

Get ready for takeoff

Report

Eunju Moon¹, Vedrana Stantić², Raffaele Masotti³, Petra Bucić⁴, Josip Kir Hromatko⁵

Team: Gravity Busters

Summer School of Science S3 2018

^{1,*}ISL, Luxembourg, ² V. Gimnazija, Zagreb, ³ Calabrese Levi, Italy, ⁴ UniZG, Croatia, ⁵ UniZG, Croatia

Abstract—This report describes a project that was done as a part of the Summer School of Science (S3) 2018 in Požega, Croatia. In the project, a helicopter arm model was built using inexpensive electrical and mechanical components, as well as open-source software. Furthermore, a mathematical model of the system was derived in order to design a PSD position controller for the arm. Finally, several experiments were conducted to find the optimal values of the controller and system variables.

Index Terms—Control system design, 1DOF helicopter, PSD controller

I. INTRODUCTION

In this world we all function on the principles of the two main types of controls: feedback and feedforward control. A control system directs and adjusts the behaviour of a machine. Feedforward controller is based on a predetermined recipe like a robot which is ordered to move 5 steps. Feedback controller bases its directions on measurements of the system state. For example, we use our eyes as sensors every day to assure that we don't bump on the wall.

In a modern machine, there is a plant which is a combination of a process and an actuator. The process is the part which is being controlled and the actuator is the part that changes the behaviour of the process. For instance, in an air conditioner, the process is temperature and the actuator is the air cooling system in a temperature control system. Besides an actuator, there is a controller which is a device that executes the control. It directs the actuator based on the input. This control system usually uses a feedback control or a feedforward control as shown in Figure I. In a feedback control system, there is a sensor in order to sense the state of the plant and send the information to the controller.

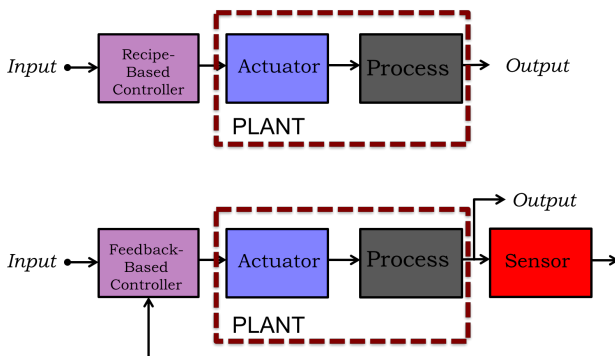


Fig. 1. Feedforward control and Feedback control [1]

The goal of this project is to make a helicopter arm stay in a certain position without being moved by disturbance.

Feedback control is used in the helicopter system to monitor the state of the helicopter and easily adapt even though there is a disturbance. Linear Difference Equations (LDEs) are used for developing a model and designing a feedback control for the helicopter. LDE is an equation that relates one element of a sequence to previous elements. Since LDE is based on a sequence, it depends on a discrete value. In the model for the helicopter, the LDE depends on time which increases in discrete steps.

The helicopter arm is attached to a potentiometer which is a sensor and the Arduino is connected to the helicopter to be used as a controller. The Arduino is also connected to a computer for manipulating and monitoring values such as control variables, direct motor speed and angle of the helicopter arm. Then, several experiments were done in order to find the optimal value of control variables in our LDEs.

Section II shows the equipment that was used in the project. This includes both hardware and software parts. Section III explains the theory used to adjust the position of the helicopter arm. This will include a further explanation about LDEs and their natural frequencies and PSD control which is linked to the methodology of our experiments. Lastly, section IV introduces experiments and their results. In the experiments, the values of control variables were changed and recorded in order to find their optimum values for a stable helicopter arm.

II. EQUIPMENT

This section is dedicated to describing the hardware and software elements of this project. The two are intertwined and one cannot be explained thoroughly without the other.

A. Hardware

A basic knowledge of the most common electrical components is crucial before starting any engineering project. It consists of what a component is and what it is used for, as well as things that cannot be overlooked. The helicopter that was built during this 10-day period consists of an Arduino board and a breadboard which connects certain elements. These elements are simple to use and very common, therefore often used on a daily basis.

1) *The resistor, the capacitor and the transistor:* These three components are often used in electrical engineering and are for that reason irreplaceable and cannot be forgotten. These are also the three most known electrical components without which many things wouldn't be able to function.

One of the most commonly used components is a resistor. A resistor is a component that limits the amount

of current which passes through it. Consequently, using resistors prevents an overwhelming amount of current to damage other electrical components. In addition to that, the resistor can be a part of something called a low-pass filter, more commonly known as an RC filter. It's called a low-pass filter because it only allows lower frequencies to pass through it. It consists of a resistor (the 'R' part) and a capacitor (the 'C' part). The capacitor is an electrical component that's made of two metal plates with an insulator in between. The capacitor is also one of the more common components.

Beside the capacitor and the resistor, another frequently used element is the transistor which is a three pin component, unlike the ones that have been mentioned thus far. There are two main types of transistors, the NPN [2] and the PNP. The letter 'p' means positive, while the letter 'n' means negative. The difference in these transistors is that one has two negative pins and one positive one, whereas the other one has two positive pins and one negative pin.

2) *The motor, potentiometer and propeller:* Other than the resistor, the capacitor and the transistor, the next three components that were used could be considered the main part of hardware in this project. A helicopter cannot be treated as a helicopter without a propeller and a motor that powers it.

A motor is an electrical machine that converts electrical energy into mechanical energy, therefore putting the propellers in motion. It is an easy to use component whose polarity can be reversed simply by switching the '-' wire with the '+' wire. In this instance, the aforementioned fact is extremely helpful because it allows one to control whether the propeller starts flying up or pushing down.

The potentiometer is a three pin adjustable resistor which can be used in the place of a position sensor. Attaching the propeller arm to the potentiometer allows complete control over the angle at which the arm of the propeller is at any given moment. This is practical information because it helps determine whether the current position of the helicopter is in any way counterproductive (e.g. wasting power while not moving in the desired direction). The potentiometer is attached directly to the micro-controller, or in this case, the Arduino. The farthest left pin is connected to the ground (GND), the middle pin to A0 while the right pin is connected to five volts (5V).

The propeller is a completely external component that can hardly be called electrical as the only connection it has with electricity is the motor which simply uses the converted mechanical energy to turn the propellers. The turning of the propellers is what causes the arm to raise and the position to change.

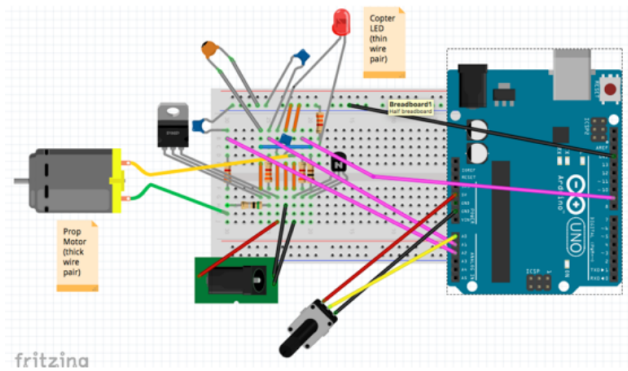


Fig. 2. A Fritzing schematic of the electronic circuit

3) *The tools:* Apart from the preceding components, some tools were used to evaluate the quality of the components, to create a disturbance, or simply to better the connections between components.

Out of all the tools that were used, the multimeter was by far the most valuable one. A multimeter is a tool used to measure current, voltage and resistance. It is used to establish whether components work, as well as how properly they are working. This can be very useful in case the verification of a component is required.

A more advanced version of the multimeter is an instrument called the oscilloscope. The oscilloscope functions the same way that the potentiometer does; it measures voltage and current. The oscilloscope can be considered advanced simply because it has the option to directly plot the current that it is measuring in real-time.

The last two devices that were used in this project were used very briefly and for completely different purposes. First of the two is the soldering iron, which is an easily warmed metal pole that is used to melt the solder and form a strong bond between electrical components. The second tool was a hairdryer that we used to create a disturbance to the propellers, as described later in the experiments.

B. Software

As is the case with hardware, before starting a project that has a software aspect, one must know the basic elements of programming or at the very least what programming is. A Python tutorial that preceded the project itself assisted in reducing the amount of confusion when first introduced to programming. The tutorial dealt with different kinds of loops as well as how to work with CSV files which is a basic element that is used globally, including this project.

CSV or comma-separated files are exactly what their name states: a text file in which commas are used to separate values. The data is most often imported from spreadsheets or database tables. Their most common use is to store data which can later be analyzed.

Plotting is a method used to envision the data that one has been presented with. That way the data becomes more accessible and it is easier to read and understand.

III. THEORY

Above all, designing a model is needed for constructing a suitable control for the helicopter. To understand and manipulate the model, it is important to understand the

concepts behind the model. In this section, the basis of our model is explained.

A. Linear Difference Equations (LDEs)

The helicopter is modelled with LDEs. As it is mentioned in the introduction, LDE relates an element of a sequence to previous element(s) of the sequence. A First-order LDE relates an element in a sequence to a previous element. First-order LDE and its general solution are given by:

$$\text{General LDE: } y[n] = \lambda[n - 1] \text{ and } y[0] = a$$

$$\text{General Solution: } y[n] = a * \lambda^n$$

where a is a constant and λ is the natural frequency that will be explained further in the later section. The value $y[0]$ is the initial value of the LDE. Second-order LDE and its general solution are:

$$\text{General LDE: } y[n] = a_1[n - 1] + a_2[n - 2]$$

$$y[0] = b_1, y[1] = b_2$$

$$\text{General Solution: } y[n] = c_1 * \lambda_1^n + c_2 * \lambda_2^n$$

where a_1, a_2, b_1, b_2, c_1 and c_2 are constants and λ_1 and λ_2 are natural frequencies. Generally, an k th LDE has k number of natural frequencies (λ). Additionally, an LDE that has more than one natural frequency can have complex numbers as their natural frequencies.

B. Natural Frequency

The natural frequency (λ) of a LDE defines the behavior of the LDE. LDE is stable (which means that as n gets to infinity, the value of $y[n]$ converges to a certain value) when $|\lambda| \leq 1$, and if λ is a negative real number, the LDE will oscillate. If $|\lambda| > 1$, the LDE is unstable which indicates that the LDE will grow unbounded. If λ is a complex number ($\lambda = a + bj$), it will oscillate. This behavior of λ is well illustrated in Figure 3. When λ is a complex number, the stableness of LDE is determined by its magnitude, M ($M = \sqrt{a^2 + b^2} = |\lambda|$). If $M \leq 1$, the LDE is stable and if $M > 1$ the LDE is unstable.

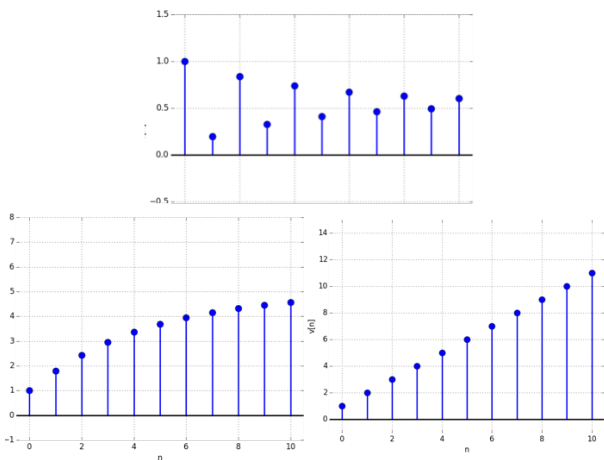


Fig. 3. (From top to bottom right) LDE with $|\lambda| \leq 1$ and $\lambda < 0$, LDE with $|\lambda| \leq 1$ and $\lambda > 0$ and, LDE with $\lambda > 1$

When LDE more than one natural frequencies, the behaviour of the LDE is defined by the λ with the highest

$|\lambda|$ in LDE. It is because the λ with the highest $|\lambda|$ will grow the fastest which results in dominating the behaviour of the LDE.

In the model for the helicopter, $y[n]$ represents the position or arm angle of the helicopter. Therefore, the natural frequency of the model should be stable or else the value of $y[n]$ will get very large and eventually be out of control. It is also important in the model to adapt quickly to an error or noise in the environment. This indicates that the natural frequency should have a suitable value so that $y[n]$ becomes stable as quickly as possible.

C. Non-linear to Linear

The LDEs used in the model are linear. A linear equation is an equation that only uses the sum of elements and multiplication with constants. It is preferable to use linear models due to their simplicity compared to non-linear equations.

However, the real-life errors are mostly non-linear. For example, the gravity that is acting against rotation force of the propeller is non-linear. The relationship between the total force on the propeller, the rotation force of the propeller and the gravity is:

$$f_{total} = f_{propeller} - f_{gravity} \cos(\theta)$$

As you can see, the gravity that affects the force of the propeller is represented as a cosine function which is not linear.

In order to solve this problem, the values in the equation are estimated and written as a linear equation. In case of the gravity force acting against the propeller force, it is possible to make a linearized approximation as shown in Figure 4.

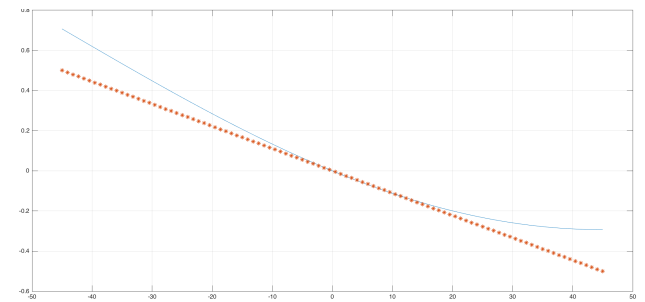


Fig. 4. Linearized approximation of gravity force

D. PSD control

PSD control stands for the three kinds of feedback controllers - Proportional, Sum and Delta Controller. The modelled version of PSD control is represented by Figure 5.

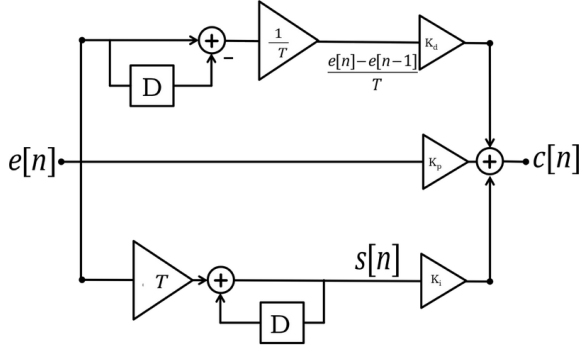


Fig. 5. PSD Control block diagram

A proportional feedback multiplies a certain constant called proportional constant (K_p) to the error, which is the difference between the desired position and the actual position. A delta feedback is related to a rate of change of the error. K_d , a derivative constant, is multiplied by a rate of change of the error in a certain time. Lastly, a summation feedback uses a value created by multiplying summation constant (K_i) to the sum of the accumulated errors. By using the PSD control, we aim for producing a model that can adjust to an error quickly by using the PSD control.

To achieve a full control of the helicopter, it is crucial to use all three of the feedbacks in our model, although this makes the model more complicated. For example, if only proportional feedback is used, large K_p is needed to make the helicopter react to an error quickly. However, this will not only consume a high current but also cause a large fan speed error.

The case where only K_d is used also has a problem. A large value of K_d is needed for quick adaptation to the error. On the other hand, a large value of K_d magnifies noise which increases the error.

Only using summation feedback will also cause a problem. If the propeller starts with the high error value, the propeller needs to exceed the desired value and make oscillations in order to be stable. In case of using proportional feedback and derivative feedback will make a so-called steady-state error. When the error is very close to 0, large K_p value is needed for sustaining the desired value. Nonetheless, large K_p value will be unstable without large K_d value, so, the system will end up with magnified noise due to large K_d value. Fortunately, using all three kinds of feedback controllers can solve these problems which result in the better model. Lastly, other than K_d , K_p and K_i values, the direct component is used in the model. This is an input value. It is possible to reduce initial error values and approach to the desired value more quickly by using the direct component.

E. Our Model

In our model, proportional plus delta (PD) control is used. Using proportional plus delta control, the motor current is described as:

$$m[n] = K_p(\theta_d[n] - \theta_a[n]) + direct + K_d \left(\frac{\theta_d[n] - \theta_a[n] - \theta_d[n-3] + \theta_a[n-3]}{3T} \right)$$

The relation between distance and velocity; velocity and acceleration; and acceleration and motor current is given by:

$$\begin{aligned} \theta_a[n] - \theta_a[n-1] &\approx T\omega_a[n-1] \\ \omega_a[n] - \omega_a[n-1] &\approx T\alpha_a[n-1] \\ \alpha_a[n] &= \gamma m[n] \end{aligned}$$

where γ is a value that makes the relationship between motor current and the acceleration. Hence, the relation between distance and motor current is given by:

$$\begin{aligned} \theta_a[n] - 2\theta_a[n-1] + \theta_a[n-2] &\approx T^2\alpha_a[n-2] \\ \theta_a[n] - 2\theta_a[n-1] + \theta_a[n-2] &\approx T^2\gamma m[n-2] \end{aligned}$$

Using the previous equation about $m[n]$, the approximated relation between K_p and K_d is given by:

$$\begin{aligned} \theta_a[n] - 2\theta_a[n-1] + \theta_a[n-2] &= T^2(K_p(\theta_d[n] - \theta_a[n]) + \\ &+ direct + K_d \left(\frac{\theta_d[n] - \theta_a[n] - \theta_d[n-3] + \theta_a[n-3]}{3T} \right)) \end{aligned}$$

If input is 0 ($\theta_d[n] = 0$, $direct = 0$), it is possible to rewrite the equation above as an LDE for $d[n]$:

$$\begin{aligned} \theta_a[n] - 2\theta_a[n-1] + (1 + \gamma T^2 K_p + \frac{\gamma * T}{3} K_d)\theta_a[n-2] + \\ - \frac{\gamma * T}{3} K_d \theta_a[n-5] = 0 \end{aligned}$$

This is a 5th-Order LDE - it is uneasy to be solved by hand, thus, the optimal values of γ , K_p and K_d can be found by using Python and simulations.

IV. EXPERIMENTS

In this section, the experiments that were done to test the derived models are going to be described.

A built physical model of a helicopter was used to test the performance of a controller designed according to a mathematical model. Mathematical models were initially simple, i.e. without gravity or friction, but their complexity increased in every experiment. The final goal was to control the position of the propeller arm even with disturbances. In order to achieve that, the first thing that was modelled was the propeller's speed and from there a formula for a position was derived. Both feedforward and feedback control were used, including direct, proportional, PS and PSD control.

An Arduino micro-controller was used for controlling the propeller. The interaction with the micro-controller is achieved using a GUI (Graphical User Interface) approachable in a web-browser through its "server" address `localhost:3000`. This platform permits the user to control the arm angle by adjusting the desired value of the control parameters, i.e. `direct`, K_p , K_d . The desired angle can be automatically alternated, from positive to the same negative value, every 5 seconds using a function called `alternate switch`.

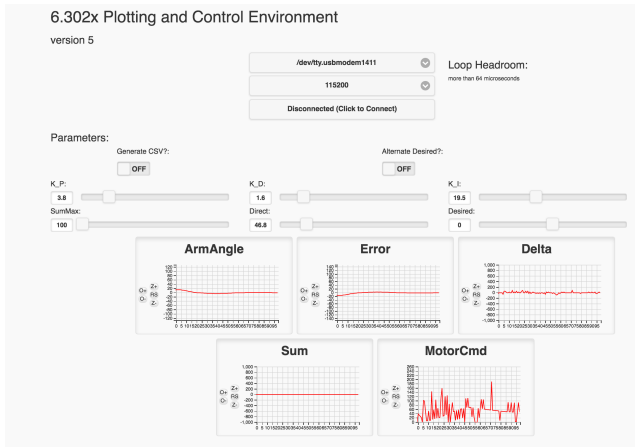


Fig. 6. GUI for the experiments

The values of error, angle, motor command (C_{cmd}) and delta are displayed in real-time. It is also possible to record the data in a .csv file (e.g. data.csv). This is really useful because the data stored in the file can be accessed and analyzed afterwards.

Figure 6 shows the GUI used for the experiments, which contains several variables the user can change. The $ArmAngle$ is the angle that the propeller makes with respect to the 0 degrees. The $Direct$ term is proportional to the amount of power that is directly applied to the propeller to increase his acceleration. The $Desired$ variable is the speed or the angle that is wanted and the difference between the desired and the measured forms the $Error$.

The K_p term is a proportional constant that is only active when there is an error. It says how fast the system is going to react to the error; the higher K_p is, the more aggressive a system is going to be. Too high values of K_p can cause problems such as oscillations. K_d is a derivative constant for the derivative term of the equation and K_i is an integral constant.

All the graphs are line graphs. The error graph shows the difference between the desired and real value, the angle graph shows the angle between the propeller and the horizon. The $MotorCmd$ graph shows the command from the Arduino (i.e. the direct plus the K_p multiplied by the error plus delta) in a range from 0 to 255. The $Delta$ term comes from the multiplication of K_d by the change in the error.

The value of the $MotorCmd$ determines the PWM that is the real control of the speed. It determines how long the electrical voltage is going to be applied between two-time samples. If it is set to 255, there is going to be high voltage all the time (5V), and if it is 0, there is going to be no voltage (0V) and therefore no electrical current.

A. First experiment

Before the first experiment, it was important to check that the model is working properly. Also, for this experiment, the plus and the minus wires that connected the propeller to the power supply had to be switched so that it would fly down and not up.

To control the model in this case, the feedforward control was used. It directly controls the amount of energy that flows through the circuit and therefore controls the speed of the propeller. The only value that was changed was the

$Direct$ component, in order to increase the speed of the propeller. The change in the propeller speed is shown in Figure 7.

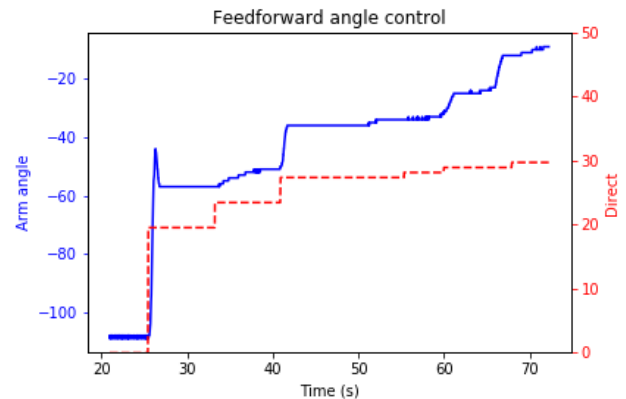


Fig. 7. Direct input and measured propeller speed

B. Second experiment

The aim of the second experiment was to position the arm of the propeller at a specific angle, in this case, the -45 degrees and +45 degrees, where 0 degrees is on the horizontal. In order to do this experiment, feedback control was used.

First, the desired angle was set to -45 degrees. Then, the right values for the direct component and the K_p were found so the propeller would be able to stay in the desired position. After that, the desired angle was changed to +45 degrees. $Direct$ and K_p values were changed, but the propeller was not able to settle at +45 degrees. When the K_p value increases too much, the system becomes unstable. This can be seen in Figure 8.

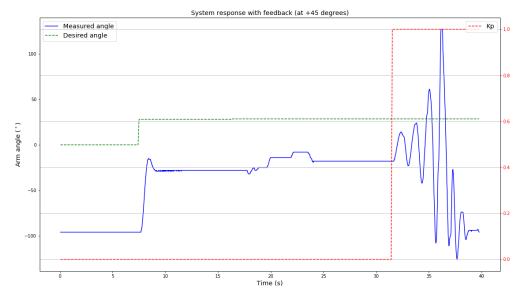


Fig. 8. The arm angle in Experiment 2

The conclusion of the experiment is that the propeller can be stable at -45 degrees with the right control values, but the same thing cannot be achieved for +45 degrees.

C. Third experiment

The goal of this experiment was to test the effect of adding the $delta$ term to the controller. In the beginning, the desired angle was set to 0 and the direct component was increased until the propeller reached -20 degrees. Then, the desired angle was changed to +8 degrees and the $alternate$ switch was turned on.

After that, K_d was set to 1 and K_p was increased slowly to see the effects that it had on the system. At K_p around

5, the propeller started to oscillate, and at $K_p = 7$, the system becomes unstable. The results are shown in 9 and 10. It can be seen that with PD control there is a steady-state error and the arm goes only from -15 degrees to 0 degrees.

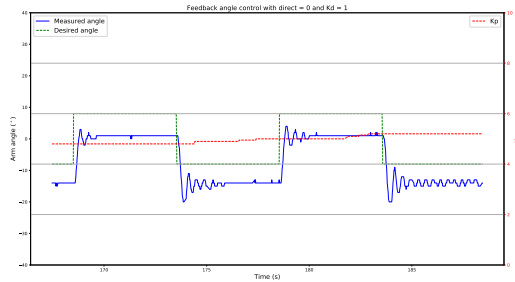


Fig. 9. The arm angle in Experiment 3

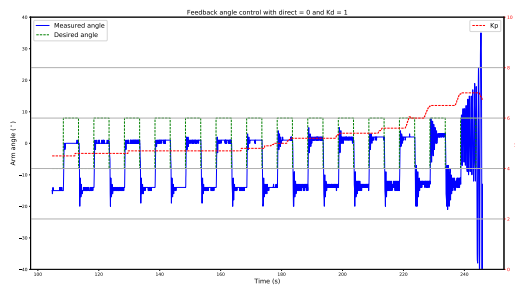


Fig. 10. The arm angle in Experiment 3

D. Fourth experiment

In order to resolve the steady-state error, PSD control had to be used. In this case, the values for the three variables had to be set, and from the previous experiment, it was known that for certain combinations of constants the system will become unstable.

Trying to find these coefficients only experimentally can be hard and dangerous for the system. For that reason, a simulation of the model was built using Matlab and Simulink. In that way, different values of the control parameters could be tested without damaging the model.

V. CONCLUSION

A. Result

The helicopter arm, which is the main part of the project, was built on a breadboard with many electrical components. This was connected to Arduino and the Arduino was connected to a computer in order to make a GUI. By using this GUI, the constant variables, errors and motor speeds were monitored and altered.

PD model with the direct component is made by applying mathematical and theoretical approach. The PD model with direct components has a low range of fluctuation and adapting fast to a change in the desired value.

Experiments show the improvements and new problems as the control model is developed. In Experiment 1, proportional gain (K_p) and direct component were used. In the

experiment, it was possible to find out that too low value of proportional gain caused a slow transition of speed as the desired speed is changed. In contrast, a too high value of proportional gain caused fluctuations in speed which made the system unstable. In Experiment 2, the goal was to position the helicopter arm in 45. It was much harder to make 45 above ground than 45 below ground due to the change in gravity force acting against the propeller force as the degree changes. The linear model of the total force was built to understand the behaviour of the helicopter arm. In Experiment 3, proportional gain (K_p), delta gain (K_d) and direct components were added together. The aim was to test and find out the most suitable values of the proportional gain and delta gain. Unfortunately, due to issues in electrical devices, this experiment was not completed.

In conclusion, the PD with the direct component model for a helicopter arm was successfully completed by the end of the project.

B. Reflection on the project

It was possible to achieve one of the goals of the project: to make a suitable control model for the helicopter. However, due to technical problems that occurred in the helicopter, testing the PD model on real helicopter arm was not completed.

There were many problems occurred in the project. Sometimes, it was really hard to find out the problem. For example, the jack for the connection between the power and the helicopter was not well-linked. At that time, the source of the problem could not be detected, and it took about a day to find out that the jack had a problem.

A model of PD control with direct component was successfully completed in the project. It will be interesting to further improve this model by completing a PSD control for the helicopter and compare the PD model and PSD model. Additionally, putting a Bluetooth system on the helicopter and controlling the helicopter without cables was suggested during the project. However, this suggestion could not be done due to the problem in the helicopter arm. It will be also interesting to make such an improvement on the helicopter.

ACKNOWLEDGEMENT

This work has been supported by the Society for Out-of-Frame Education through the project Summer School of Science (<http://drustvo-evo.hr/s3/index.html>) organized by Nikolina Šoštarić and Dora Grbavac.

REFERENCES

- [1] J. White, J. Steinmeyer, "Introduction to Control System Design", MIT. <https://www.edx.org/course/introduction-control-system-design-first-mitx-6-302-0x> (15.8.2018.)
- [2] Transistor MJE15028 datasheet. <https://www.onsemi.com/pub/Collateral/MJE15028-D.PDF> (15.8.2018.)