

# Modeliranje i sinteza regulatora daljinski upravljane letjelice

---

**Ključarić, Martin**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:766567>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-18**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Martin Ključarić**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentori:

Izv. prof. dr. sc. Vladimir Milić, mag. ing.

Student:

Martin Ključarić

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru izv. prof. dr. sc. Vladimiru Miliću radi važnog doprinosa u obliku truda, entuzijazma i redovite dostupnosti – faktora koji su uvelike olakšali izvedbu rada. Nadalje zahvaljujem se dr. sc. Matiji Krzнару te asistentu mag.ing. Branimiru Čaranu koji su više puta ponudili pomoć vezanu za posudbu nekih važnih elektroničkih komponenata za realizaciju rada te razvojnom timu iza YALMP te SDPT3 programskih paketa koji su uvelike olakšali matematičku stranu rješavanja rada. Konačno zahvaljujem se dobrom prijatelju Kristijanu Ključariću čija je ekspertiza u polju RC modelarstva omogućila ispitivanje rada u stvarnosti.

Martin Ključarić



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 24 – 06 / 1	
Ur.broj: 15 – 24 –	

## ZAVRŠNI ZADATAK

Student: **Martin Ključarić** JMBAG: **0035239671**

Naslov rada na hrvatskom jeziku: **Modeliranje i sinteza regulatora daljinski upravljane letjelice**

Naslov rada na engleskom jeziku: **Modelling and controller synthesis of a remotely controlled aircraft**

Opis zadatka:

Tema ovog završnog rada je računalno modeliranje daljinski upravljane bespilotne letjelice (zrakoplova) i sinteza PID regulatora za automatsko upravljanje oko poprečne osi. U dijelu modeliranja neće se duboko ulaziti u područje zrakoplovnog inženjerstva, odnosno neće se raditi detaljna analiza dinamike leta zrakoplova niti detaljna analiza aerodinamičkih značajki. Osnovna tema je sinteza PID regulatora primjenom linearnih matričnih nejednadžbi i njegova implementacija na mikrokontroler u zatvorenom krugu s ostalim elementima predviđenima za ugradnju u letjelicu s ciljem automatskog upravljanja letjelicom oko poprečne osi. Pri tome će se kut otklona kormila visine promatrati kao ulazna upravljačka varijabla, dok će izlazna varijabla biti kut penjanja zrakoplova. Rad će u konačnici obuhvatiti osnovne discipline iz područja mehatronike, a to su računalno modeliranje, automatsko upravljanje, odabir aktuatora i senzora uz pripadajuće elektroničko sklopovlje i programiranje.

U radu je potrebno:

1. Izraditi računalni 3D model letjelice (zrakoplova) pomoću programskog paketa Solidworks. Za tako izrađeni model, primjenom programskog alata XFLR5, napraviti procjenu aerodinamičkih značajki letjelice te odrediti matrice lineariziranog dinamičkog modela u prostoru stanja.
2. Opisati sve komponente predviđene za ugradnju značajne sa stanovišta automatskog upravljanja letjelicom, kao što su mikrokontroler, senzori, elektromotori, itd.
3. Provesti teorijska razmatranja o sintezi PID regulatora prema kvadratnom kriteriju optimalnosti primjenom linearnih matričnih nejednadžbi. Na osnovu tih razmatranja, primjenom programskog paketa Yalmip, provesti sintezu PID regulatora za automatsko upravljanje letjelicom pri čemu za ulaznu upravljačku varijablu odabrati kut otklona kormila visine.
4. Simulacijama na računalu primjenom programskog paketa MATLAB analizirati odzive lineariziranog modela letjelice u zatvorenom sustavu s projektiranim regulatorom.
5. PID regulator projektiran u prethodnoj točki ovog zadatka implementirati na mikrokontroler u zatvorenom krugu s ostalim elementima (aktuatorima i sensorima) koji su predviđeni za ugradnju u letjelicu. Također, dati smjernice za izradu fizičkog modela letjelice u cijelosti.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

Datum predaje rada:

Predviđeni datumi obrane:

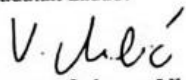
24. 4. 2024.

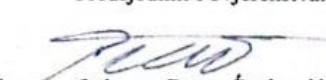
2. rok (izvanredni): 11. 7. 2024.  
3. rok: 19. i 20. 9. 2024.

2. rok (izvanredni): 15. 7. 2024.  
3. rok: 23. 9. – 27. 9. 2024.

Zadatak zadao:

Predsjednik Povjerenstva:

  
izv. prof. dr. sc. Vladimir Milić

  
izv. prof. dr. sc. Petar Čurković

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA.....	III
SAŽETAK.....	VI
SUMMARY .....	VII
1. UVOD.....	1
2. MODEL ZRAKOPLOVA .....	2
2.1. Geometrija letjelice .....	3
2.2. Matematičko modeliranje .....	5
2.2.1. Prostor stanja.....	5
2.2.2. CFD analiza .....	7
3. SINTEZA REGULATORA.....	15
3.1. Optimalni linearni kvadratni regulator.....	15
3.2. Određivanje pojačanja regulatora primjenom linearnih matričnih nejednadžbi .....	17
3.3. Računalne simulacije .....	20
4. REALIZACIJA SUSTAVA U STVARNOSTI .....	24
4.1. Elektronika i elektrika .....	24
4.1.1. Mikroupravljač.....	24
4.1.2. Senzor .....	25
4.1.3. Radio prijemnik i predajnik .....	26
4.1.4. Servo motori.....	26
4.1.5. Pogonski motor .....	27
4.1.6. Kontroler brzine vrtnje.....	28
4.1.7. Baterijski paket .....	29
4.1.8. Osnovni princip rada.....	29
4.2. RC avion .....	30
4.2.1. Krilo, horizontalni i vertikalni stabilizator.....	30
4.2.2. Tijelo aviona .....	33
4.2.3. Olova za balansiranje .....	34
4.2.4. PCB s mikroupravljačem .....	34
4.2.5. Arduino kôd regulatora .....	36
5. TESTIRANJE LETJELICE I REGULATORA .....	38
6. ZAKLJUČAK.....	40
LITERATURA.....	41
PRILOZI.....	42
I. MATLAB kôd za rješavanje LMI problema.....	43
II. MATLAB kôd za aproksimaciju veze između <i>elevatora</i> i pripadnog servo motora .....	45
III. Arduino kôd regulatora .....	46

**POPIS SLIKA**

Slika 2.1	CAD model letjelice .....	4
Slika 2.2	Avion u bokocrtu .....	4
Slika 2.3	$C_l$ vs $C_d$ polar za NACA 4412 .....	8
Slika 2.4	$C_l$ vs $\alpha$ i $C_m$ vs $\alpha$ polari za NACA 4412 .....	8
Slika 2.5	$C_l / C_d$ vs $\alpha$ polar za NACA 4412 .....	9
Slika 2.6	$C_l$ vs $C_d$ polar za NACA 0009 .....	10
Slika 2.7	$C_l$ vs $\alpha$ i $C_m$ vs $\alpha$ polari za NACA 0009 .....	10
Slika 2.8	$C_l / C_d$ vs $\alpha$ polar za NACA 0009 .....	11
Slika 2.9	Definicija aviona u svrhu analize stabilnosti u <i>xflr5</i> .....	12
Slika 2.10	Rezultati aerodinamičke analize virtualnog modela aviona .....	13
Slika 3.1	Blok dijagram sustava upravljanim PID regulatorom .....	15
Slika 3.2	I-PD forma PID regulatora .....	16
Slika 3.3	Blok dijagram povratne veze po stanjima sustava .....	17
Slika 3.4	Simulink model sustava .....	20
Slika 3.5	Rezultat simulacije za prvi skup ograničenja .....	21
Slika 3.6	Rezultat simulacije za drugi skup ograničenja .....	22
Slika 3.7	Rezultati simulacija za treći skup ograničenja .....	23
Slika 4.1	MPU6050 senzor .....	26
Slika 4.2	iRotor 5010/750kV motor .....	27
Slika 4.3	Hobby Wing Skywalker ESC .....	28
Slika 4.4	Shematski princip rada elektronike .....	30
Slika 4.5	a) Stražnja kosina b) Prednja kosina .....	31
Slika 4.6	Sučeljni spoj .....	31
Slika 4.7	Unutarnja struktura krila .....	31
Slika 4.8	Aeleroni i servo motori nakon montaže u krilu .....	32
Slika 4.9	Fleksijski mehanizam zakretanja .....	32
Slika 4.10	Krilo ojačano staklenim vlaknima .....	33
Slika 4.11	Montirani BLDC .....	33
Slika 4.12	Servo motori nakon montaže i utor za vertikalni stabilizator .....	34
Slika 4.13	Električna shema regulacijskog uređaja .....	35
Slika 4.14	PCB nakon lemljenja .....	35
Slika 4.15	Polinom izveden na temelju podataka iz tablice 4.2 .....	37
Slika 5.1	Letjelica netom prije lansiranja .....	38
Slika 5.2	Letjelica tijekom lansiranja .....	39
Slika 5.3	Letjelica u letu .....	39

---

**POPIS TABLICA**

Tablica 2.1 Legenda aerodinamičkih karakteristika..... 7  
Tablica 4.1 Neke od važnijih specifikacija razmatranih mikroupravljača ..... 25  
Tablica 4.2 Tablica zakreta servoa i elevatora (- označava zakret prema gore) ..... 37



**POPIS OZNAKA**

<b>Oznaka</b>	<b>Jedinica</b>	<b>Opis</b>
$M$	-	Machov broj
$Re$	-	Reynoldsov broj
$\alpha$	°	Napadni kut
$C_m$	-	Koeficijent posrtnog momenta
$C_l$	-	Koeficijent aerodinamičkog uzgona
$C_d$	-	Koeficijent aerodinamičkog otpora
$\mathbf{x}$		Vektor varijabli stanja
$\dot{\mathbf{x}}$		Derivacija vektora varijabli stanja sustava
$\mathbf{y}$		Vektor izlaza sustava
$\mathbf{u}$		Vektor upravljanja sustava
$\mathbf{A}$		Matrica sustava
$\mathbf{B}$		Matrica upravljanja
$\mathbf{C}$		Matrica izlaznih varijabli
$\mathbf{D}$		Matrica unaprijednog djelovanja
$\mathbf{A}_{long}$		Matrica sustava longitudinalne dinamike zrakoplova
$\mathbf{B}_{long}$		Matrica upravljanja longitudinalnom dinamikom zrakoplova
$\mathbf{C}_{long}$		Matrica izlaznih varijabli za longitudinalnu dinamiku
$\mathbf{D}_{long}$		Matrica unaprijednog djelovanja za longitudinalnu dinamiku
$X_u$	$\text{N m}^{-1} \text{s}$	Derivacija sile u smjeru $x$ -osi s obzirom na brzinu $u$
$T_u$	$\text{N m}^{-1} \text{s}$	Derivacija potisne sile s obzirom na brzinu $u$
$Z_u$	$\text{N m}^{-1} \text{s}$	Derivacija sile u smjeru $z$ -osi s obzirom na brzinu $u$
$Z_w$	$\text{N m}^{-1} \text{s}$	Derivacija sile u smjeru $z$ -osi s obzirom na brzinu $w$
$Z_q$	$\text{N rad}^{-1} \text{s}$	Derivacija sile u smjeru $z$ -osi s obzirom na kutnu brzinu $q$
$M_u$	$\text{N s}$	Derivacija momenta posrtanja s obzirom na brzinu $u$
$M_w$	$\text{N s}$	Derivacija momenta posrtanja s obzirom na brzinu $w$
$M_q$	$\text{N m rad}^{-1} \text{s}$	Derivacija momenta posrtanja s obzirom na kutnu brzinu $q$
$m$	$\text{kg}$	Masa
$I_{yy}$	$\text{mm}^4$	Aksijalni moment tromosti oko $y$ -osi
$g$	$\text{M s}^{-2}$	Gravitacijska akceleracija
$T$	$\text{N}$	Potisna sila
$u$	$\text{m s}^{-1}, \text{°}$	Brzina u smjeru $x$ -osi / upravljački signal regulatora
$v$	$\text{m s}^{-1}$	Brzina u smjeru $y$ -osi
$w$	$\text{m s}^{-1}$	Brzina u smjeru $z$ -osi
$q$	$\text{rad s}^{-1}$	Kutna brzina oko $y$ -osi
$\theta$	$\text{rad}$	Kut posrtanja
$V_{ref}$	$\text{m s}^{-1}$	Brzina u radnoj točki
$\delta_T$	-	Generička upravljačka varijabla vezana za potisak

Oznaka	Jedinica	Opis
$\delta_m$	rad	Generička upravljačka varijabla vezana za posrtni moment
$T_{\delta T}$	N	Derivacija potisne sile s obzirom na upravljačku varijablu $\delta_T$
$X_{\delta m}$	N rad <sup>-1</sup>	Derivacija sile u smjeru $x$ -osi s obzirom na upravljačku varijablu $\delta_m$
$Z_{\delta m}$	N rad <sup>-1</sup>	Derivacija sile u smjeru $z$ -osi s obzirom na upravljačku varijablu $\delta_m$
$M_{\delta m}$	N m rad <sup>-1</sup>	Derivacija momenta posrtanja s obzirom na upravljačku varijablu $\delta_m$
$K_P$	-	Pojačanje proporcionalnog djelovanja
$K_I$	-	Pojačanje integrirajućeg djelovanja
$K_D$	-	Pojačanje derivacijskog djelovanja
$r$	rad	Signal referentne veličine
$e$	rad	Signal regulacijske pogreške
$y$	rad	Stvarna (regulirana) veličina
$t$	s	vrijeme
$\mathbf{K}$		Vektor pojačanja regulatora stanja
$\bar{\mathbf{A}}$		Augmentirana matrica sustava
$\bar{\mathbf{B}}$		Augmentirana matrica upravljanja
$\mathbf{z}$		Augmentirani vektor stanja
$\xi$	rad s	Pomoćna varijabla za sintezu regulatora
$\mathbf{Y}$		Iteracijska matrica LMI metode
$\hat{\mathbf{P}}$		Iteracijska matrica LMI metode
$\mathbf{Q}$		Težinska matrica vezana za stanja sustava
$\mathbf{R}$		Težinska matrica vezana za snagu odziva sustava
$m$	-	Broj upravljačkih ulaza sustava
$n$	-	Broj stanja sustava
$p$	-	Broj izlaza sustava
$K_v$	min <sup>-1</sup> V <sup>-1</sup>	Konstanta brzine vrtnje BLDC motora
$n_n$	min <sup>-1</sup>	Nazivna brzina vrtnje BLDC motora
$K_T$	N m V <sup>-1</sup>	Konstanta momenta BLDC motora
$M_T$	N m	Nazivni moment motora
$I_{max}$	A	Maksimalna struja pražnjenja baterije
$C$	h <sup>-1</sup>	Dozvoljena stopa pražnjenja LiPo baterije
$Q$	mA h	Kapacitet LiPo baterije

---

**SAŽETAK**

Regulacijski uređaji za upravljanje zrakoplovima široko su rasprostranjeni u raznim sektorima zrakoplovstva. U uporabi su preko sto godina jer pružaju mnoge prednosti, kao što su smanjenje zamora pilota, smanjenje trošenja zrakoplova, smanjenje potrošnje goriva itd. Kako bi se projektirali, potrebno je poznavati dinamiku letjelice kojom se upravlja. Opisuje ju se matematičkim modelima, simulacijama u zračnim tunelima, a u zadnjih 20-ak godina sve više koristeći CFD (*eng. computational fluid dynamics*) softver. U skladu s navedenim, u ovom završnom radu izvedena je sinteza I-PD regulatora (još poznat kao *setpoint on I controller*) temeljenog na tzv. linearnom kvadratnom regulatoru (LQR). Proračun parametra ovog uređaja može se izvesti na razne načine, ali se u današnjici vrlo pogodnom pokazala metoda bazirana na rješavanju linearnih matričnih nejednadžbi (LMI). Ova metoda posebno je povoljna za sintezu regulatora sustava čiji se parametri ne mogu lako pronaći korištenjem starijih metoda poput Ackermannove. Opisani regulator realizirao se koristeći neke od sve moćnijih komercijalno dostupnih mikroupravljača, uz žiroskopski senzor kao povratnu vezu. Ponašanje realiziranog regulatora, a i kvaliteta LMI-LQR metode, također je provjerena na vlastoručno izrađenom RC modelu.

Ključne riječi: linearna matrična nejednadžba, linearni kvadratni regulator, I-PD regulator, daljinski upravljana letjelica

---

**SUMMARY**

Controllers for the purpose of controlling aircraft have been widely adopted in various sectors of aeronautical engineering. They have been in use for some one hundred years now, due in part to the many advantages they provide, such as reducing pilot fatigue, reducing wear on aircraft, reducing fuel consumption etc. In order to devise them, it is necessary to understand the dynamics of the aircraft one desires to control. To this end, mathematical models, air tunnel simulations and CFD (abbr. of *computational fluid dynamics*) simulations, which have become widely adopted in the last 20 years, are utilized. In accordance with the aforementioned information, an I-PD controller (also known as a *setpoint on I controller*) was designed based on the theory behind so-called linear-quadratic regulators (LQR). Calculating the parameters of this device may be done in various ways, however in today's age a method based on linear matrix inequalities (LMI) was shown to be adequate. This method is especially useful for deriving the parameters of controllers of plants whose parameters can't be easily acquired using older methods such as Ackermann's. The described controller was realized using some of the increasingly powerful commercially available microcontrollers, together with a gyroscopic sensor as a feedback loop. The behavior of the devised controller, along with the quality of the LMI-LQR method, was evaluated on a handmade RC model.

Key words: linear matrix inequality, linear-quadratic regulator, I-PD controller, remote controlled aircraft

## 1. UVOD

U ovom završnom radu bilo je potrebno izraditi 3D model letjelice (zrakoplova) u CAD-u, provesti analizu njezinih aerodinamičkih karakteristika te saznanja iz analize iskoristiti za sintezu regulatora koji će upravljati kutom posrtnanja. Nadalje, regulator je bilo potrebno realizirati uz pomoć elektroničkih komponenata.

Uređaji za regulaciju orijentacije zrakoplova upotrebljavaju se u vojnim i komercijalnim svrhama više od sto godina. Neke od prednosti ovakvih sustava su: smanjenje napora letenja, povećana preciznost letenja, smanjenje potrošnje goriva, smanjenje umora materijala aviona itd. Uz te prednosti, može se s lakoćom zaključiti zašto proizvođači letjelica inzistiraju na uvođenju takvih sustava. U ovome radu implementirat će se puno jednostavnija verzija sustava regulacije orijentacije kako bi se za buduće radove stečeno znanje moglo proširiti te se implementirati još složeniji i profinjeniji sustavi. Sinteza regulacijskih uređaja u ove svrhe može se provesti na razne načine te postoje raznolike izvedbe tih sustava (s raznim prednostima i nedostacima), međutim za ovaj rad odlučilo se implementirati jednovarijabilni PID regulator u modificiranoj formi s proporcionalnim i derivacijskim članom u povratnoj vezi. Dobro je poznato da PID regulator i njegove varijante imaju široku primjenu u raznim sektorima industrije zbog izuzetne fleksibilnosti, pouzdanosti i jednostavnosti implementacije. Općenito je glavni problem kod implementacije bilo kakvih regulatora, a tako i PID regulatora, određivanje parametara koji će sustavom upravljati optimalno. Jedna od prvih sustavnih metoda za određivanje parametara ovakvog regulatora je tzv. Ziegler-Nicholsova metoda, međutim ona pruža povoljno ponašanje regulatora samo za sustave s izuzetno sporom dinamikom (uglavnom termodinamički sustavi) te se temelji na empirijskom procesu koji zahtjeva mjerenje odziva otvorenog kruga. Za mnoge sustave, a pogotovo zrakoplove, to je uvelike nepraktično. Evolucijom elektronike omogućilo se upravljanje sustavima s puno izraženijim dinamičkim ponašanjem te je nastala potreba za pouzdanim *offline* metodama sinteze, a ne empirijskim postupcima „pogađanja i promašaja“. Mnoge takve metode rješavaju tzv. problem pozicioniranja polova (*eng. pole allocation problem*). U današnjici izuzetno lukrativnom se pokazala metoda temeljena na linearnim matričnim nejednadžbama [1] i teoriji iza linearnih kvadratnih regulatora. Glavna prednost ove metode je mogućnost pronalaženja pojačanja regulatora koji upravljaju sustavima s brojem stanja većim od dvostrukog broja izlaza ( $n \leq 2p$ ) za koje je taj proces inače vrlo zamoran izvoditi ručno jer zahtjeva iterativni postupak.

Budući da tema ovog završnog rada nije detaljna analiza dinamike leta zrakoplova, nije provedeno ručno izvođenje dinamičkog modela zrakoplova potrebnog za sintezu regulatora (a čija bi pouzdanost bila upitna), nego se poslužilo gotovim modelom iz literature [2]. U svrhu provođenja sinteze u inženjerskoj praksi, ona je također provedena prema gotovom postupku [3] temeljenom na matematici iza primjene linearnih matričnih nejednadžbi u teoriji upravljanja [1].

Postupak rješavanja zadatka se provodio sustavno. Uz poznati izgled aviona kojim se želi upravljati, napravljena je poveznica između njegovih fizikalnih karakteristika i matematičkog modela. Zatim su pronađeni parametri matematičkog modela na temelju simulacije ponašanja aviona u različitim režimima strujanja zraka koristeći CFD softver *xflr5*. Dobiveni parametri su iskorišteni za provedbu sinteze regulatora. Konačno, nakon provjere rezultata sinteze računalnom simulacijom, regulator je realiziran koristeći elektroničke komponente i C kôd u Arduino razvojnem okruženju.

## 2. MODEL ZRAKOPLOVA

Prilikom odabira modela zrakoplova bilo je potrebno naći kompromis između nekoliko aspekata: financijska isplativost, upravljivost, složenost izrade te kvaliteta leta. RC modelarstvo je hobi koji mnogi smatraju „skupim sportom“. Modeli dostupni za kupnju redovito koštaju nekoliko stotina eura. Kvalitetniji modeli napravljeni od plastike, balze, metala itd. znaju biti u ponudi za red veličine više. Također je za dobiveni model teško odrediti mnoge parametre potrebne za sintezu regulatora kao što su geometrija krila i položaj težišta. Nadalje, za mnoge jeftinije modele je, radi malih dimenzija, upitna mogućnost ugradnje elektroničkog paketa koji bi avionom upravljao, a također je teško procijeniti utjecaj dodatne težine i/ili aerodinamičkog otpora na performanse letenja. Stoga je odlučeno napraviti vlastiti model u CAD-u, a ako je ikako moguće realizirati zrakoplov fizički.

Upravljivost je najvažniji parametar vezan za kvalitetu modela. Model može imati bilo kakav dizajn sve dok je stabilan, a time upravljiv. Kako bi se ostvarila stabilnost zrakoplova potrebno je osigurati da se težište nalazi ispred centra uzgona (*eng. center of lift – COL*). U protivnom zrakoplov će težiti neupravljivom posrtanju prema nebu, što može uzrokovati lom uzgona, prevelike aerodinamičke sile i uništenje modela. Također je važno osigurati da težište nije previše udaljeno u smjeru prednjeg kraja letjelice odnosno na COL jer će s takvom konfiguracijom avion težiti posrtati prema tlu poput npr. strijele za pikado, što će uzrokovati ubranu, neplaniranu demontažu modela u kontaktu s tlom. Ukratko, avion mora letjeti na čovjeku upravljiv i predvidljiv način.

Ako je ikako moguće, konstrukcija se mora što više pojednostaviti, a da se održi dovoljna vjernost modelu korištenom u fazi sinteze regulatora. Ako se upravljački parametri zrakoplova u stvarnosti previše razlikuju od onih u fazi sinteze, regulator se neće ponašati prikladno te može uzrokovati uništenje modela. Također se moraju uvesti određena pojednostavljenja radi dostupne tehnologije koja bi se koristila u stvarnosti za izradu aviona.

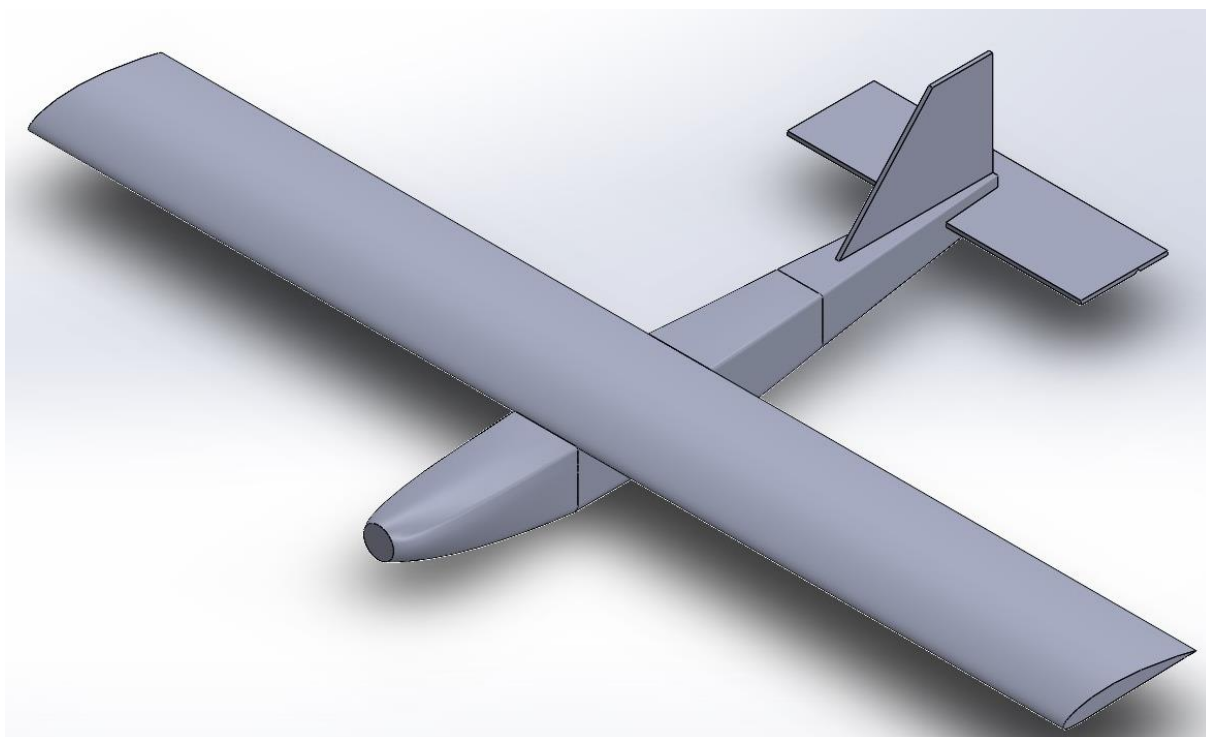
## 2.1. Geometrija letjelice

Najvažnije geometrijsko obilježje aviona je profil krila (*eng. airfoil*). Postoje mnogi različiti profili krila. Uporaba određenog profila ovisi o režimu strujanja zraka prilikom leta, optimalnoj horizontalnoj ili vertikalnoj brzini (*eng. climb rate*), nadmorskoj visini na kojoj se leti, kriterijima vezanima uz agilnost letjelice itd. Budući da je riječ o mehatroničkom radu te o manjku iskustva u polju zrakoplovnog inženjerstva, za profil krila se potražio savjet. Istraživanjem na internetu i konzultacijom s ljudima koji se dugoročno bave RC modelarstvom, zaključilo se da bi najbolje bilo upotrijebiti profil NACA 4412 radi relativno jednostavnog oblika gornjeg i donjeg *splinea* koji čine profil, što olakšava izradu. Također je riječ o profilu koji je namijenjen za uporabu u režimu letenja pri vrlo niskom Machovom broju ( $M \ll 1$ ) što je pogodno za sigurno ispitivanje funkcionalnosti letjelice i regulatora koji upravlja njom. Također je od strane konzultanata predložena uporaba relativno velikog raspona krila u iznosu od 1.5 m kako bi se ostvario povoljan uzgon za niske brzine letenja.

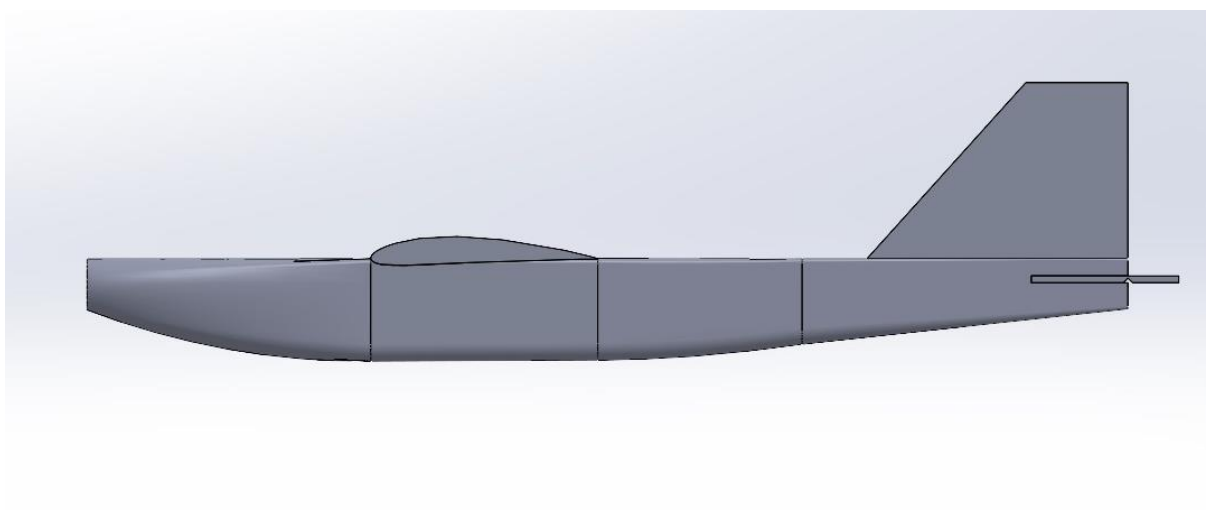
Geometrija repnih krilca (vertikalni i horizontalni stabilizatori) nije prevažna budući da ta krilca ne doprinose puno uzgonu zrakoplova za ovu namjenu. Stoga je od strane suradnika predloženo da se izrade u pravokutnom presjeku. Međutim, za potrebe analize stabilnosti korišten je profil NACA 0009. Razlog tome je činjenica da korišteni CFD softver ne podržava pravokutne profile krila za aerodinamičku analizu. Nadalje, u zrakoplovstvu je česta uporaba simetričnih profila za repna krilca jer takvi profili ne stvaraju uzgon za neutralni napadni kut [4]. Konačno, budući da je njihov aerodinamički doprinos ka ukupnom uzgonu na zrakoplov malen, nije bitno ozbiljno razmatrati koji profil će se koristiti za ovaj avion, sve dok su repna krilca jednostavna za izraditi te se relevantne dimenzije (duljina tetive krila, raspon krila) poklapaju.

Položaj motora također je važno razmotriti. Propeler prilikom rada stvara potisnu silu koja vuče avion. Međutim ako se propeler ne pozicionira ispravno može doći do stvaranja zakretnog momenta oko poprečne osi aviona (*eng. pitching moment*) zbog postojanja kraka između pravca sile potiska i centra rotacije. Kombinacijom pozicioniranja motora što bliže krilu te zakretanjem motora tako da se stvori presjecište pravca sile i centra rotacije može se minimizirati ili eliminirati utjecaj sile motora na ponašanje aviona. U protivnom moramo modelirati moment sile potiska kao poremećaj ili upravljački ulaz u sintezi regulatora.

Prilikom aerodinamičke analize ovakvih RC aviona ne obraća se pozornost na utjecaj tijela aviona na stvaranje uzgona ili otpora te mnogi softveri, izuzev onih koji se koriste za profesionalnu primjenu u zrakoplovstvu, ne nude mogućnost analize aerodinamike tijela aviona, a u slučaju softvera korištenog u ovome radu – *xflr5* – upozorava se korisnika da u aerodinamičku analizu ne uključuje geometriju tijela. To naravno ne znači da se ne mora osigurati aerodinamičnost tijela. Uzevši u obzir sve navedene zahtjeve stvoren je CAD aviona prikazan na slici 2.1 i 2.2



**Slika 2.1 CAD model letjelice**



**Slika 2.2 Avion u bokocrtu**



## 2.2. Matematičko modeliranje

Kako bi se pristupilo sintezi regulatora potrebno je izvesti matematički model dinamike aviona. Dobro je poznato da je za aerodinamičku analizu pogodno korištenje bezdimenzijske analize. Modeliranje strujanja zraka oko otvorenih profila kao što su profili krila izuzetno je teško analitičkim postupcima jer su dobivene jednadžbe izrazito teške za riješiti, a ako se uvedu određene pretpostavke i pojednostavljenja koja omogućuju rješavanje tih jednadžbi, greške su prevelike. Stoga se u polju zrakoplovstva pokazalo poželjno modeliranje određenih aerodinamičkih fenomena empirijskim postupcima koji kao rezultat daju bezdimenzijske parametre. Prednost takvog pristupa, osim vrlo dobre točnosti rezultata, je skalabilnost. Uz zadovoljavanje određenih kriterija sličnosti moguće je zaključke analize na prototipnom modelu prenijeti na model u stvarnosti. Takva filozofija se redovito uporablja u polju analize profila krila. Web stranice poput *airfoiltools.com* služe kao repozitorij svakakvih profila krila. Oni se mogu preuzeti u obliku *.dat* datoteke te uvijek dolaze u „normiranim“ dimenzijama s duljinom tetive profila jednako 1, a korisnik nakon analize u CFD softveru može profil skalirati na željene dimenzije te biti siguran u istovjetnost ponašanja istog profila, ali većih dimenzija.

Iako se koristimo točnijim i jednostavnijim pristupom modeliranja aerodinamike aviona, dobivanje modela u obliku prijenosne funkcije ili pak prostora stanja je složeno ako se nema na raspolaganju dovoljno znanje fizike zrakoplova. Ne pomaže ni činjenica da je riječ o izrazito nelinearnom sustavu. Nadalje cilj rada nije analiza fizike iza ponašanja zrakoplova, već sinteza regulacijskog uređaja. Stoga je iz literature [2] preuzet gotov, pojednostavljeni, poopćeni te za određeni režim rada linearizirani model u obliku prostora stanja. Sve što je potrebno je unijeti odgovarajuće parametre u njihove pripadne matrice te pristupiti sintezi.

### 2.2.1. Prostor stanja

Općenito, dinamika sustava se može opisati matričnim jednadžbama

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (2.1)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \quad (2.2)$$

gdje je  $\mathbf{A}$  matrica sustava,  $\mathbf{B}$  matrica upravljanja,  $\mathbf{C}$  matrica izlaza,  $\mathbf{D}$  matrica unaprijednog djelovanja (*eng. feedforward matrix*),  $\mathbf{x}$  vektor varijabli stanja,  $\mathbf{u}$  vektor upravljačkih varijabli te  $\mathbf{y}$  vektor izlaznih varijabli sustava.

Iz literature [2] longitudinalna dinamika (dinamika posrtanja) za zrakoplov u režimu letenja spomenutom u prijašnjim poglavljima, opisana je matricom sustava i matricom upravljanja koje u općem slučaju glase:

$$\mathbf{A}_{long} = \begin{bmatrix} \frac{X_u + T_u}{m} & \frac{X_w}{m} & 0 & -g \\ \frac{Z_u}{m} & \frac{Z_w}{m} & \frac{Z_q + mV_{ref}}{m} & 0 \\ \frac{M_u}{I_{yy}} & \frac{M_w}{I_{yy}} & \frac{M_q}{I_{yy}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}_{long} = \begin{bmatrix} \frac{T_{\delta_T}}{m} & \frac{X_{\delta_m}}{m} \\ 0 & \frac{Z_{\delta_m}}{m} \\ 0 & \frac{M_{\delta_m}}{I_{yy}} \\ 0 & 0 \end{bmatrix}, \quad (2.3)$$

dok su vektor varijabli stanja i vektor upravljačkih varijabli:

$$\mathbf{x}_{long} = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}, \quad \mathbf{u}_{long} = \begin{bmatrix} \delta_T \\ \delta_m \end{bmatrix}, \quad (2.4)$$

gdje su:

- $m$  – masa aviona
- $I_{yy}$  – statički moment tromosti u smjeru  $y$  - osi
- $V_{ref}$  – brzina gibanja u radnoj točki
- $u$  - brzina gibanja u smjeru  $x$  – osi
- $w$  – brzina gibanja u smjeru  $z$  – osi
- $q$  – kutna brzina u smjeru  $y$  – osi
- $\theta$  – kut posrtanja zrakoplova oko  $y$  - osi
- $\delta_T$  – upravljačka varijabla vezana za potisnu silu (*eng. throttle* ili vulgaris „gas“)
- $\delta_m$  – upravljačka varijabla vezana za moment posrtanja (*eng. pitching moment*)
- $X$  – sila u smjeru  $x$  – osi
- $Z$  – sila u smjeru  $z$  – osi
- $M$  – moment posrtanja
- $T$  – sila potiska pogonskog motora
- $X_u$  – stabilizacijski derivativ definiran kao derivacija veličine (u ovom slučaju sile  $X$ ) u ovisnosti o nekoj drugoj veličini (u ovom slučaju brzini  $u$ )

Ako se avion izvede tako da se os djelovanja sile potiska nalazi što bliže krilu, djelovanje promjene potisne sile  $T$  na promjenu „gasa“ može se zanemariti ( $\frac{T\delta_T}{m} = 0$ ). Time se također izbacuje dodatna upravljačka varijabla  $\delta_T$ . Uvrštavanjem citirane matrice sustava i matrice upravljanja u jednadžbe (2.1) i (2.2) te uz poznati oblik vektora stanja iz literature [2] dobivamo:

$$\dot{\mathbf{x}}_{long} = \mathbf{A}_{long}\mathbf{x}_{long} + \mathbf{B}_{long}\mathbf{u}_{long}, \quad (2.5)$$

$$\theta = \mathbf{C}_{long}\mathbf{x}_{long} + \mathbf{D}_{long}\mathbf{u}_{long}, \quad (2.6)$$

pri čemu je matrica  $\mathbf{A}_{long}$  jednaka onoj iz izraza (2.3), a matrica  $\mathbf{B}_{long}$  sada postaje:

$$\mathbf{B}_{long} = \begin{bmatrix} \frac{X_{\delta_m}}{m} \\ \frac{Z_{\delta_m}}{m} \\ \frac{M_{\delta_m}}{I_{yy}} \\ 0 \end{bmatrix}, \quad (2.7)$$

te nadalje matrice  $\mathbf{C}_{long}$  i  $\mathbf{D}_{long}$  postaju:

$$\mathbf{C}_{long} = [0 \quad 0 \quad 0 \quad 1], \quad \mathbf{D}_{long} = 0. \quad (2.8)$$

Broj stanja sustava je  $n = 4$ , broj upravljačkih ulaza je  $m = 1$  i broj izlaza je  $p = 1$ .

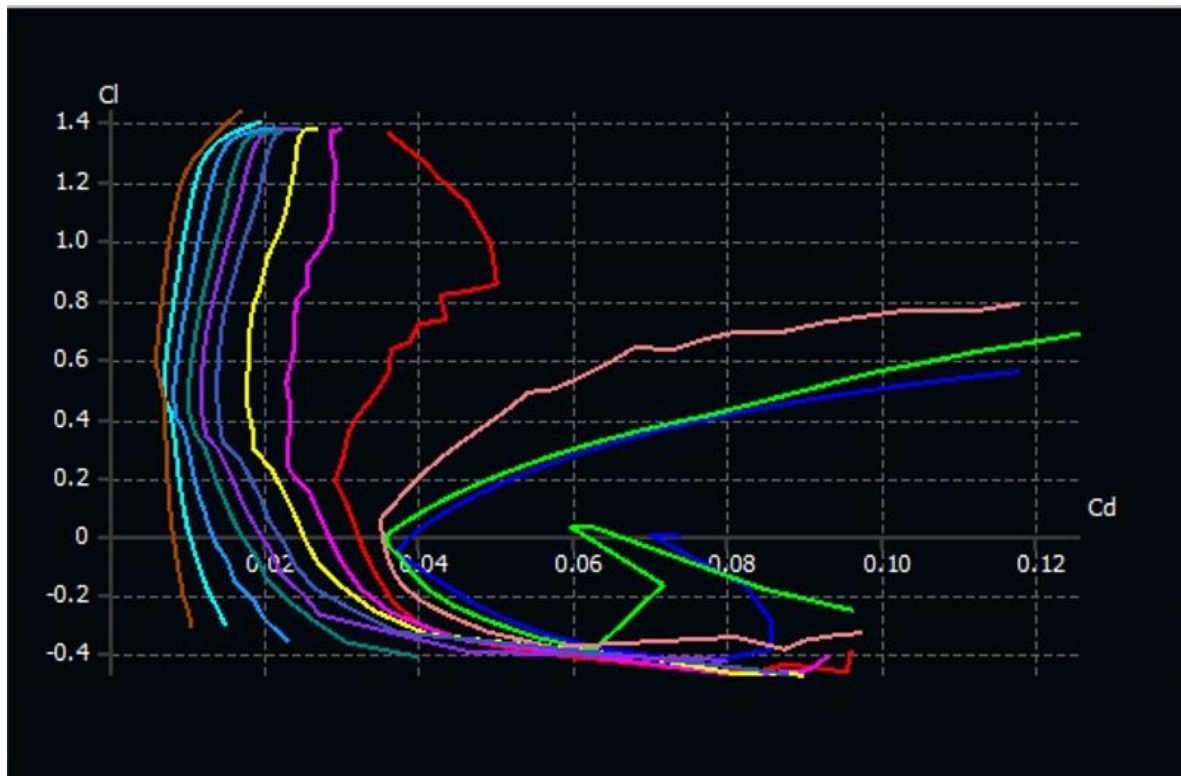
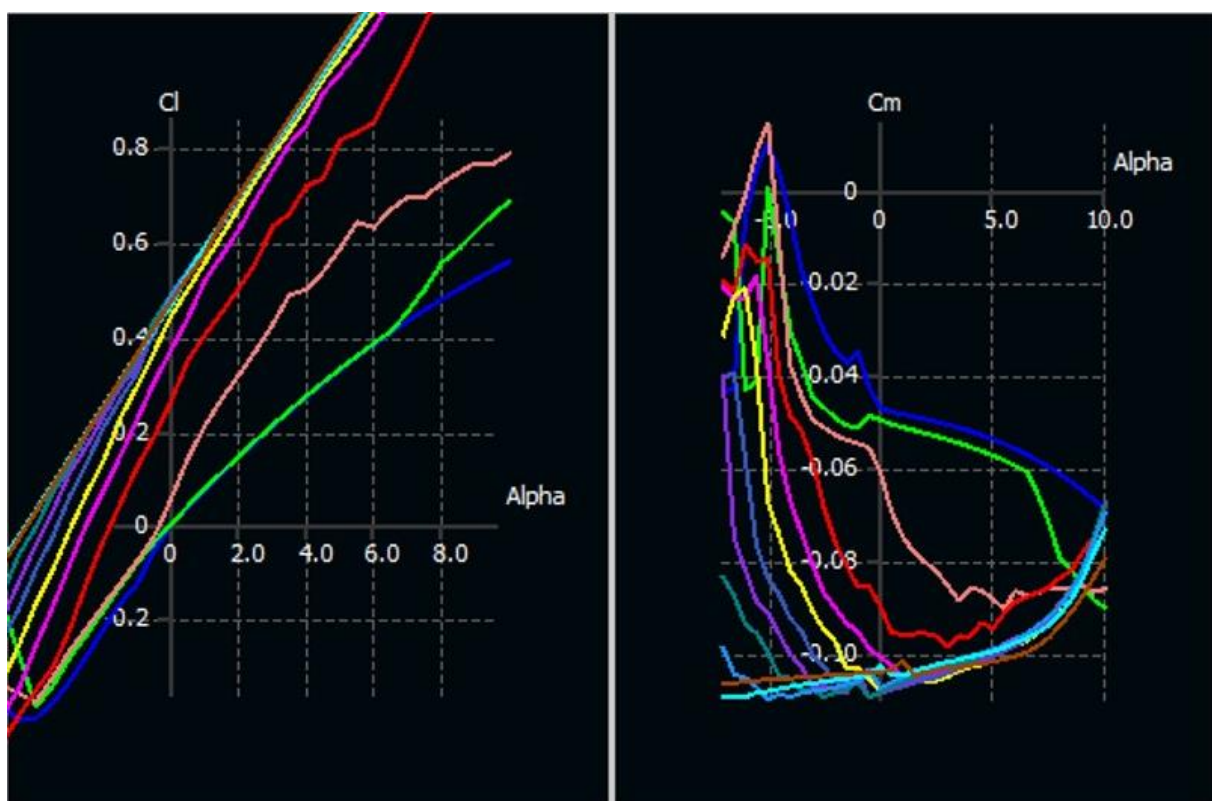
### 2.2.2. CFD analiza

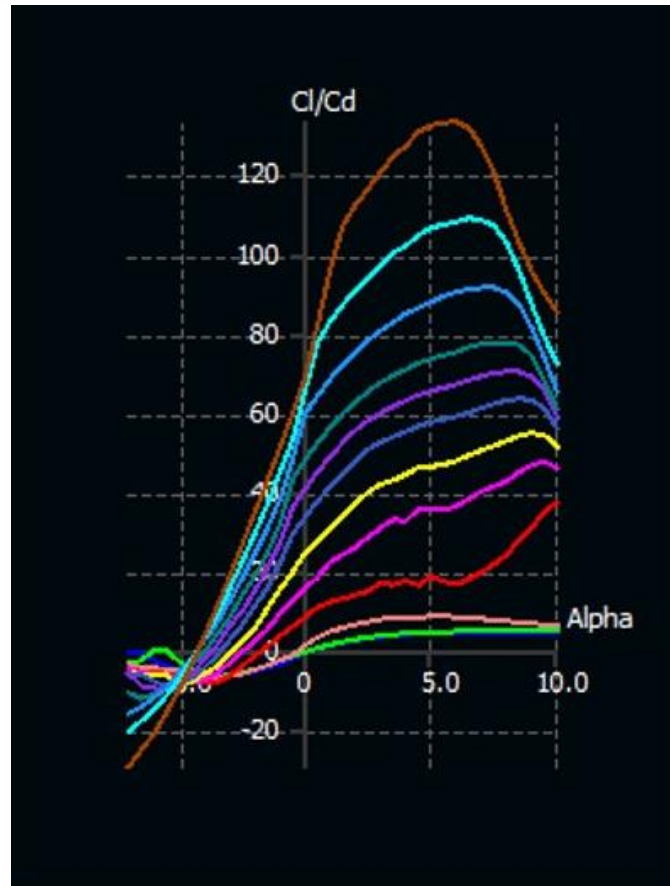
Budući da se mnogi parametri sustava (u ovom slučaju tzv. stabilizacijski derivativi) dobivaju eksperimentima u zračnom tunelu, a takav uređaj nije dostupan za ovaj rad, moramo se poslužiti softverskim rješenjima. Za analizu upotrijebio se softver *xflr5*. Program nudi analizu ponašanja profila krila i analizu stabilnosti aviona koja direktno računa matrice  $\mathbf{A}_{long}$  i  $\mathbf{B}_{long}$  za određenu radnu točku. Stoga nije potrebno ručno računati navedene koeficijente, nego je dovoljno provesti analizu u softveru.

Kako bi se u programu ostvarili svi preduvjeti za analizu stabilnosti aviona, potrebno je prvo učitati sve profile krila koji će se koristiti te ih analizirati za razne napadne kutove i Reynoldsove brojeve. Profili NACA 4412 i NACA 0009 preuzeti s *airfoiltools.com* u obliku .dat datoteke učitali su se u sekciju programa za analizu profila (tzv. *foil batch analysis*). Analiza se provela za raspone  $Re \in [20\ 000, 1\ 000\ 000]$  te kutove  $\alpha \in [-7^\circ, 10^\circ]$ . Rasponi su uzeti prema predlošku autora softvera za generičke, amaterske RC zrakoplove. Rezultati su prikazani kao *polari*  $C_l$  vs  $C_d$ ,  $C_d$  vs  $\alpha$ ,  $C_l / C_d$  vs  $\alpha$ ,  $C_m$  vs  $\alpha$ . Ova analiza postavlja temelj za analizu stabilnosti te se moraju osigurati što kvalitetniji rezultati. Budući da je riječ o numeričkoj metodi, točnije metodi koja numerički rješava Navier-Stokesove jednadžbe, ponekada se javljaju greške ili loši rezultati u određenim intervalima napadnog kuta. Prisutnost grešaka može se uočiti postojanjem jakih diskontinuiteta ili skokova u grafovima karakteristika te porukama u upravljačkoj konzoli koje upozoravaju na lošu vjernost rezultata za neki kut. Navedeni problem se rješava smanjenjem koraka promjene kuta te ponovnim pokretanjem analize za problematični interval. Ova se metoda preporučuje od strane autora programa te bitno poboljšava rezultate analize. Na slikama od slike 2.3 do slike 2.5 prikazani su rezultati aerodinamičke analize profila korištenog za glavno krilo (NACA 4412), dok su na slikama od slike 2.6 do slike 2.8 prikazani rezultati aerodinamičke analize profila korištenog za repna krila (NACA 0009). Krivulje su crtane u različitim bojama. One simboliziraju Reynoldsove brojeve za koje je analiza provedena, a mogu se razlikovati prema legendi priloženoj u tablici 2.1.

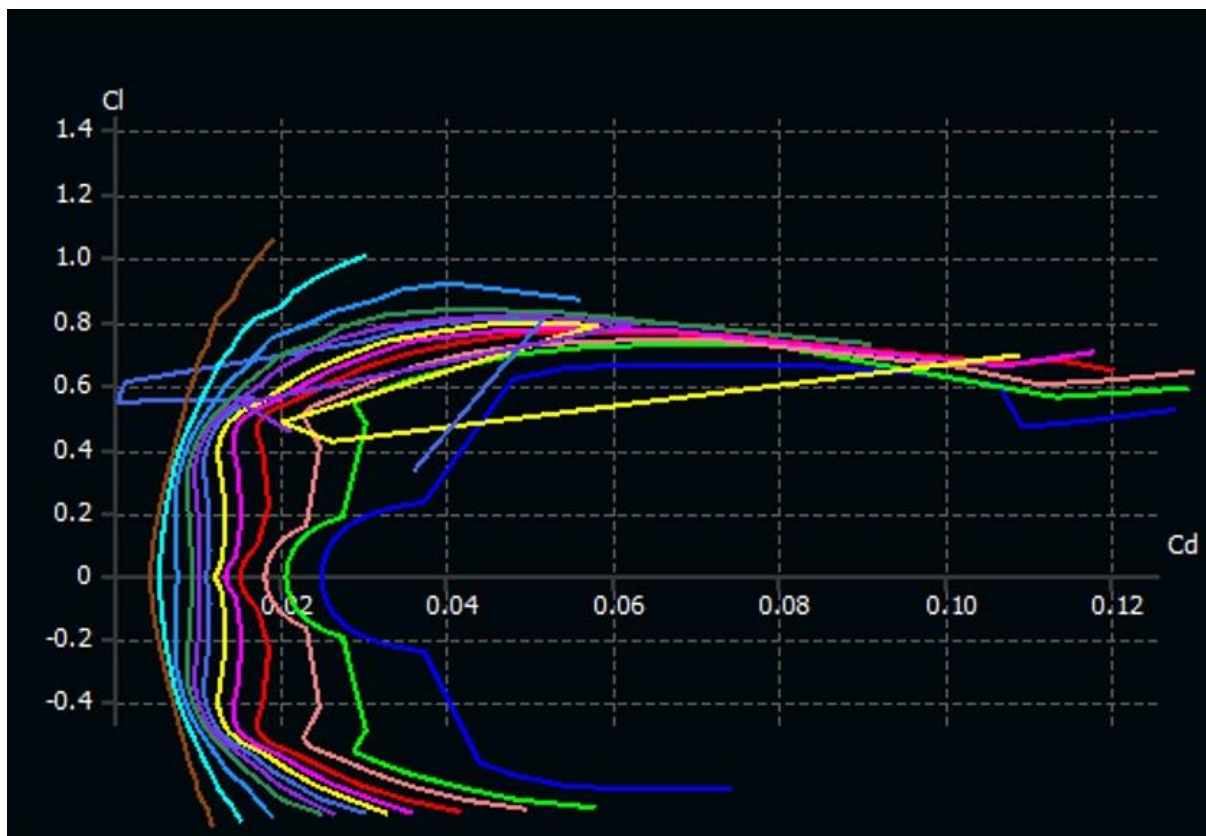
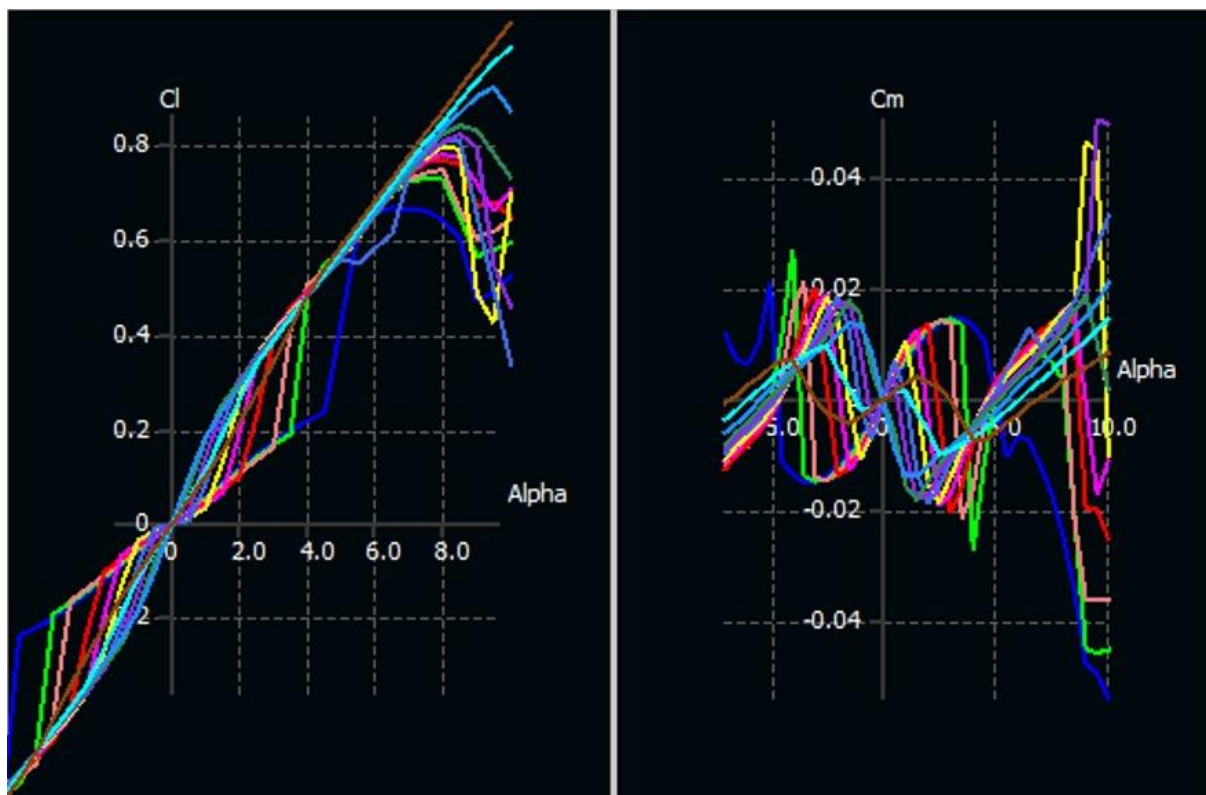
Tablica 2.1    Legenda aerodinamičkih karakteristika

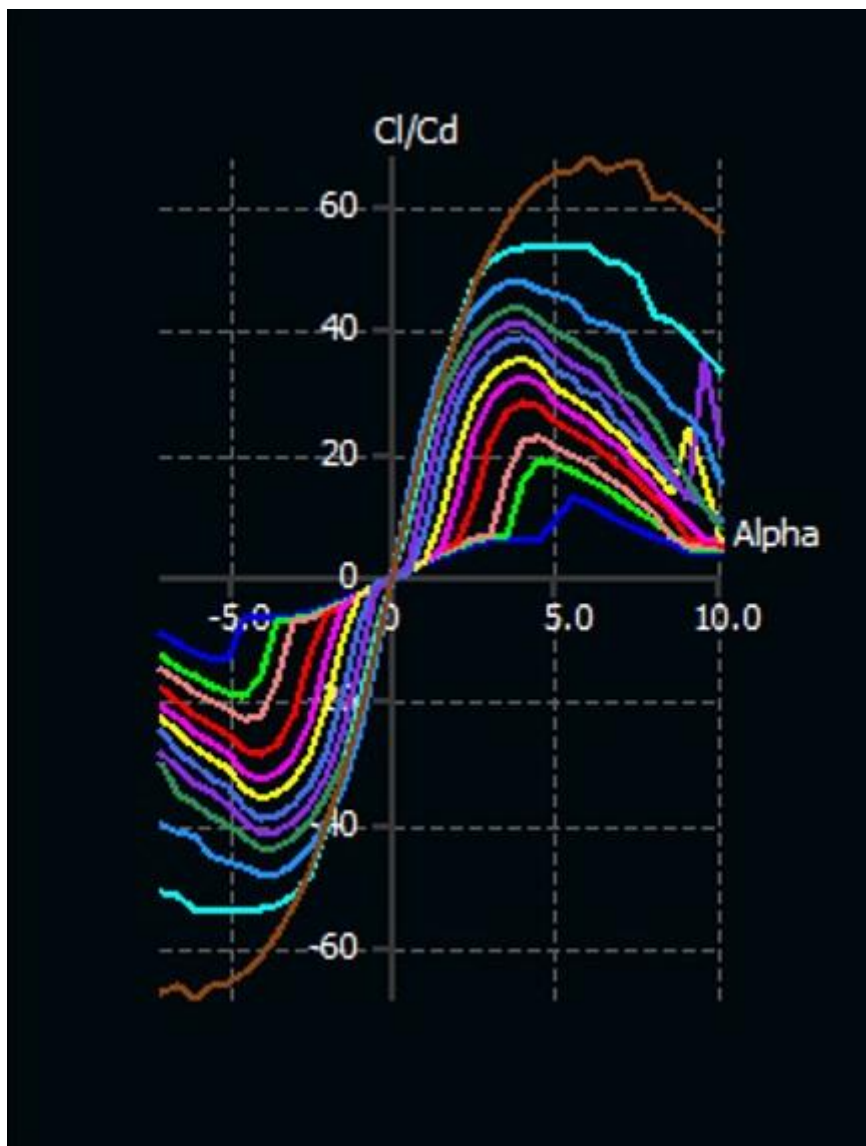
$Re \cdot 10^3$	20	30	40	60	80	100	130	160	200	300	500	1000
-----------------	----	----	----	----	----	-----	-----	-----	-----	-----	-----	------

Slika 2.3  $C_l$  vs  $C_d$  polar za NACA 4412Slika 2.4  $C_l$  vs  $\alpha$  i  $C_m$  vs  $\alpha$  polari za NACA 4412



Slika 2.5  $C_l / C_d$  vs  $\alpha$  polar za NACA 4412

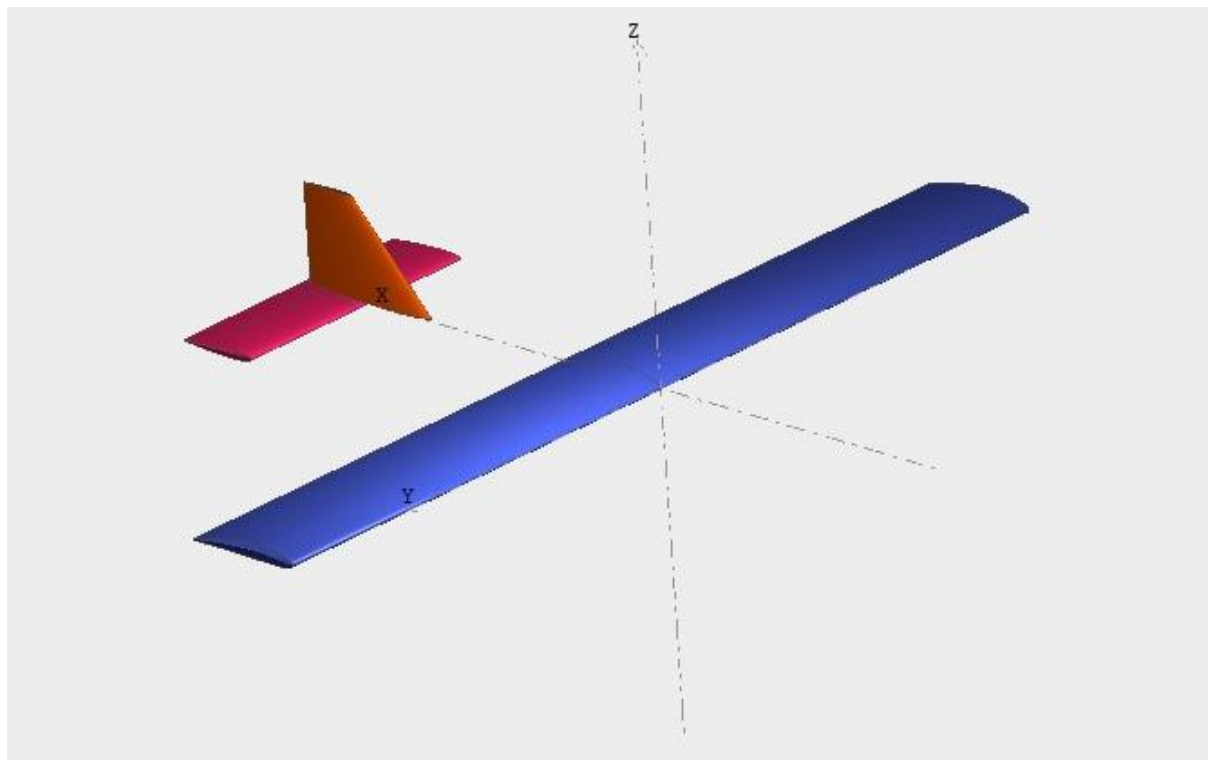
Slika 2.6  $C_l$  vs  $C_d$  polar za NACA 0009Slika 2.7  $C_l$  vs  $\alpha$  i  $C_m$  vs  $\alpha$  polari za NACA 0009



Slika 2.8  $C_l / C_d$  vs  $\alpha$  polar za NACA 0009

Prikazane aerodinamičke karakteristike profila su familije krivulja ovisne o Reynoldsovom broju za kojeg su generirane. Generirane karakteristike vrlo su slične te se može identificirati trend ponašanja s obzirom na promijenjene parametre analize. Npr. porastom napadnog kuta raste uzgon  $C_l$  do određenog graničnog kuta kod kojeg se javlja lom uzgona (pad karakteristike). Što je  $Re$  veći, to je veći maksimalni koeficijent uzgona i kritični napadni kut pri kojem se javlja njegov lom. Za strujanje u standardnim uvjetima (nominalni atmosferski tlak i temperatura, bez vjetra) veći  $Re$  odgovara većoj brzini leta. Budući da je sila uzgona proporcionalna kvadratu brzine, što je veća brzina to će biti veći uzgon krila. Također možemo primijetiti prema  $C_l / C_d$  vs  $\alpha$  grafu da uzgon drastično nadvladava otpor zraka u odabranom rasponu napadnih kutova što je uvelike poželjno.

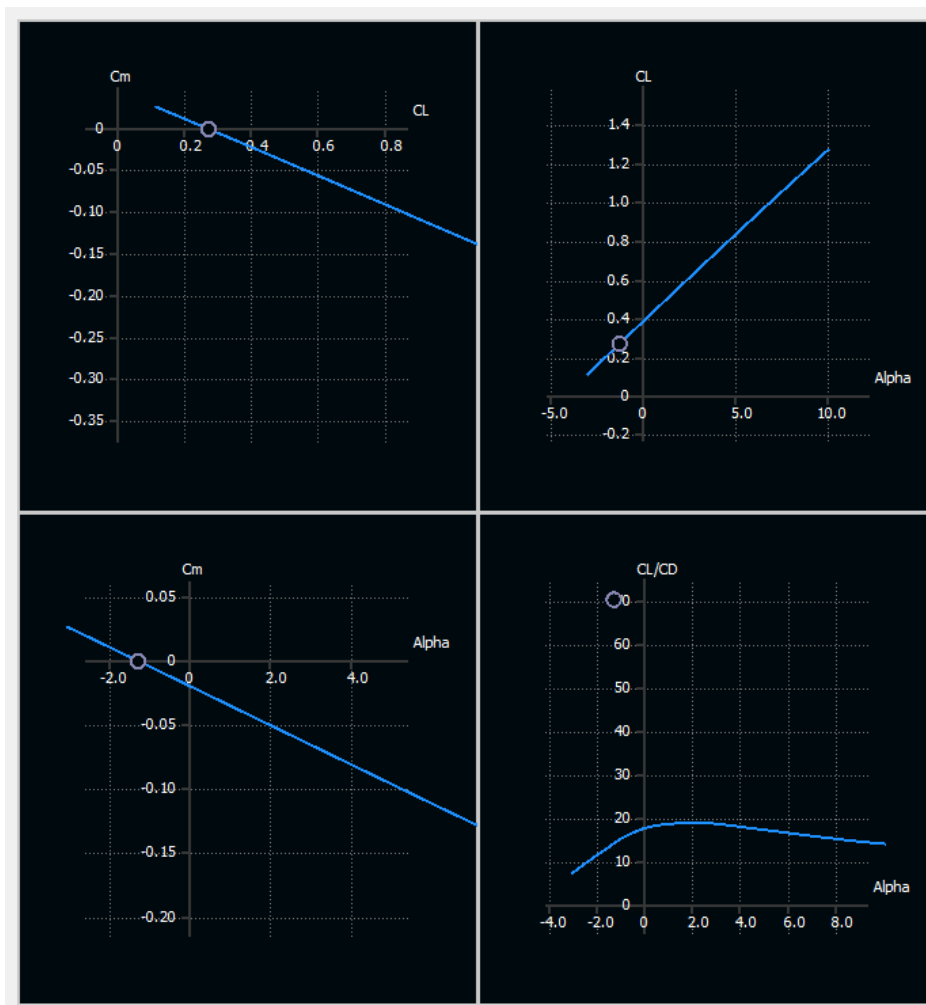
Nakon aerodinamičke analize slijedi aerodinamička analiza sâmog zrakoplova. Za početak analize potrebno je definirati oblik i raspored krila te težište aviona. Uz pomoć suradnika određeno je da bi idealni položaj težišta aviona trebao biti 65 mm u odnosu na napadnu ivicu glavnog krila, u smjeru repa. Način na koji je sve navedeno definirano može se vidjeti na slici 2.9.



Slika 2.9 Definicija aviona u svrhu analize stabilnosti u *xflr5*

Pokretanjem aerodinamičke analize u konfiguraciji *fixed lift* ( $C_l = konst.$ , preporuka autora softvera), uz prethodno navedeni raspon napadnih kuteva  $\alpha$ , dobiveni su  $C_m$  vs  $C_l$ ,  $C_m$  vs  $\alpha$ ,  $C_l$  vs  $\alpha$  i  $C_l/C_d$  vs  $\alpha$  polari prikazani na slici 2.10.





Slika 2.10 Rezultati aerodinamičke analize virtualnog modela aviona

Važno je uočiti da karakteristika  $C_m$  vs  $\alpha$  pada što ukazuje na stabilno ponašanje zrakoplova. Porastom napadnog kuta raste moment posrtanja u negativnome smjeru – efekt samoispravka napadnog kuta.

Uz učitano aerodinamičko ponašanje aviona u softver, jedino što je preostalo je pokrenuti analizu stabilnosti virtualnog modela zrakoplova. Uz postavljenu jediničnu pobudu na *elevator* te definiranje radne točke u  $\alpha = 0^\circ$ , program računa matrice  $\mathbf{A}_{long}$  i  $\mathbf{B}_{long}$  koje u konačnici iznose:

$$\mathbf{A}_{long} = \begin{bmatrix} -0.0177058 & -0.395514 & 0 & -9.81 \\ -1.24609 & -11.9184 & 14.0021 & 0 \\ -1.1448 \cdot 10^{-5} & -128.102 & -170.618 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{B}_{long} = \begin{bmatrix} -0.384582 \\ -18.2417 \\ -3302.796 \\ 0 \end{bmatrix}.$$

Polovi sustava odgovaraju svojstvenim vrijednostima matrice  $\mathbf{A}_{long}$ , a one iznose:

$$\begin{aligned}\lambda_1 &= -158.3701, \\ \lambda_2 &= -24.1638, \\ \lambda_3 &= -0.0101 + 0.6396i, \\ \lambda_4 &= -0.0101 - 0.6396i.\end{aligned}$$

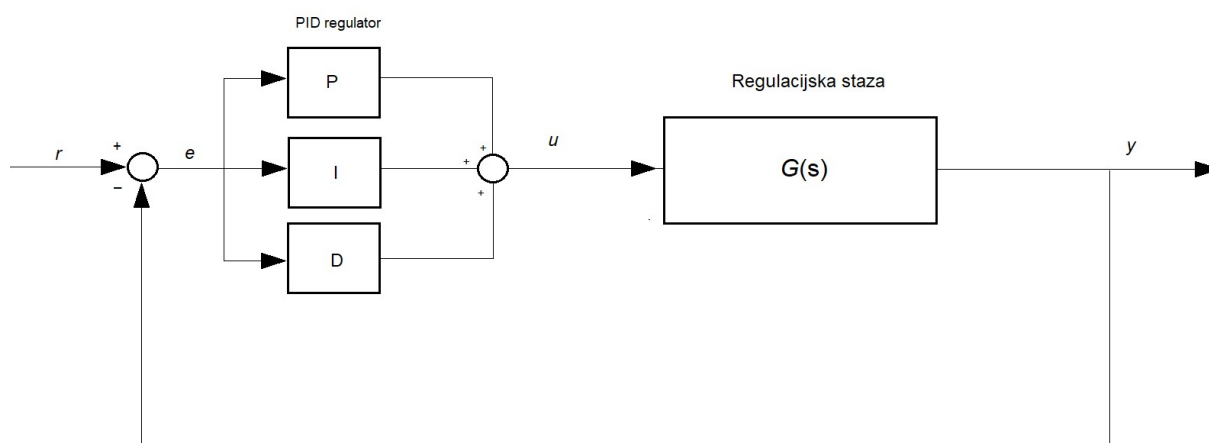
Budući da su realni dijelovi svih polova negativni, potvrđujemo zaključak o stabilnosti zrakoplova iz  $C_m$  vs  $\alpha$  karakteristike.

Valja napomenuti da dimenzije matrice  $\mathbf{B}_{long}$  u ovom i prijašnjem poglavlju nisu iste. Razlika je u tome što se u poglavlju **2.2.1** u općem slučaju modelira utjecaj momenta sile potiska uslijed promjene njezinog iznosa (*eng. throttling*) na kut posrtaanja. Trenutno *xflr5* ne nudi mogućnost modeliranja ovog fenomena stoga ne možemo iz programa dobiti iznos parametra koji ga opisuje, međutim avion možemo konstruirati tako da je pravac njegove sile potiska što bliže krilu kako bi taj moment bio što manji, čime možemo zanemariti njezin utjecaj na promjenu orijentacije aviona. Ujedno to pojednostavljenje bitno olakšava sintezu regulatora jer se MISO (*eng. multiple input, single output*) sustav pretvara a u SISO (*eng. single input, single output*). Uz ovu informaciju možemo konačno pristupiti sintezi regulatora.

### 3. SINTEZA REGULATORA

#### 3.1. Optimalni linearni kvadratni regulator

U području sinteze multivarijabilnih linearnih regulatora često se koristi tzv. linearni kvadratni regulator. Glavna mu je prednost to što u općem slučaju omogućuje upravljanje sustavom određenim s više od jednog ulaza i izlaza (naravno sve dok je sustav linearan). Takav regulator ima mnogo formi, između ostalog i PID oblik. PID regulatori se dugi niz godina implementiraju u raznim industrijskim procesima radi vrlo dobre fleksibilnosti, jednostavnosti i pouzdanosti. Opći oblik PID regulatora glasi:



Slika 3.1 Blok dijagram sustava upravljanim PID regulatorom

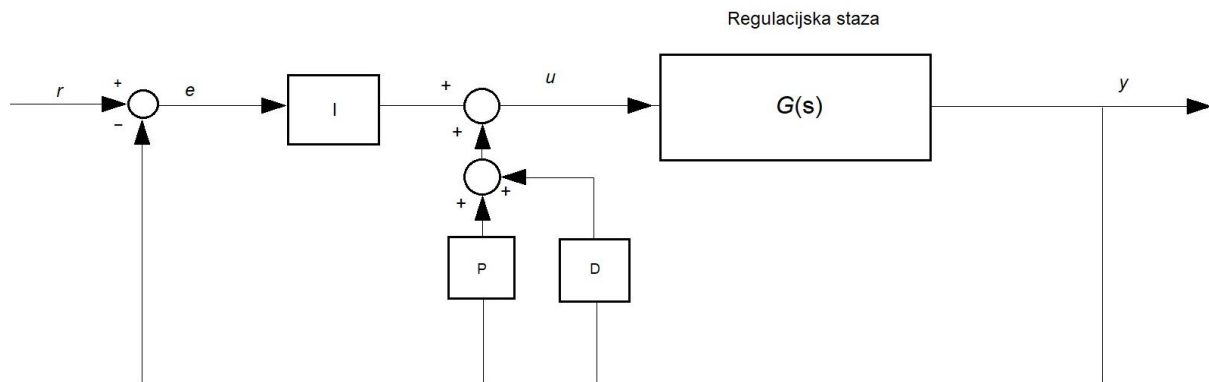
$$u = K_P e + K_I \int e dt + K_D \frac{de}{dt}, \quad (3.1)$$

gdje su:

- $u$  – upravljački signal regulatora
- $e$  – signal regulacijske pogreške
- $r$  – referentni iznos upravljane veličine
- $K_P$  – faktor pojačanja proporcionalnog djelovanja
- $K_I$  – faktor pojačanja integracijskog djelovanja
- $K_D$  – faktor pojačanja derivacijskog djelovanja

Međutim, ova forma ima nedostatak koja se manifestira uslijed skokovite promjene reference. Budući da se signal pogreške 'provlači' kroz proporcionalni i derivacijski član, nagle promjene reference iziskuju naglu promjenu upravljačkog signala, a time nepotrebno opterećenje aktuatora. Ako regulator upravljačkim površinama naredi da zauzmu drastičnu poziciju kako bi se prema zakonu regulacije prešlo u novu radnu točku, može doći do štete ili gubitka kontrole nad avionom (prisjetimo se da je riječ o lineariziranom sustavu koji je precizan u okolini radne točke). Stoga je potrebno upotrijebiti drugačiji oblik kontrolera – I-PD oblik.

I-PD regulator (*eng. setpoint on I controller*) daje isto ponašanje kao PID regulator, ali ne uzrokuje drastičnu promjenu upravljačkog signala uslijed promjene radne točke, stoga je prikladan za uporabu u ovom slučaju.



Slika 3.2 I-PD forma PID regulatora

$$u = K_I \int (r - y) dt + K_P y + K_D \frac{dy}{dt}, \quad (3.2)$$

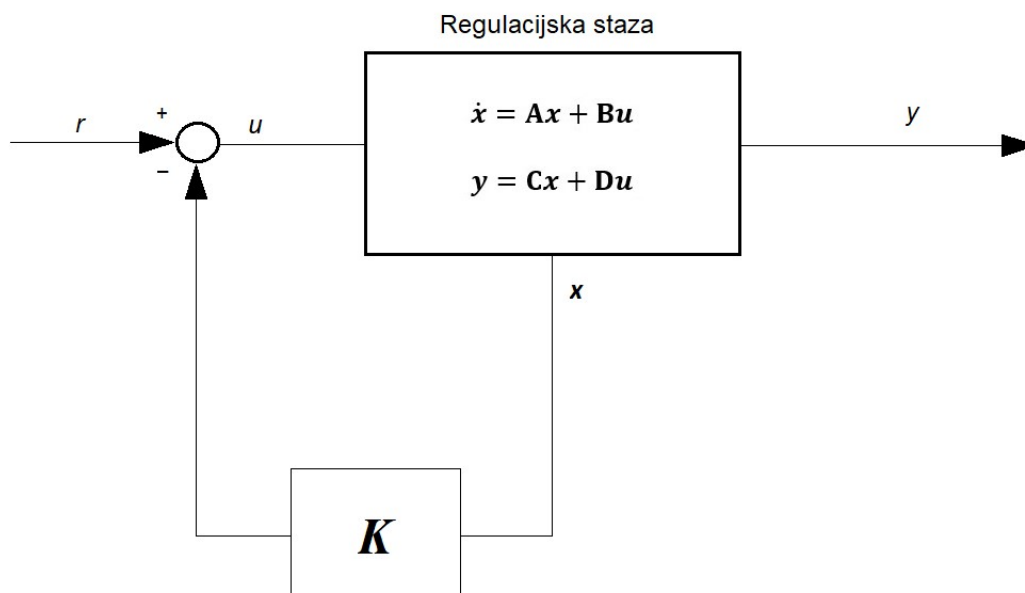
gdje su:

- $r$  – referentni iznos upravljane veličine
- $y$  – signal regulirane veličine
- $K_I$  – pojačanje integracijskog djelovanja
- $K_P$  – pojačanje proporcionalnog djelovanja
- $K_D$  – pojačanje derivacijskog djelovanja

Iz jednadžbe (3.2) i blok dijagrama na slici 3.2 vidljivo je da samo I član ima interakciju s referentnom veličinom  $r$ , dok svi ostali članovi sudjeluju uz pomoć stvarne veličine procesa  $y$ . Efekt ovakve strukture je da se nagle promjene referentne veličine projiciraju samo kroz tromi I član, a skokovite promjene uslijed djelovanja P i D člana nema.

### 3.2. Određivanje pojačanja regulatora primjenom linearnih matričnih nejednadžbi

Sintezi regulatora pristupamo prema literaturi [3]. Pun postupak uz matematičku formulaciju od početka do kraja može se pronaći u njoj. Ono što slijedi je skraćeni, praktični postupak rješavanja problema u inženjerskoj praksi za navedeni zadatak. Od sada pa nadalje, gdje god nije drugačije navedeno, vrijedi sljedeća istovjetnost notacije:  $\mathbf{A} = \mathbf{A}_{long}$ ,  $\mathbf{B} = \mathbf{B}_{long}$ ,  $\mathbf{C} = \mathbf{C}_{long}$ .



Slika 3.3 Blok dijagram povratne veze po stanjima sustava

Započnimo s regulacijom po stanjima sustava uvođenjem supstitucije  $u = -\mathbf{K}x$ . Uvrštavanjem u jednadžbe (2.1) i (2.2) te sređivanjem izraza dobivamo:

$$\dot{x} = (\mathbf{A} - \mathbf{BK})x, \quad (3.3)$$

$$y = (\mathbf{C} - \mathbf{DK})x, \quad (3.4)$$

gdje je  $\mathbf{K}$  vektor pojačanja povratne veze (*eng. feedback gain*). Nadalje, potrebno je definirati augmentirane matrice sustava  $\bar{\mathbf{A}}$  i  $\bar{\mathbf{B}}$  te vektor  $\mathbf{z}$ :

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} x \\ \xi \end{bmatrix},$$

gdje je  $\xi = \int_0^t (r - y)dt$ . Supstitucijom  $\xi$  i  $y = \mathbf{C}x$  u (3.2) dobiva se sljedeća jednakost:

$$u = K_P \mathbf{C}x + K_I \xi + K_D \mathbf{C}\dot{x} \quad (3.5)$$

Nadalje, uvrštavanjem (2.1) u (3.5) te sređivanjem dobiva se izraz

$$\begin{aligned} u &= (\mathbf{I} - K_d \mathbf{CB})^{-1} [K_P \quad K_D \quad K_I] \begin{bmatrix} \mathbf{C} & 0 \\ \mathbf{CA} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \xi \end{bmatrix} \\ &= [\bar{K}_P \quad \bar{K}_D \quad \bar{K}_I] \mathbf{S} \mathbf{z}, \end{aligned} \quad (3.6)$$

gdje je

$$\mathbf{S} = \begin{bmatrix} \mathbf{C} & 0 \\ \mathbf{CA} & 0 \\ 0 & \mathbf{I} \end{bmatrix},$$

$$\bar{K}_P = (\mathbf{I} - K_D \mathbf{CB})^{-1} K_P, \quad (3.7)$$

$$\bar{K}_D = (\mathbf{I} - K_D \mathbf{CB})^{-1} K_D, \quad (3.8)$$

$$\bar{K}_I = (\mathbf{I} - K_D \mathbf{CB})^{-1} K_I \quad (3.9)$$

Pod pretpostavkom da je matrica  $(\mathbf{I} - K_d \mathbf{CB})^{-1}$  invertibilna te da je sustav upravljiv, uz modificirani zakon regulacije  $\mathbf{u} = \mathbf{K} \mathbf{z}$  dobivamo:

$$[\bar{K}_P \quad \bar{K}_D \quad \bar{K}_I] \mathbf{S} = \mathbf{K} \quad (3.10)$$

Pod pretpostavkom da rješenje jednadžbe (3.10) postoji, moguće je izračunati pojačanja PID regulatora prema:

$$K_D = \bar{K}_D (\mathbf{I} + \mathbf{CB} \bar{K}_D)^{-1}, \quad (3.11)$$

$$K_P = (\mathbf{I} - K_D \mathbf{CB}) \bar{K}_P, \quad (3.12)$$

$$K_I = (\mathbf{I} - K_D \mathbf{CB}) \bar{K}_I \quad (3.13)$$

Sinteza regulatora provest će se prema kriteriju minimalnog gubitka performanse (*eng. minimal performance cost*):

$$J = \int_0^{\infty} (\mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (3.14)$$

gdje je  $\mathbf{Q}$  težinska matrica stanja, a  $\mathbf{R}$  težinska matrica upravljačkih varijabli. Isti učinak ostvaruje se rješavajući algebarsku Ricattijevu jednadžbu (ARE) za sustav s  $r = 0$  i  $\mathbf{P} = \mathbf{P}^T > 0$  [3]

$$\bar{\mathbf{A}}^T \mathbf{P} + \mathbf{P} \bar{\mathbf{A}} - \mathbf{P} \bar{\mathbf{B}} \mathbf{R}^{-1} \bar{\mathbf{B}}^T \mathbf{P} + \mathbf{Q} = 0 \quad (3.15)$$

Vektor optimalnog pojačanja za te uvjete glasi:

$$\mathbf{K} = -\mathbf{R}^{-1} \bar{\mathbf{B}}^T \mathbf{P} \quad (3.16)$$

Međutim, navedeni izrazi od (3.14) do (3.16) vrijede za uvjet da je broj stanja sustava manji ili jednak dvostrukom broju izlaza ( $n \leq 2p$ ). Budući da je broj stanja sustava  $n = 4$ , a broj izlaza  $p = 1$ , ovaj uvjet nije zadovoljen. Unatoč tome, moguće je da postoji vektor pojačanja  $\mathbf{K}$  kojim bi se dobro moglo upravljati sustavom, međutim izračun tog vektora može biti vrlo složen i zamoran tradicionalnim metodama. Tu dolazi LQR-LMI metoda. U ovoj metodi, umjesto optimalnog kriterija  $J$ , traži se iznos pojačanja za koji je suboptimalni kriterij  $\gamma > J$  minimalan. Sinteza se provodi prema ograničenjima definiranim sljedećim linearnim matricnim nejednadžbama [1], uz iteracijske matrice  $\hat{\mathbf{P}} = \hat{\mathbf{P}}^T \geq 0$  i  $\mathbf{Y}$  prikladne veličine ( $\dim(\hat{\mathbf{P}}) = (m \times n + p)$  i  $\dim(\mathbf{Y}) = (n + p \times n + p)$ ):

$$\begin{bmatrix} \bar{\mathbf{A}}^T \hat{\mathbf{P}} + \hat{\mathbf{P}} \bar{\mathbf{A}}^T + \bar{\mathbf{B}} \mathbf{Y} + \mathbf{Y}^T \bar{\mathbf{B}}^T & \hat{\mathbf{P}} & \mathbf{Y}^T \\ & \hat{\mathbf{P}} & -\mathbf{Q}^{-1} & 0 \\ & \mathbf{Y} & 0 & -\mathbf{R}^{-1} \end{bmatrix} \leq 0, \quad (3.17)$$

$$\hat{\mathbf{P}} \geq 0, \quad (3.18)$$

Suboptimalni vektor pojačanja iznosi:

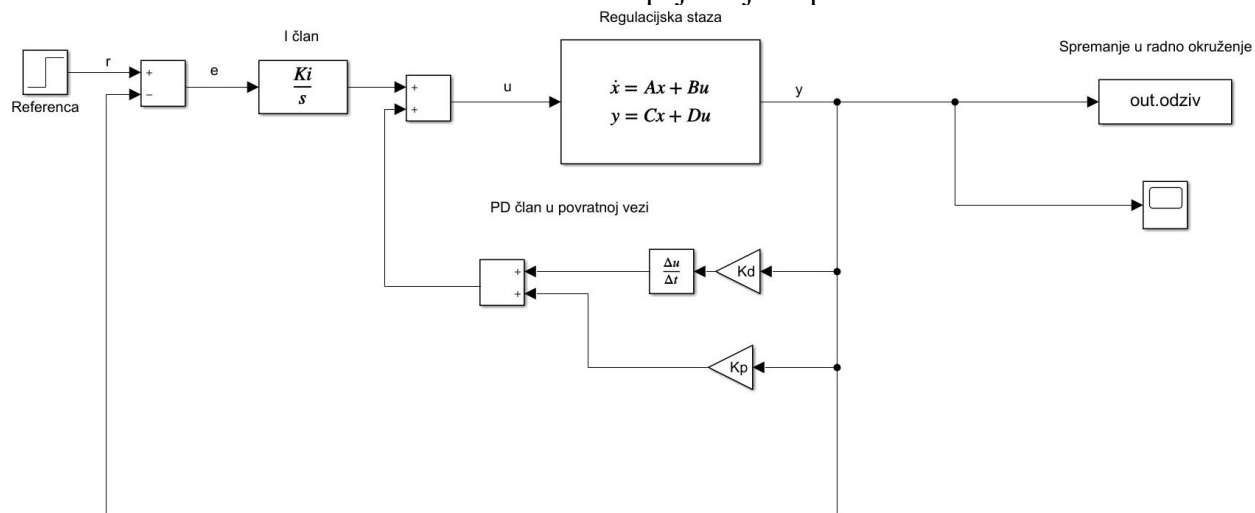
$$\mathbf{K} = \mathbf{Y} \hat{\mathbf{P}}^{-1} \quad (3.19)$$

U MATLAB-u napisana je m-skripta koja uz pomoć YALMIP [5] programskog paketa i SDPT3 [6] solvera rješava opisani LQR-LMI problem. Izvršavanjem m-kôda uz odabrani oblik  $\mathbf{Q}$  i  $\mathbf{R}$  matrica dobivamo matrice  $\mathbf{Y}$  i  $\hat{\mathbf{P}}$  te ih uvrštavamo u (3.19) kako bismo dobili pojačanje  $\mathbf{K}$ . Zatim uvrštavanjem  $\mathbf{K}$  u (3.10) dobivamo augmentirana pojačanja regulatora. Konačno, uvrštavanjem augmentiranih pojačanja u jednadžbe (3.11), (3.12) i (3.13) dobivamo pojačanja regulatora:

$$\begin{aligned} K_I &= -1.9983, \\ K_P &= 0.5886, \\ K_D &= 0.0049. \end{aligned}$$

### 3.3 Računalne simulacije

U prijašnjem poglavlju navedena je metodologija određivanja optimalnih pojačanja regulatora koja najbolje zadovoljavaju definirane kriterije, međutim vrijedi analizirati utjecaj raznih parametara korištenih u sintezi regulatora na ponašanje sustava. Koristeći MATLAB Simulink može se simulirati sustav s izračunatim pojačanjima prema slici 3.4.



Slika 3.4 Simulink model sustava

Provest će se analiza utjecaja vrijednosti težinskih matrica  $\mathbf{Q}$  i  $\mathbf{R}$ . Za linearne kvadratne regulatore, matrica  $\mathbf{Q}$  određuje brzinu prijelazne pojave, dok matrica  $\mathbf{R}$  određuje snagu koja će biti uložena u upravljanje. Za navedeni sustav sa četiri varijable stanja i jednom upravljačkom varijablom, matrice  $\mathbf{Q}$  i  $\mathbf{R}$  poprimaju oblik:

$$\mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix},$$

$$\mathbf{R} = r_{11},$$

gdje su  $q_{ii}$  i  $r_{jj}$  težine pripadnih matrica. Određivanje vrijednosti unutar ovih matrica učinit će se koristeći Brysonovo pravilo koje tvrdi sljedeće:

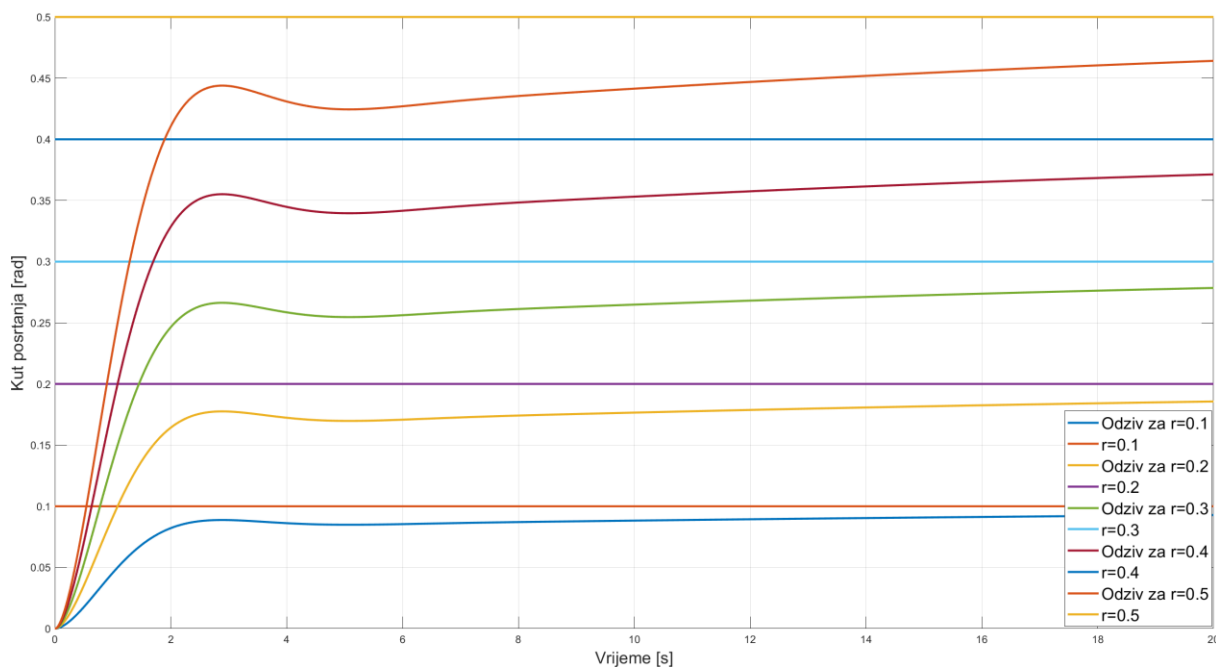
$$q_{ii} = \frac{1}{\text{maksimalno odstupanje } i - \text{te varijable stanja}^2},$$

$$r_{ii} = \frac{1}{\text{maksimalno odstupanje } j - \text{te upravljačke varijable}^2}.$$

Budući da se upravlja kutom posrtanja  $\theta$ , a on odgovara četvrtom članu vektora stanja prema jednačbi (2.4), za sve dijagonalne članove osim  $q_{44}$  stavljamo vrijednost 1, dok  $q_{44}$  pridružujemo maksimalno odstupanje u iznosu od  $5^\circ$  prema Brysonovom pravilu. Analognim



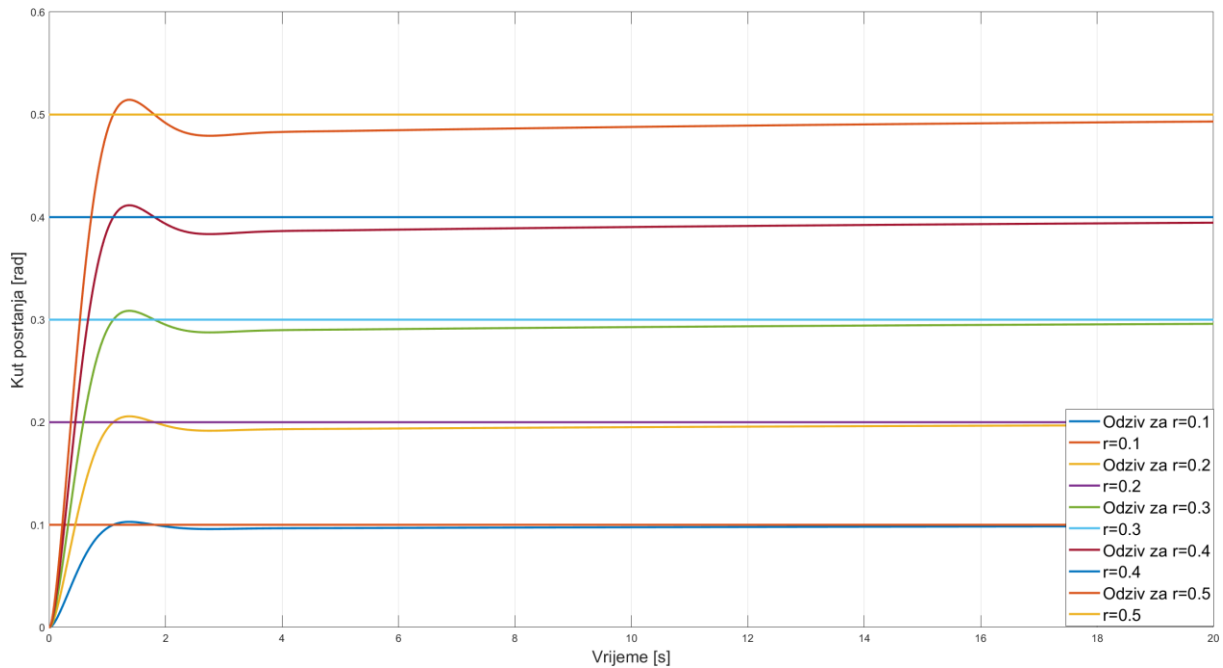
postupkom vrijednosti  $r_{11}$  pridružujemo maksimalno odstupanje u iznosu od  $1^\circ$ . Uz navedene vrijednosti  $\mathbf{Q}$  i  $\mathbf{R}$  dobivamo pojačanja koja sustavom upravljaju prema slici 3.5.



Slika 3.5 Rezultat simulacije za prvi skup ograničenja

$$K_I = -2, K_P = 0.2143 \text{ i } K_D = 0.036$$

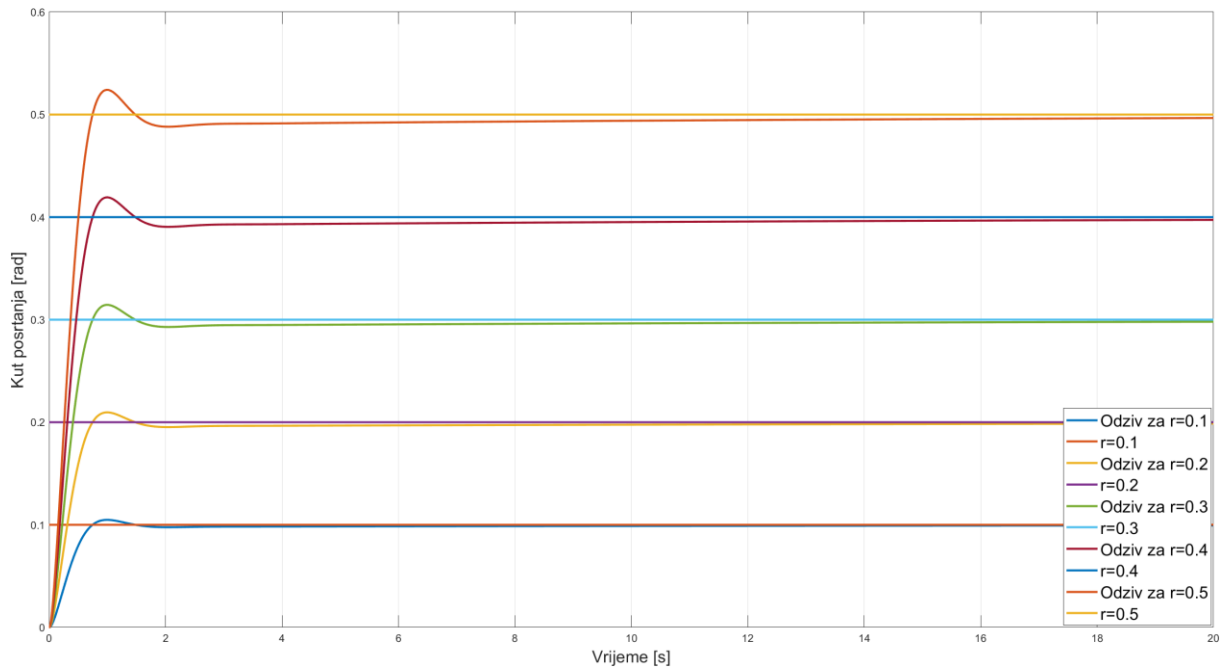
Možemo uočiti da se sustav ponaša tromo te mu je potrebno puno vremena da zauzme zadanu orijentaciju. Očito je da je zadano maksimalno dozvoljeno odstupanje kuta posrtanja preveliko što dovodi do dugog vremena trajanja prijelazne pojave te sporog približavanja referentnoj vrijednosti. Stoga se za sljedeću analizu postavilo maksimalno odstupanje kuta posrtanja u iznosu od  $1^\circ$  dok se matrica  $\mathbf{R}$  održala istom.



**Slika 3.6** Rezultat simulacije za drugi skup ograničenja

$$K_I = -0.9999, K_P = 0.4258 \text{ i } K_D = 0.0043$$

Na osnovi rezultata prikazanih na slici 3.6 može se uočiti puno bolje ponašanje sustava. Prijelazne pojave traju puno kraće te se zrakoplov puno bolje približava referentnim vrijednostima. Za veće skokove reference ostvaruje se veći prebačaj te sustavu treba više vremena da se približi referentnoj vrijednosti kuta posrtanja. Za još bolje ponašanje sustava, odabiremo još manje maksimalno odstupanje kuta posrtanja u iznosu od  $0.5^\circ$ . MEMS senzori koji se koriste u svrhe mjerenja orijentacije zrakoplova redovito mogu mjeriti vrijednosti kuteva ispod  $1^\circ$ , stoga sigurno možemo odabrati ovo ograničenje. Kao i u prijašnjem slučaju, držimo  $\mathbf{R}$  konstantnim.



Slika 3.7 Rezultati simulacija za treći skup ograničenja

$$K_I = -1.9983, K_P = 0.5886 \text{ i } K_D = 0.0049$$

Prema rezultatima prikazanim na slici 3.7 ponovo možemo primijetiti puno bolje ponašanje sustava, pogotovo za male skokove reference. Za veće skokove ponovo možemo uočiti sporije približavanje referentnoj vrijednosti nakon prebačaja, ali je ponašanje puno bolje nego u prijašnja dva slučaja. Budući da se daljnjom promjenom  $Q$  približavamo ograničenjima senzora, a uz zadanu vrijednost  $R$  se nalazimo blizu ograničenja djelovanja aktuatora, zaključujemo analizu i nastavljamo s dobivenim pojačanjima.

## 4. REALIZACIJA SUSTAVA U STVARNOSTI

Iako simulacije pokazuju povoljno ponašanje sustava, nameće se pitanje da li bi se sustav u stvarnosti ponašao sukladno. Naime, povoljno ponašanje u simulacijama ne garantira istovjetnost ponašanja sustava u stvarnosti. Razlog ovome je činjenica da modeli kojima se služimo su pojednostavljeni u odnosu na one stvarne kako bi se olakšala sinteza regulatora te kako bi se cijela problematika projekta mogla spustiti na razinu sposobnosti jednog studenta preddiplomskog studija mehatronike i robotike. Također se uz linearni model omogućuje korištenje metoda upravljanja linearnih sustava kao što su linearni prostor stanja, prijenosne funkcije, linearni regulatori itd. Kao što je navedeno više puta, zrakoplov je izrazito nelinearan sustav. Stabilizacijski derivativi u matricama **A** i **B** se u stvarnosti bitno mijenjanju s promjenom raznih parametara leta zrakoplova (Machov broj, napadni kut itd.), stoga se može očekivati ponašanje koje bitno odstupa od onog kojeg opisuje linearni model, posebice prilikom upravljanja sustavom sa referentnim signalima koji su numerički bitno veći od onih u radnoj točki (0 rad). Također se regulator mora realizirati koristeći digitalnu tehnologiju uz diskretne signale što unosi kašnjenje koje ne bi postojalo da se služimo kontinuiranim regulatorima. Ovim pristupom, unatoč navedenim manama, služilo se iz više razloga:

- Fleksibilnost - moguće izmijeniti ponašanje regulatora bez mijenjanja strujnog kruga
- Jednostavnost - ponašanje regulatora određeno je napisanim softverom, a ne električnim komponentama što bitno olakšava realizaciju
- Oblik signala korištenog senzora nije kompatibilan s kontinuiranim regulatorima (digitalni signali)
- Signal potreban za upravljanje avionom je digitalnog tipa (PWM)

Aktuatori koji se koriste u stvarnosti također ograničuju performanse sustava. Korištenjem biblioteka za upravljanje servo motorima moguće je (bez pisanja vlastitog softverskog paketa) ostvariti rezoluciju zakretanja osovine u iznosu od  $1^\circ$ . Također postoji ograničenje pomaka servo motora u iznosu od  $-90^\circ$  do  $90^\circ$  što znači da se ne može ostvariti neke zakrete upravljačkih površina koji bi se naizgled mogli ostvariti u simulacijama. Uzevši sve ovo u obzir jedino izgradnjom sličnog sustava u stvarnosti može se dobiti odgovor na stvarnu kvalitetu obavljenog zadatka u prijašnjim poglavljima

### 4.1. Elektronika i elektrika

#### 4.1.1. Mikroupravljač

Kao temelj za realizaciju regulacijskog uređaja odabran je mikroupravljač ESP32 – WROOM – 32D. Razlog tome je činjenica da je ovaj uređaj bitno napredniji i sofisticiraniji od onih koji se koriste u tradicionalnim Arduino razvojnim pločicama (ATMEGA familija čipova) te je istovremeno vrlo jeftin. Također ga je moguće programirati u Arduino okruženju. Nadalje, usporedbom specifikacija navedenog ESP32 mikroupravljača i najnaprednijeg Arduino sustava Arduino Mega mogu se dočarati prednosti ESP32 kao što je prikazano u tablici 4.1.

**Tablica 4.1** Neke od važnijih specifikacija razmatranih mikroupravljača

SPECIFIKACIJA	ESP32-WROOM-32D	Arduino Mega
Napon	3.3V	5V
<b>Frekvencija CPU</b>	<b>240 MHz</b>	<b>16 MHz</b>
I/O priključci	34	54
PWM kanali	16	15
Veličina	50 mm x 26.4 mm,	101.52 mm x 53.3 mm

Najvažnija specifikacija radi koje je odabran ovaj mikroupravljač je frekvencija upravljačke jedinice. Što je kašnjenje tijekom uzorkovanja manje, to je rad sustava bliži kontinuiranom sustavu opisanom u prijašnjim poglavljima. Što je veća frekvencija rada CPU-a, to će manje sustav kasniti i brže obavljati naredbe.

Također je povoljno imati kontroler koji radi na manjem naponu (3.3 V) budući da potrebni senzori (žiroskopi i akcelerometri) se uglavnom realiziraju u SMD tehnici za koju se često koriste naponski nivoi iznosa 3.3 V ili manje. Manji radni napon također znači da kontroler troši manje energije iz baterijskog paketa što sprečava bespotrebne gubitke energije i minimizira utjecaj upravljačke elektronike na domet aviona tj. vrijeme letenja. Sve ovo nije bez nedostataka – servo motori koji se koriste u zrakoplovu rade na naponskom nivou 5 V, što znači da je potrebno posmaknuti logički nivo gore/dolje, ali ovo nije teško ostvariti uz korištenje tzv. *logic shiftera* koji se lako mogu nabaviti i koristiti.

Osim vrlo dobre frekvencije procesora, ESP32 je fleksibilniji za programiranje nego bilo koja Arduino razvojna pločica. Na primjer, ESP32 pruža mogućnost pridruživanja hardverskih prekida na bilo koji ulazno/izlazni pin. Za usporedbu ATMEGA 2560 (mikroupravljač na Arduino Mega razvojnoj pločici) pruži samo 6 specifičnih pinova koji mogu aktivirati prekid (pinovi 2, 3, 18, 19, 20 i 21).

#### 4.1.2. Senzor

Budući da je riječ o sustavu čija je upravljana veličina kut, potrebno je imati senzor koji se prikladan za mjerenje te veličine. Za obavljanje ove zadaće odabran je senzorski paket MPU6050 prikazan na slici 4.1. Riječ je o 3-osnom žiroskopu i 3-osnom akcelerometru („6 DOF“ – 6 stupnjeva slobode) koji mjeri kutne brzine te linearne akceleracije oko 3 osi. Od 6 stupnjeva slobode čije mjerenje senzor pruža koristilo se samo jedan – kutna brzina oko  $z$  – osi čiji integral odgovara kutu posrtanja za odabrani način montiranja senzora u letjelici. Pristup mjerenja derivacije veličine kojom se upravlja te njezina integracija je vrlo povoljan budući da je prema zakonu regulacije I-PD regulatora potrebno derivirati stvarnu vrijednost mjerene veličine. U protivnome bi smo morali derivirati mjerenu veličinu, a time bismo unijeli šum u sustav što je neprihvatljivo i izrazito nepovoljno za upravljanje.

Važnije specifikacije senzora:

- Digitalni izlazni oblik akceleracija i kutnih brzina s mogućnosti korisničkog postavljanja raspona skala ( $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  i  $\pm 2000^\circ/\text{sec}$  za žiroskop te  $\pm 2g$ ,  $\pm 4$ ,  $\pm 8$  i  $\pm 16g$  za akcelerometar)

- Integrirani 16 – bitni ADC za istovremeno uzorkovanje svih žiroskopa i akcelerometra (nepotrebno trošiti procesorsku snagu na AD pretvorbu)
- Niska radna struja (3.6 mA za žiroskop i 500  $\mu$ A za akcelerometar)
- Integrirani digitalni nisko propusni filter (povoljno budući da su MEMS sustavi podložni visoko frekvencijskom šumu)
- I2C komunikacija – brzina mjerenja ovisi o brzini komunikacije
- Radni napon 3.3V



Slika 4.1 MPU6050 senzor

#### 4.1.3. Radio prijemnik i predajnik

Za radio predajnik odabran je FlySky Paladin PL18 iz jednostavnog razloga što je bio dostupan za posudbu od strane suradnika uz čiju je pomoć izrađen avion. Odabrani prijemnik je FTTr8B, sa 8 kanala za upravljanje servo motorima i glavnim motorom zrakoplova. Prijemnik se može konfigurirati preko sučelja na predajniku kako bi se određeni kanali prijemnika pridružili određenim aktuatorima na avionu te se odredio zakon njihovog upravljanja (npr. karakteristika pomaka „gljivice“ i kretanja određenog aktuatora za bilo koju upravljačku površinu, *thrust curve* motora...). Prije leta potrebno je podesiti način rada zrakoplova kako bi se ostvarilo optimalno letačko ponašanje. Svaki kanal koji je spojen na servo motore ili na upravljačku elektroniku motora upravlja se PWM signalom frekvencije 50 Hz.

#### 4.1.4. Servo motori

Odabir servo motora za RC avion ovisi o brzini letenja (tj. o iznosu aerodinamičkih sila koje djeluju na avion), konfiguraciji aviona (smještaj, tip i konstrukcija krila) te o veličini aviona. Budući da je riječ o sporo letećem avionu s relativno malim upravljačkim površinama, a i o potrebi za minimiziranjem dodatne težine, odabrani su servo motori s plastičnim zupčastim prijenosom SG90S izlaznog momenta 1.6 kg cm te raspona gibanja  $\pm 90^\circ$ . Na odabir je također imao utjecaj iskustveni savjet suradnika uz čiju pomoć je izrađen avion. Ovaj model servo motora se danas često koristi u RC modelarstvu radi male mase, pouzdanosti te financijske isplativosti.

#### 4.1.5. Pogonski motor

Prema savjetu suradnika odabran je motor bez četkica iRotor 5010/750kV prikazan na slici 4.2. Ovo je motor koji pruža razmjerno velik izlazni moment. Ako se želi konstruirati avion koji leti sporo, uz odgovarajuću konstrukciju krila, potrebno je odabrati motor koji nema visok broj okretaja kako se ne bi stvorila velika akceleracija. Veći moment takvog motora omogućuje korištenje propelera većeg promjera što poboljšava efikasnost jer nije potreban velik broj okretaja za razvijanje iste potisne sile. Stoga je potrebno koristiti motor koji ima velik moment i nizak broj okretaja te velik propeler. Nadalje, veći propeleri imaju veću iskoristivost nego manji, stoga takav izbor motora i propelera povoljno utječe na produljenje vremena letenja.

Prema podatkovnoj tablici motora, pri nominalnom napajanju iznosa 11.1 V (3S LiPo baterija) ovaj motor se u praznom hodu vrti brzinom  $8200 \text{ min}^{-1}$  uz struju 0.6 A. Za maksimalno opterećenje pruža 15 A struje.

Općenito, za istosmjerne motore bez četkica vrijede sljedeće zakonitosti:

$$n_n = K_v U, \quad (4.1)$$

gdje je  $K_v$  naponska konstanta motora dimenzije  $\text{min}^{-1} \text{ V}^{-1}$ . Umnoškom napona napajanja i ovog parametra dobivamo  $n = 8250 \text{ min}^{-1}$  što se poklapa s podacima iz proizvođačevih tablicama. Moment motora može se dobiti preko sljedeće zakonitosti:

$$M_T = K_T U, \quad (4.2)$$

gdje je  $K_T$  momentna konstanta motora dimenzije  $\text{Nm V}^{-1}$ .  $K_T$  može se izračunati prema:

$$K_T = \frac{60}{2\pi K_v} \approx 0,0127 \text{ Nm V}^{-1} \quad (4.3)$$

Prema navedenim zakonitostima moment motora iznosi  $M_T = 0.141 \text{ Nm}$  za nazivni napon.



Slika 4.2 iRotor 5010/750kV motor

#### 4.1.6. Kontroler brzine vrtnje

Budući da je odabrani motor tipa BLDC, a takvim motorima je potreban trofazni napon, potrebna je odgovarajuća upravljačka elektronika koja će omogućiti naponsku pretvorbu iz DC u AC. Tu zadaću ima tzv. kontroler brzine vrtnje (eng. *electronic speed controller*, skraćeno ESC). Osim energetske pretvorbe omogućuje upravljanje brzinom vrtnje motora. Upravljanje se obavlja pomoću PWM signala iz prijemnika. Odabran je ESC HobbyWing Skywalker V2 maksimalne struje 60 A (prikazan na slici 4.3). Budući da je maksimalna struja motora 15 A, ESC zadovoljava potrebe te može pogoniti motor. Odabran je puno jači ESC radi dodatne sigurnosti te mogućnosti nadogradnje pogona u budućnosti.



Slika 4.3 Hobby Wing Skywalker ESC



#### 4.1.7. Baterijski paket

Za napajanje cjelokupnog sustava potreban je prijenosni izvor istosmjerne struje. U tu svrhu odabrana je LiPo baterija odgovarajućeg kapaciteta naboja i izlazne struje. Na odabir baterije utječe potrebni napon (broj ćelija baterije), njezin kapacitet u mAh te *current rating*.

Najveće trošilo u sustavu je pogonski motor. Pri maksimalnom opterećenju struja koja je motoru potrebna iznosi 15 A. Uz pretpostavku da je potrošnja ostalih komponenata zanemariva u odnosu na potrošnju motora te uz poznavanje standardnih iznosa kapaciteta baterija, može se metodom iteracije doći do odabira kapaciteta baterije provjerom maksimalnog mogućeg vremena dobave električne energije. Tako je za navedenu struju odabrana baterija kapaciteta  $Q = 3300$  mAh koja uz navedene pretpostavke pruža vrijeme letenja 13.2 min. Budući da se avion neće letjeti pod stalnim maksimalnim iznosom potiska, odabrani kapacitet bi trebao pružati razmjerno dugo vrijeme letenja.

Također je potrebno osigurati da baterija može pružiti potrebnu struju. Relevantni parametar u ovome slučaju je *current rating*, *discharge rate* ili jednostavno *C – rating*. Radi sigurnosnih razloga, te radi mogućnosti montiranja jače elektronike u budućnosti, odabran je rating 60C koji omogućava pražnjenje baterije maksimalnom strujom 198 A prema zakonitosti:

$$I_{max} = CQ \quad (4.4)$$

Broj ćelija odabran je prema željenoj vrtnji motora. Za  $8200 \text{ min}^{-1}$  potreban je napon u iznosu 11.1 V. Budući da su ćelije LiPo baterija proizvedene s izlaznim naponom 3.7 V, za 11.1 V potrebno je imati 3 ćelije, tj. baterija mora biti 3S.

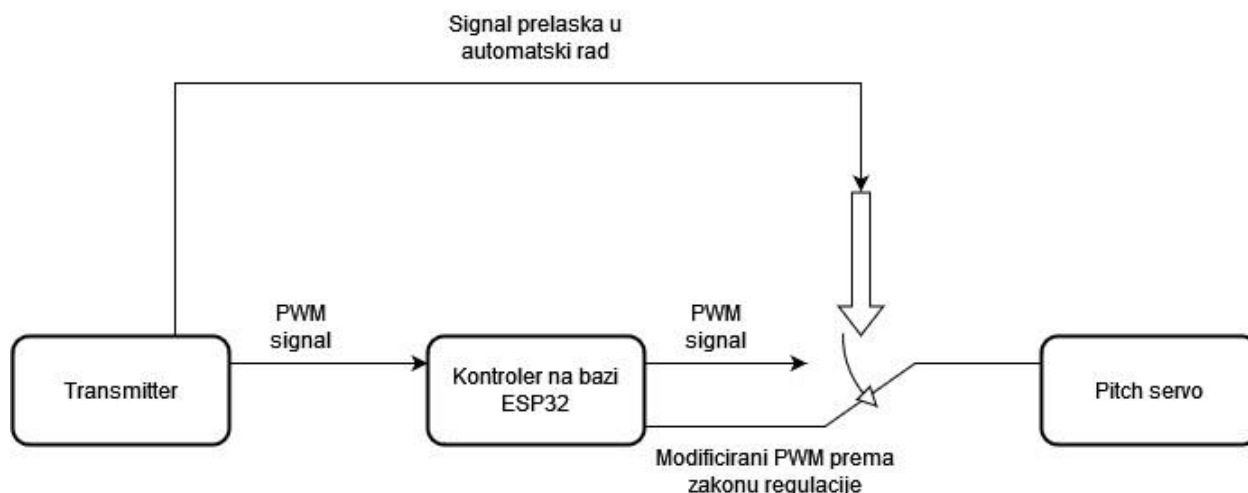
#### 4.1.8. Osnovni princip rada

U nominalnom radu RC avion upravljan je ručnim radio predajnikom navedenim u prijašnjem poglavlju. Očitanjem upravljačke naredbe na joystickovima ručnog upravljača predajnik kodira upravljačku informaciju i šalje ju prijemniku putem radio valova. Prijemnik zatim dekodira podatke i generira odgovarajući PWM signal koji vratila odgovarajućih servo motora postavlja u zadanu orijentaciju. Ta promjena pozicije vratila motora se preko žičanog mehanizma preslikava na upravljačke površine – *aelerone*, *elevator* i *rudder*. Svi motori se spajaju direktno na transmitter osim pitch motora koji je spojen na ESP32. ESP32 je zatim spojen na transmitterov PWM, 5V i GND izvod za komunikaciju i napajanje. Kada je cijeli sustav u ručnom načinu rada, ESP32 PWM signal prosljeđuje na servo motor koristeći hardverski prekid na sljedeći način:

1. Hardverski prekid se aktivira promjenom logičkog stanja na odgovarajućem pinu (reakcija na bridove)
2. Aktivacijom prekida u prekidnoj podrutini provjerava se je li priključak za koji je vezan prekid u visokom ili niskom logičkom stanju
3. Očitano logičko stanje na priključku postavlja se na drugi (upravljački) pin mikroupravljača koji je spojen na upravljački vod *pitch* servo motora

Dobiveni efekt je da ESP32 u ručnom načinu rada prosljeđuje PWM signal od prijemnika prema motoru.

Aktivacijom sklopke na ručnom upravljaču stvara se naponski signal koji odgovara visokom naponskom stanju na jednom od priključaka prijemnika. Hardverskim prekidom koji reagira na promjenu logičke razine napona prekida se prosljeđivanje PWM signala, inicijalizira se programski paket za upravljanje servo motorom putem kôda u upravljaču (Servo biblioteka) te se pokreće timer koji određuje koliko često se mjeri kut posrtanja i računa upravljački signal u softverski implementiranom I-PD regulatoru. Ponovnom aktivacijom sklopke zaustavlja se i resetira timer, deinicijalizira se programski paket za upravljanje servo motorom i ponovo se uključuje prosljeđivanje PWM signala. Opisani princip rada dodatno je objašnjen shemom prikazanom na slici 4.4.



Slika 4.4 Shematski princip rada elektronike

## 4.2. RC avion

Prema zahtjevima navedenima u poglavlju 2.1 bilo je potrebno pristupiti konstrukciji aviona. Glavni dijelovi od kojih se sastoji zrakoplov su krila (glavno i stabilizacijsko), električna i elektronika (regulator, prijemnik, baterija, servo i pogonski motori).

### 4.2.1. Krilo, horizontalni i vertikalni stabilizator

Glavno krilo zrakoplova raspona je u iznosu od 1.55 m. Potreban je dovoljan raspon kako bi avion mogao generirati dovoljan uzgon pri nižim brzinama leta. U ovom projektu poželjne su niže brzine gibanja kako u slučaju gubitka kontrole ne bi došlo do katastrofalne štete na zrakoplovu.

Radi niske cijene i lake oblikovljivosti krilo je izrađeno od ekstrudiranog polistirena – XPS pod trgovačkim nazivom *Climapor* ili *Depron*. Ovaj se materijal u zadnjih deset godina često koristi u svrhe RC modelarstva radi navedenih povoljnih karakteristika. Nabavljeni XPS površine 2.5 m<sup>2</sup> i debljine 6 mm se izrezao na 4 komada dimenzija 77.5 cm x 20 cm. Radi načina na koji je isporučen XPS u dućanu (dvije ploče duljine 1.25 m) krilo se moralo izraditi u dva dijela te sučeljno spojiti ljepilom. Ljepilo korišteno u proizvodnji krila moralo je biti kompatibilno s poroznim polimerima kao što su XPS, stoga je korišteno ono pod trgovačkim nazivom *UHU Por*. Korištenjem nekompatibilnih ljepila došlo bi do rastapanja materijala.

Nakon rezanja XPS-a na odgovarajuće dimenzije, na prednjoj i stražnjoj strani komada morala se brusiti kosina kao što je prikazano u slici 4.5. Njena je funkcija omogućiti bolje uvjete za adheziju ljepila. Na tim kosinama će se kasnije spojiti gornja i donja polovica krila.



**Slika 4.5** a) Stražnja kosina b) Prednja kosina

Izbrušeni komadi XPS-a koji tvore donju polovicu krila zatim su sučeljno spojeni *UHU Por* ljepilom (slika 4.6). Na mjestu spajanja dodatno je nalijepljena tkanina od staklenih vlakana koristeći dvokomponentno ljepilo.



**Slika 4.6** Sučeljni spoj

Također su ucrtane pozicije četiri letvica (jedna od drva, a ostale tri od XPS-a, plave linije na slici 4.6). Te letvice se pričvršćuju *UHU Por* ljepilom za sučeljno spojenu podlogu od XPS-a. Drvena letvica služi za ukrućivanje krila na savijanje te je postavljena na ~30% duljine tetive krila jer se tamo javljaju najveća naprezanja. Ostale tri letvice od XPS-a služe za ostvarivanje željenog oblika profila krila.

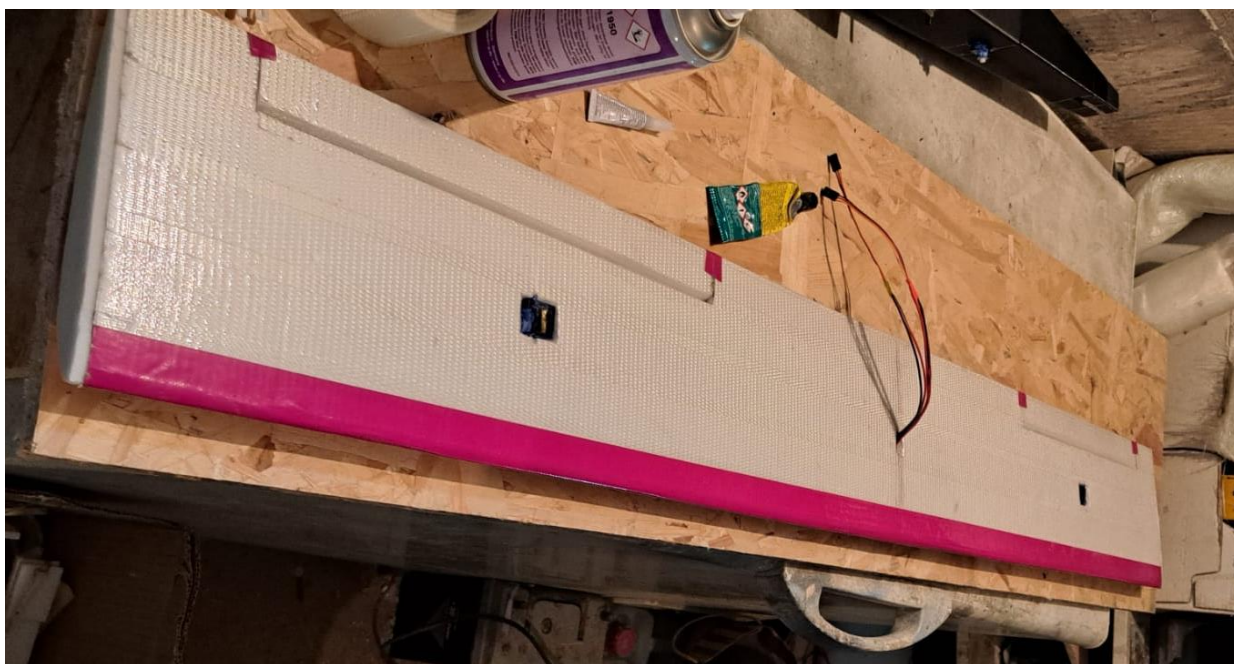


**Slika 4.7** Unutarnja struktura krila

Nakon uspješne izrade donje polovice krila, komadi XPS-a koji odgovaraju gornjoj polovici savinuti su preko letvica te su zalijepljeni za njih i za prethodno spomenuta skošenja. Nadalje, gornji dio krila je također sučeljno spojen *UHU Por*-om te je mjesto spoja ojačano identičnom metodom kao onom navedenom za donji dio krila. Prikaz strukture gotovog glavnog krila na kojoj su jasno vidljive potporne letvice može se vidjeti na slici 4.7.

Prema naputku suradnika uz čiju pomoć je izrađen zrakoplov, točna geometrija horizontalnog i vertikalnog stabilizatora nije bitna za ovaj RC avion. Stoga je horizontalni stabilizator napravljen u obliku pravokutnika dimenzija 130 mm x 400 mm, debljine 6 mm, dok je vertikalni stabilizator napravljen tako da je visine 155 mm, debljine 6 mm, minimalne i maksimalne širine 90 mm i 230 mm.

*Rudder, elevator* i *aeroni* su izrađeni izrezivanjem iz krila (slika 4.8). Na odgovarajuće upravljačke površine učvršćena je plastična „žabica“ te metalni prihvatni element koji služi za prihvat čelične žice koja uz servo motore tvori mehanizam aktucije upravljačkih površina. Rotacija upravljačkih površina ostvarena je fleksijskim mehanizmom (slika 4.9) od ljepljive trake. Također su urezani utori za servo motore koji upravljaju *aeronima*.



Slika 4.8 Aeleroni i servo motori nakon montaže u krilu



Slika 4.9 Fleksijski mehanizam zakretanja

Konačno, kako bi se dodatno ukrutilo krila i upravljačke površine, sva krila omotana su ljepljivom trakom ojačanom staklenim vlaknima (slika 4.10). Dodatno je na prednju ivicu glavnog krila zalijepljena obična ljepljiva traka kako bi se taj dio dodatno učvrstio.



Slika 4.10 Krilo ojačano staklenim vlaknima

#### 4.2.2. Tijelo aviona

Tijelo aviona isprintano je FDM 3D printerom koristeći PLA i PETG filament. Cijelo tijelo isprintano je u tri dijela radi visinskog ograničenja korištenog 3D printera. Dijelovi su zalijepljeni ljepljivom trakom trgovačkog naziva *Cyanofix*. Debljina isprintanog modela iznosi 2 mm radi maksimalnog balansa čvrstoće na udarce i mase trupa. Ručnom pilom izrezani su dosjedi za glavno krilo te vertikalni stabilizator. Također su ispiljeni utori za montažu *pitch* i *yaw* servo motora (slika 4.12) te su izbušene rupe za provlačenje energetskih vodova BLDC motora.



Slika 4.11 Montirani BLDC



**Slika 4.12** Servo motori nakon montaže i utor za vertikalni stabilizator

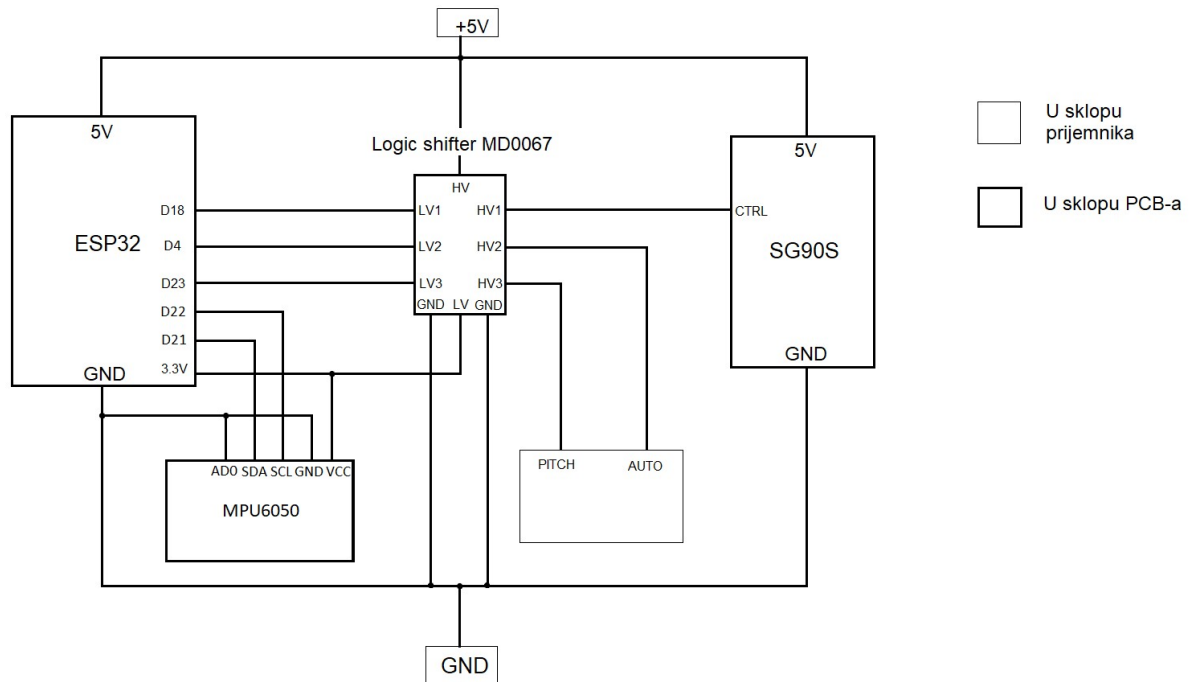
S unutarnje strane tijela nalazi se šupljina za postavljanje elektroničkih komponenti. Komponente poput baterije i ESC su učvršćene na komad *šperploče* uz pomoć *Velcro* čičaka. Drvene ploče s navedenim komponentama su učvršćene za unutrašnjost aviona vrućim ljepilom. Potrebno je dobro odabrati poziciju komponenta kako bi se ostvario stabilni položaj težišta aviona. BLDC motor učvršćen je tvorničkim vijcima za komad *šperploče* te je montiran na tijelo aviona *Cyanofixom* (slika 4.11). Dodani su i prijemnik te PCB s mikroupravljačem. Konačno, dodana su dva šuplja karbonska štapa s prednje i stražnje strane dosjeda krila za koje će se, uz pomoć gumene trake, učvrstiti krilo.

#### **4.2.3. Olova za balansiranje**

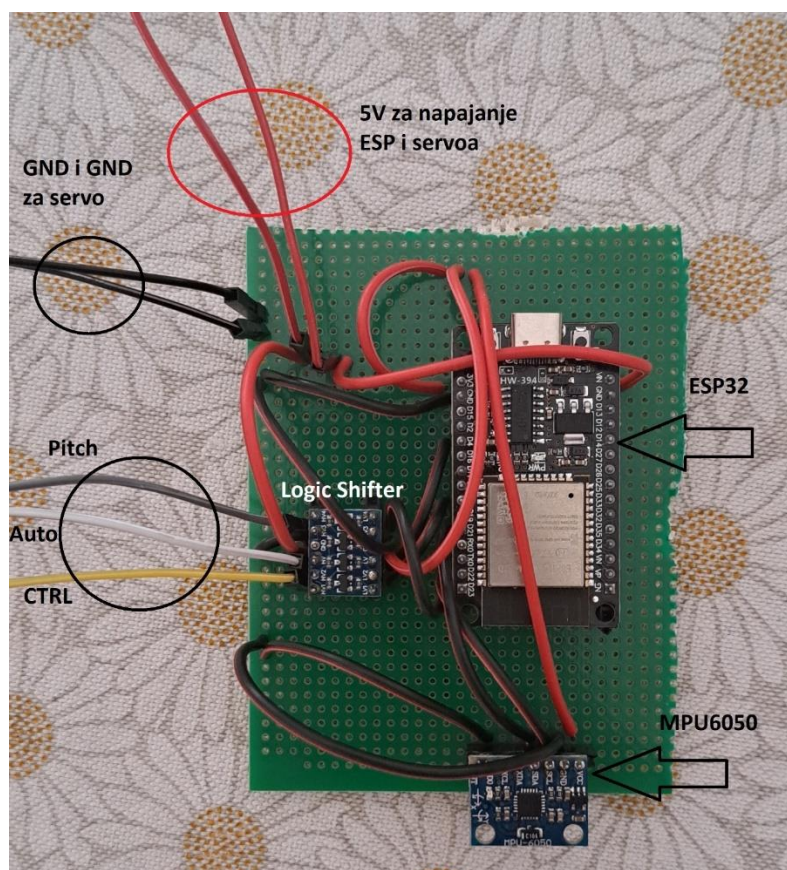
Za odabrani raspored elektroničkih komponenta i krila, težište se nalazilo u nepovoljnom položaju koje bi značajno otežalo upravljanje letjelicom. Kako bi se ovaj problem riješio, dva olova ukupne mase 40 g su se montirala na nos zrakoplova na udaljenosti od 23 cm od napadne ivice glavnog krila. Uz pomoć olova, položaj težišta pozicionirao se na 6.5 cm od napadne ivice krila, u smjeru repa. Time se ostvarilo ponašanje zrakoplova što bliže aerodinamičkoj analizi prikazanoj na slici 2.10.

#### **4.2.4. PCB s mikroupravljačem**

U svrhu lakšeg pozicioniranja i učvršćivanja regulacijske elektronike u model aviona, relevantni elektronički sklopovi složeni su na zajedničku tiskanu pločicu. Komponente su spojene koristeći bakrene kablīce te THT (*eng. through hole technology*) lemljenjem. Električna shema spajanja relevantnih elektroničkih elemenata na tiskanu ploču prikazana je na slici 4.13, a gotov izgled sklopa nakon lemljenja prikazan je na slici 4.14.



Slika 4.13 Električna shema regulacijskog uređaja



Slika 4.14 PCB nakon lemljenja

Sustav se sastoji od 3 glavne cjeline – mikroupravljača, servo motorčića, MEMS senzora i sklopa za posmak logičke razine. Prijemnik preko PITCH kanala predaje 5V PWM

signal prema mikroupravljaču. Budući da je mikroupravljač izveden na naponskoj razini 3.3V, a prijemnik i servo rade na 5V, potrebno je sniziti logičku razinu na niži radni napon ESP32 putem sklopa za posmak logičke razine MD0067. Taj se PWM signal preko prekidne rutine šalje na upravljački pin servo motora odgovornog za pitch (naravno nakon ponovnog posmaka na odgovarajući 5V naponski nivo). Efekt je da ESP32 u ručnom radu služi kao „middle man“ koji prosljeđuje signal generiran u prijemniku. U trenutku kada se detektira promjena logičkog stanja na AUTO pinu (koji odgovara sklopki na ručnom upravljaču) ESP32 prekida prosljeđivanje PWM signala iz prijemnika te PWM postavlja prema zakonu regulacije. Također se provodi analogni postupak posmaka logičke razine i u ovom slučaju. U trenutku kada se desi ponovna promjena stanja na AUTO pinu (što odgovara prelasku iz automatskog u ručni rad), PWM signal iz prijemnika se ponovo prosljeđuje, a regulacija se gasi. Najveći problem kod ostvarivanja ovakvog načina rada jest činjenica da je signal na AUTO pinu također PWM frekvencije 50 Hz, širine poluvala u iznosu od 1 ms za nepritisnuto, a 2 ms za pritisnuto stanje. Razlog tome je način izvedbe prijemnika te pretpostavka od strane proizvođača da će korisnik koristiti dostupne sklopke za aktivaciju servo motora koji upravljaju određenim sporednim funkcijama (npr. pomicanje zakrilca, nogara za slijetanje, zračnih kočnica...). Stoga ako korisnik želi znati logičko stanje sklopke, mora mjeriti duljinu trajanja poluvala PWM signala.

#### 4.2.5. Arduino kôd regulatora

Kôd je sačinjen od tri prekidne rutine – dva hardverska prekida koji reagiraju na promjenu naponske razine te timer za osiguravanje željenog vremena uzorkovanja. Odabrano vrijeme uzorkovanja mora biti manje od 10 ms jer za bilo koje veće vrijeme uzorkovanja opada točnost metode računanja integrala. Eksperimentalno je utvrđeno da vrijeme učitavanja mjerenja na žiroskopu uz korištenje mikroupravljač 2.5 ms, stoga je odabrano dvostruko veće vrijeme uzorkovanja. Vrijeme izvođenja naredbi u `loop` (izuzev mjerenja na žiroskopu) je zanemarivo malo zbog jednostavnosti naredba (aritmetičke operacije, logičke operacije, pridruživanje vrijednosti varijabli...).

Jedan hardverski prekid (`pwmInterrupt`) služi za prosljeđivanje upravljačkog signala prema servo motoru. Prekidna rutina provjerava stanje napona na PITCH pinu (`pwmInput`, priključak D23) te postavlja identično logičko stanje na CTRL pin (`pitchPin`, priključak D18) koji je spojen na sami servo motor. Drugi hardverski prekid se aktivira promjenom naponskog nivoa na AUTO pinu (`autoSetPin`, priključak D4) te postavlja logičku varijablu `isAuto` na vrijednost koja odgovara ručnom (`false`) ili automatskom (`true`) radu. Također povezuje `Servo` biblioteku te analogno prekida njezin rad u slučaju ponovnog prelaska u ručni rad.

Timer nakon svakih odbrojenih 10 ms aktivira prekidnu rutinu koja postavlja logičku varijablu `measureAndControl` u stanje `true`. To znači da se može izvesti mjerenje i računanje upravljačkog signala u `loopu`. Razlog zašto se I-PD rutina provodi u `loopu`, a ne u prekidnoj rutini je zbog potrebe za što manjim zadržavanjem u prekidnoj rutini, što može dovesti do nepravilnog rada mikroupravljača.

U `setup` se odvija inicijalizacija prekida, timera, načina rada pinova te žiroskopa. U `loop` se ništa ne odvija osim kada je pokrenut timer koji regulira frekvenciju uzorkovanja. Uzorkovanje se provodi kontinuirano bez obzira na način rada kako bi mikroupravljaču uvijek bila dostupna informacija u osnovnoj orijentaciji letjelice. U trenutku aktivacije prekidne rutine timera postavlja se stanje `true` na logičku varijablu `measureAndControl`. Kondicionalna naredba `if` se aktivira te se izvodi potprogram koji mjeri linearne i kutne brzine aviona. Kutna brzina koja odgovara osi posrtanja aviona ( $z$  – os senzora) se integrira kako bi se dobio kut



posrtanja. Ako se uređaj nalazi u automatskom radu, računaju se P, I i D djelovanja prema zakonu regulacije I-PD regulatora. Također je uključen *anti-windup* kôd koji sprečava „namatanje“ integratora unatoč zasićenju aktuatora tj. kada se elevator nalazi na maksimumu svojeg hoda. Preuzeti dinamički model zrakoplova kao upravljačku varijablu koristi kut zakreta *elevatora*. Budući da je servo odgovoran za *elevator* spojen preko mehanizma na tu upravljačku površinu, potrebno je dobiti vezu između zakreta servoa i upravljačke površine. Mjerenjem kuta otklona od horizontale *elevatora* u odnosu na kut zakreta servo motora te aproksimacijom polinomom 3. reda dobila se potrebna matematička veza. Nakon što se izračuna izlaz iz regulatora, koristeći napomenuti polinom računa se kut zakreta servo motora te se zadaje na `pitchServo.write()` funkciju koja zakreće vratilo servo motora u potrebnu orijentaciju.

**Tablica 4.2** Tablica zakreta servoa i elevatora (- označava zakret prema gore)

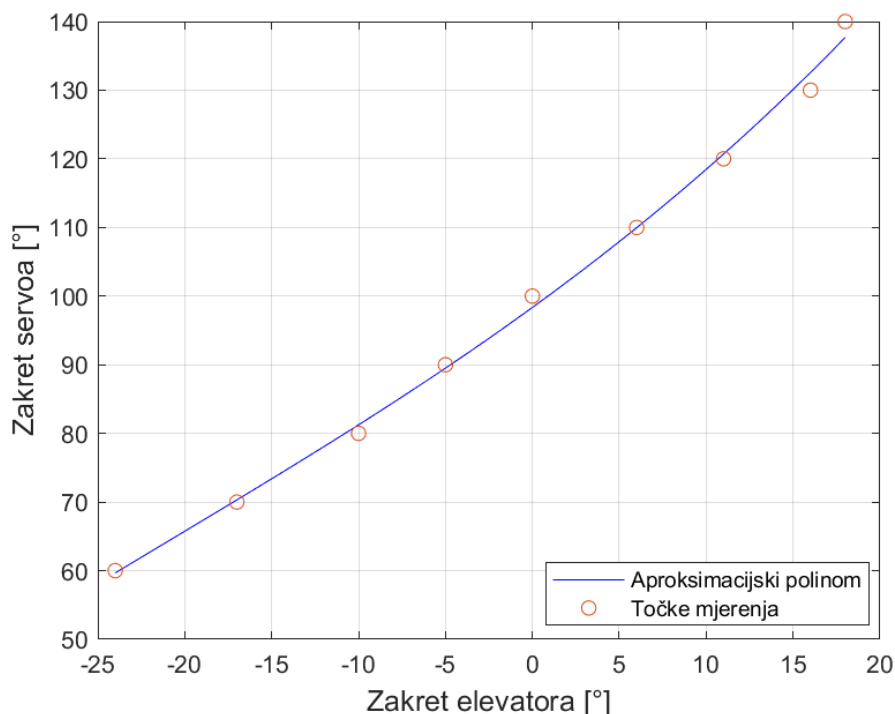
Servo	60°	70°	80°	90°	100°	110°	120°	130°	140°
Elevator	-24°	-17°	-10°	-5°	0°	6°	11°	16°	18°

Koristeći `polyfit()` u MATLAB-u uz podatke iz tablice 4.2, dobiva se sljedeći polinom trećeg stupnja:

$$y_s = 0.00023774 x_e^3 + 0.0151x_e^2 + 1.8371x_e + 98.3472$$

gdje je:

- $y_s$  – kut zakreta servo motora
- $x_e$  – kut zakreta elevatora



**Slika 4.15** Polinom izveden na temelju podataka iz tablice 4.2

## 5. TESTIRANJE LETJELICE I REGULATORA

Prilikom testiranja susrelo se s problemima vezanima za prelazak iz ručnog u automatski rad. Kôd koji upravlja ručnim i automatskim radom radi pouzdano, međutim prilikom prelaska iz automatskog u ručni rad, javljaju se problemi. Iz razloga vezanih za način na koji prekidi u ESP32 rade, ne dolazi do ponovnog uključivanja prekida koji upravlja prosljeđivanjem PWM signala u glavnom programu, stoga nije bilo moguće isprobati zrakoplov s uključenim regulatorom bez riskiranja integriteta letjelice. Ubuduće, kako bi se omogućio pouzdani prelaz iz ručnog u automatski rad, biti će potrebno pronaći alternativni način za ostvarivanje ovog cilja ili se koristiti drugačijim softverskim (ili hardverskim) rješenjem.

Performanse letjelice ispitane su letom kod modelarske piste Lučko. Probni let zrakoplova u stvarnim uvjetima prikazan je na slikama 5.1, 5.2 i 5.3. Zrakoplov je upravljao tijekom serije manevara koji su ispitali njegove performanse prilikom *pitch*, *roll* i *yaw* kretanja. Radi dobrog balansiranja zrakoplova i dobrog „trimminga“ upravljačkih površina, iskustvo letenja je bilo vrlo ugodno i upravljivo, međutim može se primijetiti blaga *nose down* tendencija koja je uzrokovana konfiguracijom težišta u odnosu na COL. Zrakoplov nije napravljen u svrhe visoke manevrabilnosti, ali unatoč tome posjeduje određene agilne kvalitete koje uvelike olakšavaju letenje. Kao što je očekivano, velik raspon krila omogućuje stabilno letenje čak prilikom niskih brzina što olakšava polijetanje i slijetanje. Iako je predviđeno vrijeme letenja u iznosu od 13.2 min na maksimalnom potisku, probni let je trajao 8 minuta i 20 sekundi nakon čega je došlo do gubitka pogona na BLDC motoru. Razlog tome je neistinitost deklariranog kapaciteta baterije. Stvarni kapacitet iznosi približno 1800 mAh – 45.5% manje nego što je očekivano. Radi navedenih karakteristika, sigurno se može zaključiti da će za buduće regulacijske sustave zrakoplov poslužiti kao robustan i kvalitetan ispitni primjerak (uz kvalitetniju bateriju).



Slika 5.1 Letjelica netom prije lansiranja



**Slika 5.2** Letjelica tijekom lansiranja



**Slika 5.3** Letjelica u letu

## 6. ZAKLJUČAK

U ovom završnom radu izrađen je računalni model daljinski upravljano zrakoplova te je provedena sinteza I-PD regulatora za automatsko upravljanje oko poprečne osi. Računalni 3D model načinjen je pomoću Solidworks CAD programskog paketa prema naputcima suradnika s iskustvom u modelarstvu daljinski upravljanih zrakoplova. Za tako izrađeni model, primjenom programskog alata za računalnu analizu dinamike fluida *xflr5* procijenjene su aerodinamičke značajke zrakoplova te su dobivene matrice lineariziranog dinamičkog modela u prostoru stanja. Na osnovu teorijskih razmatranja o sintezi PID regulatora prema kvadratnom kriteriju optimalnosti primjenom linearnih matričnih nejednadžbi, koristeći programski paket YALMIP provedena je sinteza regulatora za automatsko upravljanje zrakoplovom pri čemu je za ulaznu upravljačku varijablu odabran kut otklona kormila *elevatora*. Simulacijama na računalu primjenom programskog paketa MATLAB analizirani su odzivi lineariziranog modela u zatvorenoj regulacijskoj petlji s projektiranim regulatorom. Projektirani regulator implementiran je na mikroupravljaču u zatvorenom krugu s ostalim elementima koji su predviđeni za ugradnju u letjelicu. Na kraju, izrađen je i fizički model letjelice te su izvršena preliminarna testiranja u stvarnim uvjetima. Iz svega prethodno opisanog može se zaključiti da je zadaća završnog rada u potpunosti ispunjena. U svrhu dodatnih aktivnosti u pogledu fizičkog ispitivanja sustava, postavljen je čvrst temelj za nadogradnju rada u budućnosti.

Tijekom rješavanja zadatka bilo je potrebno upotrijebiti svo stečeno znanje iz teorije upravljanja i regulacije, računalnih alata, objektnog programiranja, programiranja mikroprocesorskih sustava i tehnologija izrade. Također je bilo potrebno naučiti neke nove vještine i upoznati se s poljima fizike i matematike koje se nije istraživalo tijekom preddiplomskog studija mehatronike i robotike, kao što su aerodinamika letjelica, linearne matrične nejednadžbe itd.

Iako se može izvući određeno zadovoljstvo s uspješnim izvođenjem zadatka, ima dosta mjesta za unaprjeđenje rada. Poradit će se na boljoj izvedbi softvera za regulator. Također će biti potrebno puno bolje istražiti sposobnosti ESP32 i ne oslanjati se samo na površinske karakteristike kao vodilju za odabir mikroupravljača, nego i na programerske i *real-time* aspekte njihova funkcioniranja. Također bi bilo dobro koristiti profinjeniji MEMS senzor koji bi bolje i pouzdanije mogao mjeriti potrebne varijable stanja za upravljanje zrakoplova. Konačno, ubuduće bi bilo potrebno koristiti bolji CFD softver. Budući da je temelj za izvedbu složenijih i boljih sustava regulacije rada zrakoplova čvrsto postavljen u ovom radu, ubuduće se može implementirati regulator koji će letjelicom autonomno letjeti po zadanoj putanji ili čak tražiti termalne vrtloge koristeći satelitsku kartu kojima bi se značajno moglo sniziti potrošnju energije tijekom leta.

Volio bih se još jedan put zahvaliti mentoru izv. prof. dr. sc. Vladimiru Miliću jer je uvijek bio dostupan i voljan pružiti savjet vezan za rješavanje zadatka kada je to bilo potrebno. Također bih ga volio pohvaliti radi entuzijazma kojim je pristupio ovome radu unatoč manjku iskustva s ovakvim tipom letjelice. Vrlo mi je drago što sam uz pomoć ovog uistinu zanimljivog rada uspio ojačati repertoar znanja stečenog tijekom studiranja na preddiplomskom studiju mehatronike i robotike te se nadam da će u budućnosti ono omogućiti puno efikasnije rješavanje sličnih problema.

---

**LITERATURA**

- [1] Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V. Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia, 1994.
- [2] Durham, W. Aircraft Flight Dynamics and Control. John Wiley & Sons, Chichester, 2013.
- [3] Pradhan, J. K., Ghosh, A. Multi-input and multi-output proportional-integral-derivative controller design via linear quadratic regulator-linear matrix inequality approach. IET Control Theory and Application, Volume 9, Issue 14, pp. 2140 - 2145, 2015.
- [4] Vertical stabilizer – rudder, NASA, raspoloživo na <https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/rud.html>, [16.8.2024.]
- [5] Lofberg, J. YALMIP : a toolbox for modeling and optimization in MATLAB. 2004 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, pp. 284 - 289, 2004.
- [6] Tutuncu, R. H., Toh, K. C., Todd, M. J. Solving semidefinite-quadratic-linear programs using SDPT3, Mathematical Programming Ser. B, 95, pp. 189 - 217, 2003.

---

**PRILOZI**

- I. MATLAB kôd za rješavanje LMI problema
- II. MATLAB kôd za aproksimaciju veze između *elevatora* i pripadnog servo motora
- III. Arduino kôd regulatora

## I. MATLAB kôd za rješavanje LMI problema

```

%% Sinteza regulatora RC aviona
% Prema istraživanju: Multi-input and multi-output
%           proportional-integral-derivative controller
%           design via linear quadratic regulator-linear
%           matrix inequality approach
% Autori istraživanja: Jatin K. Pradhan1, Arun Ghosh

% Matrice prostora stanja sustava: A, B (dobivena analizom stabilnosti u xflr)

A = [-0.0177058      0.395514      0      -9.81;
     -1.24609      -11.9184      14.0021      0;
     1.1448e-05     -128.102     -170.618      0;
     0              0              1              0];
B = [0.384582 -18.2417 -3302.796 0]';

% Vektor stanja: x = [u w q theta]', upravljamo s kutom posrtnanja theta,
% stoga matrica izlaza C glasi:
C = [0 0 0 1];

% Potrebne varijable za sloznost dimenzionalnosti nadolazećih matrica
p =1; % broj izlaza
m =1; % broj ulaza
n =4; % broj stanja

% Augmentirane matrice A i B prema (11)
A_aug = [A zeros(n, p);-C zeros(p, p)];
B_aug = [B; zeros(p, m)];

% optimizacijske varijable Y i P LMI metode:

Y = sdpvar(m, n+p);
P = sdpvar(n+p, n+p);

% Težinske matrice Q i R za LQR, Brysonovo pravilo
Q = 1*diag([1 1 1 1 1/((1*pi/180)^2)]);
R = 1/((1*pi/180)^2);

% Linearna matricna nejednadžba prema (5)
lmi = [A_aug*P+P*A_aug'+B_aug*Y+Y'*B_aug' P Y';P -Q^-1 zeros(n+p,m);Y zeros(m,n+p)
-R^-1];
% Uvjeti optimizacije prema (5)
% Komentar - P>=0 umjesto P>0 zato što striktnu nejednakost > solver ne
% podržava
lmi_uvjeti = [lmi<=0, P>=0];

% Postavke solvera
options = sdpsettings('solver','sdpt3');

% Pozivanje solvera sdpt3
solvesdp(lmi_uvjeti, -trace(P), options);

% Prebacivanje u matlabove varijable iz varijabli solvera
Y = double(Y);
P = double(P);

% State feedback prema 3.2
K = Y/P;

```

---

```
% Matrica S prema (15) za prebačaj pojačanja iz state feedback u PID formu
S = [C 0; C*A 0; zeros(p,n), 1];

% Augmentirana matrica pojačanja PID prema (16)
K_aug = K/S;

% Konačna pojačanja prema (17)
Kd = K_aug(2)*(1 + C*B*K_aug(2))^-1
Kp = (1 - Kd*C*B)*K_aug(1)
Ki = (1 - Kd*C*B)*K_aug(3)

% Provjera stabilnosti sustava u regulacijskoj petlji
eig(A_aug + B_aug*K)

% Stabilnost sustava prije implementacije regulatora
eig(A)
```



---

**II. MATLAB kôd za aproksimaciju veze između *elevatora* i pripadnog servo motora**

```
% interpolacija točaka za elevator

pitch = [60:10:140]; % raspon kuteva pitch servoa
elevator = [-24 -17 -10 -5 0 6 11 16 18]; % raspon kuteva za elevator

polinom = polyfit(elevator, pitch, 3);
% vektor u kojem se spremaju koeficijenti polinoma
x = elevator(1):0.1:elevator(end);
% vektor elevator kuteva za plottanje karakteristike
f = polyval(polinom, x);
% računanje vrijednosti aproksimacijske funkcije
%f1 = polinom(1)*x.^3 + polinom(2)*x.^2 + polinom(3)*x.^1 + polinom(4)*x.^0;
f2 = 0.00023774*x.^3 + 0.0151*x.^2 + 1.8371*x.^1 + 98.3472*x.^0;
% ovaj izraz je isto kao prijašnji zakomentirani izraz
plot(x, f2, 'b', elevator, pitch, 'o') % crtanje grafa
legend('Aproksimacijski polinom', 'Točke mjerenja'); % legenda
xlabel('Zakret elevatora [°]', 'FontSize', 12); % oznaka x osi
ylabel('Zakret servoa [°]', 'FontSize', 12); % oznaka y osi
grid on % uključena mreža
```

### III. Arduino kôd regulatora

```
// I-PD controller za daljinski upravljani avion
// Martin Ključarić 0035239671
// Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu 2024. godina

#include <ESP32Servo.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#define pitchPin 18 // CTRL pin na shemi
#define pwmInput 23 // PITCH pin na shemi
#define autoSetPin 4 // AUTO pin na shemi

Adafruit_MPU6050 mpu; // definiranje senzora
Servo pitchServo; // definiranje varijable vezane za pitch servo motor
float pitch = 0, pitchVel = 0, pitchRef = 0, dt = 0.01, u = 0, u_sat = 0, error =
0, integral = 0, previous_integral = 0, proportional = 0, derivative = 0,
Ki = -1.9983, Kp = 0.5886, Kd = 0.0049; // inicijalizacija svih varijabli
float upperLimit = 0.31416, lowerLimit = -0.4189; // limiti elevatora
float rtd = 180/3.14159, dtr = 3.14159/180;
// konverzijske varijable iz radijana u stupnjeve i obratno
double pitch_angle;

hw_timer_t * timer_cfg = NULL;
// varijabla za definiciju timera za uzorkovanje
volatile bool isAuto = false, measureAndControl = false;
// definicija kontrolnih varijabli korištenih u prekidima (volatile)
volatile long timestart=0, timeend=0;
// varijable za mjerenje položaja AUTO gumba

// ----- PREKIDI -----//

// PREKID ZA TIMER ZA UZORKOVANJE I RAČUNANJE PID

void IRAM_ATTR timer_ISR(){
    measureAndControl = true;
    // kada dođe vrijeme uzorkovanja setira se flag i u loopu se računa i uzorkuje
    //(izbjegavanje zadržavanja u prekidnoj rutini)
}

// PREKID ZA PROSLJEĐIVANJE PWM SIGNALA

void IRAM_ATTR pwmInterrupt(){
    // prekid za prosljeđivanje PWM signala u ručnom radu
    if(digitalRead(pwmInput) == HIGH)
    // ako je stanje 1, daje 1, inače daje 0
        digitalWrite(pitchPin, HIGH);
    else if(digitalRead(pwmInput) == LOW)
    // oponaša PWM signal iz transmittera tijekom ručnog rada
        digitalWrite(pitchPin, LOW);
}

// PREKID ZA POSTAVLJANJE U AUTOMATSKI RAD

void IRAM_ATTR setToAutomatic(){
    if(digitalRead(autoSetPin) == HIGH){
    // gumb daje PWM signal na AUTO pinu, duljina poluvla 1ms za nepritisnuto stanje,
    // duljina 2ms za pritisnuto
        timestart = millis();
    // bilježenje vremena kada dođe pozitivni prid PWM-a
```

```
    }
    else if(digitalRead(autoSetPin) == LOW){
        timeend = millis() - timestart;
        // mjerenje vremena kada dođe negativni brid PWM-a, računanje vremena trajanje
        // poluvala
        if(isAuto == false && timeend == 2){
            // ako nije u automatskom radu i poluval traje 2ms
            pitchServo.attach(pitchPin);
            // uključi servo biblioteku
            isAuto = true;
            // bilježi da je rad automatski
            detachInterrupt(digitalPinToInterrupt(pwmInput));
            // micanje prekida koji je prosljeđivao PWM iz recievera - ESP ne vraća prekid ako
            // se ovo napravi u programu....

        }

        else if(isAuto == true && timeend == 1){
            // ako je u automatskom radu i poluval traje 1ms
            pitchServo.detach();
            // otpoji servo biblioteku
            isAuto = false;
            // bilježi da je rad ručni
            attachInterrupt(digitalPinToInterrupt(pwmInput), pwmInterrupt, CHANGE);
            // ponovo inicijaliziraj prekid za prosljeđivanje PWM-a iz recievera

        }
    }
}

// SETUP PINOVA, TIMERA, SENZORA
void setup() {
    pinMode(pwmInput, INPUT);
    // postavljanje pwmInput pina kao ulaz
    attachInterrupt(digitalPinToInterrupt(pwmInput), pwmInterrupt, CHANGE);
    // postavljanje hardware prekida na pwmInput pin
    pinMode(pitchPin, OUTPUT);
    // definiranje pitchPina kao izlaz
    pinMode(autoSetPin, INPUT);
    // postavljanje pina za prebačaj iz ručnog u automatski rad i obratno
    attachInterrupt(digitalPinToInterrupt(autoSetPin), setToAutomatic, CHANGE);
    // interrupt za prebačaj načina rada
    timer_cfg = timerBegin(1000000);
    // postavljanje frekvencije timera (1MHz)
    timerAttachInterrupt(timer_cfg, &timer_ISR);
    // attachiranje prekida timer_isr na prekid timer_cfg
    timerAlarm(timer_cfg, 10000, true, 0);
    // postavljanje alarma svake 10ms, brojanje prema gore (true) i reset nakon
    // brojanja do maksimalne vrijednosti (0)

// ----- Postavljanje senzora----- //

    mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
```

```
// setiranje radnog područja akcelerometra te unutarnjeg filtera
mpu.setGyroRange(MPU6050_RANGE_500_DEG);
mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);

// ----- //

}

void loop() {

    if(measureAndControl == true){
// ako se okrene flag za mjerenje u timer_ISR()
        sensors_event_t a, g, temp;
// mjerenje vrijednosti i spremanje u a, g, temp
        mpu.getEvent(&a, &g, &temp);

        pitch = pitch + g.gyro.z * dt * dtr;
// kut je integral kutne brzine (množimo s pi/180 da dobijemo radijane)
        pitchVel = g.gyro.z * dtr;
// kutna brzina (množimo s pi/180 da dobijemo radijane)
        error = pitchRef * dtr - pitch;
// reg. pogreška

        //I-PD rutina
        if(isAuto == true){
            proportional = pitch;
            derivative = pitchVel;
            integral = integral + error * dt;

            u = Ki*integral + Kp*proportional + Kd*derivative;

            if(u > upperLimit){
// ako je u preko gornje granice
                u = upperLimit;
// u je jednak gornjoj granici
                integral = previous_integral;
// integral je jednak prijašnjem (ne integriramo više)
            }
            else if(u < lowerLimit){
// ako je u preko donje granice
                u = lowerLimit;
// u je jednak donjoj granici
                integral = previous_integral;
// integral je jednak prijašnjem (ne integriramo više)
            }

            u = u*rtd;

            previous_integral = integral;
// spremanje trenutne vrijednosti integratora u pomoćnu varijablu (koja služi za
// prekid integriranja u zasićenju)
            pitch_angle = (0.00023774 * pow(u,3) + 0.0151 * pow(u,2) + 1.8371 *
pow(u,1) + 98.3472);
//pretvorba iz zakreta krilca (u) u zakret servoa (pitch)
            pitchServo.write((int) (pitch_angle));
// piši pitch na servo, pretvori iz radijana u stupnjeve
        }
        measureAndControl = false;
// resetiranje logičke varijable
    }
}
```