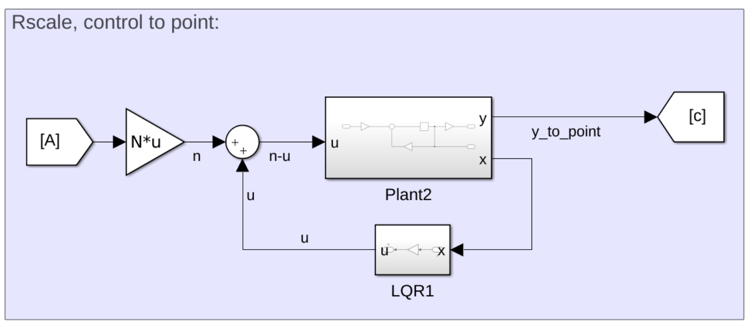

Figure 7: Rscale control to point model in Simulink.

In the Matlab system can be represented by the state space function:

```
>>sys = ss(A–B*K,B*N,C,D)
```

The Close-loop impulse and step response are presented in the following figure 8, 9.

2.5 Study of system behavior

In case Longitudinal dynamic it's possible to measure θ (pitch angle). There is no specific conditions for deflector angle actuator. The $R = 0.1$ matrix we will keep equal to 0.1. In the following figure 10, there are impulse responses for different Q matrices. Where we change x parameter from 1 to 500 with 50 as step.

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & x \end{bmatrix}$$

The following Q and R matrices will suffice for our purposes. These values will be used in the following simulations.

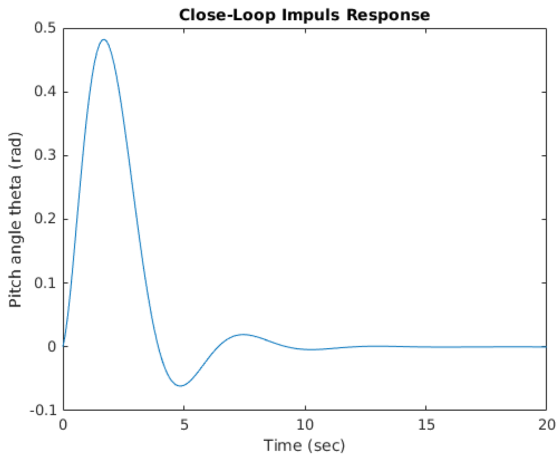

Figure 8: Close-loop impulse response.

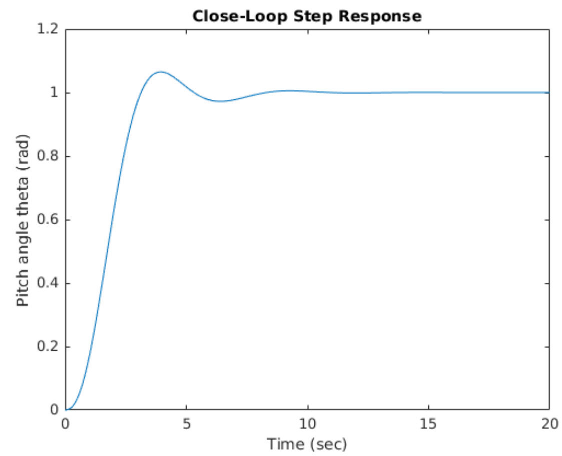

Figure 9: Close-loop step response.

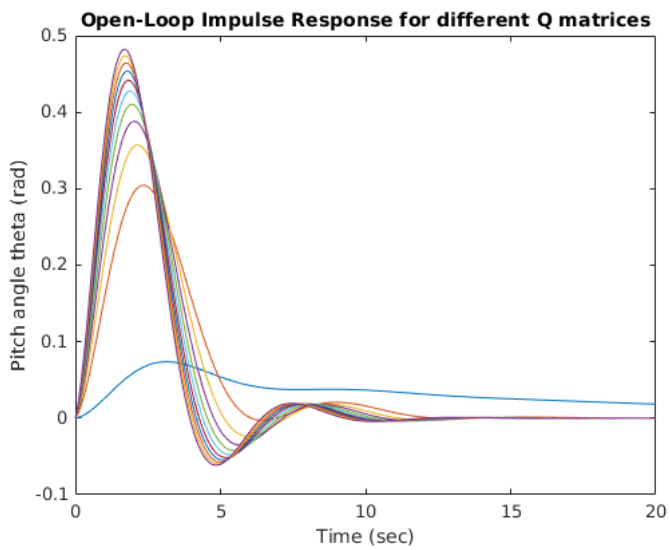

Figure 10: Impulse response for different Q matrices.

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 500 \end{bmatrix} \quad R = 0.1$$

2.6 Observer's proposal

The LQR controller can be used if we have information about the whole state x. A Kalman filter can be used to reconstruct the state from the y measurement. The following system of equations 15 representing Kalman filter.

$$\begin{aligned} \frac{d}{dt}\hat{x} &= A\hat{x} + Bu + Kf(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \tag{15}$$

Simulink model of Kalman filter is presented in the following diagram 11.

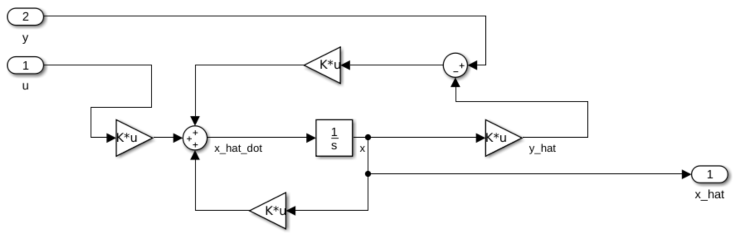

Figure 11: Kalman filter implementation in Simulink

Kalman filter use knowledge about disturbance of state and measurement noise magnitudes. Vd matrix representing covariance of state disturbance and Vn matrix is covariance of measurement noise. There are more options how to calculate Kf gain. Define Vd and Vn matrices we can use the same **lqr** function as follow:

```
Vd = .01*eye(3);
Vn = 1;
Kf = (lqr(A',C',Vd,Vn))';
```

Simulink model of system is presented in the following diagram 12.

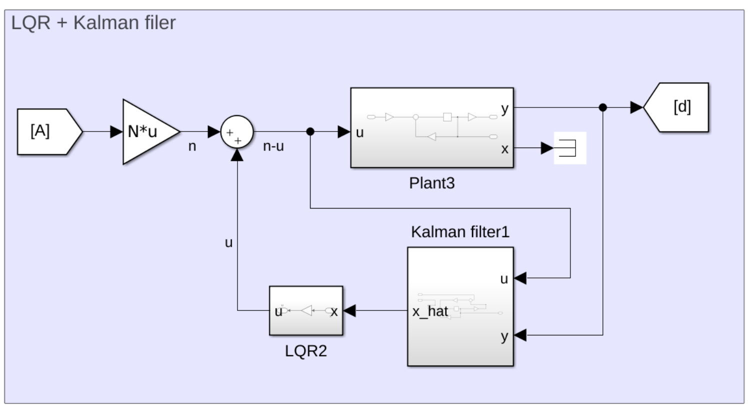

Figure 12: Rscale control to point model in Simulink.

Implementing Kalman filter in Matlab:

```
Akf = A-Kf*C;
Bkf = [B Kf];
Ckf = eye(3);
Dkf = 0*[B Kf];
sys_kf = ss(Akf, Bkf, Ckf, Dkf);
```

2.7 Study of system behavior with an observer (simulation)

The real system control contain measurement noise and disturbance. In model 13 were used w_d and w_n disturbance and noise inputs as Gaussian white noise.

The whole system with LQG implementation, disturbances and noise can be describe as following system of equations 17.

$$\epsilon = x - \hat{x} \tag{16}$$

$$\begin{bmatrix} \dot{x} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - KfC \end{bmatrix} \cdot \begin{bmatrix} x \\ \epsilon \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -Kf \end{bmatrix} \cdot \begin{bmatrix} w_d \\ w_n \end{bmatrix} \tag{17}$$

The whole model is controlled by placing eigenvalues in $A - BK$ and $A - KfC$ by K and Kf matrices.

Simulink model is presented in the following diagram 13.

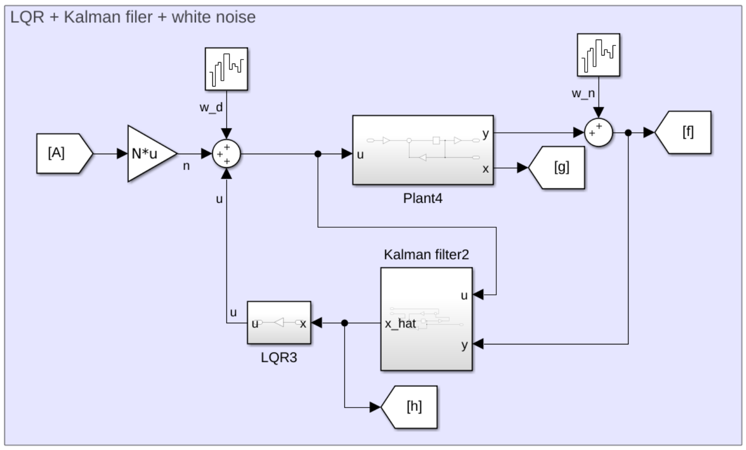

Figure 13: LQG simulink model

The following graph 14 shows the correct operation of the LQG controller. Kalman filter correctly estimates the state with permissible noise level.

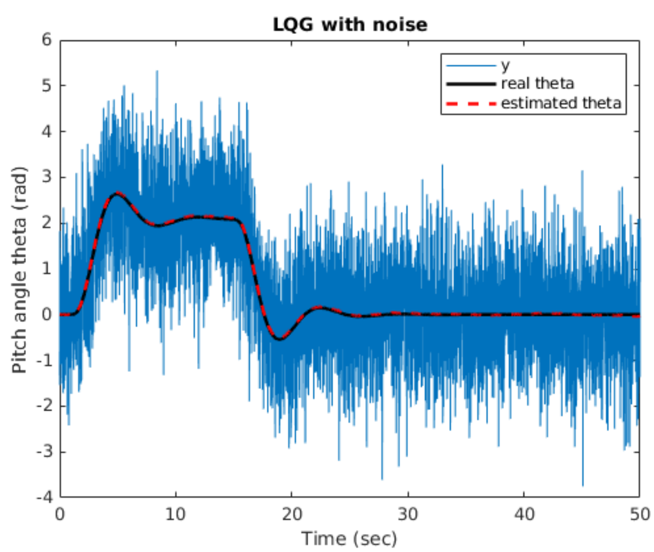

Figure 14: LQG regulator output

With increasing noise amplitude, the model is able to remain functional. This can be verified by changing the disturbance and noise magnitude in the Simulink model.

2.8 Evaluation of the whole task and conclusion

LQG can be used to control the pitch angle of an aircraft. LQG is sufficiently resistant to the disturbance and measurement noise. Implementation in the Matlab/Simulink environment sufficiently clear and easily scalable. All Simulink models and Matlab scripts are available in appendix.

3 System Identification

System name: Helicopter azimuth

List of files:

measure.m	Script for measuring I/O data
main_params_plot.m	Define parameters and plot data
model_data_measure.slx	Model for data measurement
model.slx	Main model
model_simple.slx	Simplify model
heli_data.mat	Measured data for parameter estimation
model_estimation_session.mat	Parameter estimation session

3.1 System analysis and generation I/O data

The following Simulink model was used to measure the data 15

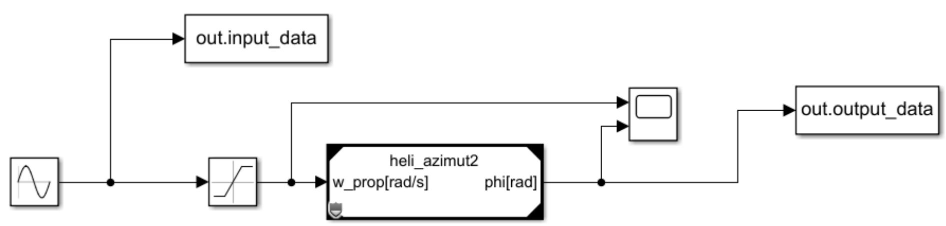

Figure 15: Simulink model for data measurement

3 types of input data were used: sine function, step response, pulse function 16.

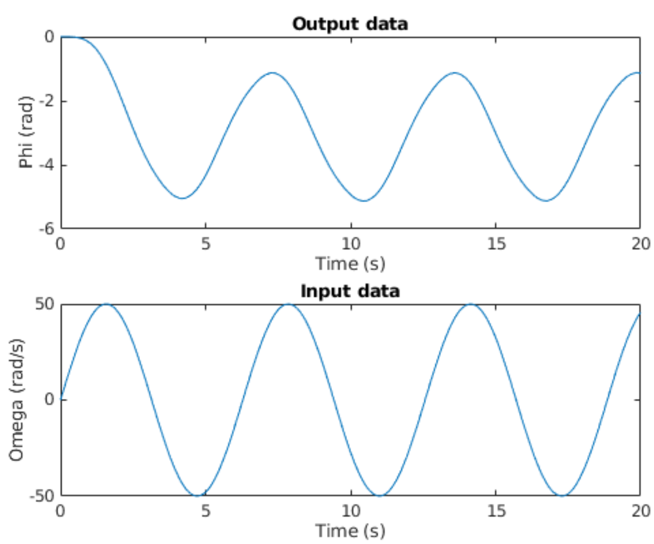

Figure 16: Simulink model for data measurement

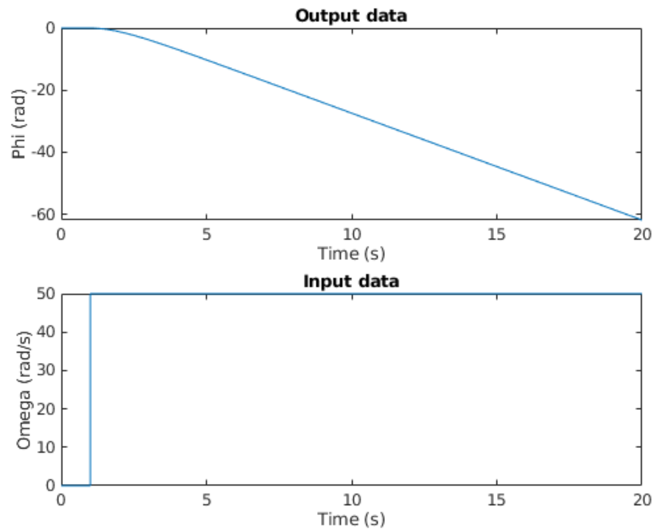

Figure 17: Simulink model for data measurement

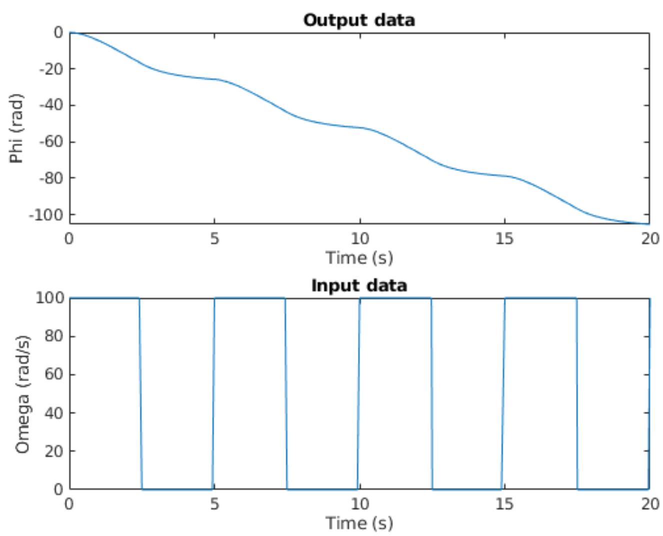

Figure 18: Simulink model for data measurement

3.2 Model description

a	Side length of flat plate perpendicular to flow $[m]$
θ	Turn angle $[rad]$
m	Mass $[kg]$
ω	Angular velocity of propeller $[rad/s]$
L	Length $[m]$
I	Moment of inertia $[kg/m^2]$
b	Friction coefficient [-]
$C_x \approx 1.28$	Drag coefficient [-]
$\rho \approx 1.2$	Mass density of the fluid $[kg/m^3]$
A	Propeller area $[m^2]$
$S = a^2$	Area of flat plate $[m^2]$

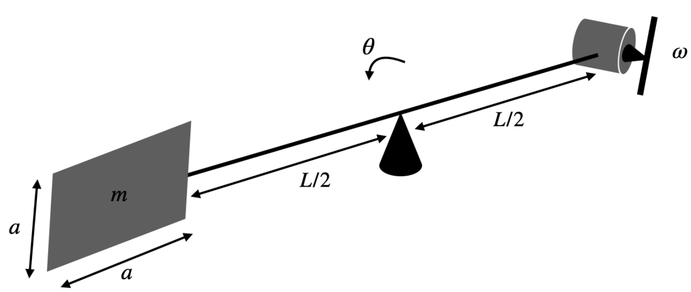

Figure 19: Model helicopter azimuth

Math model of helicopter azimuth described by equation 19.

$$I\ddot{\theta} = M_{prop} - (M_f + M_d) \tag{18}$$

$M_d = \frac{1}{2}C_x\rho S\left(\frac{L}{2}\right)^3\dot{\theta}^2$ - drag equation.
$M_f = b\dot{\theta}$ - friction momentum.
$M_p = \frac{\rho AL}{4}\omega^2$ - momentum from propeller thrust.

$$\ddot{\theta} = \frac{\rho AL}{4I}\omega^2 - \frac{b}{I}\dot{\theta} - \frac{1}{2I}C_x\rho S\left(\frac{L}{2}\right)^3\dot{\theta}^2 \tag{19}$$

Parameters to estimation: $p_0 = \frac{AL}{I}$, $p_1 = \frac{b}{I}$, $p_2 = \frac{SL^3}{I}$
Simulink model represented in following diagram 20.

3.3 Parameter estimation

The model 20 doesn't working with Parameter Estimation App. The problem is power variables. That's why I was forced to remove powers and simplify a model to 21.

The results are saved and can be checked in **model_estimation_session.mat**.

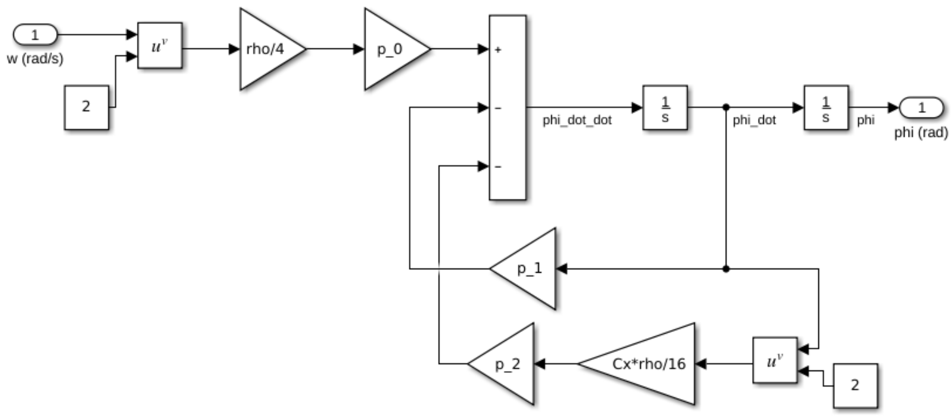

Figure 20: Simulink model of helicopter azimuth

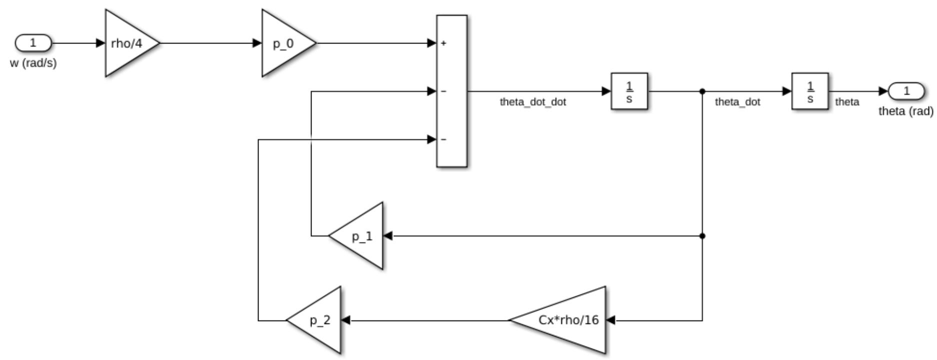

Figure 21: Simulink simplify model of helicopter azimuth

3.4 Feed-forward

3.5 System Identification (Black-box)

3.6 Evaluation of the whole task and conclusion

The parameter estimation app did not work with powers of the parameters, therefore the model was simplified from 20 to 21. Estimation approached the output data but not exactly.

4 HIL

System name: One wheel skateboard

List of files:

main.m	Parameters initialization
model.slx	Main model
dataOneWheel.mat	Data from accelerometer from mobile Matlab app

Datasheets:

adxl335.pdf	Accelerometer
lts_6-np.pdf	LEM sensor
moc23series.pdf	DC motor

4.1 Selection and description of the controlled system

Modeled system is one wheel skateboard 22.

Figure 22: One Wheel [3]

The simplify model of the one wheel skateboard system features a DC motor, accelerometer and control unit 23.

4.2 Model of a controlled system in Simulink

Controlled system is DC motor with digital controlled H-bridge 24

4.3 Models of sensors and actuators

As a sensors were used encoder with 2 channels A and B to measure position and velocity of dc motor. LEM sensor to measure current in dc motor 25. And accelerometer sensor that provide accelerations in 3-axis. Accelerations recalculated to angle (pitch) for stabilizing a system and provide acceleration from dc motor 26.

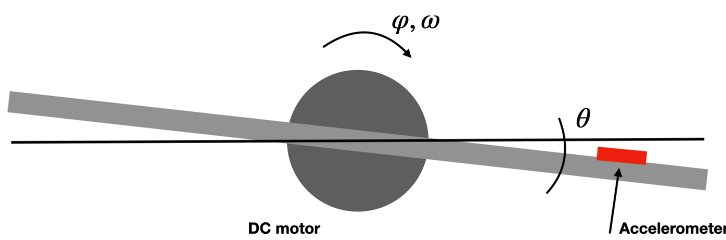

Figure 23: One Wheel simplify model

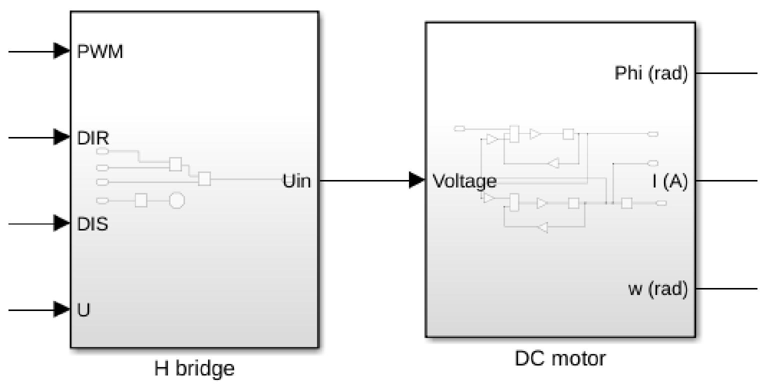

Figure 24: Controlled system Simulink model

4.4 Control unit and signal adaptation

Control Unit process acceleration data, LEM sensor data and motor actual velocity 27.

4.5 Tests

4.5.1 Correct setup

Correct setup with acceleration data from mobile Matlab application 29.

4.5.2 Changing parameters

Change rotor inertia from $J = 2189 \cdot 10^{-6}[kg/m^2]$ to $J = 1[kg/m^2]$. Result is present in 30 graph.

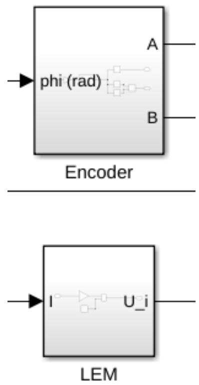

Figure 25: Encoder and LEM sensors

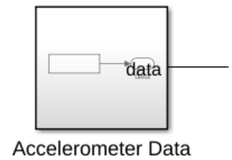

Figure 26: Accelerometer sensor

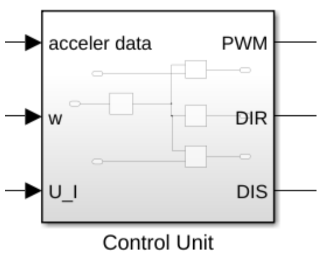

Figure 27: Control sub-block

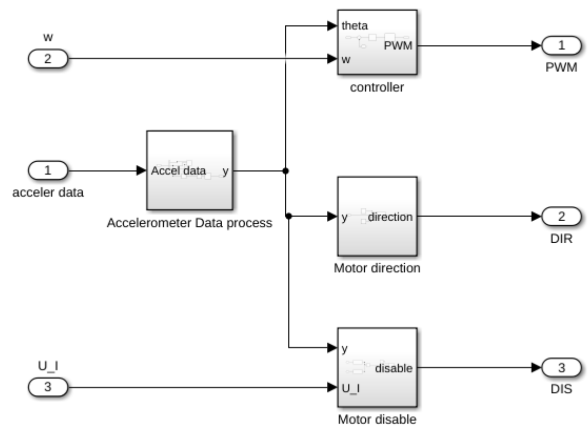

Figure 28: Control unit

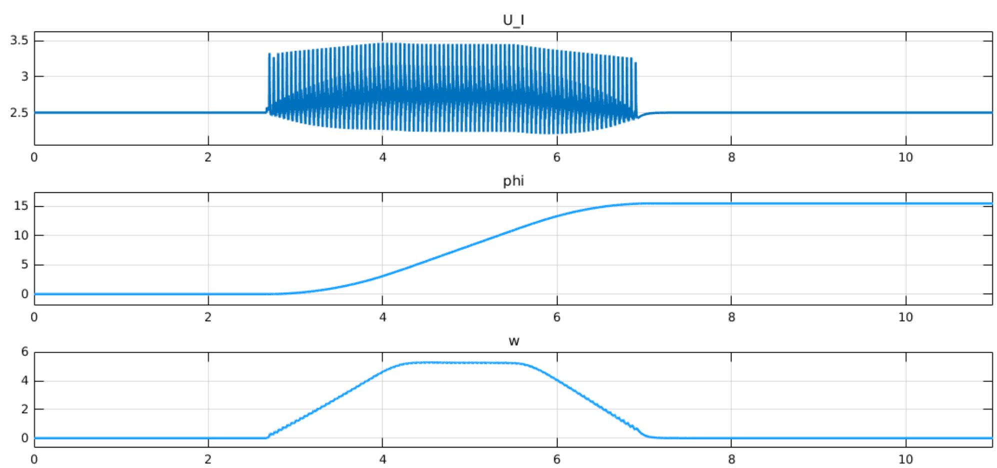

Figure 29: Correct setup test

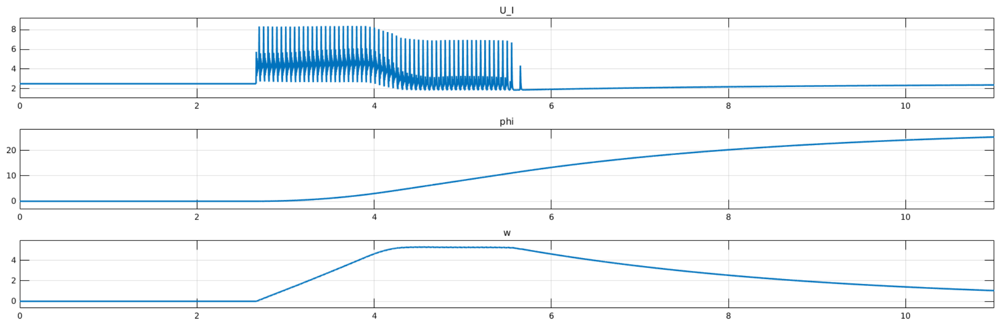

Figure 30: Changing parameters test

4.5.3 Fault setup

Change the maximum pitch angle from $\varphi_m ax = 10[rad]$ to $\varphi_m ax = 3[rad]$ 31.

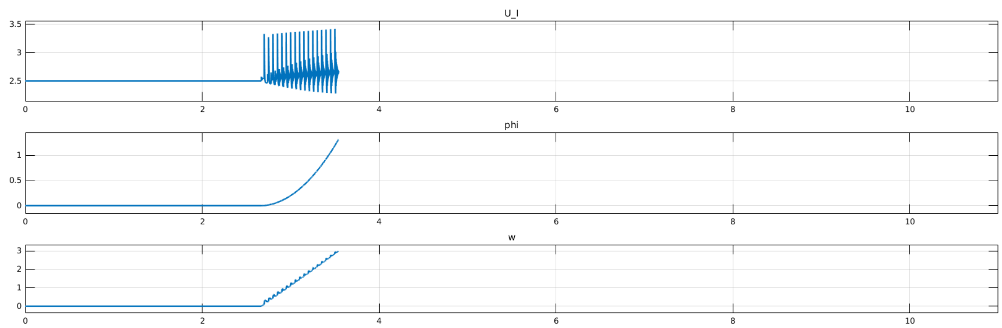

Figure 31: Fault setup test

4.6 Evaluation of the whole task and conclusion

To control a real one wheel, it is necessary to use a gyroscope together with an accelerometer. The accuracy of the encoder model was insufficient to control the model, so an analog signal was used. Datasheet of used actuator and sensor are included.

Bibliography

[1] Labane Chrif and Zemalache Meguenni Kadda. Aircraft control system using lqg and lqr controller with optimal estimation-kalman filter design. *Procedia Engineering*, 80:245 – 257, 2014. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2014.09.084. URL http://www.sciencedirect.com/science/article/pii/S1877705814011771. 3rd International Symposium on Aircraft Airworthiness (ISAA 2013).

[2] Rscale function. URL http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_rscale.

[3] One wheel. URL https://onewheel.com/.