AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

KATEDRA AUTOMATYKI I ROBOTYKI

Capstone Project I

A one-degree of freedom model of a tethered helicopter.

Authors: Jan Gliwa, Emilian Rejman, Dominik Stachura
Course of study: Automatic Control and Robotics, Cyber-Physical Systems
Tutor: dr inż. Andrzej Tutaj

Kraków, 2019

Introduction

The goal of the project was to identify and evaluate the LQR steering process of laboratory station. The station consisted of helicopter model with restriction to one vertical axis of freedom. We had to identify the hardware components and its parameters, evaluate mathematical model, verify the real data with calculated model and to fit the steering process so that the real helicopter is steerable.

Hardware description

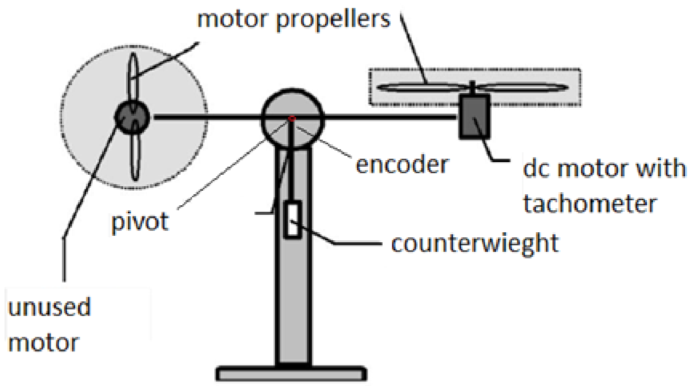

Figure 1. Graphical description of the hardware of laboratory station.

1) DC motor – the model consisted of two Brushless motors with propellers attached to motor shaft
2) Encoder – attached in the middle of the rotary axis
3) Tachometer – used to identify the actual speed of dc motor
4) Counterweight – to introduce bigger oscillations to the station

Sensor calibration

Propeller velocity measurement scaling

The first step was to identify the gain of measuring card from tachometer signal. Firstly, we set random control signal, which was sent to the engine. We measured the raw real voltage using multimeter on tachometer terminals and compared it with value transformed by the measurement card displayed in manual MATLAB model provided by the producent.

The raw value was almost twice smaller than the displayed value, which gave us gain of measurement card equal to 2.

From tachometer datasheet we got the resolution of tachometer equal to 0.52 / 1000 RPM. We introduced those two values into our model.

We compared propeller velocity value calculated in the program with propeller velocity measured using device. Obtained values gave nearly identical results, which confirmed correct identification and calibration.

Identification of dependency between rotational velocity and control value

The measurements were conducted, and the results were as followed:

Table 1. Achieved propeller velocity depending on control signal.

control value [-]	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5
propeller velocity [RPM]	-2280	-1950	-1600	-1173	-670	0	670	1220	1620	1990	2250

Polynomial coefficients of approximation were calculated basing on measurements.

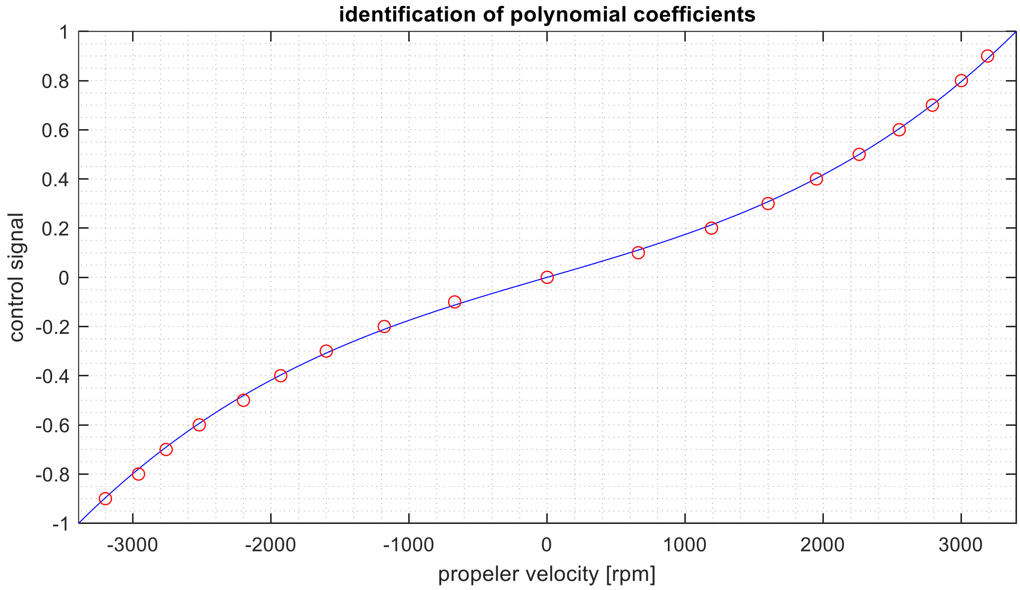

Figure 2. Dependence between control signal and propeller velocity.

Polynomial coefitiants were RPM(u)^-1 -> u(RPM) because that function was needed to provide to the model.

The assumption was made that this model can be described as first order inertial model. The parameter G=K/T was calculated with the usage of *fminsearch* function to which we introduced the real values and the values obtained from the model.

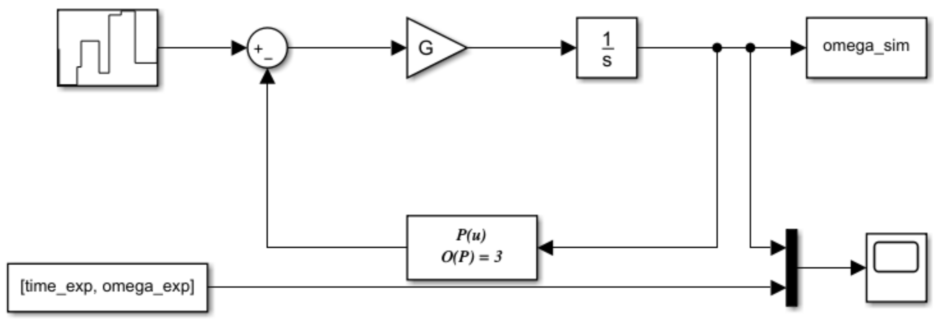

Figure 3. Model of propeller velocity calculation using control signal.

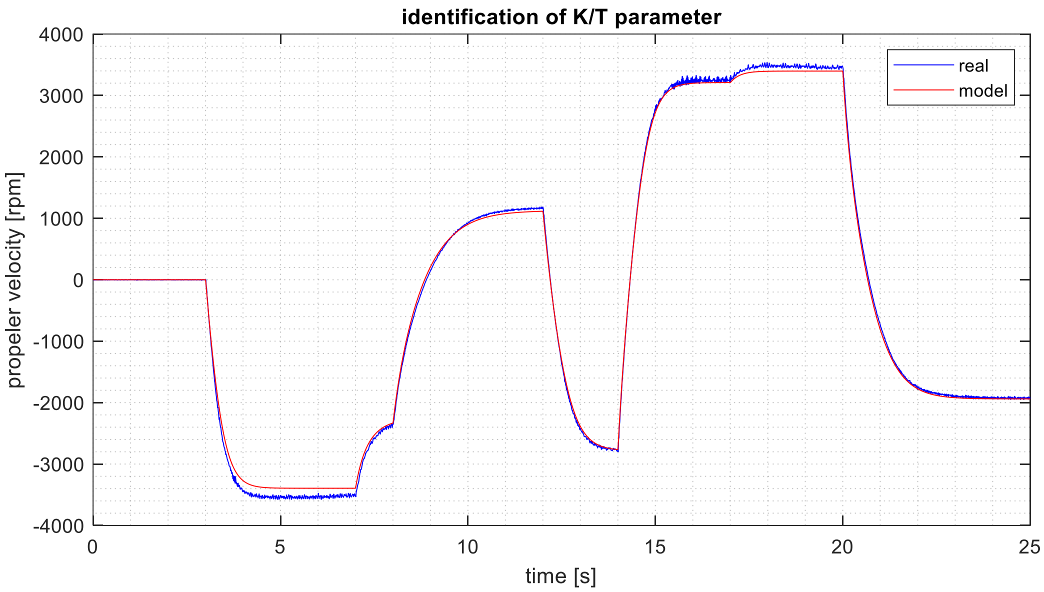

Figure 4. Identification of G parameter.

Obtained $G=\frac{K}{T}=7{,}356*10^3$.

Verification was made on different values to confirm the correctness of identification.

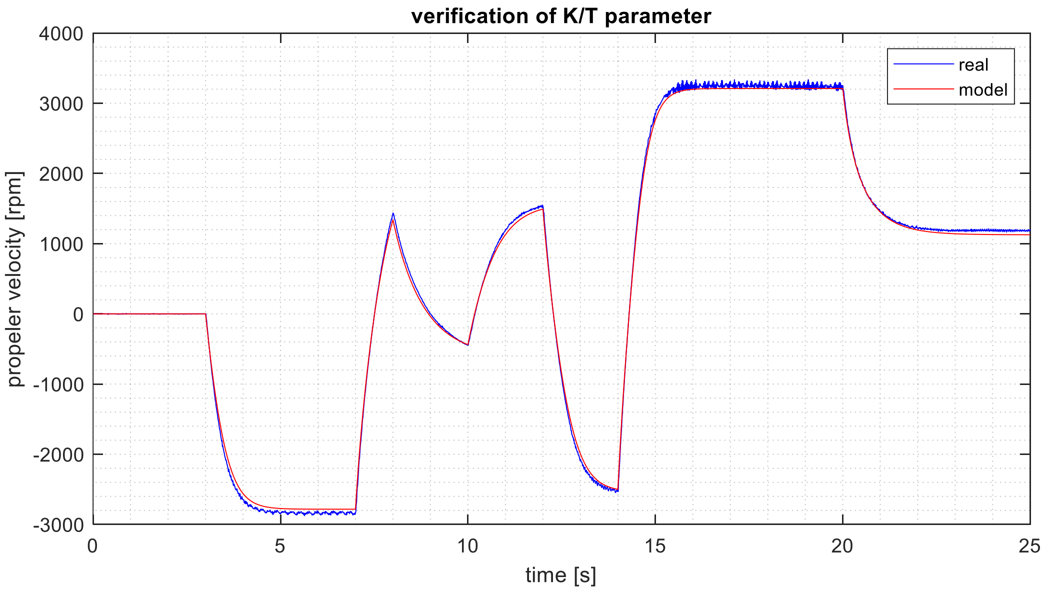

Figure 5. Verification of obtained G value.

Identification of the throttle of DC motor with propellers in respect to propeller velocity.

In order to identify the throttle of DC motor with propellers in respect to propeller velocity, the system was brought into the balance. Then, the propeller velocity and mass used to set the object in equilibrium were measured.

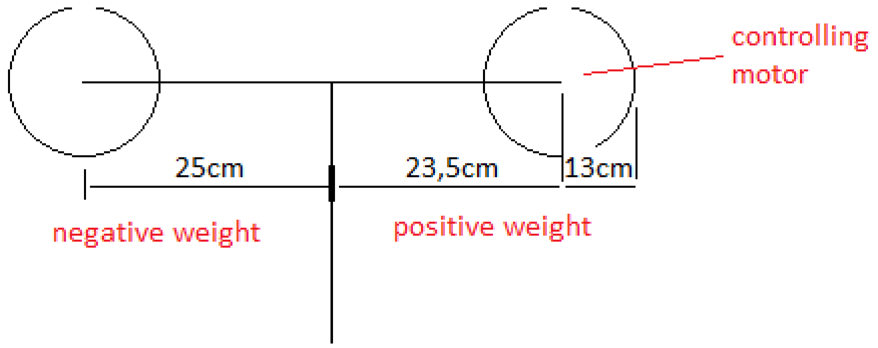

Figure 6. Weight signs and model dimensions.

Table 2. Propeller velocity, weight and control signal measurement.

u [-]	0.67	0.52	0.31	-0.32	-0.52	-0.74
propeller velocity [RPM]	2670	2250	1650	-1670	2300	-2800
mass [g]	44	29	14	-14	-29	-44

Polynomial coefficients of approximation were calculated basic of measurements.

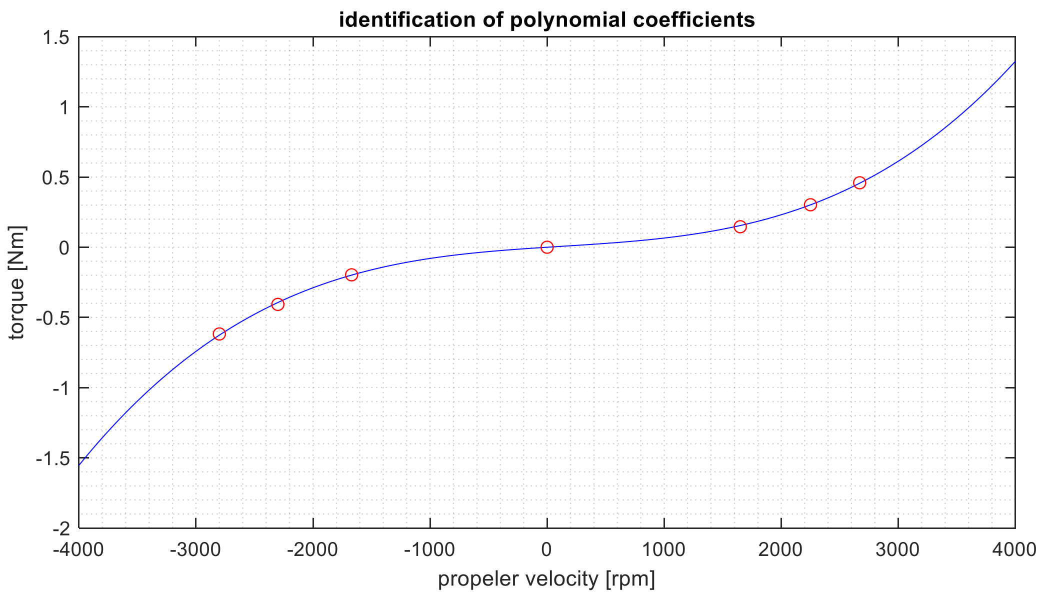

Figure 7. Torque from propeller velocity (calculated using weights and beam length).

Nonlinear model

The nonlinear model of our laboratory station can be described with following formula:

$$J\varepsilon = dF_{thrust} - Gsin(\varphi - \varphi_o) - D\omega$$

where:

$J -$ inertia of the plant
$\varepsilon -$ angular acceleration of the beam
$d -$ length from pivot point to the point where the throttle from the DC motor works=25.5cm
$F_{thrust} -$ throttle from DC motor
$G -$ moment of unbalance of the beam
$\varphi -$ beam rotation angle in a vertical plane
$\varphi_o -$ the unbalance angle depends on the system geometry equal to-58rad
$D -$ viscous friction coefficient
$\omega -$ angular velocity of the beam

The equation was transformed to:

$$0 = \Delta\ddot{\varphi} + \frac{D}{J}\Delta\dot{\varphi} + \frac{G}{J}\Delta\varphi$$
$$\frac{D}{J} = 2\vartheta\omega_n$$
$$\frac{G}{J} = {\omega_n}^2$$

where:

$$\omega_n - \text{neutral frequency of the plant}$$

Mass was hanged bringing balance to the object setting the beam in horizontal position. The obtained weight was equal to 27g. The G parameter was calculated using following formula:

$$Gsin(\varphi - \varphi_o) = g \cdot mass \cdot d_2$$

where:

$$d_2 - distance\ where\ mass\ was\ ganged\ from\ pivot\ point$$

The value of G equals to:

$$\mathrm{G} = 0.1122\frac{m^2kg}{s^2}$$

Next, the period was calculated by introducing an impulse to the system and calculating the difference between oscillation time:

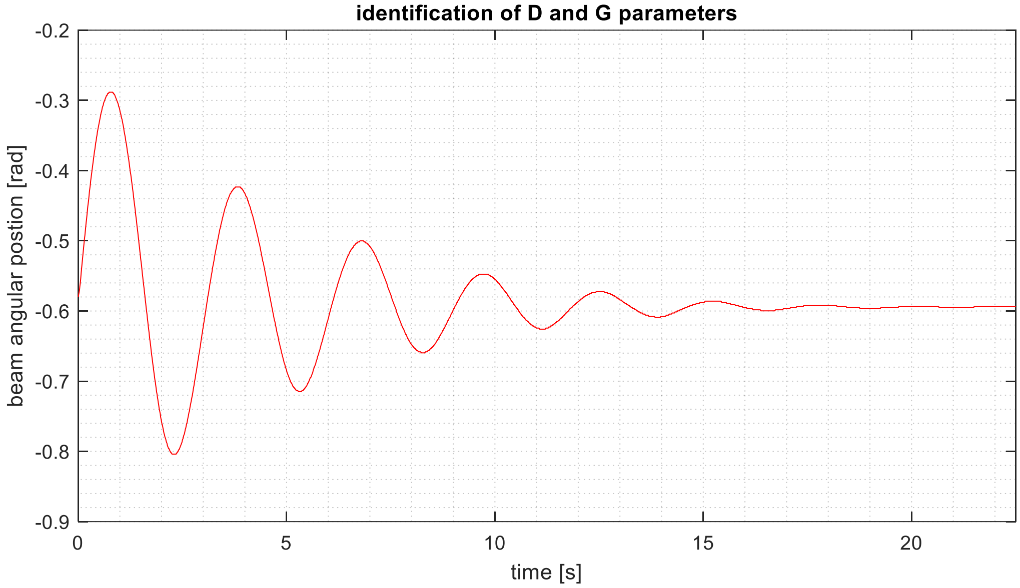

Figure 8. Dependence between beam angular position and dumping oscillations period.

Then, a *Logarithmic decrement method* for dumping was used to calculate the ϑ and ω_n.

That's how $\omega_d = \frac{2\pi}{T} = 2.0601 \text{ rad}$, was calculated which is *damped natural frequency*.

Next, damping ratio was counted:

$$\vartheta = \frac{1}{\sqrt{1 + (\frac{2\pi}{\ln(\frac{x0}{x1})})^2}} = 0.0978$$

And with this value the ω_n – *damped natural frequency* – could be calculated:

$$\omega_\mathrm{n} = \frac{\omega_\mathrm{d}}{\sqrt{1-\vartheta^2}} = 2.07 \text{ rad}$$

In case of harmonica dumbing oscillations the size of decrement and logarithmic decrement is constant in time, that is why any two amplitudes could have been used.

Now the Inertia of the plant could be counted using the equation:

$$J = \frac{G}{{\omega_\mathrm{n}}^2} = 0.0266 \text{ kg} * \text{m}^2$$

And finally, the *D* value could be calculated with equation:

$$D = 2\vartheta\omega_\mathrm{n}J = 0.0108\ kg * rad * m^2$$

Our nonlinear model is represented in Simulink with below figure:

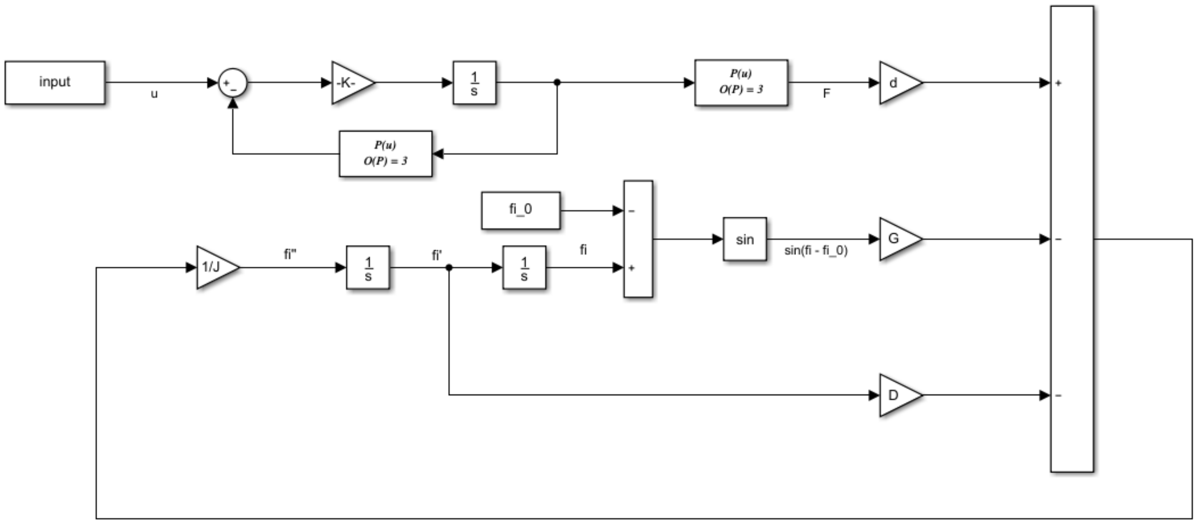

Figure 9. Nonlinear model built in Simulink.

Verification of nonlinear model

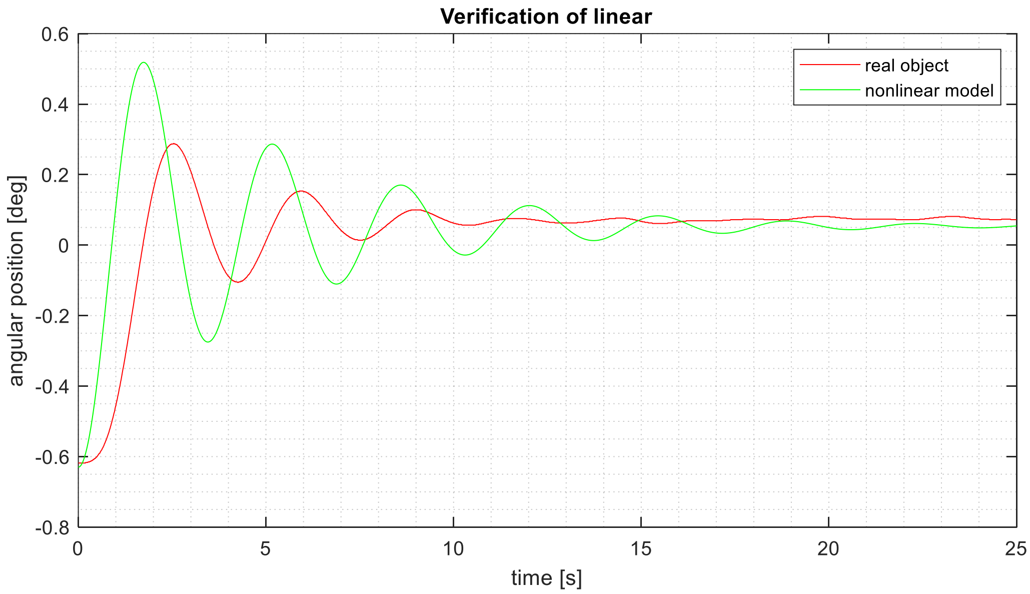

Figure 10. Verification of nonlinear model.

Linearization of the model in the equilibrium point

We started linearization by calculating state matrices:

x1 – beam angular position

x2 – beam angular velocity

x3 – propeller angular speed

Making derivatives equal to zero enables calculating stationary values in linearization point:

$$\begin{cases} \dot{x_1} = 0 \\ \dot{x_2} = 0 \\ \dot{x_3} = 0 \end{cases} \leftrightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \\ x_3 = 1860rpm \end{cases}$$

$$J\varepsilon = dF - Gsin(\varphi - \varphi_0) - D\omega$$

$$F = w_2(n)$$

where:

n – propeler angular speed

w_2 – polynomial aproximated to the experiment

$$J\varepsilon = dF(n) - Gsin(\varphi - \varphi_0) - D\omega$$

$$\begin{cases} \dot{\varphi} = \omega \\ \dot{\omega} = \frac{d}{J} w_2(n) - \frac{G}{J}\sin(\varphi - \varphi_0) - \frac{D}{J}\omega \\ \dot{n} = (-w_1(n) + u)K \end{cases}$$

$$x = \begin{cases} \varphi \\ \omega \\ n \end{cases}$$

With these equations of state, we can create A and B matrixes:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{G}{J}\sin(\varphi - \varphi_0) & -\frac{D}{J} & \frac{d}{J} w_2(n)' \\ 0 & 0 & -Kw_1(n)' \end{bmatrix}$$

where:

w_2 – polynomial of force dependence on rotational speed

w_1 – polynomial of speed dependence on the set signal

By substituting calculated values of all parameters, the following values of linearization have been achieved:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -3.4016 & -0.4045 & 0.0020 \\ 0 & 0 & -2.0739 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 7376.5 \end{bmatrix}, C = [0 \quad 0 \quad 1], D = [0]$$

Verification of linear model with experimental data

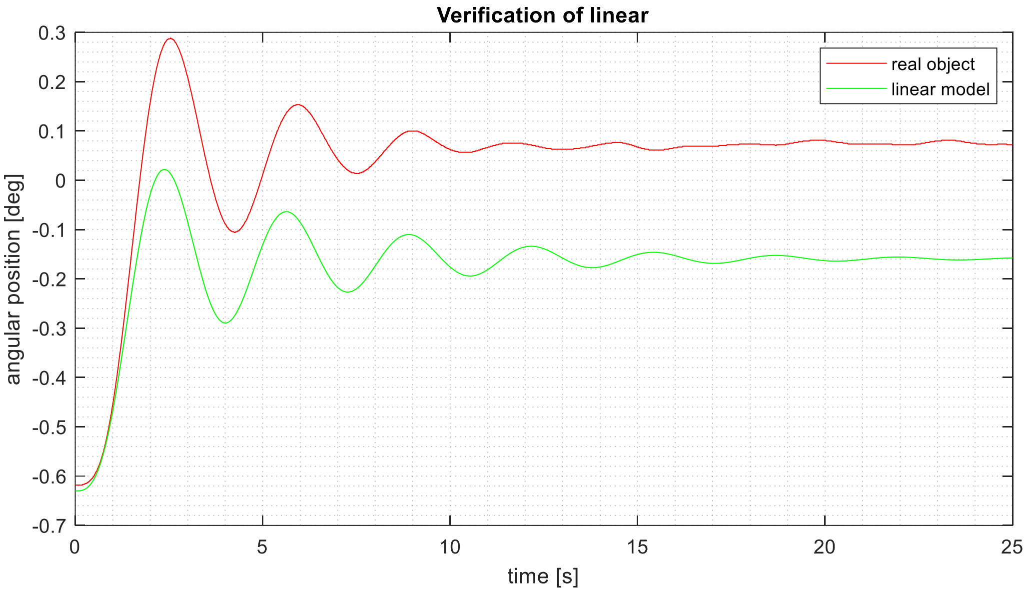

Figure 11. Verification of linear model.

Control algorithm selection and implementation

Because there is access to all state variables, it was decided, that the LQ control will be selected and implemented to the system.

Cost Function for the LQ controller:

$J = \int_0^{oo} (x^T Q x + u^T R u) dt$

Sometimes the cost function is given in the following form:

$$J = \int_0^{oo} (x^T Q x + u^T R u + \ 2x^T N u) dt$$

Which is used by MATLAB and frequency domain versions of LQR.

When we want to minimize the energy consumed for the control action, at the stage of minimizing the cost function, the value related to the control action is also considered.

$\Delta u(t) = -K \Delta x(t), \qquad K \epsilon R^{1x3}$

$$\Delta u(t) = u(t) - u^*$$

$$\Delta x(t) = A \Delta x(t) + B \Delta u(t)$$

$$\Delta x(t) = x(t) - x^*$$

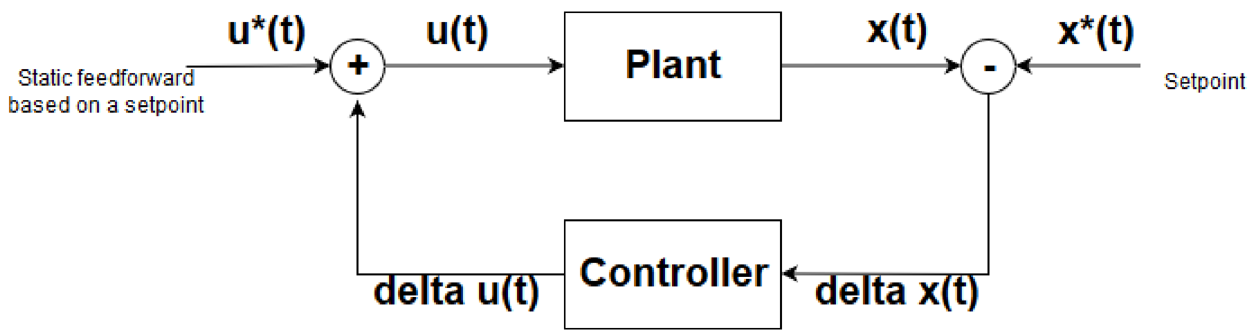

Figure 12. Model of control closed-loop control for nonlinear system.

Permissible deviations from the given balance point were provided by us to the script.

The LQR parameters calculations were made based on linear model.

The following costs for each variable were set:

$$q1 = \frac{1}{deg^2}$$
$$q2 = \frac{1}{(10deg)^2}$$
$$q3 = \frac{1}{600^2}$$

q1 – the deviation from the set point beam angular position
q2 – the deviation from the set point beam angular velocity
q3 – the deviation from the set point RPM of propeller

The cost matrix is equal to:

$$QQ = \begin{bmatrix} q1 & 0 & 0 \\ 0 & q2 & 0. \\ 0 & 0 & q3 \end{bmatrix}$$

The smallest permissible deviation is q1 because this value has the highest priority.

LQR coefficients were calculated using MATLAB function *lqr*.

The following gains for LQ controller have been achieved:

$$\begin{cases} \varphi_{gain} = 15.0324 \\ \omega_{gain} = 5.4780 \\ n_{gain} = 0.0015 \end{cases}$$

Interpolation of u and propeller velocity to linearize the equilibrium point for different beam angle values.

The interpolation was made using measurements below:

Table 3. Propeller velocity, beam position and control signal measurement.

u [-]	0.15	0.25	0.45	0.55
beam position [rad]	-0.49	-0.33	0.305	0.71
propeller velocity [RPM]	960	1425	2125	2450

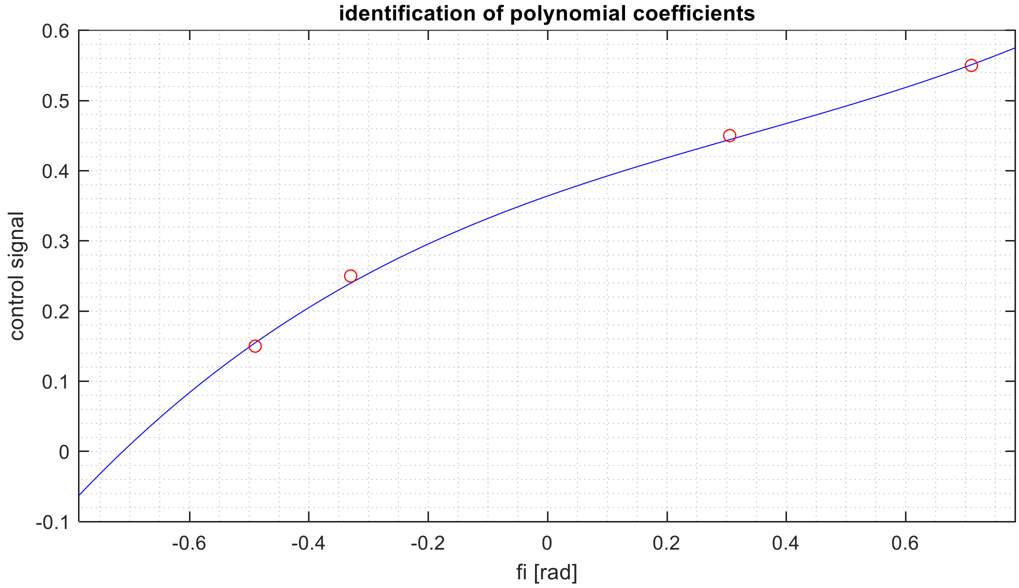

Figure 13. Control signal from set point of angular beam position.

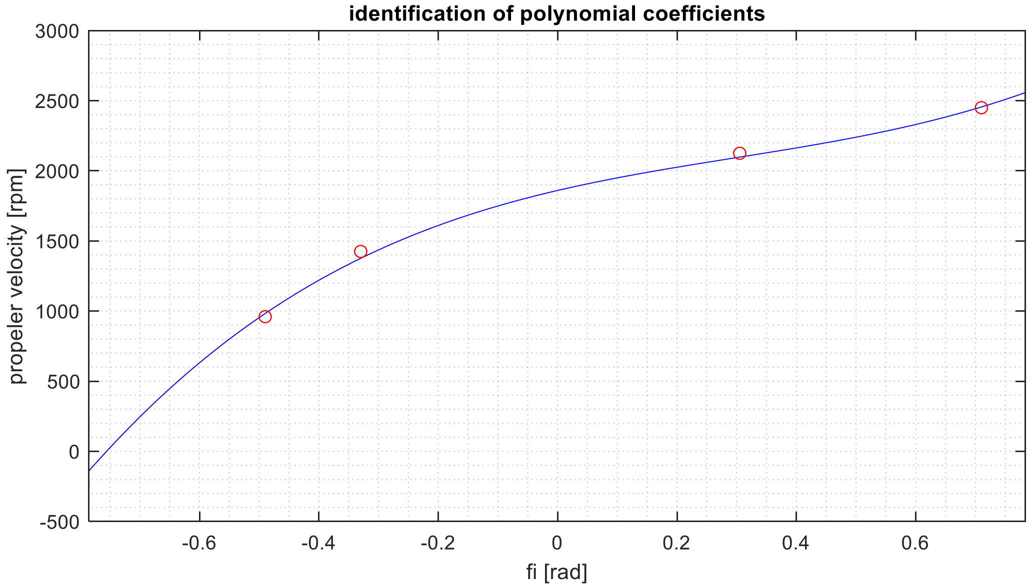

Figure 14. Propeller velocity from set point of angular beam position.

Verification of the calculated LQR parameters on real system.

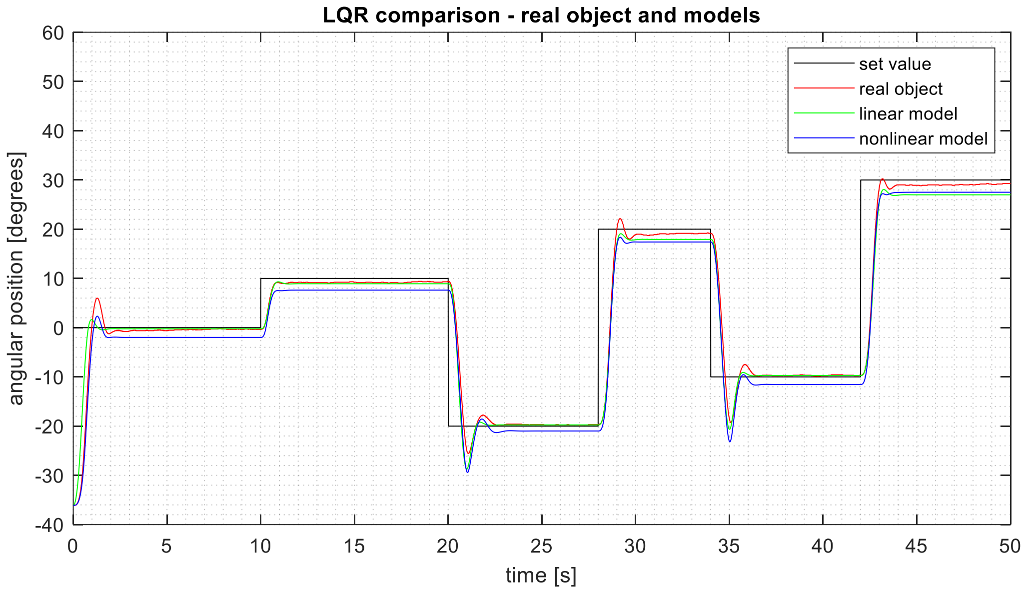

Figure 15. Step responses of the system and models with use of LQ controller.

Luenberger observer

The only one variable of laboratory station which is not measured directly is velocity of the beam. It has big quantization noise appearing from differences in sample period of encoder and program.

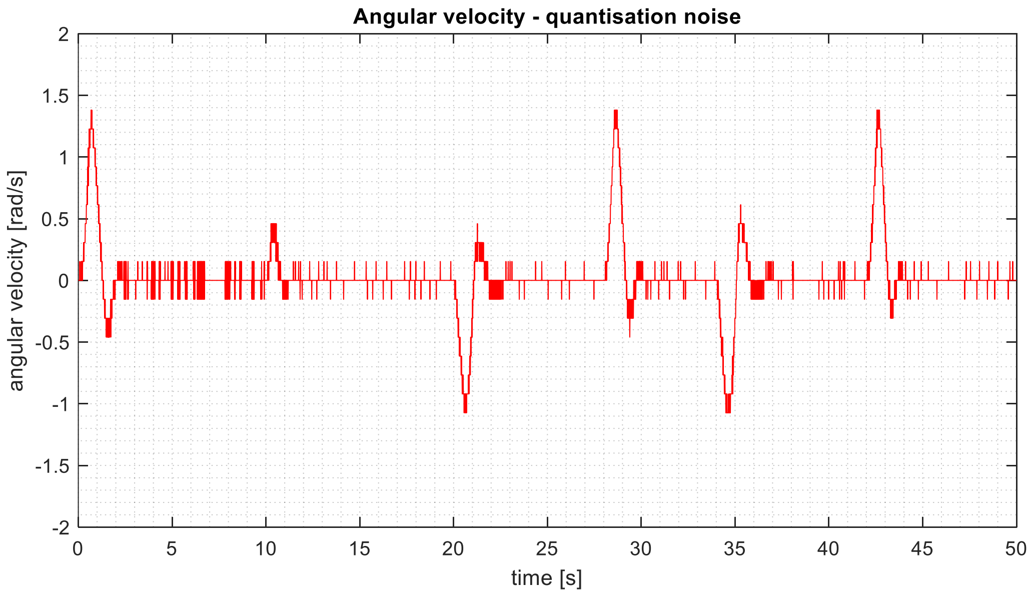

Figure 16. Quantization noise visualization.

We decided to implement Luenberger Observer to calculate beam velocity more efficiently, in order to obtain better behavior of the system which uses LQ controller.

The plant can be described using the following formula:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

Observer (state estimation) state-space execution:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t))$$
$$\hat{y}(t) = C\hat{x}(t)$$

Calculating difference between real and estimated state-space variables:

$$\dot{\tilde{x}}(t) = \dot{\hat{x}}(t) - \dot{x}(t) = A\hat{x}(t) + Bu(t) - Ax(t) - Bu(t) + L\big(y(t) - \hat{y}(t)\big) =$$
$$= A\big(\hat{x}(t) - x(t)\big) + L\big(y(t) - \hat{y}(t)\big) =$$
$$= A\big(\hat{x}(t) - x(t)\big) + L\big(Cx(t) - C\hat{x}(t)\big) =$$
$$= A\big(\hat{x}(t) - x(t)\big) - LC(\hat{x}(t) - x(t)) =$$
$$= (A - LC)\big(\hat{x}(t) - x(t)\big) = (A - LC)\tilde{x}$$

This equation is represented by the model in Simulink:

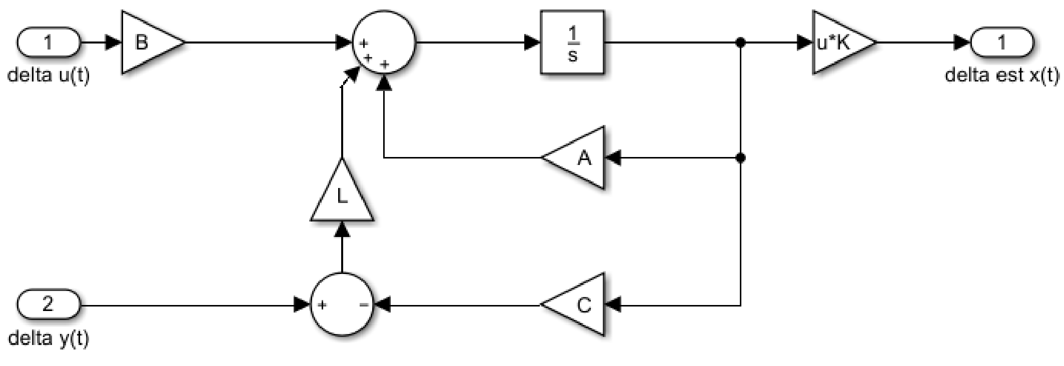

Figure 17. Luenberger Observer implementation in Simulink.

The C matrix uses those two variables, which are obtained directly using sensors and is equal to:

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

The (A-LC) matrix must be asymptotically stable, it means the eigenvalues of this matrix must be positioned on the left half-plane.

The received L matrix is equal to:

$$L = \begin{bmatrix} 1.6455 & 0 \\ -3.0166 & 0.0020 \\ 0 & -1.0639 \end{bmatrix}$$

The implementation of the system with LQ controller and Luenberger Observer looked like this:

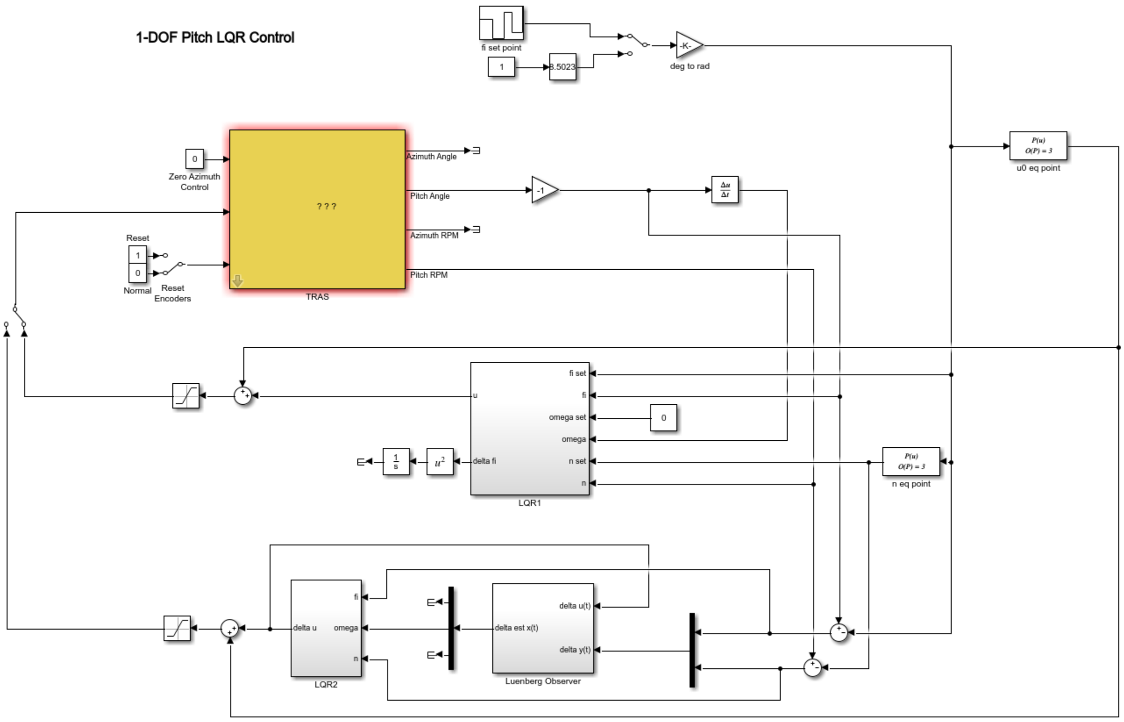

Figure 18. System implementation in Simulink.

Real-world experiments

The system has been tested for three different alternatives. In the first one, angular velocity of the beam has been obtained by calculating the derivative of angular position of the beam (Luenberger Observer was not used). In the second, Luenberger Observer has been used to estimate angular velocity of the beam, whereas in the third, all three state variables have been estimated.

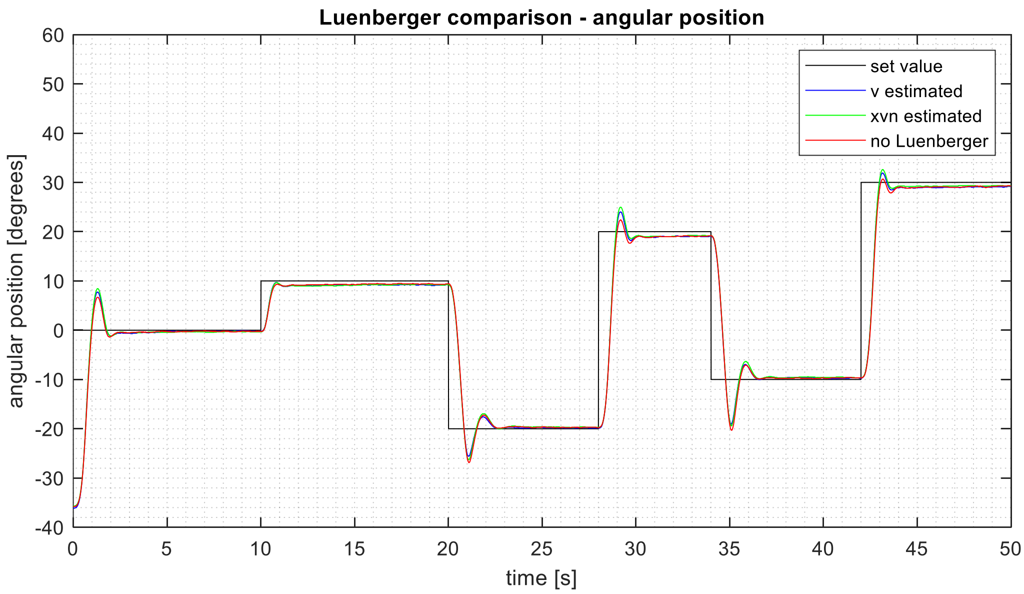

Figure 19. Step responses for three alternatives.

To compare those three options, we calculated an integral error of beam angular position deviation. It can be seen, that the error is the smallest when Luenberger Observer for all three state variables is used and the biggest when Luenberger Observer is not used, but the differences are not huge.

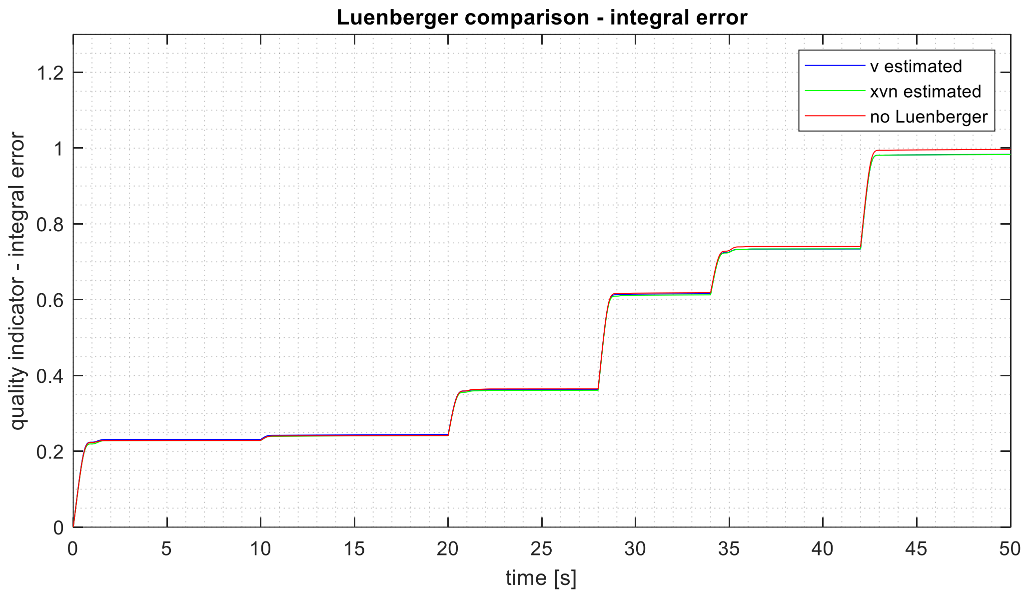

Figure 20. Integral error comparison for three alternatives.

Finally, the beam angular velocity has been compared. It can be noticed, that the quantization noise disappears at the cost of huge picks at the moment, when set point value is being changed.

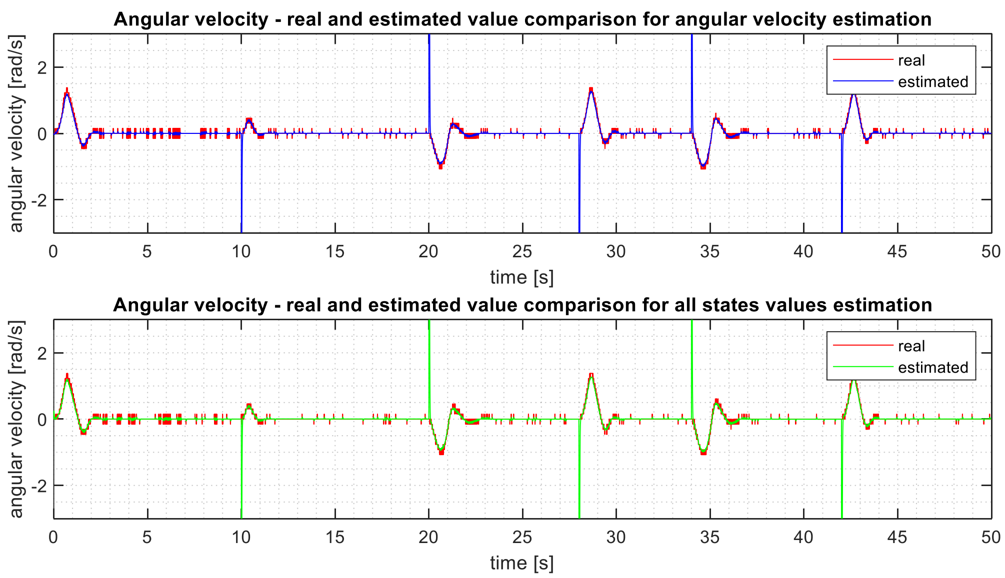

Figure 20. Angular velocity of the beam comparison.

Summary

All things considered, milestones of the project have been reached. Starting with sensors calibration and scaling, through formulating the mathematical model of the plant, identifying model parameters, linearizing the model, selecting and implementing the control algorithm, ending with driving real-world experiments. What has been done additionally is implementing and testing the Luenberger Observer.

The LQ controller (*linear-quadratic regulator)* made the plant controllable with very satisfactory result. Plant was reaching it's set angle with high dynamics in both ways maintaining high accuracy in all working angles (not only linearized one).

Additionally, the Luenberger Observer was implemented to eliminate the quantization error of beam angular velocity coming from low resolution of encoder. The results coming from estimating all state space variables (beam position, beam velocity and propeller rotational speed) gave us the best results, making the integral error the smallest. Although, there was not a big difference compared to the clear LQ controller (where the Luenberger Observer has not been used).

The set of experiments and measurements was done to make the regulator as good as possible with given equipment and the verification and comparison tests were done.

For LQ controller it's needed to have accurate mathematical model, the more precise it is, the better the regulation will be – though it's really hard to create exact mathematical model for more complex plants. For our plant, where model was created only from a few components and the steering was supposed to consider position and angular velocity of the beam (with variable weights in regulator) the LQ controller was meeting all requirements controller.

Additional development of regulator could be done by adding the integral part to the LQR creating the LQRI controller, resulting in removing steady state error which is present in normal LQ controller. To develop the project, predictive controller can be implemented, or the system can be expanded to two degree of freedom model.