

# Upravljanje bespilotnom četverrotorskom letjelicom primjenom vremenski promjenjivog regulatora stanja

---

Periša, Fran

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:235:603374>

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

*Download date / Datum preuzimanja:* **2025-01-16**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Fran Periša**

ZAGREB, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

## ZAVRŠNI RAD

Mentor:

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

Student:

Fran Periša

ZAGREB, 2024.

“Tada će me zvati, ali se ja neću odazvati;  
tražit će me, ali me neće naći.  
Jer su mrzili spoznaju  
i nisu izabrali Gospodnjeg straha  
niti su poslušali moj savjet,  
nego su prezreli svaku moju opomenu.  
Zato će jesti plod svojeg vladanja  
i nasititi se vlastitih savjeta.  
Jer glupe će ubiti njihovo odbijanje,  
a nemar će upropastiti bezumne.  
A tko sluša mene, bezbrižan ostaje  
i spokojno živi bez straha od zla.”

(Izr 1, 28-33)

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

Zahvaljujem se svom mentoru **izv. prof. dr. sc. Vladimiru Miliću** što je prihvatio biti mojim mentorom te se iskreno zahvaljujem na pruženim savjetima, komentarima i izdvojenom vremenu prilikom pisanja ovoga rada. Ovaj rad posvećujem svojoj obitelji; onoj koja me prati odozgora te onoj koja je trenutno uz mene.

Fran Periša



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

## ZAVRŠNI ZADATAK

Student: **Fran Periša**

JMBAG: **0035237656**

Naslov rada na hrvatskom jeziku: **Upravljanje bespilotnom četverorotorskom letjelicom primjenom vremenski promjenjivog regulatora stanja**

Naslov rada na engleskom jeziku: **Control of a quadrotor unmanned aerial vehicle using a time-varying state controller**

Opis zadatka:

Tema ovog završnog rada je primjena metode za slijeđenje referentne trajektorije bespilotne letjelice s četiri rotora koja se temelji na linearnom vremenski promjenjivom zakonu upravljanja uz kvadratni kriterij optimalnosti. Iako je teorija koja stoji iza sinteze ovakvog zakona upravljanja dobro poznata i može se smatrati standardnom još uvijek je intenzivno područje istraživanja u različitim primjenama kao što su bespilotne letjelice, mobilni roboti i autonomna vozila zbog svoje relativne jednostavnosti, a često služi i kao usporedna metoda za kompleksnije nelinearne metode. Glavna ideja pristupa u ovom radu je nelinearni model letjelice aproksimirati linearnim diferencijalnim jednadžbama pogreške u prostoru stanja pri čemu su matrica koeficijenata i matrica ulaza vremenski promjenjive.

U radu je potrebno:

1. Provesti teorijska razmatranja o nužnim i dovoljnim uvjetima optimalnog upravljanja linearnim vremenski promjenjivim sustavima uz kvadratni kriterij optimalnosti i metodama za rješavanje pripadajuće Riccatijeve jednadžbe.
2. Izvesti diferencijalne jednadžbe koje opisuju kinematiku i dinamiku bespilotne letjelice s četiri rotora koje su značajne sa stanovišta sinteze zakona upravljanja za slijeđenje referentne trajektorije.
3. Za male perturbacije oko referentne trajektorije, nelinearni matematički model linearizirati u oblik vremenski promjenljivog prostora stanja koji je prikladan za sintezu linearnog vremenski promjenljivog zakona upravljanja prema kvadratnom kriteriju optimalnosti.
4. Za sintezu vremenski promjenljivog zakona upravljanja prema kvadratnom kriteriju optimalnosti načiniti odgovarajuću vlastitu funkciju u nekom standardnom matematičkom programskom alatu (npr. u MATLAB-u) u kojoj će se rješavati pripadajuća Riccatijeva jednadžba koja daje vremenski promjenjivu matricu pojačanja regulatora.
5. Načiniti simulacijske modele u odgovarajućem matematičkom programskom alatu (npr. u MATLAB-u) u kojima će se direktno numerički rješavati diferencijalne jednadžbe primjenom ugrađenih funkcija. Simulacijama na računalu analizirati ponašanje zatvorenog sustava upravljanja u slučajevima nelinearnog i lineariziranog modela četverorotorske letjelice za slučaj nekoliko različitih referentnih trajektorija.

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 Ćurković

# Sadržaj

Sadržaj	i
Popis slika	iii
Popis tablica	iv
Popis simbola	v
Popis kratica	viii
Sažetak	ix
Abstract	x
<b>1 Uvod</b>	<b>1</b>
<b>2 Teorijska razmatranja</b>	<b>2</b>
2.1 Optimalno upravljanje . . . . .	2
2.1.1 Kriteriji optimalnosti . . . . .	2
2.1.2 Ograničenja tipa jednakosti . . . . .	4
2.1.3 Nužni uvjeti optimalnog upravljanja . . . . .	9
2.2 Hamilton-Jacobi-Bellmanova jednadžba . . . . .	10
2.3 Linearni kvadratični regulator . . . . .	14
2.3.1 Primjena Hamilton-Jacobi-Bellmanove jednadžbe na kvadratični problem . . . . .	14
2.4 Vremenski varijantan linearni kvadratični regulator . . . . .	16
2.4.1 Linearni kvadratični regulator i slijeđenje referentne trajektorije . . . . .	17
2.5 Metode rješavanja Riccatijeve jednadžbe . . . . .	20
2.5.1 Schurova dekompozicija . . . . .	20
2.5.2 Newtonova metoda . . . . .	23
2.5.3 Eulerova metoda . . . . .	26
2.5.4 Dormand-Prince metoda . . . . .	29
<b>3 Izvod dinamike pogreške kroz model četverorotorske letjelice</b>	<b>31</b>
3.1 Matematičko modeliranje četverorotorske letjelice . . . . .	31
3.1.1 Kinematički model . . . . .	33

---

3.1.2	Dinamički model . . . . .	35
3.2	Dinamika pogreške . . . . .	38
<b>4</b>	<b>Simulacijska razmatranja</b>	<b>42</b>
4.1	Prva trajektorija . . . . .	43
4.2	Druga trajektorija . . . . .	46
<b>5</b>	<b>Zaključak</b>	<b>50</b>
	<b>Literatura</b>	<b>51</b>
<b>A</b>	<b>Matlab programski kodovi</b>	<b>53</b>
A.1	Funkcije . . . . .	53
A.2	Definiranje parametara i varijabli . . . . .	55
A.3	Izvršavanje simulacija i grafički prikaz simulacijskih rezultata . . . . .	61



# Popis slika

3.1	Prikaz bespilotne letjelice i pripadnih značajki. . . . .	32
4.1	Prikaz slijeđenja prve trajektorije. . . . .	44
4.2	Pogreške slijeđenja referentnih položaja i orijentacija za linearizirani model (prva trajektorija). . . . .	44
4.3	Iznosi upravljačkih varijabli lineariziranog modela za slijeđenje prve trajektorije. . . . .	45
4.4	Usporedba položaja i orijentacija lineariziranog modela s referentnima (prva trajektorija). . . . .	45
4.5	Uvećan prikaz dijela prve trajektorije. . . . .	46
4.6	Prikaz slijeđenja druge trajektorije. . . . .	47
4.7	Pogreške slijeđenja referentnih položaja i orijentacija za linearizirani model (druga trajektorija). . . . .	48
4.8	Iznosi upravljačkih varijabli lineariziranog modela za slijeđenje druge trajektorije. . . . .	48
4.9	Usporedba položaja i orijentacija lineariziranog modela s referentnima (druga trajektorija). . . . .	49
4.10	Uvećan prikaz dijela druge trajektorije. . . . .	49

# Popis tablica

2.1	Tablica vrijednosti - Primjer 4. (Eulerova metoda). . . . .	28
4.1	Parametri bespilotne letjelice. . . . .	42

# Popis simbola

$\hat{e}$	jedinični vektor u pripadnoj osi koordinatnog sustava	-
$A_s$	broj vremenskih koraka	-
$A$	matrica sustava	-
$B$	ulazna matrica	-
$f_B$	vektor sila koje djeluju na bespilotnu letjelicu	-
$H_f$	Hessijanova matrica	-
$I$	matrica inercija	-
$K$	matrica pojačanja	-
$m$	ograničenja funkcije	-
$Q$	težinska matrica stanja	-
$R_T$	inverz matrice transformacije	-
$R$	težinska matrica upravljanja	-
$R_\phi, R_\theta, R_\psi$	matrica rotacije	-
$S$	matrica rješenja pripadajuće Riccatijeve jednadžbe	-
$u$	vektor upravljačkih (ulaznih) varijabli	-
$v$	vektor derivacija translacijskih položaja	-
$w$	vektor kutnih brzina	-
$x$	vektor varijabli stanja	-
$\mathcal{H}$	Hamiltonian matrica	-
$\mathcal{O}$	lokalna pogreška odsijecanja	-
$\mathcal{U}$	Euklidski prostor	-

$f$	funkcija sustava	-
$f_t$	ukupna sila potiska	N
$I$	jedinična matrica	-
$I_{xx}, I_{yy}, I_{zz}$	moment inercije oko pripadnih osi	kg · m <sup>2</sup>
$J$	funkcija cilja	-
$L$	trenutni trošak	-
$l$	duljina pojedinog kraka bespilotne letjelice	m
$m$	masa	kg
$N$	broj vremenskih koraka	-
$O$	članovi višeg reda Taylorovog razvoja	-
$p$	kutna brzina u koordinatnoj osi $x_B$	rad/s
$q$	kutna brzina u koordinatnoj osi $y_B$	rad/s
$r$	kutna brzina u koordinatnoj osi $z_B$	rad/s
$t$	vrijeme	s
$u$	translacijska brzina u koordinatnoj osi $x_B$	m/s
$v$	translacijska brzina u koordinatnoj osi $y_B$	m/s
$w$	translacijska brzina u koordinatnoj osi $z_B$	m/s

### Konstante

$\pi$	pi	$\approx 3.14159$
$g$	gravitacijsko ubrzanje	$\approx 9.80665 \text{ m/s}^2$

### Grčka slova

$\lambda$	Lagrangeov multiplikator	-
$\lambda_s$	svojstvene vrijednosti	-
$\phi$	Eulerov kut zakreta - valjanje	rad
$\phi_f$	krajnji trošak	-

$\psi$	Eulerov kut zakreta - skretanje	rad
$\tau_B$	ukupan moment sile bespilotne letjelice	-
$\theta$	Eulerov kut zakreta - poniranje	rad

### Indeksi

$(\cdot)_0$	početni uvjet, početni trenutak	-
$(\cdot)_\lambda$	parcijalna derivacija prema Lagrangeovom multiplikatoru	-
$(\cdot)_g$	ravnotežna varijabla	-
$(\cdot)_{\text{ref}}$	referentna varijabla	-
$(\cdot)_u$	parcijalna derivacija prema vektoru upravljanja	-
$(\cdot)_x$	parcijalna derivacija prema vektoru stanja	-
$(\cdot)_B$	komponenta koordinatnog sustava bespilotne letjelice	-
$(\cdot)_f$	krajnji trenutak	-
$(\cdot)_I$	komponenta inercijskog koordinatnog sustava	-
$(\cdot)_t$	parcijalna derivacija prema vremenu	-
$(\cdot)_x$	djelovanje u osi $x$	-
$(\cdot)_y$	djelovanje u osi $y$	-
$(\cdot)_z$	djelovanje u osi $z$	-

# Popis kratica

**DP** Dormand–Prince.

**HJB** Hamilton-Jacobi-Bellman.

**IVP** Initial value problem.

**LQ** Linear quadratic.

**LQR** Linear quadratic regulator.

**LTV** Linear time-varying.

**ODE** Ordinary differential equation.

**RK** Runge–Kutta.

**TVLQR** Time-varying linear quadratic regulator.

**UAV** Unmanned aerial vehicle.

# Sažetak

Kroz ovaj je rad cilj uspješno primijeniti metodu upravljanja za praćenje referentne trajektorije na matematičkom modelu četverorotorske bespilotne letjelice, koristeći linearan vremenski promjenjiv regulator stanja temeljen na kvadratnom kriteriju optimalnosti. Kroz početak rada se uvode pojmovi i objašnjenja vezani uz teoriju optimalnog upravljanja poput Hamilton-Jacobi-Bellmanove jednačbe te diferencijalne i algebarske Riccatijeve jednačbe. Nadalje se opisuju metode rješavanja istih te postupak primjene i linearizacije nelinearnog sustava u oblik vremenski promjenljivog prostora stanja u koordinatama pogreške. Slijedi matematički opis kinematike i dinamike bespilotne letjelice te primjena prethodno objašnjene teorije na navedeni model kroz računalni alat *Matlab*. Definiranjem vlastitih te korištenjem postojećih funkcija u navedenom alatu, simuliraju se i uspoređuju slijeđenja dvaju referentnih trajektorija za linearizirani i nelinearizirani model letjelice te prikazuju značajke ponašanja lineariziranog modela, kao i upravljanja korištena za spomenute modele.

**Ključne riječi:** autonomna četverorotorska letjelica, linearni vremenski promjenjivi sustav, linearni kvadratični regulator, Riccatijeva diferencijalna jednačba, slijeđenje referentne trajektorije

# Abstract

The aim of this bachelor's thesis is to successfully apply a control method for reference trajectory tracking to the mathematical model of a quadrotor unmanned aerial vehicle, using a linear time-varying state regulator based on a quadratic optimality criterion. Thesis begins by introducing concepts and explanations related to optimal control theory, such as the Hamilton-Jacobi-Bellman equation and both differential and algebraic Riccati equations. The methods for solving these equations are then described, along with the process of linearizing the nonlinear system into a time-varying state-space representation in error coordinates. This is followed by a mathematical description of the kinematics and dynamics of the unmanned aerial vehicle, and the application of the previously explained theory to this model using the computational tool *Matlab*. By defining user functions and applying built-in functions and functions from *Matlab* toolboxes, simulations are performed to compare the tracking of two reference trajectories for both the linearized and nonlinear models of the aircraft. The results highlight the behavior of the linearized model and the control strategies used for both models.

**Keywords:** autonomous quadrotor, linear time-varying system, linear quadratic regulator, Riccati differential equation, reference trajectory tracking



# 1 Uvod

Razvoj tehnologije doveo je do široke primjene autonomnih bespilotnih letjelica. Primjetna je primjena istih u vojnom sektoru, poljoprivredi, inspekcijama infrastrukture, dostavnih usluga i slično.

Kako bi se unaprijedile njihove performanse, u smislu slijeđenja željenih trajektorija što većom preciznošću, rasla je te još uvijek raste potreba za novim sustavima upravljanja kojima će se osigurati optimalne performanse u raznim nepredvidivim uvjetima leta. U tom kontekstu, optimalno upravljanje može osigurati precizno i učinkovito ponašanje takvih letjelica.

U optimalnom upravljanju, jednom od važnijih instanci se smatra linearni kvadratični regulator (LQR), kojim se optimiziraju performanse sustava uz minimizaciju pogreške stanja i upotrebe upravljačke energije. Linearni kvadratični regulator za linearne vremenski promjenjive (LTV) sustave omogućuje navedeno pri čemu dinamika sustava varira tokom vremena. Metodologija sinteze LQR-a za LTV sustave započinje linearizacijom nelinearnog sustava, u ovom slučaju drona, u vremenski promjenjiv sustav u koordinatama pogreške, pri čemu se pogreške stanja nastoje minimizirati.

Uz linearni kvadratični regulator, za navedene se primjene može koristiti PID regulacija, koju karakterizira jednostavnost, ali manja preciznost u odnosu na LQR, dok druge metode upravljanja linearnim te nelinearnim sustavima poput modelskog prediktivnog upravljanja te *backstepping* upravljanja zahtijevaju veću računalnu snagu te su zahtjevnije za implementaciju nego LQR. Cilj ovog rada je primijeniti spomenuti vremenski promjenjiv linearni kvadratični regulator (TVLQR) za upravljanje autonomnom bespilotnom letjelicom s četiri rotora s ishodom praćenja referentne trajektorije ostvarenjem optimalnih performansi te upravljanja.

U ovom radu se kroz prvo poglavlje općeniti pojmovi i metode vezani uz optimalno upravljanje [1, 2, 3, 4] te se opisuju neke od metoda za rješavanje pripadajuće Riccatijeve jednadžbe [5, 6, 7]. U drugom se poglavlju izvodi matematički model četverrotorske letjelice - drona, kroz izvod kinematičkog i dinamičkog modela [8, 9, 10, 11, 12, 13] što u konačnici rezultira matematičkim modelom pogodnim za primjenu TVLQR-a za praćenje referentne trajektorije. U zadnjem se poglavlju vrše simulacije nelineariziranog modela drona te lineariziranog modela istog u koordinatama dinamike pogreške čime se korištenjem postojećih, ali i vlastito izrađenih funkcija, računaju varijable stanja te upravljačke varijable kroz zadano simulacijsko vrijeme u računalnom alatu *Matlab* korištenjem stvarnih parametara drona [14], čime se dobivaju i grafički prikazuju shodna ponašanja obaju modela.

## 2 Teorijska razmatranja

### 2.1 Optimalno upravljanje

Optimalno upravljanje, kao grana teorije upravljanja (*eng. control theory*), jedna je od najznačajnijih grana iste. Račun varijacija, razvijen kroz 17. stoljeće, vuče korijene optimalnog upravljanja kojega su nadalje kroz 18. stoljeće razvijali matematičari Euler i Lagrange te kasnije kroz 19. stoljeće Hamilton, Jacobi, Weierstrass i mnogi drugi.

Tijekom 1950-ih godina razvija se i grana dinamičkog programiranja od strane Bellmana, na kojoj počivaju nove formulacije poznate Hamilton-Jacobijeve jednadžbe.

Račun varijacija, kao i praktični problemi optimalnog upravljanja, dobili su na važnosti kroz radove Pontryagina, čiji se poznati teorem minimuma smatra pretečom matematičke pozadine optimalnog upravljanja. Teorija optimalnog upravljanja, s velikim značajem na sintezi upravljanja dinamičkim sustavima više varijabli, to jest, multivarijabilnim dinamičkim sustavima, primjenjuje se na nekolicinu sustava. Takvi sustavi prisutni su kroz sve grane gospodarstva, od ekonomskih sustava, medicine i demografije pa sve do kompleksnih tehnoloških sustava.

Glavni pojam optimalnog upravljanja je kriterij optimalnosti, koji se naziva i funkcijom cilja. Glavna svrha je istog minimizirati ili maksimizirati te u skupu mogućih rješenja dobiti optimalno rješenje koje zadovoljava željni kriterij.

#### 2.1.1 Kriteriji optimalnosti

Kriteriji optimalnosti razmatraju se kao minimalni ili maksimalni s ciljem optimizacije sustava koji se upravlja. Navedeni sustav zapisuje se na sljedeći način

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.1)$$

s početnim uvjetima  $\mathbf{x}(t_0) = \mathbf{x}_0$  u intervalu  $t_0 \leq t \leq t_f$ .

- $\mathbf{x}(t)$  označava vektor varijabli stanja,
- $\mathbf{u}(t)$  označava vektor upravljačkih varijabli,
- $t_0$  označava početni trenutak vremena,
- $t_f$  označava završni trenutak vremena.

Neki od kriterija optimalnosti su:

1. **Problem minimalnog vremena**, gdje je cilj pronaći vektor upravljačkih varijabli  $\mathbf{u}(t)$  koji uspijeva prenijeti stanje  $\mathbf{x}(t_0)$  u stanje  $\mathbf{x}(t_f)$  u što kraćem vremenu  $\mathbf{T}$ . Vektor upravljačkih varijabli je najčešće ograničen  $\mathbf{u}(t) \in U$ , gdje je  $U$  vektor dozvoljenih upravljačkih varijabli). Funkcija cilja koja se minimizira glasi

$$J(\mathbf{u}(t), \mathbf{x}_0) = \int_{t_0}^{t_f} dt = \mathbf{T}. \quad (2.2)$$

2. **Problem konačne vrijednosti**, gdje je cilj pronaći vektor upravljačkih varijabli  $u(t)$  koji uspijeva prenijeti stanje  $\mathbf{x}(t_0)$  u željeno stanje  $\mathbf{x}(t_i)$  uz minimalni trošak, to jest, minimiziranu funkciju cilja. Funkcija cilja koja se minimizira glasi

$$J(\mathbf{u}(t), \mathbf{x}_0) = [\mathbf{x}(t_f) - \mathbf{x}(t_i)]^T \mathbf{Q} [\mathbf{x}(t_f) - \mathbf{x}(t_i)], \quad (2.3)$$

gdje je  $\mathbf{Q}$  kvadratna pozitivno semi-definitna matrica.

3. **Problem minimalnog utroška energije**, gdje je cilj pronaći vektor upravljačkih varijabli  $\mathbf{u}(t)$  koji uspijeva prenijeti stanje  $\mathbf{x}(t_0)$  u stanje  $\mathbf{x}(t_f)$  uz minimalni utrošak energije. Funkcija cilja koja se minimizira glasi

$$J(\mathbf{u}(t)) = \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t), \quad (2.4)$$

gdje je  $\mathbf{R}$  kvadratna pozitivno definitna matrica.

4. **Problem praćenja referentne trajektorije**, gdje je cilj pronaći vektor upravljačkih varijabli  $\mathbf{u}(t)$  koji uspijeva pratiti zadanu referentnu trajektoriju kroz određen vremenski interval  $t_f - t_0$ . Funkcija cilja koja se minimizira glasi

$$J = \int_{t_0}^{t_f} [\mathbf{x}(t) - \mathbf{x}_{ref}(t)]^T \mathbf{Q} [\mathbf{x}(t) - \mathbf{x}_{ref}(t)] dt, \quad (2.5)$$

gdje izraz  $\mathbf{x}(t) - \mathbf{x}_{ref}(t)$  označava pogrešku praćenja referentne trajektorije. Kao i prije,  $\mathbf{Q}$  označava kvadratnu pozitivno semi-definitnu matricu.

5. **Bolzin problem** je općeniti zapis kriterija optimalnog upravljanja. Funkcija cilja glasi

$$J = \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt + \phi_f(\mathbf{x}(t_f)), \quad (2.6)$$

gdje  $L(\mathbf{x}(t), \mathbf{u}(t))$  označava trenutni trošak, to jest, trošak koji se sumira po svakom trenutku zadanog vremenskog intervala, dok  $\phi_f(\mathbf{x}(t_f))$  označava krajnji trošak, to jest, trošak evaluiran na kraju trajektorije. Bolzin se problem može svesti na **problem optimalnog upravljanja** ako je  $\phi_f(\mathbf{x}(t_f)) = 0$ . S druge

strane, uzimajući u obzir formulaciju (2.5) i nuliranje  $\mathbf{x}_d(t)$  ili konstantnost istog, u kombinaciji s (2.4) također se dobiva **problem optimalnog upravljanja**:

$$\begin{aligned} J &= \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt + \phi_f(\mathbf{x}(t_f)) \\ &= \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) \\ &= \int_{t_0}^{t_f} [\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t)] dt. \end{aligned} \quad (2.7)$$

### 2.1.2 Ograničenja tipa jednakosti

Neka je  $L(\mathbf{x}, \mathbf{u})$  trenutni trošak gdje su

- $\mathbf{u} = [u_1 \quad u_2 \quad \cdots \quad u_m]^T \in \mathbb{R}^m$  vektor upravljačkih (ulaznih) varijabli,
- $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T \in \mathbb{R}^n$  vektor varijabli stanja.

Cilj optimizacijskog problema je odrediti vektor upravljačkih varijabli koji minimizira  $L(\mathbf{x}, \mathbf{u})$  te zadovoljava ograničenje tipa jednakosti

$$f(\mathbf{x}, \mathbf{u}) = 0. \quad (2.8)$$

Za određivanje lokalnog minimuma potrebno je odrediti nužne i dovoljne uvjete minimuma koji zadovoljavaju  $f(\mathbf{x}, \mathbf{u}) = 0$  [1]. Kroz nalaženje istih bit će uveden pojam Lagrangeovih multiplikatora. Za dobivanje nužnih i dovoljnih uvjeta minimuma će se derivacija indeksa performansi  $dL$  aproksimirati Taylorovim razvojem, do prvog reda za dobivanje nužnih uvjeta minimuma te do drugog reda za dobivanje dovoljnih uvjeta.

Može se krenuti s izvodom za dobivanje **nužnih uvjeta minimuma**. U stacionarnoј točki, to jest, rješenju uvjeta (2.8), je  $dL = 0$  kroz prvi red aproksimacije po varijabli  $d\mathbf{u}$ , dok je  $df = 0$ . Može se pisati totalni diferencijal kroz vektorski zapis

$$\begin{aligned} dL &= L_{\mathbf{x}} d\mathbf{x} + L_{\mathbf{u}} d\mathbf{u}, \\ df &= f_{\mathbf{x}} d\mathbf{x} + f_{\mathbf{u}} d\mathbf{u}, \end{aligned} \quad (2.9)$$

te za kritičnu (stacionarnu) točku, kako je i pisano,  $dL$  mora biti jednak nuli uz  $df = 0$ ,

$$\begin{aligned} dL &= L_{\mathbf{x}} d\mathbf{x} + L_{\mathbf{u}} d\mathbf{u} = 0, \\ df &= f_{\mathbf{x}} d\mathbf{x} + f_{\mathbf{u}} d\mathbf{u} = 0. \end{aligned} \quad (2.10)$$

Navedeni zapisi mogu se zapisati i u obliku općeg totalnog diferencijala

$$\begin{aligned} dL &= \sum_{j=1}^n \frac{\partial L}{\partial x_j} dx_j + \sum_{j=1}^m \frac{\partial L}{\partial u_j} du_j, \\ df_i &= \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} dx_j + \sum_{j=1}^m \frac{\partial f_i}{\partial u_j} du_j, \end{aligned} \quad (2.11)$$

gdje je  $i = 1, \dots, n$ .

$L_{\mathbf{x}}$  i  $L_{\mathbf{u}}$  su vektori, to jest, vektori parcijalnih derivacija prema varijablama vektora  $\mathbf{x}$  i  $\mathbf{u}$

$$L_{\mathbf{x}} = \left[ \frac{\partial L}{\partial x_1} \quad \frac{\partial L}{\partial x_2} \quad \dots \quad \frac{\partial L}{\partial x_n} \right], \quad L_{\mathbf{u}} = \left[ \frac{\partial L}{\partial u_1} \quad \frac{\partial L}{\partial u_2} \quad \dots \quad \frac{\partial L}{\partial u_m} \right], \quad (2.12)$$

dok su  $f_{\mathbf{x}}$  i  $f_{\mathbf{u}}$  Jacobijeve matrice parcijalnih derivacija vektora  $f$  prema varijablama vektora  $\mathbf{x}$  i  $\mathbf{u}$

$$f_{\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad f_{\mathbf{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \dots & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \dots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}. \quad (2.13)$$

Kako je iz (2.8) vektor varijabli stanja  $\mathbf{x}$  određen za dani vektor upravljačkih varijabli  $\mathbf{u}$ , tako se može zaključiti, ali i vrijedi za njihove derivacije. Za izraz  $dL$  iz (2.10) može se unijeti prošireni zapis derivacije varijabli stanja  $d\mathbf{x}$ , ako je poznato da je Jacobijeva matrica člana  $f_{\mathbf{x}}$  nesingularna (regularna)

$$d\mathbf{x} = -f_{\mathbf{x}}^{-1} f_{\mathbf{u}} d\mathbf{u}, \quad (2.14)$$

te slijedi

$$dL = (L_{\mathbf{u}} - L_{\mathbf{x}} f_{\mathbf{x}}^{-1} f_{\mathbf{u}}) d\mathbf{u}. \quad (2.15)$$

Postavljanjem  $dL = 0$  za dani  $d\mathbf{u}$  može se pisati

$$L_{\mathbf{u}} - L_{\mathbf{x}} f_{\mathbf{x}}^{-1} f_{\mathbf{u}} = 0. \quad (2.16)$$

Zapis (2.16) je nužan uvjet za nalaženje minimuma. Po uzoru na [1] se mogu uvidjeti još dva načina da se izvede izraz (2.16). Jednadžbe iz (2.9) mogu se zapisati u matičnom obliku

$$\begin{bmatrix} dL \\ df \end{bmatrix} = \begin{bmatrix} L_{\mathbf{x}} & L_{\mathbf{u}} \\ f_{\mathbf{x}} & f_{\mathbf{u}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} = 0. \quad (2.17)$$

Kako je i prije napisano, navedeni zapis u matičnom obliku odgovara izrazima za definiciju stacionarne točke. Redci matrice (matrica koja sadrži  $L_{\mathbf{x}}, L_{\mathbf{u}}, f_{\mathbf{x}}, f_{\mathbf{u}}$ ) moraju biti linearno zavisni tako da postoji  $n$  vektor  $\lambda$ , koji se nazivaju Lagrangeovi multiplikatori,  $\lambda \in \mathbb{R}^n$ . Prethodno spomenuta matrica je dimenzijski definirana kao  $(n+1) \times (n+m)$  te se linearna zavisnost redaka iste definira također kao nepotpuni rang matrice, to jest, rang  $< (n+1)$  te je determinanta matrice jednaka nuli. Napomena: sustav jednadžbi (2.17) je sustav linearnih jednadžbi. Prema napisanome vrijedi

$$[1 \ \lambda^T] \begin{bmatrix} L_{\mathbf{x}} & L_{\mathbf{u}} \\ f_{\mathbf{x}} & f_{\mathbf{u}} \end{bmatrix} = 0. \quad (2.18)$$

Gornji se izraz može zapisati

$$\begin{aligned} L_{\mathbf{x}} + \lambda^T f_{\mathbf{x}} &= 0, \\ L_{\mathbf{u}} + \lambda^T f_{\mathbf{u}} &= 0. \end{aligned} \quad (2.19)$$

Rješavanjem istog dobiva se rješenje

$$\lambda^T = -L_{\mathbf{x}} f_{\mathbf{x}}^{-1}. \quad (2.20)$$

Ubacivanjem gornjeg izraza u prethodni se ponovno dobiva izraz za nalaženje minimuma (uvjeta za kritičnu (stacionarnu) točku) (2.16). Nadalje, idući način za nalaženje izraza (2.16) uključuje Hamiltonian

$$\boxed{\mathcal{H}(\mathbf{x}, \mathbf{u}, \lambda) = L(\mathbf{x}, \mathbf{u}) + \lambda^T f(\mathbf{x}, \mathbf{u})}. \quad (2.21)$$

Za nalaženje  $\mathbf{x}$ ,  $\mathbf{u}$  i  $\lambda$  potrebno je razmotriti da je prirast u  $\mathcal{H}$  ovisan o prirastima kroz  $\mathbf{x}$ ,  $\mathbf{u}$  i  $\lambda$  prema

$$d\mathcal{H} = \mathcal{H}_{\mathbf{x}} d\mathbf{x} + \mathcal{H}_{\mathbf{u}} d\mathbf{u} + \mathcal{H}_{\lambda} d\lambda. \quad (2.22)$$

Korisno je napomenuti da je zapis  $\mathcal{H}_{\lambda}$  istovjetan zapisu  $\frac{\partial \mathcal{H}}{\partial \lambda}$ ; zapisuje se

$$\mathcal{H}_{\lambda} = \frac{\partial \mathcal{H}}{\partial \lambda} = f(\mathbf{x}, \mathbf{u}). \quad (2.23)$$

Kako se traži vektor upravljačkih varijabli koji zadovoljava uvjet (2.8) istovjetan zapis se može pisati preko člana prirasta Hamiltoniana

$$\mathcal{H}_{\lambda} = 0, \quad (2.24)$$

te se Hamiltonian svodi na funkciju cilja (indeks performansi)

$$\mathcal{H}|_{f=0} = L. \quad (2.25)$$

Pogodno je prisjetiti se da ako je  $f = 0$ , tada vrijedi izraz (2.14) prema kojem je  $d\mathbf{x}$  dobiven preko  $d\mathbf{u}$ , no pogodnije je da derivacija vektora varijabli stanja i derivacija vektora upravljačkih varijabli ne budu uparene na takav način te se odabire  $\lambda$  takav da vrijedi

$$\mathcal{H}_{\mathbf{x}} = 0. \quad (2.26)$$

Tada prema (2.22) prirast  $d\mathbf{x}$  ne doprinosi prirastu  $d\mathcal{H}$ . Podsjetnik:  $\mathcal{H}_{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \mathbf{x}}$ . Prema tome se definira

$$\mathcal{H}_{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \mathbf{x}} = L_{\mathbf{x}} + \lambda^T f_{\mathbf{x}} = 0, \quad (2.27)$$

iz kojeg slijedi (2.20). Kako vrijede formulacije (2.24) i (2.26) tako iz (2.22) vrijedi

$$d\mathcal{H} = \mathcal{H}_{\mathbf{u}} d\mathbf{u} = L. \quad (2.28)$$

Kako je od prije poznato (prema (2.25)) tada za postizanje stacionarne točke preostaje uvjet stacionarnosti

$$\mathcal{H}_{\mathbf{u}} = 0, \quad (2.29)$$

te se mogu zapisati nužni uvjeti za točku minimuma funkcije cilja  $L(\mathbf{x}, \mathbf{u})$  koje zadovoljavaju ograničenje tipa jednakosti  $f(\mathbf{x}, \mathbf{u}) = 0$

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial \lambda} &= f = 0, \\ \frac{\partial \mathcal{H}}{\partial \mathbf{x}} &= L_{\mathbf{x}} + \lambda^T f_{\mathbf{x}} = 0, \\ \frac{\partial \mathcal{H}}{\partial \mathbf{u}} &= L_{\mathbf{u}} + \lambda^T f_{\mathbf{u}} = 0, \end{aligned} \quad (2.30)$$

gdje je  $\mathcal{H}$  definiran prema izrazu (2.21).

Kroz gornje načine dobivanja nužnih uvjeta minimuma se mogla uvidjeti važnost Lagrangeovih multiplikatora, koja se može opisati na sljedeći način. Kroz jednadžbe (2.30) se redom izračunavaju  $\mathbf{x}$ , pa  $\lambda$  te posljednje  $\mathbf{u}$ , no određivanje  $\lambda$  nije direktno od interesa, ali potreban je za određivanje minimalne vrijednosti funkcije cilja  $L$ .

Prema jednadžbi (2.14) vidljiva je direktna poveznica  $d\mathbf{x}$  i  $d\mathbf{u}$  koja ne može biti zanemarena.

Uvođenjem nepoznate varijable  $\lambda$  se dobiva dodatan stupanj slobode te se preko  $\lambda$  može dobiti željena nezavisnost između porasta  $d\mathbf{x}$  i  $d\mathbf{u}$ . Parcijalnim derivacijama  $\mathcal{H}$  prema varijablama  $\lambda$ ,  $\mathbf{x}$  te  $\mathbf{u}$  se sve svodi na dobivanje kritične točke, to jest, [1] minimizacija  $L(\mathbf{x}, \mathbf{u})$  prema ograničenjima  $f(\mathbf{x}, \mathbf{u}) = 0$  zamijenjena je minimizacijom Hamiltoniana  $\mathcal{H}(\mathbf{x}, \mathbf{u}, \lambda)$  bez ograničenja.

Slijedi određivanje **dovoljnih uvjeta minimuma** funkcije cilja  $L(\mathbf{x}, \mathbf{u})$  prema kojima će se moći sa sigurnošću reći je stacionarna (kritična) točka, određena prema (2.30), stvaran minimum. Raspisuje se Taylorov razvoj drugog reda prema  $dL$  te  $df$

$$dL = \begin{bmatrix} L_{\mathbf{x}} & L_{\mathbf{u}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} d\mathbf{x}^T & d\mathbf{u}^T \end{bmatrix} \begin{bmatrix} L_{\mathbf{xx}} & L_{\mathbf{xu}} \\ L_{\mathbf{ux}} & L_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + O(3), \quad (2.31)$$

$$df = \begin{bmatrix} f_{\mathbf{x}} & f_{\mathbf{u}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} d\mathbf{x}^T & d\mathbf{u}^T \end{bmatrix} \begin{bmatrix} f_{\mathbf{xx}} & f_{\mathbf{xu}} \\ f_{\mathbf{ux}} & f_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + O(3), \quad (2.32)$$

gdje su

- $O(3)$  članovi trećeg reda,
- $L_{\mathbf{u}} = \frac{\partial L}{\partial \mathbf{u}}$ ,
- $L_{\mathbf{uu}} = \frac{\partial^2 L}{\partial \mathbf{u}^2}$  - naziva se matricom zakrivljenosti,
- $L_{\mathbf{xu}} = \frac{\partial^2 L}{\partial \mathbf{u} \partial \mathbf{x}}$  te slijedno za ostale,

- $f_{\mathbf{x}\mathbf{u}} = \frac{\partial^2 f}{\partial \mathbf{u} \partial \mathbf{x}}$  te slijedno za ostale.

Sve navedene matrice su Hessijanove matrice koja se definira u općem obliku kao

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (2.33)$$

Diferencijal drugog reda (Hessijan) Hamiltoniana

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 \mathcal{H}}{\partial \mathbf{x}^2} & \frac{\partial^2 \mathcal{H}}{\partial \mathbf{x} \partial \mathbf{u}} \\ \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u} \partial \mathbf{x}} & \frac{\partial^2 \mathcal{H}}{\partial \mathbf{u}^2} \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\mathbf{xx}} & \mathcal{H}_{\mathbf{xu}} \\ \mathcal{H}_{\mathbf{ux}} & \mathcal{H}_{\mathbf{uu}} \end{bmatrix}. \quad (2.34)$$

Prema [1] se uvodi

$$\begin{bmatrix} 1 & \lambda^T \end{bmatrix} \begin{bmatrix} dL \\ df \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\mathbf{x}} & \mathcal{H}_{\mathbf{u}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} d\mathbf{x}^T & d\mathbf{u}^T \end{bmatrix} \begin{bmatrix} \mathcal{H}_{\mathbf{xx}} & \mathcal{H}_{\mathbf{xu}} \\ \mathcal{H}_{\mathbf{ux}} & \mathcal{H}_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} + O(3), \quad (2.35)$$

gdje vrijedi (2.30). Uvrštavanjem izraza

$$d\mathbf{x} = -f_{\mathbf{x}}^{-1} f_{\mathbf{u}} d\mathbf{u} + O(2), \quad (2.36)$$

u izraz (2.35) dobiva se

$$dL = \frac{1}{2} d\mathbf{u}^T \begin{bmatrix} -f_{\mathbf{u}}^T f_{\mathbf{x}}^{-T} & I \end{bmatrix} \begin{bmatrix} \mathcal{H}_{\mathbf{xx}} & \mathcal{H}_{\mathbf{xu}} \\ \mathcal{H}_{\mathbf{ux}} & \mathcal{H}_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} -f_{\mathbf{x}}^{-1} f_{\mathbf{u}} \\ I \end{bmatrix} d\mathbf{u} + O(3). \quad (2.37)$$

Za potvrdu minimuma,  $dL$  treba biti pozitivan za sve priraste  $d\mathbf{u}$ , a osigurava se ako je matrica zakrivljenosti  $L_{\mathbf{uu}}$  za  $f(\mathbf{x}, \mathbf{u}) = 0$  pozitivno definitna. Određuje se prema

$$\begin{aligned} L_{\mathbf{uu}}^f &\equiv L_{\mathbf{uu}|f} = \begin{bmatrix} -f_{\mathbf{u}}^T f_{\mathbf{x}}^{-T} & I \end{bmatrix} \begin{bmatrix} \mathcal{H}_{\mathbf{xx}} & \mathcal{H}_{\mathbf{xu}} \\ \mathcal{H}_{\mathbf{ux}} & \mathcal{H}_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} -f_{\mathbf{x}}^{-1} f_{\mathbf{u}} \\ I \end{bmatrix} \\ &= \mathcal{H}_{\mathbf{uu}} - f_{\mathbf{u}}^T f_{\mathbf{x}}^{-T} \mathcal{H}_{\mathbf{xu}} - \mathcal{H}_{\mathbf{ux}} f_{\mathbf{x}}^{-1} f_{\mathbf{u}} + f_{\mathbf{u}}^T f_{\mathbf{x}}^{-T} \mathcal{H}_{\mathbf{xx}} f_{\mathbf{x}}^{-1} f_{\mathbf{u}}. \end{aligned} \quad (2.38)$$

Ako vrijedi ograničenje  $f(\mathbf{x}, t) = 0$  može se pisati  $L_{\mathbf{uu}} = \mathcal{H}_{\mathbf{uu}}$  pri čemu je  $\mathcal{H}_{\mathbf{uu}}$  druga parcijalna derivacija Hamiltoniana prema  $\mathbf{u}$  te mora biti zadovoljena pozitivna definitnost,  $\mathcal{H}_{\mathbf{uu}} \succ 0$ . Iako nije glavna tema ovog rada, uvodi se jednostavan primjer koji objedinjuje sve navedeno za nužne i dovoljne uvjete minimuma funkcije  $L(\mathbf{x}, \mathbf{u})$ . Primjer je preuzet iz [1].



## Primjer 1. - Ploha drugog reda s linearnim ograničenjima

Neka je dan trenutni trošak

$$L(x_1, u_1) = \frac{1}{2}[x_1 \ u_1] \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ u_1 \end{bmatrix} + [0 \ 1] \begin{bmatrix} x_1 \\ u_1 \end{bmatrix}, \quad (2.39)$$

$$f(x_1, u_1) = x_1 - 3 = 0. \quad (2.40)$$

Hamiltonian glasi

$$\mathcal{H} = L + \lambda^T f = \frac{1}{2}x_1^2 + x_1u_1 + u_1^2 + u_1 + \lambda(x_1 - 3), \quad (2.41)$$

gdje je  $\lambda$  skalar. Uvjeti za stacionarnu točku poznati su prema (2.30)

$$\mathcal{H}_\lambda = x_1 - 3 = 0, \quad (2.42)$$

$$\mathcal{H}_x = x_1 + u_1 + \lambda = 0, \quad (2.43)$$

$$\mathcal{H}_u = x_1 + 2u_1 + 1 = 0. \quad (2.44)$$

Rješavanje istih dovodi do rješenja  $x_1 = 3$ ,  $u_1 = -2$ , and  $\lambda = -1$ . Stacionarna točka glasi

$$(x_1, u_1)^* = (3, -2). \quad (2.45)$$

Kako je Jacobijan  $f_u$  jednak nuli, izraz (2.38) se svodi na  $L_{uu}^f = \mathcal{H}_{uu}$ .  $\mathcal{H}_{uu}$  je druga parcijalna derivacija Hamiltoniana prema  $\mathbf{u}$ , to jest, prva parcijalna derivacija  $\mathcal{H}_u$  prema  $\mathbf{u}$ .

$$L_{uu}^f = \mathcal{H}_{uu} = 2 > 0. \quad (2.46)$$

Minimum je potvrđen.

### 2.1.3 Nužni uvjeti optimalnog upravljanja

Promatra se sustav nelinearnih diferencijalnih jednadžbi

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t_0 \leq t \leq t_f. \quad (2.47)$$

Nužni uvjeti optimalnog upravljanja mogu se izvesti preko Bolzinog problema (2.6), metode Lagrangeovih multiplikatora te uvođenjem Hamiltoniana na sljedeći način. Definira se Bolzin problem

$$J = \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))dt + \phi_f(\mathbf{x}(t_f)), \quad (2.48)$$

prema kojem je cilj minimizacija funkcije cilja uz određeni vektor upravljačkih varijabli  $\mathbf{u}(t)$ . Korištenjem metode Lagrangeovih multiplikatora i definiranjem Hamiltoniana (2.21)

$$\boxed{\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t) = L(\mathbf{x}(t), \mathbf{u}(t), t) + \lambda^T f(\mathbf{x}(t), \mathbf{u}(t), t)}, \quad (2.49)$$

se nakon matematičkih operacija parcijalnih derivacija može pisati izraz

$$J = \phi_f(\mathbf{x}(t_f), t_f) - \lambda^T(t_f) \cdot \mathbf{x}(t_f) + \lambda^T(t_0) \cdot \mathbf{x}(t_0) + \int_{t_0}^{t_f} \left[ \mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t) + \dot{\lambda}^T(t) \cdot \mathbf{x}(t) \right] dt, \quad (2.50)$$

koji je sada, za postizanje lokalnog minimuma, kao i prije potrebno derivirati te izjednačiti s nulom za priraste  $d\mathbf{x}$  i  $d\mathbf{u}$  te se u konačnici dobivaju **nužni uvjeti optimalnog upravljanja**

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) = \frac{\partial \mathcal{H}}{\partial \lambda}, \\ \dot{\lambda} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}} - \lambda^T \cdot \frac{\partial f}{\partial \mathbf{x}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \\ 0 = \frac{\partial \mathcal{H}}{\partial \mathbf{u}} + \lambda^T \cdot \frac{\partial f}{\partial \mathbf{u}} = \frac{\partial \mathcal{H}}{\partial \mathbf{u}}, \end{cases} \quad (2.51)$$

prilikom čegu su rubni i početni uvjeti

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \lambda(t_f) = \frac{\partial \phi_f}{\partial \mathbf{x}}(t_f). \quad (2.52)$$

## 2.2 Hamilton-Jacobi-Bellmanova jednadžba

Jedna od najvažnijih formulacija dinamičkog programiranja (više informacija kroz [2], str.53-96.) je Hamilton-Jacobi-Bellmanova jednadžba, skraćeno HJB jednadžba. HJB jednadžba optimalnog upravljanja je proširenje HJB jednadžbe iz analitičke (klasične) mehanike te svako ime (ili slovna oznaka) označava matematičara koji je dao doprinos ovoj poznatoj formulaciji koja je ustanovljena tijekom 1950-ih godina prošlog stoljeća.

HJB jednadžba je nelinearna parcijalna diferencijalna jednadžba koja pruža nužne i dovoljne uvjete za optimalno upravljanje uz uzimanje u obzir funkcije cilja te najčešće primjene ima u planiranju i praćenju određene trajektorije te u medicinskom i financijskom sektoru.

Neka se razmatra nelinearni dinamički sustav

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.53)$$

gdje su kao i prije

- $\mathbf{x}(t)$  vektor varijabli stanja,
- $\mathbf{u}(t)$  vektor upravljačkih varijabli,
- $f$  funkcija koja opisuje dinamiku sustava.

Kao što je već napisano, u optimalnom upravljanju cilj je pronalazak optimalnog upravljanja  $\mathbf{u}^*$  koje se uvrštava unutar sustava (2.53) te pronalazak optimalne trajektorije  $\mathbf{x}^*$  koja se promatra na vremenskom intervalu  $t \in [t_0, t_f]$ . Cilj je minimizirati također otprije poznatu funkciju cilja definiranu kao problem optimalnog upravljanja

$$J = \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t) dt + \phi_f(\mathbf{x}(t_f), t_f), \quad (2.54)$$

te dobiti njezin minimum  $J^*$ , koji se naziva **optimalan trošak**, a navedeno se matematički može zapisati

$$\begin{aligned} J^* &= \min_{\mathbf{u} \in \mathcal{U}} J \\ &= \min_{\mathbf{u} \in \mathcal{U}} \left\{ \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t) dt + \phi_f(\mathbf{x}(t_f), t_f) \right\}. \end{aligned} \quad (2.55)$$

Podsjetnik:

- $L(\mathbf{x}(t), \mathbf{u}(t), t)$  označava trenutni trošak,
- $\phi_f(\mathbf{x}(t_f), t_f)$  označava krajnji trošak,
- $\mathcal{U}$  označava Euklidski prostor,  $\mathbf{u}(t) \in \mathcal{U}$ ,  $\mathcal{U} \subset \mathbb{R}^m$ .

Postoje dva slučaja navedenog problema - problem konačnog vremena (*eng. finite time horizon problem*) i problem beskonačnog vremena (*eng. infinite time horizon problem*). Kod problema konačnog vremena  $t_f$  je određena vrijednost u vremenu dok je kod problema  $t_f$  beskonačan, što znači da nema krajnjeg troška te se funkcija cilja zapisuje

$$J(\mathbf{x}) = \int_{t_0}^{\infty} L(\mathbf{x}(t), \mathbf{u}(t), t) dt. \quad (2.56)$$

Digitalni uređaji rade u diskretnom vremenu, to jest, sve se promatra u odvojenim, diskretnim trenutcima vremena (koracima), gdje će se oni označavati slovom  $k$  dok će se ukupan broj vremenskih koraka označiti slovom  $N$ . Sustav (2.53) se u diskretnom vremenu može zapisati kao

$$\mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k]), \quad (2.57)$$

gdje

- $\mathbf{x}[k]$  označava stanje sustava u vremenskom koraku  $k$ ,
- $\mathbf{u}[k]$  označava upravljanje u vremenskom koraku  $k$ ,
- $\mathbf{x}[k+1]$  označava stanje sustava u budućem vremenskom koraku  $k+1$ .

Jednadžba (2.54) se također može zapisati u diskretnom vremenu

$$\tilde{J} = \sum_{k=0}^{N-1} L([k], \mathbf{u}[k]) + \phi_f(\mathbf{x}[N]). \quad (2.58)$$

Ipak, u ovom trenutku promatrat će se kontinuirani nelinearni dinamički sustav (2.53) čije će rješenje biti HJB jednadžba.

Prema Bellmanovom principu je svaki segment optimalne trajektorije na intervalu  $[t, t_f]$  optimalan te se isti interval  $[t, t_f]$  može zapisati kao  $[t, t + \Delta t]$  te  $[t + \Delta t, t_f]$ . Zanimljivo je razmatranje kada  $\Delta t \rightarrow 0$ .

Prvi korak je definiranje optimalne funkcije cilja (2.55) preko navedenih intervala, što znači da će doći do razdjelje integrala na dva integrala spomenutih vremenskih intervala. Nakon toga potrebno je definirati trošak u intervalu  $[t, t + \Delta t]$  te uzeti u obzir sve prethodne korake s ciljem pronalaska najbolje *strategije* od trenutka  $t$  te uključiti to u HJB jednadžbu. Oznaka  $\tau$  podrazumijeva vrijeme unutar intervala  $[t, t_f]$ ,  $t \leq \tau \leq t_f$ . Vektor upravljačkih varijabli je naravno unutar toga intervala ( $\mathbf{u}(\tau)$ ). Zapisuje se

$$J^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(\tau) \in \mathcal{U}} \left\{ \phi_f(\mathbf{x}(t_f), t_f) + \int_t^{t_f} L(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \right\}, \quad (2.59)$$

te se integral razdjeljuje na dva intervala

$$J^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(\tau) \in \mathcal{U}} \left\{ \phi_f(\mathbf{x}(t_f), t_f) + \int_t^{t+\Delta t} L(\mathbf{x}, \mathbf{u}, \tau) d\tau + \int_{t+\Delta t}^{t_f} L(\mathbf{x}, \mathbf{u}, \tau) d\tau \right\}. \quad (2.60)$$

Drugi integral može se zapisati kao  $J^*(\mathbf{x}(t + \Delta t), t + \Delta t)$  koji će prema svojim granicama moći uključiti krajnji trošak te gornji izraz prelazi u zapis

$$J^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(\tau) \in \mathcal{U}} \left\{ \int_t^{t+\Delta t} L(\mathbf{x}, \mathbf{u}, \tau) d\tau + J^*(\mathbf{x}(t + \Delta t), t + \Delta t) \right\}. \quad (2.61)$$

Nadalje se  $J^*(\mathbf{x}(t + \Delta t), t + \Delta t)$  može razviti u Taylorov razvoj, prilikom čega se zadržavaju linearni članovi

$$\begin{aligned} J^*(\mathbf{x}(t + \Delta t), t + \Delta t) &\approx J^*(\mathbf{x}(t), t) + \left[ \frac{\partial J^*}{\partial t}(\mathbf{x}(t), t) \right] \Delta t \\ &\quad + \left[ \frac{\partial J^*}{\partial \mathbf{x}}(\mathbf{x}(t), t) \right] (\mathbf{x}(t + \Delta t) - \mathbf{x}(t)). \end{aligned} \quad (2.62)$$

Kako  $\Delta t \rightarrow 0$  mogu se nadalje uvesti neke aproksimacije. Prva po redu je aproksimacija preko koje se pretpostavlja da se  $L(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)$  ne mijenja značajno kroz interval  $[t, t + \Delta t]$

$$L(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \approx L(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.63)$$

čime se dobiva

$$\begin{aligned} \int_t^{t+\Delta t} L(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau &\approx \int_t^{t+\Delta t} L(\mathbf{x}(t), \mathbf{u}(t), t) d\tau \approx L(\mathbf{x}(t), \mathbf{u}(t), t) \int_t^{t+\Delta t} d\tau \\ &\approx L(\mathbf{x}(t), \mathbf{u}(t), t) \cdot (\Delta t), \end{aligned} \quad (2.64)$$

dok je druga aproksimacija Taylorov razvoj prvog reda za  $\mathbf{x}(t + \Delta t) - \mathbf{x}(t)$  iz izraza (2.62). Za izraz  $\mathbf{x}(t + \Delta t)$  Taylorov razvoj prvog reda oko  $t$  se zapisuje

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \frac{d\mathbf{x}(t)}{dt} \Delta t. \quad (2.65)$$

Nadalje

$$\mathbf{x}(t + \Delta t) - \mathbf{x}(t) \approx \frac{d\mathbf{x}(t)}{dt} \Delta t, \quad (2.66)$$

čime se u gornju jednadžbu uvrštava izraz (2.53)

$$\mathbf{x}(t + \Delta t) - \mathbf{x}(t) \approx f(\mathbf{x}(t), \mathbf{u}(t), t) \Delta t. \quad (2.67)$$

Konačno, primjenom aproksimacijskih formulacija (2.64) i (2.67) te (2.62) se može pisati

$$\begin{aligned} J^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(t) \in \mathcal{U}} \{ & L(\mathbf{x}(t), \mathbf{u}(t), t) \Delta t + J^*(\mathbf{x}(t), t) \\ & + J_t^*(\mathbf{x}(t), t) \Delta t + J_x^*(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t) \Delta t \}, \end{aligned} \quad (2.68)$$

pri čemu se koristi skraćeni zapis parcijalnih derivacija preko indeksa koji označavaju po kojoj se varijabli derivira.

Iz gornje jednadžbe moguće je iz minimizacije maknuti članove koji ne ovise o vektoru upravljačkih varijabli  $\mathbf{u}(t)$  te pokratiti s lijeve i desne strane članove  $J^*(\mathbf{x}(t), t)$

$$0 = J_t^*(\mathbf{x}(t), t) + \min_{\mathbf{u}(t) \in \mathcal{U}} \{ L(\mathbf{x}(t), \mathbf{u}(t), t) + J_x^*(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t) \}. \quad (2.69)$$

Vidljivo je da je izvedena jednadžba parcijalna prema  $J^*(\mathbf{x}(t), t)$  te se izračunava unatrag kroz vrijeme s početnim uvjetom (koji je krajnji trošak)

$$J^*(\mathbf{x}(t_f), t_f) = \phi_f(\mathbf{x}(t_f)), \quad (2.70)$$

prilikom čega trebaju postojati kombinacije  $\mathbf{x}(t_f)$  i  $t_f$  koje zadovoljavaju niz ograničenja  $\mathbf{m}(\mathbf{x}(t_f), t_f) = 0$ . Definira se **Hamiltonian**

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), J_x^*, t) = L(\mathbf{x}(t), \mathbf{u}(t), t) + J_x^*(\mathbf{x}(t), t) f(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.71)$$

koje se ubacuje u izraz (2.69) te se definira **Hamilton-Jacobi-Bellman (HJB) jednadžba**

$$\boxed{-J_t^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(t) \in \mathcal{U}} \mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), J_x^*(\mathbf{x}(t), t), t)}, \quad \forall \mathbf{x}, \forall t \in [t_0, t_f], \quad (2.72)$$

koja je ujedno nužan i dovoljan uvjet optimalnosti te se kako je spomenuto (2.69) rješava unatrag kroz vrijeme različitim numeričkim metodama od kojih je među poznatijima metoda konačnih razlika (na primjer, Eulerova diskretizacija). Rješenje HJB jednadžbe

pruža optimalan trošak  $J^*(\mathbf{x}(t), t)$  koji predstavlja minimum funkcije cilja od početnog do konačnog stanja kroz određeno vrijeme uz optimalno upravljanje koje se definira

$$\begin{aligned} \mathbf{u}^*(\mathbf{x}(t), t) &= \arg \min_{\mathbf{u} \in \mathcal{U}} \{L(\mathbf{x}(t), \mathbf{u}(t), t) + J_x^*(\mathbf{x}(t), \mathbf{u}(t), t)f(\mathbf{x}(t), \mathbf{u}(t), t)\} \\ &= \arg \min_{\mathbf{u}(t) \in \mathcal{U}} \mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), J_x^*(\mathbf{x}(t), t), t). \end{aligned} \quad (2.73)$$

Pokazani izvod HJB jednadžbe napisan je po uzoru na [3].

## 2.3 Linearni kvadratični regulator

Poznata je dinamika sustava opisana skupom linearnih diferencijalnih jednadžbi te je funkcija cilja kvadratna; takav problem naziva se linearni kvadratični (LQ) problem. Rješenje takvog problema pruža linearni kvadratični regulator koji zaprima težinske matrice  $\mathbf{Q}$  i  $\mathbf{R}$  pri čemu

- $\mathbf{Q} = \mathbf{Q}^T$ ,  $\mathbf{Q} \succeq 0$ ,  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  je simetrična pozitivno semidefinitna matrica, kojom se definira koliko se *kažnjavaju* odstupanja stanja sustava od željenih vrijednosti,
- $\mathbf{R} = \mathbf{R}^T$ ,  $\mathbf{R} \succ 0$ ,  $\mathbf{R} \in \mathbb{R}^{m \times m}$  je simetrično pozitivno definitna matrica, kojom se definira koliko se *kažnjavaju* upravljački naponi, to jest, zahtjev za potrebnom energijom kojom se sustav dovodi u željeno stanje,

te se preko navedenih težinskih matrica pronalazi optimalan vektor upravljačkih varijabli  $\mathbf{u}(t)$  kojim se minimizira funkcija cilja, to jest, pronalazi optimalan trošak. Za početak potrebno je primijeniti HJB jednadžbu na kvadratični problem u cilju formulacije linearnog kvadratičnog regulatora (LQR).

### 2.3.1 Primjena Hamilton-Jacobi-Bellmanove jednadžbe na kvadratični problem

Razmatra se slučaj konačnog vremena; neka je poznat linearan vremenski invarijantan dinamički sustav opisan

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (2.74)$$

pri čemu

- $\mathbf{A}$  označava matricu sustava,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,
- $\mathbf{B}$  označava ulaznu matricu,  $\mathbf{B} \in \mathbb{R}^{n \times m}$ ,
- $\mathbf{x}(t)$  označava vektor varijabli stanja,  $\mathbf{x}(t) \in \mathbb{R}^n$ ,

- $\mathbf{u}(t)$  označava vektor ulaznih varijabli,  $\mathbf{u}(t) \in \mathbb{R}^m$ ,

te je krajnji trošak  $\phi_f(\mathbf{x}(t_f)) = \mathbf{x}^T \mathbf{Q}_{t_f} \mathbf{x}$  i trenutni trošak  $L(\mathbf{x}, \mathbf{u}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}$ . Matrica  $\mathbf{Q}_{t_f}$  je pozitivno definitna matrica što osigurava da optimalan trošak  $J^*$  bude pozitivan.

Neka se uzme u obzir jednačba (2.69) i u istu uvrste izrazi trenutnog troška i krajnjeg troška (za koje se kaže da je kvadratni zbog množenja stanja, to jest, upravljačkih varijabli) i opis dinamike sustava (2.74)

$$0 = J_t^*(\mathbf{x}(t), t) + \min_{\mathbf{u}(t) \in \mathcal{U}} \{ \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + J_x^*(\mathbf{x}(t), t)(\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) \}, \quad (2.75)$$

pri čemu optimalan trošak glasi

$$J^*(\mathbf{x}, t) = \mathbf{x}^T \mathbf{S}(t) \mathbf{x}, \quad \mathbf{S}(t) = \mathbf{S}^T(t) \succ 0, \quad (2.76)$$

prilikom čega parcijalne derivacije  $J_t^*$  i  $J_x^*$  izraza (2.75) zapisuju

$$J_t^* = \frac{\partial J^*}{\partial t} = \mathbf{x}^T \dot{\mathbf{S}}(t) \mathbf{x}, \quad J_x^* = \frac{\partial J^*}{\partial \mathbf{x}} = 2\mathbf{x}^T \mathbf{S}(t). \quad (2.77)$$

Gornje raspisane jednačbe neka se primjene na (2.75)

$$\begin{aligned} 0 &= \mathbf{x}^T \dot{\mathbf{S}}(t) \mathbf{x} + \min_{\mathbf{u} \in \mathcal{U}} \{ \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + 2\mathbf{x}^T \mathbf{S}(t)(\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}) \} \\ &= \mathbf{x}^T \dot{\mathbf{S}}(t) \mathbf{x} + \min_{\mathbf{u} \in \mathcal{U}} \{ \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + 2\mathbf{x}^T \mathbf{S}(t) \mathbf{A} \mathbf{x} + 2\mathbf{x}^T \mathbf{S}(t) \mathbf{B} \mathbf{u} \}. \end{aligned} \quad (2.78)$$

Cilj je pronalazak optimalnog upravljanja  $\mathbf{u}^*$ . Kako je funkcija cilja

$$J(\mathbf{x}, \mathbf{u}) = \int_{t_0}^{t_f} [\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt + \mathbf{x}^T \mathbf{Q}_{t_f} \mathbf{x}, \quad (2.79)$$

kvadratna, to jest, konveksna funkcija, za koje vrijedi da je lokalni minimum ujedno i globalni minimum, moguć je pronalazak globalnog optimalnog  $\mathbf{u}^*$  preko parcijalne derivacije izraza (2.78) prema  $\mathbf{u}$  i izjednačavanjem s nulom (izraz je već jednak nuli). Članovi koji ne ovise o  $\mathbf{u}$  su jednaki nuli, dok

$$\frac{\partial}{\partial \mathbf{u}} (\mathbf{u}^T \mathbf{R} \mathbf{u}) = 2\mathbf{R} \mathbf{u}, \quad \frac{\partial}{\partial \mathbf{u}} (2\mathbf{x}^T \mathbf{S}(t) \mathbf{B} \mathbf{u}) = 2\mathbf{B}^T \mathbf{S}(t) \mathbf{x}. \quad (2.80)$$

Slijedi

$$0 = 2\mathbf{R} \mathbf{u}^* + 2\mathbf{B}^T \mathbf{S}(t) \mathbf{x}, \quad (2.81)$$

te konačno

$$\boxed{\begin{aligned} \mathbf{u}^*(\mathbf{x}, t) &= -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t) \\ &= -\mathbf{K}(t) \mathbf{x}. \end{aligned}} \quad (2.82)$$

Prema (2.49) i (2.51) se zapisuju nužni kriteriji optimalnog upravljanja za kvadratni kriterij optimalnosti

$$\boxed{\begin{aligned}\frac{\partial \mathcal{H}}{\partial \lambda} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \\ \dot{\lambda}(t) &= -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \\ \frac{\partial \mathcal{H}}{\partial \mathbf{u}} &= 2\mathbf{R}\mathbf{u} + \mathbf{B}^T \lambda(t).\end{aligned}} \quad (2.83)$$

Neka se izraz (2.82), koji je definiran kao optimalno globalno upravljanje (minimum) može primijeniti na izraz (2.78) jer isti zahtjeva minimalan  $\mathbf{u}$  za računanje te se na lijevu stranu izlučuje  $\mathbf{x}^T$ , a na desnu stranu  $\mathbf{x}$  jer je svaki član bio množen njima.

$$\begin{aligned}0 &= \mathbf{x}^T [\dot{\mathbf{S}}(t) + \mathbf{Q} + \mathbf{S}(t)\mathbf{B}^T \mathbf{R}^{-1} \mathbf{R} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t)] + 2\mathbf{S}(t)\mathbf{A} - 2\mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t)]\mathbf{x} \\ &= \mathbf{x}^T [\dot{\mathbf{S}}(t) + \mathbf{Q} + \mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t)] + 2\mathbf{S}(t)\mathbf{A} - 2\mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t)]\mathbf{x} \\ &= \mathbf{x}^T [\dot{\mathbf{S}}(t) + \mathbf{Q} - \mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t) + 2\mathbf{S}(t)\mathbf{A}]\mathbf{x} \\ &= \mathbf{x}^T [\dot{\mathbf{S}}(t) + \mathbf{Q} - \mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t) + \mathbf{S}(t)\mathbf{A} + \mathbf{A}^T \mathbf{S}(t)]\mathbf{x}\end{aligned} \quad (2.84)$$

Da se osigura minimizacija funkcije cilja  $J$  za sva stanja  $\mathbf{x}$  potrebno je izraz unutar kvadratične forme izjednačiti s nulom, čime je definirana obična **diferencijalna Riccatijeva jednadžba**

$$\boxed{-\dot{\mathbf{S}}(t) = \mathbf{Q} - \mathbf{S}(t)\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}(t) + \mathbf{S}(t)\mathbf{A} + \mathbf{A}^T \mathbf{S}(t)}, \quad (2.85)$$

te je  $\mathbf{S}(t) \in \mathbb{R}^{n \times n}$  rješenje numeričke integracije uz dani konačni uvjet  $\mathbf{S}(t_f) = \mathbf{Q}_{t_f}$ .

Za dobivanje optimalnog rješenja linearnog kvadratnog problema, postojanje rješenja Riccatijeve jednadžbe (2.85) uvijek je nužan i dovoljan uvjet. Za slučaj da se promatra problem beskonačnog vremena ( $t_f = \infty$ ), zanemaruju se sve ovisnosti o vremenu te se za početak iz izraza (2.75) može ukloniti član ovisan o vremenu iz čega u konačnici istom sljednošću kao i u izvodu kada je vrijeme konačno, slijedi izraz

$$\boxed{0 = \mathbf{Q} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} + \mathbf{S}\mathbf{A} + \mathbf{A}^T \mathbf{S}}. \quad (2.86)$$

Prikazan izraz naziva se **algebarska Riccatijeva jednadžba** čije je optimalno upravljanje  $\mathbf{u}^* = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}$  uz pretpostavku da optimalan trošak kvadratične forme glasi  $J^*(\mathbf{x}) = \mathbf{x}^T \mathbf{S}\mathbf{x}$ .

## 2.4 Vremenski varijantan linearni kvadratični regulator

Prethodno definirani izvodi također vrijede ako je dinamika sustava

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (2.87)$$



pri čemu se naravno varijable stanja mijenjaju s vremenom, ali je zanimljiva vremenska varijantnost matrice sustava  $\mathbf{A}$  i matrice ulaza  $\mathbf{B}$ . Slično tome, matrice troška  $\mathbf{Q}$  i  $\mathbf{R}$  također mogu varirati kroz vrijeme. To je prilično iznenađujuće, budući da je klasa vremenski varijabilnih linearnih sustava prilično opća klasa sustava. Uglavnom se ne zahtijevaju nikakve pretpostavke o tome kako vremenska ovisnost ulazi u sustav, osim možda da ako su matrice  $\mathbf{A}$  i  $\mathbf{B}$  diskontinuirane kroz vrijeme te bi se tada morale koristiti odgovarajuće tehnike za točno integriranje diferencijalne jednadžbe [4].

### 2.4.1 Linearni kvadratični regulator i slijeđenje referentne trajektorije

Kako je prije spomenuto, postoje problemi konačnog vremena i beskonačnog vremena. U problemima beskonačnog vremena, kako vrijeme teži beskonačnosti, stanja sustava teže (konvergiraju) prema nuli ili ravnotežnoj točki (*eng. equilibrium point*) zato što je optimalno upravljanje takvo da stabilizira sustav te minimizira funkciju cilja kroz beskonačan vremenski interval. S druge strane, problem praćenja referentne trajektorije je problem konačnog vremena koji se pojavljuje u stvarnim primjenama u robotici, autonomnim vozilima i slično gdje je cilj pratiti željenu referentnu trajektoriju od početnog do konačnog stanja kroz zadani vremenski interval.

Prvo će se osmotriti lokalna stabilizacija nelinearnog sustava zadanog  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$  prilikom čega postoji ravnotežna točka, koja je ujedno točka prema kojoj sustav teži,  $(\mathbf{x}_g, \mathbf{u}_g)$  za koju vrijedi ravnotežno stanje  $\dot{\mathbf{x}} = f(\mathbf{x}_g, \mathbf{u}_g) = 0$ . Neka se uvede razlika trenutnog stanja i konačnog stanja kao i trenutnog i željenog upravljanja, to jest, koordinate pogreške

$$\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_g, \quad \bar{\mathbf{u}} = \mathbf{u} - \mathbf{u}_g. \quad (2.88)$$

Bitno je primjetiti

$$\begin{aligned} \dot{\bar{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_g, \\ \dot{\bar{\mathbf{x}}} &= \dot{\mathbf{x}}. \end{aligned} \quad (2.89)$$

Uvodi se aproksimacija (linearizacija) sustava  $\dot{\bar{\mathbf{x}}} = f(\mathbf{x}, \mathbf{u})$  Taylorovim razvojem prvog reda oko ravnotežne točke (čime se sagledava kako se sustav *ponaša* oko ravnotežne točke te navedeno vrijedi za male perturbacije oko iste) prilikom čega se zadržavaju samo linearni članovi

$$\dot{\bar{\mathbf{x}}} = f(\mathbf{x}, \mathbf{u}) \approx f(\mathbf{x}_g, \mathbf{u}_g) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_g, \mathbf{u}_g)} (\mathbf{x} - \mathbf{x}_g) + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_g, \mathbf{u}_g)} (\mathbf{u} - \mathbf{u}_g). \quad (2.90)$$

Neka se definiraju Jacobijeve matrice

$$\mathbf{A} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_g, \mathbf{u}_g)}, \quad \mathbf{B} = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_g, \mathbf{u}_g)}, \quad (2.91)$$

te se može pisati

$$\dot{\bar{\mathbf{x}}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{B}\bar{\mathbf{u}}, \quad (2.92)$$

prilikom čega je moguće definirati funkciju cilja te optimalan trošak, također u koordinatama pogreške

$$J(\bar{\mathbf{x}}(0)) = \int_0^\infty [\bar{\mathbf{x}}^T \mathbf{Q} \bar{\mathbf{x}} + \bar{\mathbf{u}}^T \mathbf{R} \bar{\mathbf{u}}] dt, \quad \mathbf{Q} = \mathbf{Q}^T \succeq 0, \mathbf{R} = \mathbf{R}^T \succ 0, \quad (2.93)$$

$$J^*(\bar{\mathbf{x}}) = \bar{\mathbf{x}}^T \mathbf{S} \bar{\mathbf{x}}, \quad (2.94)$$

iz čega proizlazi optimalno upravljanje u koordinatama pogreške

$$\begin{aligned} \bar{\mathbf{u}}^* &= -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} \bar{\mathbf{x}} \\ &= -\mathbf{K} \bar{\mathbf{x}}, \end{aligned} \quad (2.95)$$

to jest, optimalno upravljanje

$$\mathbf{u}^* = \mathbf{u}_g - \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} (\mathbf{x} - \mathbf{x}_g). \quad (2.96)$$

Vremenski varijantan linearni kvadratični regulator (*eng. Time-varying linear quadratic regulator* - **TVLQR**) je primjenjiv za linearizaciju oko referentne trajektorije nelinearnog sustava.

Neka je poznata referentna (željena) trajektorija<sup>1</sup>,  $\mathbf{x}_{\text{ref}}(t)$ ,  $\mathbf{u}_{\text{ref}}(t)$  za vremenski interval  $t \in [t_0, t_f]$ . Kao i za koordinate pogreške prema ravnotežnoj točki, sada se definiraju koordinate pogreške između trenutne trajektorije i referentne trajektorije i pripadnih upravljanja

$$\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_{\text{ref}}(t), \quad \bar{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_{\text{ref}}(t), \quad (2.97)$$

prilikom čega vrijedi

$$\dot{\bar{\mathbf{x}}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{ref}} = f(\mathbf{x}, \mathbf{u}) - f(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}}). \quad (2.98)$$

Po uzoru na (2.90) se sada radi linearizacija raspisom Taylorovog razvoja prvog reda oko referentne trajektorije

$$\dot{\bar{\mathbf{x}}} \approx f(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}}) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})} (\mathbf{x} - \mathbf{x}_{\text{ref}}) + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})} (\mathbf{u} - \mathbf{u}_{\text{ref}}) - f(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}}), \quad (2.99)$$

gdje vrijede Jacobijeve matrice

$$\mathbf{A} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})}, \quad \mathbf{B} = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_{\text{ref}}, \mathbf{u}_{\text{ref}})}, \quad (2.100)$$

te nadalje

$$\dot{\bar{\mathbf{x}}}(t) = \mathbf{A}(t)\bar{\mathbf{x}}(t) + \mathbf{B}(t)\bar{\mathbf{u}}(t). \quad (2.101)$$

<sup>1</sup>Vremenski promjenjiva krivulja opisana točkama koje se gibaju u prostoru ili ravnini

Dobiven izraz sličan je (2.92), no postoje razlike. Kao prvo, linearizacija je vremenski varijantna, dok je s druge strane linearizacija valjana za bilo koje stanje duž izvedive trajektorije (ne samo za fiksne točke), jer se koordinatni sustav kreće zajedno s trajektorijom [4].

Definira se funkcija cilja gdje je uračunat krajnji trošak jer se radi o vremenski varijantnom problemu te optimalan trošak

$$J(\bar{\mathbf{x}}(t_0), t_0) = \int_{t_0}^{t_f} [\bar{\mathbf{x}}^T(t) \mathbf{Q} \bar{\mathbf{x}}(t) + \bar{\mathbf{u}}^T(t) \mathbf{R} \bar{\mathbf{u}}(t)] dt + \bar{\mathbf{x}}^T(t_f) \mathbf{Q}_{t_f} \bar{\mathbf{x}}(t_f), \quad (2.102)$$

$$J^*(\bar{\mathbf{x}}, t) = \bar{\mathbf{x}}^T(t) \mathbf{S}(t) \bar{\mathbf{x}}(t), \quad (2.103)$$

uz  $\mathbf{Q} = \mathbf{Q}^T \succeq 0$ ,  $\mathbf{R} = \mathbf{R}^T \succ 0$ ,  $\mathbf{Q}_{t_f} = \mathbf{Q}_{t_f}^T \succ 0$ . Kako vrijedi Bellmanov princip, početak vremenskog intervala može početi u bilo kojem vremenskom trenutku intervala  $t \in [t_0, t_f]$ . Važno je napomenuti da se težinskoj matrici krajnjeg troška  $\mathbf{Q}_{t_f}$  pridodaju veće težine (parametri matrice) u odnosu na težinsku matricu  $\mathbf{Q}$  zbog težnje za konvergencijom trenutne trajektorije referentnoj trajektoriji kako nadolazi vremenski trenutak  $t_f$  ( $\bar{\mathbf{x}}(t_f) \rightarrow 0$ ). Matrica pojačanja  $\mathbf{K}$  postaje vremenski varijantna zbog potrebe za korekcijom težnje za optimalnošću kroz svaki vremenski trenutak; formulacije optimalnog upravljanja u koordinatama pogreške, kao i optimalnog upravljanja, su iste poput (2.95) i (2.96) samo su vremenski varijantne

$$\bar{\mathbf{u}}^*(t) = -\mathbf{K}(t) \bar{\mathbf{x}}, \quad (2.104)$$

te

$$\begin{aligned} \mathbf{u}^*(t) &= \mathbf{u}_{\text{ref}}(t) - \mathbf{K}(t)(\mathbf{x} - \mathbf{x}_{\text{ref}}(t)) \\ &= \mathbf{u}_{\text{ref}}(t) - \mathbf{R}^{-1} \mathbf{B}^T(t) \mathbf{S}(t)(\mathbf{x}(t) - \mathbf{x}_{\text{ref}}(t)). \end{aligned} \quad (2.105)$$

Objašnjenjem problema konačnog vremena i problema beskonačnog vremena, slijedi njihovo povezivanje u cilju osiguranja stabilnosti sustava kroz cijelu trajektoriju (jer ipak je stabilnost pojam koji se veže s odlaskom vremena u beskonačnost).

Trajektorija je vremenski varijantna krivulja te se definira kroz  $t \in [t_0, \infty]$ , no cilj je doseći ravnotežno stanje u određeno vrijeme u kojemu će sustav biti i ostati stabilan daljnjim tokom vremena (prema str.18).

Za spomenuto ravnotežno stanje se preko algebarske Riccatijeve jednadžbe (2.86), koja se koristi za problem beskonačnog vremena (koristi se algebarska Riccatijeva jednadžba jer ravnotežno stanje ostaje ravnotežno s daljnjim tokom vremena u beskonačnost) te se dobiva  $\mathbf{S} = \mathbf{S}_{\text{are}}$  te će dobiveno biti inicijalizacija za problem konačnog vremena koji se rješava unatrag kroz vrijeme (prema str.18).  $\mathbf{S}_{\text{are}}$  se postavlja kao krajnji, to jest, u ovom slučaju, početni uvjet u vremenskom trenutku  $t_f$  ( $\mathbf{S}_{\text{are}} = \mathbf{S}(t_f)$ ), nakon čega slijedi numeričko rješavanje obične diferencijalne Riccatijeve jednadžbe (2.85) unatrag kroz vrijeme do vremena  $t_0$  u cilju dobivanja  $\mathbf{S}(t)$  te slijedno  $\mathbf{K}(t)$  i  $\mathbf{u}^*(t)$  (2.105).

## 2.5 Metode rješavanja Riccatijeve jednadžbe

Rješenje Riccatijeve jednadžbe, koja prema tipu problema može biti algebarska (sustav nije vremenski varijantan) ili diferencijalna (sustav je vremenski varijantan), moguće je izračunati kroz niz različitih numeričkih metoda.

Algebarska Riccatijeva jednadžba je algebarska jednadžba, što znači da je za sustave manjeg reda rješiva klasičnim analitičkim putem, matričnim manipulacijama, no za sustave većeg reda, primjenjuju se numeričke metode.

Kroz rad će se obraditi numerička metoda dobivanja rješenja preko **Schurove dekompozicije** te **Newtonova metoda**, dok će se za dobivanje rješenja diferencijalne Riccatijeve jednadžbe obraditi numerička **Eulerova metoda** te **Dormand - Prince (Runge-Kutta (4,5)) metoda**.

### 2.5.1 Schurova dekompozicija

Nazvana prema njemačkom matematičaru Issaiju Schuru, Shurova dekompozicija, pripada području matričnih dekompozicija. Definicije, teoremi te primjer su preuzeti i obrađeni prema [5]. Za njezino shvaćanje će se za početak uvesti nekoliko definicija.

Neka  $\mathbf{A}_s \in \mathbb{F}^{m \times n}$  označava  $m \times n$  matricu sa članovima polja  $\mathbb{F}$ . Polje  $\mathbb{F}$  označava realne brojeve  $\mathbb{R}$  ili kompleksne brojeve  $\mathbb{C}$ . Nadalje, notacije  $\mathbf{A}_s^T$  i  $\mathbf{A}_s^H$  će označavati transponiranu i konjugirano - transponiranu matricu, dok će notacija  $\mathbf{A}_s^+$  označavati Moore - Penrosov pseudoinverz matrice  $\mathbf{A}_s$ . Za  $\mathbf{A}_s \in \mathbb{R}^{n \times n}$  neka se spektar (set  $n$  svojstvenih vrijednosti  $\lambda_s$ ) označi kao  $\sigma(\mathbf{A}_s)$ .

#### Definicija 2.1

$\mathbf{A}_s \in \mathbb{R}^{n \times n}$  je ortogonalna ako  $\mathbf{A}_s^T = \mathbf{A}_s^{-1}$ . ■

#### Definicija 2.2

$\mathbf{A}_s \in \mathbb{C}^{n \times n}$  je unitarna ako  $\mathbf{A}_s^H = \mathbf{A}_s^{-1}$ . ■

Neka  $\mathbf{J} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \mathbb{R}^{2n \times 2n}$  gdje  $I$  označava jediničnu matricu (matricu identiteta)  $n$ -tog reda. Napomena:  $\mathbf{J}^T = \mathbf{J}^{-1} = -\mathbf{J}$ .

#### Definicija 2.3

$\mathbf{A}_s \in \mathbb{R}^{2n \times 2n}$  je Hamiltonian (matrica) ako  $\mathbf{J}^{-1} \mathbf{A}_s^T \mathbf{J} = -\mathbf{A}_s$ . ■

#### Definicija 2.4

$\mathbf{A}_s \in \mathbb{R}^{2n \times 2n}$  je simplektička matrica ako  $\mathbf{J}^{-1} \mathbf{A}_s^T \mathbf{J} = \mathbf{A}_s^{-1}$ . ■

Nadalje slijedi nekoliko teorema prema kojima se dobiva dublje razumijevanje prema Schurovoj dekompoziciji.

**Teorem 2.1**

1.) Neka  $\mathbf{A}_s \in \mathbb{R}^{2n \times 2n}$  bude Hamiltonian. Tada  $\lambda_s \in \sigma(\mathbf{A}_s)$  implicira  $-\lambda_s \in \sigma(\mathbf{A}_s)$  s istom višestrukošću. 2.) Neka  $\mathbf{A}_s \in \mathbb{R}^{2n \times 2n}$  bude simplektička. Tada  $\lambda_s \in \sigma(\mathbf{A}_s)$  implicira  $1/\lambda_s \in \sigma(\mathbf{A}_s)$  s istom višestrukošću. ■

**Teorem 2.2**

Neka  $\mathbf{A}_s \in \mathbb{R}^{2n \times 2n}$  bude Hamiltonian (ili simplektička). Neka  $\mathbf{U} \in \mathbb{R}^{2n \times 2n}$  bude simplektička matrica. Tada je  $\mathbf{U}^{-1}\mathbf{A}_s\mathbf{U}$  Hamiltonian (ili simplektička). ■

**Teorem 2.3 Schurova kanonička forma.**

Neka  $\mathbf{A}_s \in \mathbb{R}^{n \times n}$  ima svojstvene vrijednosti  $\lambda_{s_1}, \dots, \lambda_{s_n}$ . Tada postoji unitarno slična  $\mathbf{U}$  takva da je  $\mathbf{U}^H\mathbf{A}_s\mathbf{U}$  gornje trokutasta matrica sa svojstvenim vrijednostima  $\lambda_{s_1}, \dots, \lambda_{s_n}$  na dijagonali, prikazanim redosljedom. ■

Idući podatak je zanimljiv. Moguće je raditi samo sa realnim matricama redukcijom na kvazi gornje trokutastu formu sa sljedećim blokovima na dijagonali;  $2 \times 2$  blokovima, što odgovara kompleksno - konjugiranim svojstvenim vrijednostima od  $\mathbf{A}_s$  te  $1 \times 1$  blokovima što odgovara realnim svojstvenim vrijednostima od  $\mathbf{A}_s$ . Takva se forma naziva realnom Schurovom formom te se zapisuje  $\mathbf{A}_s = \mathbf{U}\mathbf{T}\mathbf{U}^T$ , a transformacija  $\mathbf{A}_s$  u realnu Schurovu formu se dobiva QR iteracijskim algoritmom (dekompozicijom).

**Teorem 2.4 Realna Schurova forma.**

Neka  $\mathbf{A}_s \in \mathbb{R}^{n \times n}$ . Tada postoji ortogonalno slična  $\mathbf{U}$  (jer se radi o slučaju skupa  $\mathbb{R}$ , dok je za skup  $\mathbb{C}$ , tada bi  $\mathbf{U}$  bila unitarna) takva da je  $\mathbf{T} = \mathbf{U}^T\mathbf{A}_s\mathbf{U}$  kvazi gornje trokutasta matrica. ■

Ako se iz prethodnog teorema  $\mathbf{U}^T\mathbf{A}_s\mathbf{U}$  podijeli na  $\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}$  gdje  $T_{11} \in \mathbb{R}^{k_s \times k_s}, 0 < k_s \leq n$ , tada se prvih  $k_s$  vektora matrice  $\mathbf{U}$  naziva Schurovim vektorima koji odgovaraju  $\sigma(T_{11}) \subseteq \sigma(\mathbf{A}_s)$ .

Slijedi primjena na algebarsku Riccatijevu jednadžbu (2.86). Napomena:  $\mathbf{A}_s$  iz uvoda iz Schurove dekompozicije sada odgovara Hamiltonian matrici  $\mathcal{H}$  koja će se svoditi na realnu Schurovu formu te se uvode otprije poznate oznake matrica vezanih za Riccatijevu jednadžbu ( $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}, \mathbf{S}$ ). Uvodi se oznaka  $\mathbf{G} = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$ . Sve matrice unutar

Hamiltonieve su  $\mathbb{R}^{n \times n}$  te  $\mathbf{G} = \mathbf{G}^T \geq 0$ . Formira se Hamiltonian matrica

$$\mathcal{H} = \begin{bmatrix} \mathbf{A} & -\mathbf{G} \\ -\mathbf{Q} & -\mathbf{A}^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \quad (2.106)$$

Prethodne definicije osiguravaju da  $\mathcal{H}$  nema imaginarnih svojstvenih vrijednosti. Prema tome, u obzir se uzima Teorem (2.4) prema kojemu postoji ortogonalna  $\mathbf{U} \in \mathbb{R}^{2n \times 2n}$  te se  $\mathcal{H}$  postavlja u realnu Schurovu formu

$$\mathbf{U}^T \mathcal{H} \mathbf{U} = \mathbf{T} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \quad (2.107)$$

gdje  $T_{ij} \in \mathbb{R}^{n \times n}$ . Moguće je postaviti realne spektre od  $T_{11}$  kao negativne te realne spektre od  $T_{22}$  kao pozitivne. Tada se  $\mathbf{U}$  može formulirati preko četiri  $n \times n$  bloka

$$\mathbf{U} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}. \quad (2.108)$$

Tada slijedi idući teorem.

### Teorem 2.5

Prema gornjoj notaciji i pretpostavkama,

- 1.)  $U_{11}$  je invertibilna te  $\mathbf{S} = U_{21}U_{11}^{-1}$  biva rješenjem (2.86),  $\mathbf{S} = \mathbf{S}^T \succeq 0$ .
- 2.)  $\sigma(T_{11}) = \sigma(\mathbf{A} - \mathbf{G}\mathbf{S})$ . ■

Slijedi primjer.

### Primjer 2.

Schurov pristup rješavanju algebarske Riccatijeve jednadžbe nije prikladan za "ručno" rješavanje, no za shvaćanje prethodno napisanog priložen je primjer. Poznata je jednadžba

$$\mathbf{A}^T \mathbf{S} + \mathbf{S} \mathbf{A} - \mathbf{S} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} + \mathbf{Q} = 0, \quad (2.109)$$

pri čemu  $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ,  $\mathbf{R} = 1$ ,  $\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$ . Tada

$$\mathcal{H} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -2 & -1 & 0 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{5}}{10} & -\frac{3\sqrt{5}}{10} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{\sqrt{5}}{10} & -\frac{3\sqrt{5}}{10} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{3\sqrt{5}}{10} & \frac{\sqrt{5}}{10} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{3\sqrt{5}}{10} & \frac{\sqrt{5}}{10} & \frac{1}{2} \end{bmatrix}. \quad (2.110)$$

$\mathbf{U}$  je ortogonalna matrica kojom  $\mathcal{H}$  postavljamo u realnu Schurovu formu

$$\mathbf{T} = \mathbf{U}^T \mathcal{H} \mathbf{U} = \begin{bmatrix} -1 & 0 & 1 & -\frac{1}{2} \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.111)$$

iz čega slijedi rješenje  $\mathbf{S}$  iz linearne matricne jednadžbe  $\mathbf{S}U_{11} = U_{21}$  to jest

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{5}}{10} \\ -\frac{1}{2} & -\frac{\sqrt{5}}{10} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{3\sqrt{5}}{10} \\ -\frac{1}{2} & -\frac{3\sqrt{5}}{10} \end{bmatrix}. \quad (2.112)$$

Rješenje je  $\mathbf{S} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$  te se može jednostavno provjeriti da je spektar od  $\mathbf{A} - \mathbf{G}\mathbf{S}$ , to jest,  $\sigma(\mathbf{A} - \mathbf{G}\mathbf{S})$ , jednak  $\{-1, -1\}$  što je bilo vidljivo iz  $T_{11}$ .

## 2.5.2 Newtonova metoda

Newtonova metoda, ili kako se još naziva, Newton-Raphsonova metoda, matematička je metoda za numeričko rješavanje jednadžbi. Kao većina diferencijalnog računa, bazirana je na ideji linearne aproksimacije. Tematika je obrađena prema [6] te je ujedno iz iste preuzet pokazni primjer Newtonove metode. Uobičajena Newtonova metoda za nalaženje nultočke  $x$  funkcije  $f(x) = 0$  opisuje se na sljedeći način

- Za inicijalizaciju se odabire početan uvjet  $x_0$ , to jest, početna aproksimacija budućeg rješenja  $x$
- Definira se niz aproksimacija  $(x_i)$  definiranih prema

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i \geq 0. \quad (2.113)$$

Tada, ako je  $x_0$  dobro odabran, sekvence  $\{x_i\}$  konvergiraju prema nultočki  $x$  te je konvergencija kvadratna (broj točnih decimala se približno udvostručuje svakom iteracijom) ako  $f'(x) \neq 0$ .

Newtonova metoda jednostavno se implementira na kontinuiranu te diskretnu algebarsku Riccatijevu jednadžbu, no nadalje će se obraditi sve vezano uz kontinuiranu algebarsku Riccatijevu jednadžbu. Neka se uzme u obzir (2.86). Polazeći od početnog aproksimativnog rješenja  $\mathbf{S}_0$ , dobivena rješenja se iteriraju dok ne dođe do konvergencije; to se postiže rješavanjem Ljapunovljeve jednadžbe u svakoj iteraciji. Može se pisati  $\mathbf{S} = \mathbf{S}_0 + (\mathbf{S} - \mathbf{S}_0)$  te uvrstiti u (2.86).

$$(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0)^T\mathbf{S} + \mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0) = -\mathbf{S}_0\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0 - \mathbf{Q} + (\mathbf{S} - \mathbf{S}_0)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T(\mathbf{S} - \mathbf{S}_0).$$

Pod pretpostavkom da je  $\mathbf{S} - \mathbf{S}_0$  malog iznosa (to jest, da je početno aproksimativno rješenje dobro), može se zanemariti zadnji član na desnoj strani gornje jednadžbe. Time se dobiva Ljapunovljeva jednadžba za iduću po redu aproksimaciju  $\mathbf{S}_1$ :

$$(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0)^T\mathbf{S}_1 + \mathbf{S}_1(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0) = -\mathbf{S}_0\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}_0 - \mathbf{Q}.$$

Pod pretpostavkom da je  $\mathbf{S}_1$  bolja aproksimacija od  $\mathbf{S}_0$  (to jest,  $\|\mathbf{S} - \mathbf{S}_1\| \ll \|\mathbf{S} - \mathbf{S}_0\|$ ), postupak se nastavlja dok ne dođe do konvergencije. Prethodno opisan postupak se zapisuje

1. Odabir početne aproksimacije  $\mathbf{S}_0$ .
2. Iterativni izračun  $\mathbf{S}_k$  rješavanjem Ljapunovljeve jednadžbe:

$$(\mathbf{A} - \mathbf{GS}_k)^T \mathbf{S}_{k+1} + \mathbf{S}_{k+1}(\mathbf{A} - \mathbf{GS}_k) = -\mathbf{S}_k \mathbf{GS}_k - \mathbf{Q}, \quad k = 0, 1, 2, \dots$$

gdje  $\mathbf{G} = \mathbf{BR}^{-1}\mathbf{B}^T$ .

3. Postupak se nastavlja dok ne dođe do konvergencije.

Za daljnje razmatranje definira je još jedan tip Newtonove metode prilikom čega se koristi Fréchetova derivacija (koja neće biti implicitno izvođena). Neka se definira operator

$$R_C(\mathbf{S}) = \mathbf{SA} + \mathbf{A}^T \mathbf{S} - \mathbf{SGS} + \mathbf{Q}, \quad (2.114)$$

i na isti primjeni perturbacija  $Z$ ,  $\mathbf{S} \rightarrow \mathbf{S} + Z$

$$R_C(\mathbf{S} + Z) = (\mathbf{S} + Z)\mathbf{A} + \mathbf{A}^T(\mathbf{S} + Z) - (\mathbf{S} + Z)\mathbf{G}(\mathbf{S} + Z) + \mathbf{Q}, \quad (2.115)$$

nakon raspisivanja

$$R_C(\mathbf{S} + Z) = \mathbf{SA} + Z\mathbf{A} + \mathbf{A}^T \mathbf{S} + \mathbf{A}^T Z - \mathbf{SGS} - \mathbf{SGZ} - Z\mathbf{GS} - Z\mathbf{GZ} + \mathbf{Q}, \quad (2.116)$$

te je primjetno da su članovi bez  $Z$ -a ustvari  $R_C(\mathbf{S})$ , dok se kvadratni član  $Z\mathbf{GZ}$ , pošto je vrlo niske vrijednosti, miče. Zapisuje se

$$R_C(\mathbf{S} + Z) - R_C(\mathbf{S}) = Z\mathbf{A} + \mathbf{A}^T Z - \mathbf{SGZ} - Z\mathbf{GS}. \quad (2.117)$$

Uvodi se aproksimacija  $R_C(\mathbf{S} + Z) \approx R_C(\mathbf{S}) + R'_{\mathbf{S}}(Z)$  te se iz iste, prema prethodnome, definira Fréchetova derivacija

$$\begin{aligned} R'_{\mathbf{S}}(Z) &= Z\mathbf{A} + \mathbf{A}^T Z - \mathbf{SGZ} - Z\mathbf{GS} \\ &= (\mathbf{A} - \mathbf{GS})^T Z + Z(\mathbf{A} - \mathbf{GS}). \end{aligned} \quad (2.118)$$

Neka se također uvede aproksimacija

$$R_C(\mathbf{S}_{i+1}) \approx R_C(\mathbf{S}_i) + R'_{\mathbf{S}_i}(\Delta_i), \quad (2.119)$$

gdje  $\mathbf{S}_{i+1} = \mathbf{S}_i + \Delta_i = \mathbf{S} + Z$ ,  $\Delta = Z$ . Kako je cilj da  $\mathbf{S}_{i+1}$  bude što bolja aproksimacija stvarnog rješenja te ujedno bolja od prethodne, teži se prema tome da operator  $R_C(\mathbf{S}_{i+1}) = 0$  te slijedi Ljapunovljeva jednadžba

$$\begin{aligned} R'_{\mathbf{S}_i}(\Delta_i) + R_C(\mathbf{S}_i) &= 0, \\ (\mathbf{A} - \mathbf{GS})^T Z + Z(\mathbf{A} - \mathbf{GS}) &= 0, \\ \mathbf{A}_i^T \Delta_i + \Delta_i \mathbf{A}_i + R_C(\mathbf{S}_i) &= 0, \quad i \geq 0. \end{aligned} \quad (2.120)$$



Napomena:  $\mathbf{A}_i = (\mathbf{A} - \mathbf{G}\mathbf{S})$  te  $\mathbf{G} = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$ . Navedeno se može ukratko zapisati kroz nekoliko koraka

1. Neka su poznati  $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}$ .
2. Izračun  $\mathbf{G} = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$ .
3. Računanje sekvenci  $\{\mathbf{S}_i\}$  dok ne dođe do konvergencije ( $i \geq 0$ ):
  - (a) Postavljanje početne aproksimacije  $\mathbf{S}_0 = \mathbf{S}_0^T$  da  $\mathbf{A} - \mathbf{G}\mathbf{S}_0$  bude stabilna matrica (svojtvene vrijednosti s negativnim realnim dijelom).
  - (b) Izračun  $\mathbf{A}_i = \mathbf{A} - \mathbf{G}\mathbf{S}_i$ .
  - (c) Izračun devijacije  $\Delta_i$  putem Ljapunovljeve jednadžbe

$$\mathbf{A}_i^T \Delta_i + \Delta_i \mathbf{A}_i + R_C(\mathbf{S}_i) = 0.$$

- (d) Izračun  $\mathbf{S}_{i+1} = \mathbf{S}_i + \Delta_i$ .
- (e) Nastavak izračuna (iteracija) prema zadanoj toleranciji  $\epsilon$ ,

$$\frac{\|\mathbf{S}_{i+1} - \mathbf{S}_i\|_F}{\|\mathbf{S}_i\|_F} \leq \epsilon. \quad (2.121)$$

gdje norma  $\|\dots\|_F$  označava Frobeniusovu normu

$$\|M\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}, \quad M \in \mathbb{F}^{m \times n}. \quad (2.122)$$

$M$  je matrica, dok  $\mathbb{F}$  označava skup  $\mathbb{R}$  ili  $\mathbb{C}$ .

Prikazana Newtonova metoda poznata je kao ista u inkrementalnom obliku. Računski je manje zahtjevna nego klasična Newtonova metoda te pruža veću točnost zbog rješavanja Ljapunovljeve jednadžbe za  $\Delta_i$ , a ne direktno za konačno rješenje  $\mathbf{S}_i$ . Slijedi pokazni primjer Newtonove metode.

### Primjer 3.

Razmatra se (2.86) te je poznato

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = 1.$$

Za početak se izračunava  $\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ . Kreće nulta iteracija ( $i = 0$ ) te se za početnu aproksimaciju postavlja  $\mathbf{S}_0 = \begin{bmatrix} 0.4 & 0.1 & 0.1 \\ 0.1 & 0.3 & 0.0 \\ 0.1 & 0.0 & 0.2 \end{bmatrix}$ . Izračunava se sljedeće

$$\Delta_0 = \begin{bmatrix} -0.0248 & -0.0302 & -0.0369 \\ -0.0302 & -0.0426 & 0.0103 \\ -0.0369 & 0.0103 & -0.0224 \end{bmatrix}, \quad \mathbf{S}_1 = \mathbf{S}_0 + \Delta_0 = \begin{bmatrix} 0.3752 & 0.0698 & 0.0631 \\ 0.0698 & 0.2574 & 0.0103 \\ 0.0631 & 0.0103 & 0.1776 \end{bmatrix}.$$

Relativna promjena iznosi:  $\frac{\|\mathbf{S}_1 - \mathbf{S}_0\|}{\|\mathbf{S}_0\|} = 0.1465$ . Započinje prva iteracija ( $i = 1$ ) te slijedi

$$\Delta_1 = \begin{bmatrix} -0.0020 & -0.0015 & -0.0010 \\ -0.0015 & -0.0011 & -0.0008 \\ -0.0010 & -0.0008 & -0.0005 \end{bmatrix}, \quad \mathbf{S}_2 = \mathbf{S}_1 + \Delta_1 = \begin{bmatrix} 0.3732 & 0.0683 & 0.0620 \\ 0.0683 & 0.2563 & 0.0095 \\ 0.0620 & 0.0095 & 0.1770 \end{bmatrix}.$$

Relativna promjena iznosi:  $\frac{\|\mathbf{S}_2 - \mathbf{S}_1\|}{\|\mathbf{S}_1\|} = 0.0086$ . Započinje druga iteracija ( $i = 2$ ) te slijedi

$$\Delta_2 = 10^{-5} \begin{bmatrix} -0.4561 & -0.3864 & -0.2402 \\ -0.3864 & -0.3311 & -0.2034 \\ -0.2402 & -0.2034 & -0.1265 \end{bmatrix}, \quad \mathbf{S}_3 = \mathbf{S}_2 + \Delta_2 = \begin{bmatrix} 0.3732 & 0.0683 & 0.0620 \\ 0.0683 & 0.2563 & 0.0095 \\ 0.0620 & 0.0095 & 0.1770 \end{bmatrix}.$$

Relativna promjena iznosi:  $\frac{\|\mathbf{S}_3 - \mathbf{S}_2\|}{\|\mathbf{S}_2\|} = 2.1709 \times 10^{-5}$ . Iteracija je gotova zbog dovoljno niske relativne promjene rješenja (neka se pretpostavi da je tolerancija bila  $\epsilon = 10^{-4}$ ).

### 2.5.3 Eulerova metoda

Numeričko rješavanje običnih diferencijalnih jednadžbi (*eng. ordinary differential equations - ODE*) klasičan je problem numeričke matematike. Kroz povijest se razvijao niz metoda koje rješavaju ovu zadaću, od kojih je među najjednostavnijima Eulerova metoda, koja je ujedno najjednostavnija Runge Kutta metoda. Problem početne vrijednosti (Cauchyjev problem, to jest, *eng. Initial value problem - IVP*) za obične diferencijalne jednadžbe se zapisuje

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0. \quad (2.123)$$

Gornji izraz se može aproksimirati na sljedeći način

$$\frac{dy(t)}{dt} \approx \frac{y_{n+1} - y_n}{\Delta t}, \quad (2.124)$$

čijim se uvrštavanjem u (2.123) može pisati

$$\frac{y_{n+1} - y_n}{\Delta t} = f(t_n, y_n). \quad (2.125)$$

Konačno se može definirati opća formula Eulerove unaprijedne metode

$$y_{n+1} = y_n + \Delta t f(t_n, y_n). \quad (2.126)$$

Eulerova unaprijedna metoda bazirana je na skraćenom, to jest, odsječenom razvoju Taylorovog reda  $y$  oko  $t = t_n$

$$y(t_n + \Delta t) \equiv y_{n+1} = y(t_n) + \Delta t \left. \frac{dy}{dt} \right|_{t_n} + \mathcal{O}(\Delta t^2) = y_n + \Delta t f(y_n, t_n) + \mathcal{O}(\Delta t^2). \quad (2.127)$$

Eulerova metoda je metoda prvog reda, što opisuje činjenica da je lokalna pogreška odsijecanja<sup>2</sup> po vremenskom koraku proporcionalna kvadratu veličine vremenskog koraka (oznaka  $\mathcal{O}(\Delta t^2)$ ), dok je ukupna pogreška proračuna kroz cijeli vremenski interval proporcionalna veličini vremenskog koraka (oznaka  $\mathcal{O}(\Delta t)$ ). Na sličan se način može definirati Eulerova unazadna metoda, koja se numerički računa unatrag kroz vrijeme te sve napisano za pogreške odsijecanja vrijedi i za ovaj slučaj. Aproksimacija prve derivacije funkcije glasi

$$\frac{dy(t)}{dt} \approx \frac{y_n - y_{n-1}}{\Delta t}, \quad (2.128)$$

te uvrštavanjem iste u (2.123) se dobiva

$$\frac{y_n - y_{n-1}}{\Delta t} = f(t_n, y_n), \quad (2.129)$$

to jest, istovjetno

$$\frac{y_{n+1} - y_n}{\Delta t} = f(t_{n+1}, y_{n+1}). \quad (2.130)$$

iz čega slijedi opća formula za Eulerovu unazadnu metodu

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1}). \quad (2.131)$$

Slijedi pokazni primjer Eulerove metode, osmišljen od strane autora ovog rada.

#### Primjer 4.

Potrebno je pomoću Eulerove unaprijedne metode riješiti diferencijalnu jednadžbu prvog reda  $\frac{dy}{dt} = 3y + t$ ,  $(t_0, y_0) = (0, 0.5)$ ,  $t \in [0, 2]$ . Neka je ukupan vremenski interval podijeljen na deset vremenskih koraka.

Kao prvo, potrebno je definirati vremenski korak  $\Delta t$  na sljedeći način:

---

<sup>2</sup>Pogreška koja nastaje zbog prekida numeričkog izračuna prije nego što je dovršen

- $\Delta t = \frac{2-0}{10} = 0.2$ .

Također je potrebno odrediti iznos  $f(t_0, y_0)$  za početne uvjete:

- $f(t_0, y_0) = f(0, 0.5) = 3(0.5) + 0 = 1.5$ .

Svaki vremenski korak je definiran prema:  $t_{n+1} = t_n + \Delta t$ . Prema (2.126) se može zapisati:

- $t_1 = t_0 + \Delta t = 0.2, y_1 = y_0 + \Delta t f(t_0, y_0) = 0.8,$
- $t_2 = t_1 + \Delta t = 0.4, y_2 = y_1 + \Delta t f(t_1, y_1) = 1.32,$
- $t_3 = t_2 + \Delta t = 0.6, y_3 = y_2 + \Delta t f(t_2, y_2) = 2.192,$
- $t_4 = t_3 + \Delta t = 0.8, y_4 = y_3 + \Delta t f(t_3, y_3) = 3.6272,$
- $t_5 = t_4 + \Delta t = 1.0, y_5 = y_4 + \Delta t f(t_4, y_4) = 5.96352,$

i tako dalje do desetog koraka ...

Izračunato je rješenje dane diferencijalne jednadžbe uz početne uvjete

$$y(t) = 0.6111e^{3t} - \frac{t}{3} - \frac{1}{9},$$

prema kojemu će biti poznate stvarne vrijednosti rješenja te će se iste usporediti s aproksimiranim rješenjima dobivenima Eulerovom unaprijednom metodom (do  $n = 5$ ). Vidljivo prema tablici 2.1, podjelom vremenskog intervala na deset vremenskih koraka se

Tablica 2.1: Tablica vrijednosti - Primjer 4. (Eulerova metoda).

Trenutak ( $t_n$ )	Aproksimacija ( $y_n$ )	Stvarna vrijednost ( $y(t_n)$ )	Greška  (%)
0.0	0.5	0.5	0.000
0.2	0.8	0.9357	14.503
0.4	1.32	1.7845	26.030
0.6	2.192	3.3858	35.260
0.8	3.6272	6.3585	42.956
1.0	5.96352	11.8298	49.590

aproksimacija vrijednosti funkcije, s porastom vremenskih koraka, značajno razlikuje od stvarne vrijednosti zadane funkcije te greška u petom koraku iznosi skoro 50%, što je izrazito velika numerička pogreška. Stoga je prilikom korištenja Eulerove metode bitno odabrati značajnu veličinu željenih vremenskih koraka da se izbjegne istaknut problem.

## 2.5.4 Dormand-Prince metoda

Kroz primjenu Eulerove metode se koristila informacija o nagibu to jest, derivaciji  $y$  u nekom vremenskom trenutku u cilju ekstrapolacije rješenja idućeg vremenskog trenutka te je lokalna pogreška odsijecanja bila  $\mathcal{O}(\Delta t^2)$ , što je označavalo numeričku metodu prvog reda.

Općenito, metode gdje lokalna pogreška odsijecanja iznosi  $\mathcal{O}(\Delta t^{p+1})$  se nazivaju metodama  $p$ -tog reda. Runge-Kutta (skraćeno - RK $p$ ) metoda, nazvana je po uzoru na njemačke matematičare Carla D.T. Rungu i Wilheima Kuttu.

Taylorov razvoj uključuje nalaženje derivacija višeg reda, što može biti zahtjevno te računski *skupo*. Spomenuti problem olakšavaju RK metode koje diferencijalnu jednadžbu dijele na niz diferencijalnih jednadžbi prvog reda, čime se izbjegava računanje derivacija višeg reda te se aproksimiraju vrijednosti istih u točno određenim vremenskim trenucima. Prethodno objašnjena Eulerova metoda spada u RK metodu prvog reda, RK1.

U nastavku je ukratko opisana klasična **RK4** metoda čija metodologija vrijedi za ostale RK metode. Izvod priloženih formula neće se obraditi, ali je isti moguće pogledati u [7].

- $k_1 = f(t_n, y_n)$ ,
- $k_2 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2}k_1)$ ,
- $k_3 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2}k_2)$ ,
- $k_4 = f(t_n + \Delta t, y_n + \Delta t \cdot k_3)$ ,
- $y_{n+1} = y_n + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4)$ ,  $i \geq 0$ ,  $\Delta t \geq 0$ .

RK4 metoda aproksimira funkcijsku krivulju između dvaju točaka (to jest, unutar vremenskog koraka) polinomom četvrtog reda gdje se zahtijevaju četiri aproksimacije nagiba unutar vremenskog koraka. Nalaze se četiri tangente zadane krivulje te se računa njihov prosjek kao aproksimacija krivulje unutar vremenskog intervala (to jest, dvije točke). Oznake  $k_1, k_2, k_3, k_4$  označavaju nagibe tangenti na početku vremenskog koraka ( $k_1$ ), polovici vremenskog koraka ( $k_2, k_3$ ) te kraju vremenskog koraka ( $k_4$ ). Uzimanjem prosjeka istih se dobiva aproksimacija krivulje unutar cijelog vremenskog koraka.

Kako je RK4 metoda četvrtog reda, lokalna pogreška odsijecanja iste je  $\mathcal{O}(\Delta t^5)$ , dok je ukupna pogreška  $\mathcal{O}(\Delta t^4)$ . Kao i prije,  $y_{n+1}$  označava aproksimaciju stvarnog iznosa  $y(t_{n+1})$ . **Dormand-Prince** metoda (skraćeno, DP) korištena je u mnogim ugrađenim funkcijama za rješavanje običnih diferencijalnih jednadžbi te pripada RK metodama. Preciznije, DP metoda je RK5(4) metoda, što znači da pruža par rješenja, aproksimacije funkcijske krivulje između dvaju točaka polinomom četvrtog reda te polinomom petog

reda. Razlika između rješenja (dvaju aproksimacija) se koristi za procjenu pogreške čime se može definirati idući vremenski korak s ciljem bolje aproksimacije krivulje. Ako je razlika (pogreška) velikog iznosa, tada se idući vremenski korak smanjuje u odnosu na prijašnji, dok se za slučaj male vrijednosti razlike (pogreške) idući vremenski korak povećava.

- $k_1 = \Delta t f(t_n, y_n)$ ,
- $k_2 = \Delta t f(t_n + \frac{1}{5}\Delta t, y_n + \frac{1}{5}k_1)$ ,
- $k_3 = \Delta t f(t_n + \frac{3}{10}\Delta t, y_n + \frac{3}{40}k_1 + \frac{9}{40}k_2)$ ,
- $k_4 = \Delta t f(t_n + \frac{4}{5}\Delta t, y_n + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3)$ ,
- $k_5 = \Delta t f(t_n + \frac{8}{9}\Delta t, y_n + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4)$ ,
- $k_6 = \Delta t f(t_n + \Delta t, y_n + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 - \frac{46732}{5247}k_3 + \frac{49}{176}k_4 - \frac{5103}{18656}k_5)$ ,
- $k_7 = \Delta t f(t_n + \Delta t, y_n + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6)$ .

Iznos  $y_{n+1}$  računa se prema

$$y_{n+1} = y_n + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6,$$

čime je dano rješenje RK4 metode. Nadalje slijedi

$$z_{n+1} = y_n + \frac{5179}{57600}k_1 + \frac{7571}{16695}k_3 + \frac{393}{640}k_4 - \frac{92097}{339200}k_5 + \frac{187}{2100}k_6 + \frac{1}{40}k_7,$$

te je gornji izraz rješenje RK5 metode. Kako je prethodno napisano, slijedi definiranje razlike između dvaju aproksimativnih rješenja RK4 i RK5 metode. Razlika neka bude apsolutna

$$|z_{n+1} - y_{n+1}| = \left| \frac{71}{57600}k_1 - \frac{71}{16695}k_3 + \frac{71}{1920}k_4 - \frac{17253}{339200}k_5 + \frac{22}{525}k_6 - \frac{1}{40}k_7 \right|.$$

Lokalna pogreška odsijecanja za RK4 iznosi  $\mathcal{O}(\Delta t^5)$ , dok lokalna pogreška odsijecanja za RK5 iznosi  $\mathcal{O}(\Delta t^6)$ . Optimalan vremenski korak  $\Delta t_{opt}$  dan je izrazom

$$\Delta t_{opt} = \left( \frac{\epsilon \Delta t}{2|z_{n+1} - y_{n+1}|} \right)^{\frac{1}{5}} \Delta t,$$

gdje  $\epsilon$  označava toleranciju, to jest, maksimalno prihvatljivu pogrešku numeričkog rješenja pri svakom vremenskom koraku.

# 3 Izvod dinamike pogreške kroz model četverrotorske letjelice

Pojam bespilotne letjelice (*eng. Unmanned aerial vehicle* - UAV) ili drugim nazivljem, drona, biva razmatran od rane 1783. godine. Naravno, ne u smislu bespilotne letjelice modernog doba, već kao balona pogonjenog vrućim zrakom. Upravo su spomenuti baloni bili prve zračne letjelice koje nisu zahtijevale prisutnost čovjeka potrebnog za upravljanje. Država Austrija prva je upotrijebila iste za ratnu upotrebu.

Još 1849. godine, Austrija je koristila bespilotne balone s eksplozivom u napadu na Veneciju, no napad je bio bezuspješan. Tijekom Drugog svjetskog rata je Njemačka razvila jedan od prvih krstarećih projektila, pod oznakom V-1. Nadalje, razvoj se nastavio i tijekom hladnog rata, posebno za potrebe izviđanja i testiranja projektila, od kojih je među poznatijima model Ryan Firebee, izrađen pod okriljem SAD-a. Upotreba bespilotnih letjelica primarno je bila, a i trenutno je orijentirana, ratovanju, iako se upotrebljavaju u drugim sektorima, no ne u tolikoj mjeri.

## 3.1 Matematičko modeliranje četverrotorske letjelice

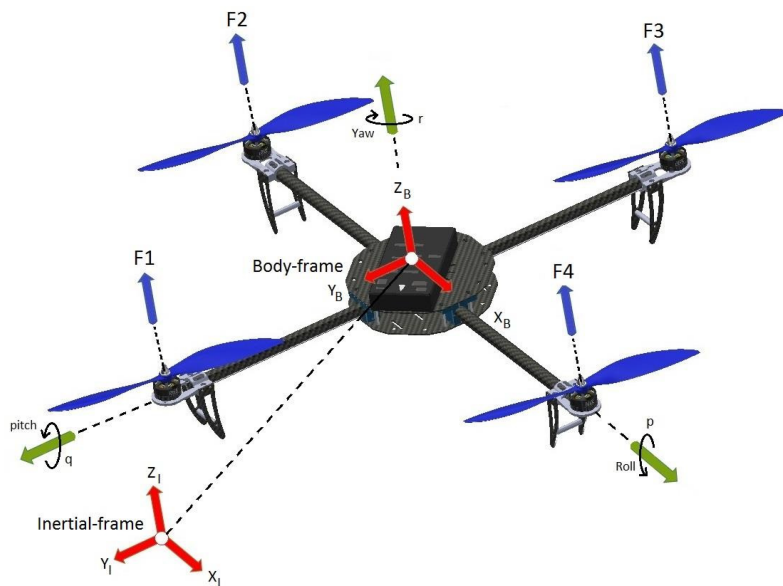
Slijedi izvod modela četverrotorske bespilotne letjelice, pri čemu sva izlaganja istog slijede reference [8], [9]. U nastavku rada će se radi jednostavnosti koristiti izraz dron. Za početak je potrebno uvesti koordinatne sustave vezane za dron te zemljinu površinu. Uz navedeno, objasniti će se oznake korištene kroz matematički izvod modela, koji će obuhvatiti kinematiku i dinamiku drona.

Kinematika, kao grana mehanike, proučava gibanje tijela, bez sagledavanja uzroka gibanja, dok dinamika, također kao grana mehanike, proučava zakone gibanja tijela s obzirom na sile koje ih uzrokuju.

Koordinatni sustav zemljine površine, to jest, inercijski koordinatni sustav, onaj je gdje je prvi Newtonov zakon<sup>1</sup> valjan. Prema slici 3.1, preuzetoj iz [10], neka se indeksom  $B$  označi koordinatni sustav drona (*eng. body frame*) te indeksom  $I$  inercijski koordinatni sustav. Dron je sustav sa šest stupnjeva slobode gibanja; tri su translacijska stupnja te tri rotacijska stupnja slobode gibanja. Pod translacijskim stupnjevima slobode gibanja smatraju se kretanje duž osi  $x$  (pomak unaprijed - unatrag), osi  $y$  (pomak ulijevo - udesno)

---

<sup>1</sup>Svako tijelo ostaje u stanju mirovanja ili jednolikoga gibanja po pravcu dok ga neka vanjska sila ne prisili da to stanje promijeni to jest, ako je vanjska sila jednaka nuli te ako je masa tijela stalna, tada se brzina gibanja tijela ne mijenja



Slika 3.1: Prikaz bespilotne letjelice i pripadnih značajki.

te osi  $z$  (pomak gore - dolje). Rotacijskim stupnjevima slobode gibanja se smatraju valjanje (*eng. roll*), poniranje (*eng. pitch*) te skretanje (*eng. yaw*). Navedeni stupnjevi slobode gibanja opisuju poziciju drona u prostoru, točnije, inercijskom desnokretnom koordinatnom sustavu.

Neka se dron smatra potpuno krutim tijelom, čime se gibanje samog drona promatra u koordinatnom sustavu drona, a opisuje u inercijskom koordinatnom sustavu pri čemu se kaže da je koordinatni sustav drona razmatran u odnosu na inercijski koordinatni sustav.

*Roll*, *Pitch* te *Yaw* nazivaju se Eulerovim kutevima, nazvanima prema matematičaru Leonhardu Euleru te se općenito koriste za definiranje orijentacije krutog tijela u trodimenzionalnom prostoru. Kroz rad će se koristiti **ZYX** kombinacija Eulerovih kuteva. Navedena notacija jednostavno označava redosljed rotacija oko pripadnih osi te je također vidljiva na slici 3.1

- Rotacija koordinatnog sustava drona  $(x_B, y_B, z_B)$  oko inercijske  $z_I$ -osi pod kutem  $\psi \in [-\pi, \pi]$  (*Yaw*),
- Rotacija koordinatnog sustava drona  $(x_B, y_B, z_B)$  oko inercijske  $y_I$ -osi pod kutem  $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  (*Pitch*),
- Rotacija koordinatnog sustava drona  $(x_B, y_B, z_B)$  oko inercijske  $x_I$ -osi pod kutem  $\phi \in [-\pi, \pi]$  (*Roll*).



Navedeno se matematički može opisati preko matrica rotacija

$$\mathbf{R}_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}, \quad (3.2)$$

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix}. \quad (3.3)$$

Množenjem istih se prema odabranom redosljedu rotacija definira ukupna matrica rotacija kojom se definira kako je, u ovom slučaju dron, orijentiran u odnosu na inercijski koordinatni sustav

$$\begin{aligned} \mathbf{R}_{zyx}(\psi, \theta, \phi) &= \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi) \\ &= \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}. \end{aligned} \quad (3.4)$$

Radi jednostavnijeg zapisa uvedena su pojednostavljenja

$$\begin{aligned} s(\theta) &= \sin(\theta), \\ c(\theta) &= \cos(\theta), \\ s(\phi) &= \sin(\phi), \\ c(\phi) &= \cos(\phi), \\ s(\psi) &= \sin(\psi), \\ c(\psi) &= \cos(\psi). \end{aligned}$$

Napomena: prilikom množenja matrica rotacija ne vrijedi svojstvo komutativnosti<sup>2</sup>, što znači  $\mathbf{R}_{zyx}(\psi, \theta, \phi) \neq \mathbf{R}_{xyz}(\phi, \theta, \psi)$ .

### 3.1.1 Kinematički model

Kako su definirane potrebne definicije za izvod kinematičkog modela, može se započeti s izvodom istog. Za početak, umjesto indeksa  $I$  u označavanju inercijskih osi, neka se piše  $x = x_I$ ,  $y = y_I$  te  $z = z_I$ . Definira se vektor opisa translacijskog i rotacijskog

<sup>2</sup>Svojstvo neke binarne matematičke operacije da se rezultat ne mijenja ako operandi zamijene mjesta

položaja (orijentacije) u inercijskom koordinatnom sustavu,  $[x \ y \ z \ \phi \ \theta \ \psi]^T$  te vektor translacijskih i kutnih brzina u koordinatnom sustavu drona,  $[u \ v \ w \ p \ q \ r]^T$ . Kako je napisano,  $[u \ v \ w]^T$  označava vektor translacijskih brzina u osima  $x_B$ ,  $y_B$  te  $z_B$ , slijedno. Također,  $[p \ q \ r]^T$  označava vektor kutnih brzina u odnosu na osi  $x_B$ ,  $y_B$  te  $z_B$ , slijedno. Kinematika se promatra kao translacijska i rotacijska. Translacijska kinematika definira se prema

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{R}_{zyx}(\psi, \theta, \phi) \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (3.5)$$

te se jednostavnije zapisuje kao

$$\mathbf{v} = \mathbf{R}_{zyx}(\psi, \theta, \phi) \cdot \mathbf{v}_B, \quad (3.6)$$

gdje je pogodno primijetiti da matrica rotacija vrijedi i za relacije derivacija translacijskih položaja. Kako je definirano, ako je vektor prikazan u koordinatnom sustavu krutog tijela te se želi prikazati u inercijskom koordinatnom sustavu, potrebno ga je pomnožiti s matricom rotacije. Matrice rotacije su ortogonalne matrice, što znači da je operacija inverza jednaka transponiranju. Stoga, u slučaju da se poznati vektor u inercijskom koordinatnom sustavu želi prikazati u koordinatnom sustavu krutog tijela, potrebno je isti pomnožiti s transponiranom matricom rotacije. Sada kada je razjašnjeno navedeno svojstvo, slijedi izvod rotacijske kinematike.

Poznate su kutne brzine u inercijskom koordinatnom sustavu (vremenske derivacije Eulerovih kutova)  $[\dot{\psi}, \dot{\theta}, \dot{\phi}]^T$  te je potrebno navedene kutne brzine izraziti u koordinatnom sustavu drona kao  $p$ ,  $q$ ,  $r$ . Stoga, kutne brzine u koordinatnom sustavu drona bit će jedanke vremenskim derivacijama Eulerovih kutova pomnoženih pripadajućom matricom rotacije u cilju transformacije rotacije iz inercijskog u koordinatni sustav drona. Kao i prije, konvencija rotacija je **ZYX**, no potrebno je primijetiti da se kroz ovaj izvod vektor kutne brzine preslikava iz inercijskog koordinatnog sustava u koordinatni sustav drona te je stoga slijed rotacija XYZ

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_z^T(\psi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_z^T(\psi) \mathbf{R}_y^T(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}, \quad (3.7)$$

čime se nakon raspisivanja formulira jednadžba

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & s(\phi)c(\theta) \\ 0 & -s(\phi) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (3.8)$$

Težnja je da se vektor kutnih brzina u koordinatnom sustavu drona izrazi u inercijskom koordinatnom sustavu te nakon inverza matrice transformacije slijedi

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (3.9)$$

gdje je  $t(\theta) = \tan(\theta)$ . Za detaljniji izvod i objašnjenje izvednog izraza, čitatelju se preporučuje [11]. Gornji se izraz kraće može zapisati kao

$$\boldsymbol{\omega} = \mathbf{R}_T \cdot \boldsymbol{\omega}_B. \quad (3.10)$$

Raspisivanjem jednadžbi translacijske kinematike (3.5) te rotacijske kinematike (3.9) konačno se prikazuje kinematički model drona

$$\begin{cases} \dot{x} = u [c(\psi)c(\theta)] + v [c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)] + w [c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi)], \\ \dot{y} = u [s(\psi)c(\theta)] + v [s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi)] + w [s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)], \\ \dot{z} = -u [s(\theta)] + v [c(\theta)s(\phi)] + w [c(\theta)c(\phi)], \\ \dot{\phi} = p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)], \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)], \\ \dot{\psi} = q \left( \frac{s(\phi)}{c(\theta)} \right) + r \left( \frac{c(\phi)}{c(\theta)} \right). \end{cases} \quad (3.11)$$

### 3.1.2 Dinamički model

Slijedi izvod dinamičkog modela drona. Prema drugom Newtonovom zakonu<sup>3</sup>, može se matrično zapisati iduća jednadžba

$$\mathbf{f}_B = m(\boldsymbol{\omega}_B \wedge \mathbf{v}_B + \dot{\mathbf{v}}_B), \quad (3.12)$$

gdje  $\mathbf{f}_B = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$  označava ukupnu silu koja djeluje na dron,  $m$  označava ukupnu masu drona te oznaka  $\wedge$  vektorski produkt. Potrebno je primijetiti indeks  $B$  koji kao i prije, označava koordinatni sustav drona te napomenuti da jednadžba vrijedi pod pretpostavkom da je ishodište koordinatnog sustava drona smješteno točno u njegovom težištu. Raspisom jednadžbe (3.12) može se pisati

$$\begin{cases} f_x = m(qw - rv + \dot{u}), \\ f_y = m(-pw + ru + \dot{v}), \\ f_z = m(pv - qu + \dot{w}). \end{cases} \quad (3.13)$$

<sup>3</sup>Promjena količine gibanja razmjerna je djelovanju sile, a odvija se u smjeru iste

Neka se gornji izraz zapiše kao

$$\begin{cases} \dot{u} = \frac{f_x}{m} + rv - qw, \\ \dot{v} = \frac{f_y}{m} + pw - ru, \\ \dot{w} = \frac{f_z}{m} + qu - pv. \end{cases} \quad (3.14)$$

Komponente  $f_x$ ,  $f_y$  te  $f_z$  se prema (3.12) mogu dodatno definirati kao

$$\mathbf{f}_B = -mg\mathbf{R}^T(\psi, \theta, \phi) \cdot \hat{\mathbf{e}}_z + f_t \hat{\mathbf{e}}_{z_B}, \quad (3.15)$$

prilikom čega oznaka  $\hat{\mathbf{e}}$  označava jedinični vektor u pripadnoj osi koordinatnog sustava, dok  $g$  označava gravitacijsku akceleraciju. Ukupna sila potiska svih propelera označena je oznakom  $f_t$  ( $= F_1 + F_2 + F_3 + F_4$ ), što je vidljivo prema slici 3.1. Uvrštavanjem (3.15) u (3.14) konačno se zapisuje

$$\begin{cases} \dot{u} = g[s(\theta)] + rv - qw, \\ \dot{v} = -g[c(\theta)s(\phi)] + pw - ru, \\ \dot{w} = -g[c(\theta)c(\phi)] + f_t + qu - pv. \end{cases} \quad (3.16)$$

Nadalje, prema Eulerovoj jednadžbi se definira ukupan moment sile drona

$$\boldsymbol{\tau}_B = \boldsymbol{\omega}_B \wedge (\mathbf{I} \cdot \boldsymbol{\omega}_B) + \mathbf{I} \cdot \dot{\boldsymbol{\omega}}_B, \quad (3.17)$$

gdje  $\boldsymbol{\tau}_B = [\tau_x \ \tau_y \ \tau_z]^T \in \mathbb{R}^3$ . Pod pretpostavkom da je dron simetričan oko svojih osi, matrica inercija  $\mathbf{I} \in \mathbb{R}^3$  definirana je prema

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (3.18)$$

Raspisom jednadžbe (3.17) dobiva se

$$\begin{cases} \tau_x = qrI_{zz} - rqI_{yy} + \dot{p}I_{xx}, \\ \tau_y = -prI_{zz} + rpI_{xx} + \dot{q}I_{yy}, \\ \tau_z = pqI_{yy} - qpI_{xx} + \dot{r}I_{zz}. \end{cases} \quad (3.19)$$

Neka se gornji izraz zapiše kao

$$\begin{cases} \dot{p} = qr \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{\tau_x}{I_{xx}}, \\ \dot{q} = pr \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{\tau_y}{I_{yy}}, \\ \dot{r} = pq \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{\tau_z}{I_{zz}}. \end{cases} \quad (3.20)$$

Ako se izrazi (3.16) i (3.20) zapišu zajedno, dobiva se dinamički model drona u vlastitom koordinatnom sustavu  $(x_B, y_B, z_B)$

$$\begin{cases} \dot{u} = g[s(\theta)] + rv - qw, \\ \dot{v} = -g[c(\theta)s(\phi)] + pw - ru, \\ \dot{w} = -g[c(\theta)c(\phi)] + f_t + qu - pv, \\ \dot{p} = qr \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{\tau_x}{I_{xx}}, \\ \dot{q} = pr \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{\tau_y}{I_{yy}}, \\ \dot{r} = pq \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{\tau_z}{I_{zz}}. \end{cases} \quad (3.21)$$

Gibanje drona nije idealno; postoje mnogi poremećaji koji otežavaju željeno gibanje drona, poput udara vjetra, žiroskopskih momenata te mnogih aerodinamičnih i aeroelastičnih pojava, na primjer "učinka tla", prilikom čega se smanjuje otpor zraka i povećava uzgon letjelice. Kroz izraze iz (3.21) su zanemareni navedeni poremećaji te će kroz nastavak rada također ostati zanemareni.

Matematički opis drona može biti dan prema (3.11) te (3.21)

$$\begin{cases} \dot{x} = u [c(\psi)c(\theta)] + v [c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)] + w [c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi)], \\ \dot{y} = u [s(\psi)c(\theta)] + v [s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi)] + w [s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)], \\ \dot{z} = -u [s(\theta)] + v [c(\theta)s(\phi)] + w [c(\theta)c(\phi)], \\ \dot{\phi} = p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)], \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)], \\ \dot{\psi} = q \left( \frac{s(\phi)}{c(\theta)} \right) + r \left( \frac{c(\phi)}{c(\theta)} \right), \\ \dot{u} = g[s(\theta)] + rv - qw, \\ \dot{v} = -g[c(\theta)s(\phi)] + pw - ru, \\ \dot{w} = -g[c(\theta)c(\phi)] + f_t + qu - pv, \\ \dot{p} = qr \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{\tau_x}{I_{xx}}, \\ \dot{q} = pr \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{\tau_y}{I_{yy}}, \\ \dot{r} = pq \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{\tau_z}{I_{zz}}, \end{cases} \quad (3.22)$$

no takav model nije pogodan za upravljanje istim. Stoga se prema drugom Newtonovom zakonu iz (3.12) i (3.15) zapisuje

$$\mathbf{R}_{zyx}(\psi, \theta, \phi) \cdot \mathbf{f}_B = m\dot{\mathbf{v}} = -mg\hat{\mathbf{e}}_z + f_t \mathbf{R}_{zyx}(\psi, \theta, \phi) \cdot \hat{\mathbf{e}}_{z_B}. \quad (3.23)$$

Pažljivo prateći gornju jednadžbu te pretpostavljajući da je pod pretpostavkom malih kutova, vremenska derivacija Eulerovih kutova jednaka vektoru kutnih brzina u koordinatnom sustavu drona ( $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \approx [p \ q \ r]^T$ ), može se pisati konačan, upravljački pogodan, **matematički model** drona

$$\begin{cases} \ddot{x} = \frac{f_t}{m}[c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi)], \\ \ddot{y} = \frac{f_t}{m}[s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)], \\ \ddot{z} = -g + \frac{f_t}{m}[c(\theta)c(\phi)], \\ \ddot{\phi} = \dot{\theta}\dot{\psi} \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{\tau_x}{I_{xx}}, \\ \ddot{\theta} = \dot{\phi}\dot{\psi} \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{\tau_y}{I_{yy}}, \\ \ddot{\psi} = \dot{\phi}\dot{\theta} \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{\tau_z}{I_{zz}}. \end{cases} \quad (3.24)$$

## 3.2 Dinamika pogreške

Za prikaz matematičkog modela drona u vremenski promjenjivom prostoru stanja u koordinatama dinamike pogreške, potrebno je za početak definirati varijable stanja i sam zapis prostora stanja.

Dron je podaktuiran sustav, što objašnjava činjenica da sadrži šest stupnjeva slobode gibanja, no samo četiri izlaza (skretanje, poniranje, valjanje te potisak) kojima se upravlja električnim motorima preko četiri ulazne upravljačke varijable,  $[u_1 \ u_2 \ u_3 \ u_4]^T = [f_t \ \tau_x \ \tau_y \ \tau_z]^T$ . Sila potiska  $f_t$  ubrzava dron oko osi  $z_B$ , dok okretni momenti sile djeluju oko pripadnih osi  $x_B$ ,  $y_B$  te  $z_B$ . Kako dron može postići bilo koju točku  $x, y, z$  prostora promjenom kuta gibanja te potiska, kao izlaz spomenute četiri ulazne upravljačke varijable, prema [12] je odabran vektor  $[x \ y \ z \ \psi]^T$ . Prikazani vektor će biti referentna trajektorija drona te se također može zapisati kao  $[x_{ref}(t) \ y_{ref}(t) \ z_{ref}(t) \ \psi_{ref}(t)]^T$ .

Dvostrukim vremenskim derivacijama željene (referentne) trajektorije se prema (3.24) mogu dobiti ostale vrijednosti preostalih stanja i ulaza u smislu željenih stanja. Prema napisanome, tada se mogu izračunati (prema [13])

$$\begin{aligned} \phi_{ref} &= \arctan \left( \frac{c(\theta_{ref})(\ddot{x}s(\psi_{ref}) - \ddot{y}c(\psi_{ref}))}{g + \ddot{z}} \right), \\ \theta_{ref} &= \arctan \left( \frac{\ddot{x}c(\psi_{ref}) + \ddot{y}s(\psi_{ref})}{g + \ddot{z}} \right), \\ u_{1ref} &= \frac{m(\ddot{z} + g)}{c(\theta)c(\phi)}, \end{aligned} \quad (3.25)$$

te nakon toga, kada su poznate sve referentne varijable stanja te referentna ulazna varijabla  $u_{1ref}$ , mogu se izračunati preostali ulazi  $\tau_x$ ,  $\tau_y$ ,  $\tau_z$  prema (3.24). Tada se nelinearni matematički model drona može linearizirati oko referentne trajektorije; računajući Jacobijeve matrice  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$  (prema str.19) za vektor referentnih varijabli stanja te vektor referentnih ulaznih varijabli, slijedno. Kako je napisano, prvo je potrebno prema matematičkom modelu drona (3.24) definirati varijable stanja

$$\mathbf{x} = [x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z} \quad \phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi}]^T \in \mathbb{R}^{12}, \quad (3.26)$$

te se uvodi supstitucija

$$\begin{aligned} x_1 &= x \xrightarrow{\frac{d}{dt}} \dot{x}_1 = x_2, \\ x_2 &= \dot{x} \xrightarrow{\frac{d}{dt}} \dot{x}_2 = \ddot{x} = \frac{u_1}{m} [c(x_{11})s(x_9)c(x_7) + s(x_{11})s(x_7)], \\ x_3 &= y \xrightarrow{\frac{d}{dt}} \dot{x}_3 = x_4, \\ x_4 &= \dot{y} \xrightarrow{\frac{d}{dt}} \dot{x}_4 = \ddot{y} = \frac{u_1}{m} [s(x_{11})s(x_9)c(x_7) - c(x_{11})s(x_7)], \\ x_5 &= z \xrightarrow{\frac{d}{dt}} \dot{x}_5 = x_6, \\ x_6 &= \dot{z} \xrightarrow{\frac{d}{dt}} \dot{x}_6 = \ddot{z} = -g + \frac{u_1}{m} [c(x_9)c(x_7)], \\ x_7 &= \phi \xrightarrow{\frac{d}{dt}} \dot{x}_7 = x_8, \\ x_8 &= \dot{\phi} \xrightarrow{\frac{d}{dt}} \dot{x}_8 = \ddot{\phi} = x_{10}x_{12} \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right) + \frac{u_2}{I_{xx}}, \\ x_9 &= \theta \xrightarrow{\frac{d}{dt}} \dot{x}_9 = x_{10}, \\ x_{10} &= \dot{\theta} \xrightarrow{\frac{d}{dt}} \dot{x}_{10} = \ddot{\theta} = x_8x_{12} \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \frac{u_3}{I_{yy}}, \\ x_{11} &= \psi \xrightarrow{\frac{d}{dt}} \dot{x}_{11} = x_{12}, \\ x_{12} &= \dot{\psi} \xrightarrow{\frac{d}{dt}} \dot{x}_{12} = \ddot{\psi} = x_8x_{10} \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{u_4}{I_{zz}}, \end{aligned} \quad (3.27)$$

kojom se matematički model drona sveo na skup običnih diferencijalnih jednadžbi prvog reda. Za dobivanje zapisa (2.101) neka se izvedu matrice  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$ , kako je navedeno,

prema str.19.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{2,7} & 0 & A_{2,9} & 0 & A_{2,11} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{4,7} & 0 & A_{4,9} & 0 & A_{4,11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{6,7} & 0 & A_{6,9} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{8,10} & 0 & A_{8,12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{10,8} & 0 & 0 & 0 & 0 & A_{10,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{12,8} & 0 & A_{12,10} & 0 & 0 & 0 \end{bmatrix}, \quad (3.28)$$

gdje su

- $A_{2,7} = \frac{u_1}{m} [c(x_7)s(x_{11}) - c(x_{11})s(x_7)s(x_9)],$
- $A_{2,9} = \frac{u_1}{m} [c(x_7)c(x_9)c(x_{11})],$
- $A_{2,11} = \frac{u_1}{m} [c(x_{11})s(x_7) - c(x_7)s(x_9)s(x_{11})],$
- $A_{4,7} = -\frac{u_1}{m} [c(x_7)c(x_{11}) + s(x_7)s(x_9)s(x_{11})],$
- $A_{4,9} = \frac{u_1}{m} [c(x_7)c(x_9)s(x_{11})],$
- $A_{4,11} = \frac{u_1}{m} [s(x_7)s(x_{11}) + c(x_7)c(x_{11})s(x_9)],$
- $A_{6,7} = -\frac{u_1}{m} [c(x_9)s(x_7)],$
- $A_{6,9} = -\frac{u_1}{m} [c(x_7)s(x_9)],$
- $A_{8,10} = x_{12} \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right),$
- $A_{8,12} = x_{10} \left( \frac{I_{yy} - I_{zz}}{I_{xx}} \right),$
- $A_{10,8} = x_{12} \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right),$
- $A_{10,12} = x_8 \left( \frac{I_{zz} - I_{xx}}{I_{yy}} \right),$
- $A_{12,8} = x_{10} \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right),$
- $A_{12,10} = x_8 \left( \frac{I_{xx} - I_{yy}}{I_{zz}} \right),$



$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ B_{2,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ B_{4,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ B_{6,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{8,2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & B_{10,3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{12,4} \end{bmatrix}, \quad (3.29)$$

gdje su

- $B_{2,1} = \frac{1}{m}[s(x_7)s(x_{11}) + c(x_7)c(x_{11})s(x_9)],$
- $B_{4,1} = -\frac{1}{m}[c(x_{11})s(x_7) - c(x_7)s(x_9)s(x_{11})],$
- $B_{6,1} = \frac{1}{m}[c(x_7)c(x_9)],$
- $B_{8,2} = 1/I_{xx},$
- $B_{10,3} = 1/I_{yy},$
- $B_{12,4} = 1/I_{zz}.$

Sada se opće forme matrica  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$ , mogu evaluirati za prethodno definirane, kao i naknadno izračunate, vektore referentnih varijabli stanja te vektor ulaznih varijabli. Također, nakon definiranja opće forme spomenutih matrica, sustav se može konačno zapisati u obliku dinamike pogreške u vremenski varijantnom prostoru stanja

$$\dot{\bar{\mathbf{x}}}(t) = \mathbf{A}(t)\bar{\mathbf{x}}(t) + \mathbf{B}(t)\bar{\mathbf{u}}(t), \quad (3.30)$$

te započeti upravljanje.

## 4 Simulacijska razmatranja

Potrebno je prikazati slijeđenje zadane referentne trajektorije nelineariziranog modela drona (3.27) te lineariziranog modela zapisanog kroz koordinate pogreške (3.30). Za početak se, vidljivo prema tablici 4.1, definiraju parametri drona, preuzeti iz [14].

Ovisno o referentnim trajektorijama, matrice  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$ , mogu biti vremenski promjenjive i obrnuto. Ako su akceleracije (druge derivacije referentne trajektorije položaja  $x$ ,  $y$ ,  $z$ ) vremenski promjenjive, tada će i same spomenute matrice biti vremenski promjenjive te shodno matrica pojačanja  $\mathbf{K}(t)$  te optimalno upravljanje  $\mathbf{u}^*(t)$ .

Kako je navedeno na stranici 19., rješenje  $\mathbf{S}(t)$  pripadajuće Riccatijeve jednadžbe, a samim time i matricu pojačanja  $\mathbf{K}(t)$  te optimalno upravljanje, moguće je dobiti pristupom rješavanja diferencijalne Riccatijeve jednadžbe unatrag kroz vrijeme zadavanjem početnog uvjeta, to jest, matrice  $\mathbf{S}_{are}$ , dobivene rješavanjem algebarske Riccatijeve jednažbe. Prema tome se matrica pojačanja definira u stvarnom vremenu tijekom gibanja drona.

Drugi pristup je rješavanje algebarske Riccatijeve jednadžbe kroz svaki vremenski korak vremenskog intervala, čime se slijedno dobiva matrica pojačanja u svakom vremenskom koraku vremenskog intervala. Tada se navedena matrica interpolira regulatorom u stvarnom vremenu za potrebe upravljanja drona. Navedeni pristup je numerički manje zahtjevan od prethodno navedenog pristupa te će se iz istog razloga koristiti u nastavku za potrebe simulacije u računalnom alatu *Matlab*. U *Matlabu* je algebarsku Riccatijevu jednadžbu moguće riješiti ugrađenom naredbom `lqr(A,B,Q,R)`.

Tablica 4.1: Parametri bespilotne letjelice.

Parametar	Oznaka	Iznos	Mjerna jedinica
Masa	$m$	1.5	kg
Gravitacijsko ubrzanje	$g$	9.807	m/s <sup>2</sup>
Duljina pojedinog kraka	$l$	0.225	m
Momenti inercije	$I_{xx}$	0.022	kg·m <sup>2</sup>
	$I_{yy}$	0.022	kg·m <sup>2</sup>
	$I_{zz}$	0.0018	kg·m <sup>2</sup>

Za dobivanje vremenski promjenjive matrice pojačanja, to jest, u ovom slučaju iznose iste u svakom vremenskom trenutku, jedno od mogućih rješenja je definiranje pripadne funkcije u *Matlabu* u kojoj će se za definirani vremenski interval i vremenski korak iteracijom kroz svaki vremenski korak popunjavati u početku inicijalizirana

nul-matrica izvršavanjem naredbe `lqr`. Navedeno se može vidjeti u Prilogu A u funkciji pod nazivom `dobivanje_pojacanja` kroz linije 11-23, kao i svi kodovi korišteni za dobivanje i grafičko prikazivanje simulacijskih rezultata.

## 4.1 Prva trajektorija

Neka je zadana trajektorija

$$\begin{cases} x_{ref} = 5t, \\ y_{ref} = 5\cos(t), \\ z_{ref} = t, \\ \psi_{ref} = 0, \end{cases} \quad (4.1)$$

kojom se definira prostorna spirala. Preko zadanih trajektorija moguće je izračunati ostale referentne varijable stanja za potpun opis referentne trajektorije, kako je opisano u sekciji 3.2. Navedeni postupak se odrađivao putem *Matlabovog Symbolic* paketa.

Za dobivanje matrice pojačanja linearnog kvadratičnog regulatora, odabrane su sljedeće težinske matrice

$$\mathbf{Q} = 10 \times I_{12}, \quad \mathbf{R} = 0.1 \times I_4, \quad (4.2)$$

gdje je  $I_i$  jedinična kvadratna matrica dimenzija  $i \times i$ .

Nakon potpunog definiranja referentne trajektorije, to jest, svih dvanaest referentnih varijabli stanja, potpuno se definiraju vremenski promjenjive matrice  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$  uvrštavanjem referentnih varijabli stanja u (3.28) te (3.29). Uz ugrađenu *Matlabovu* funkciju za rješavanje običnih diferencijalnih jednadžbi `ode45`, rješava se sustav (3.30), to jest, ako se isti raspiše za direktno računanje varijabli stanja, može se pisati

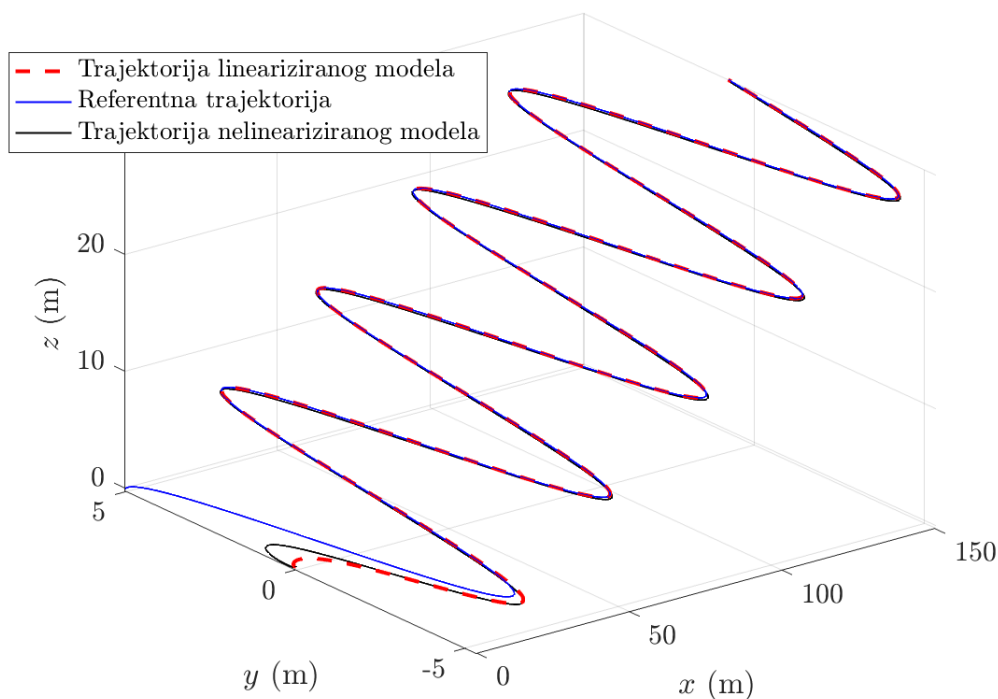
$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_{ref} + (\mathbf{A} - \mathbf{BK})\mathbf{x} - (\mathbf{A} - \mathbf{BK})\mathbf{x}_{ref}, \quad (4.3)$$

prilikom čega su naravno svi članovi jednadžbe vremenski varijantni.

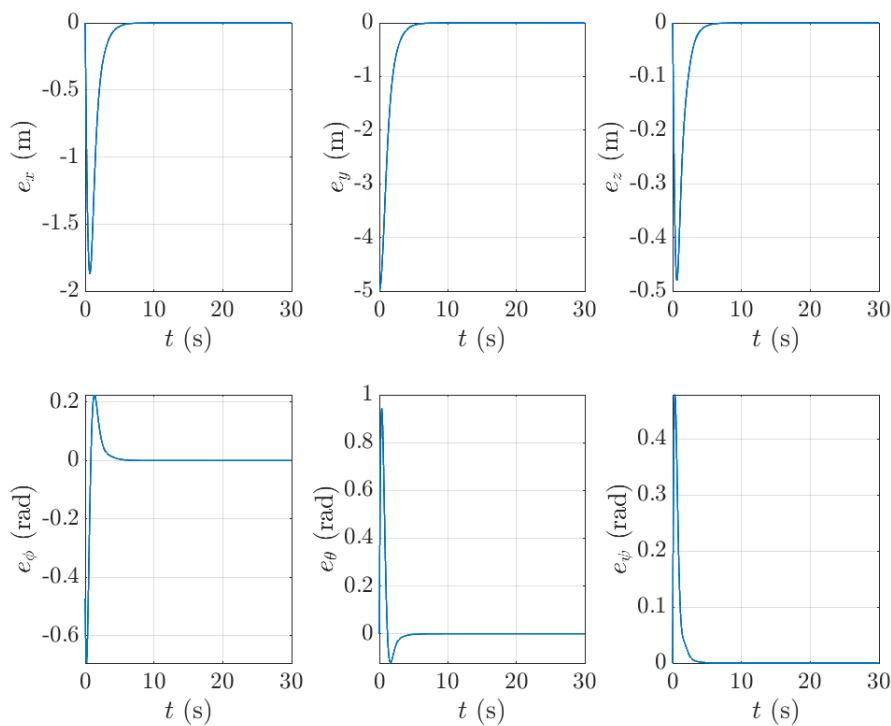
Optimalno upravljanje dobiveno prema izrazu (2.105) se primjenjuje kroz zatvoreni sustav upravljanja za slučajeve nelinearnog i lineariziranog modela drona. U nastavku su prikazani simulacijski rezultati za navedenu prvu trajektoriju, kroz koje se mogu vidjeti prikazi slijeđenja referentne trajektorije lineariziranog i nelineariziranog modela, iznosi upravljačkih varijabli lineariziranog modela te pogreške slijeđenja referentnih varijabli položaja i orijentacije istog. Početni uvjeti svih dvanaest varijabli stanja jednaki su nuli,

$$\mathbf{x}(0) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T, \quad (4.4)$$

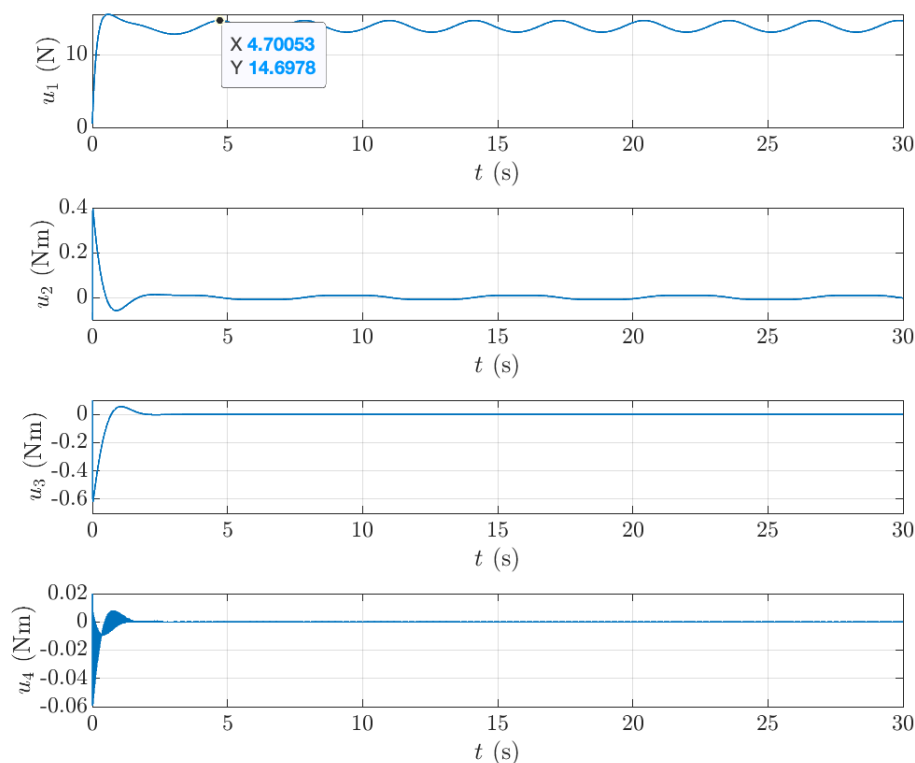
te je trajanje simulacije 30 sekundi.



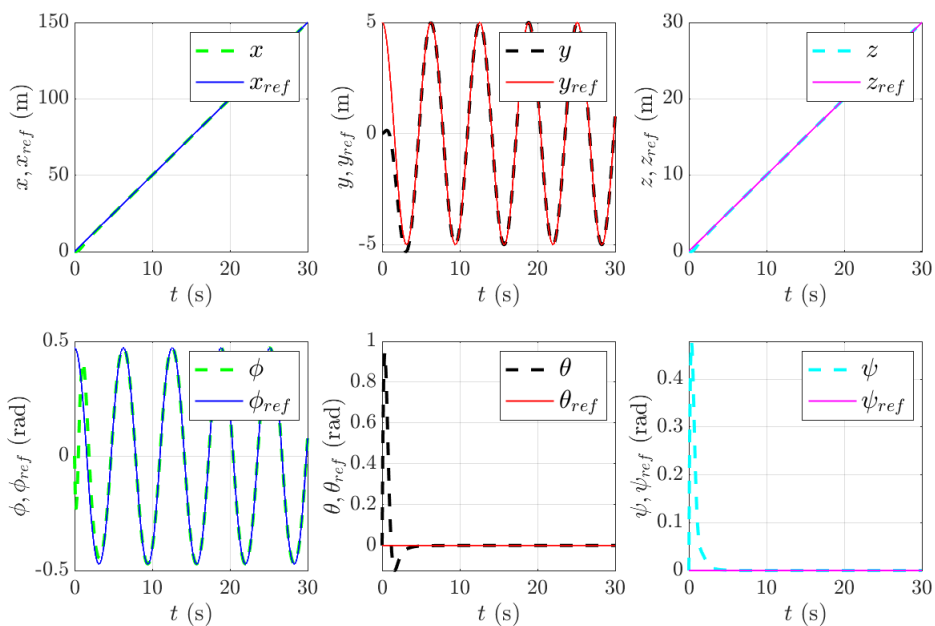
Slika 4.1: Prikaz slijeđenja prve trajektorije.



Slika 4.2: Pogreške slijeđenja referentnih položaja i orijentacija za linearizirani model (prva trajektorija).

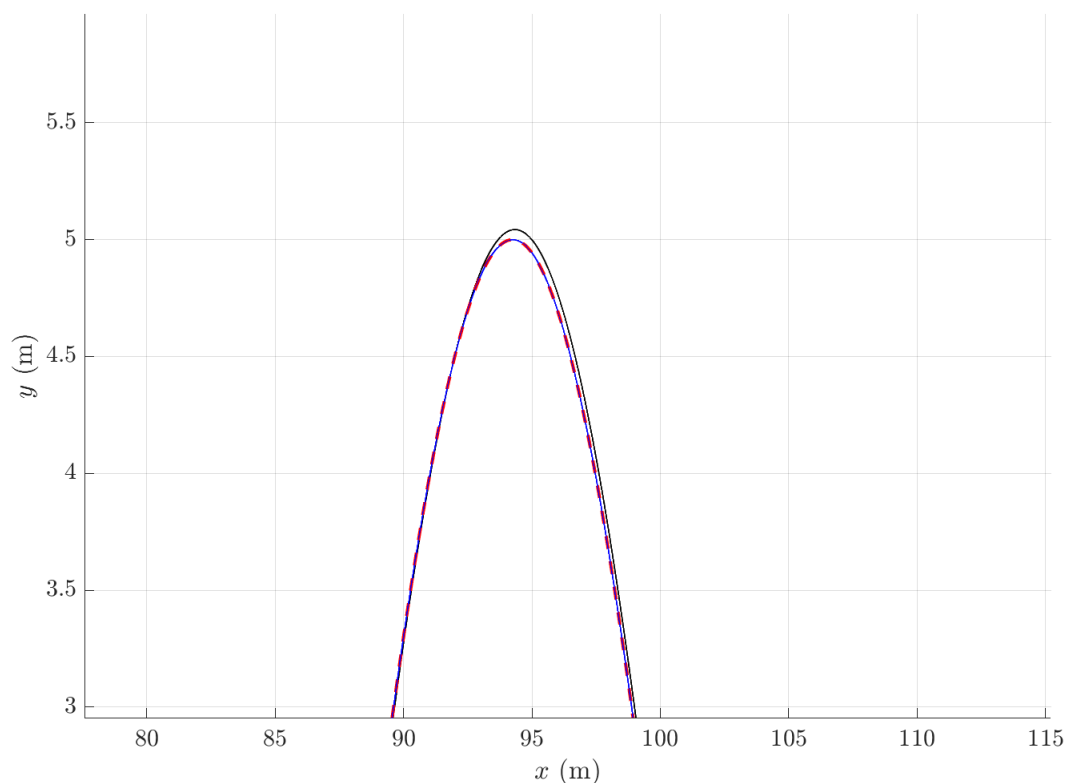


Slika 4.3: Iznosi upravljačkih varijabli lineariziranog modela za slijeđenje prve trajektorije.



Slika 4.4: Usporedba položaja i orijentacija lineariziranog modela s referentnima (prva trajektorija).

Slikom 4.1 prikazano je slijeđenje referentne trajektorije za linearizirani i nelinearizirani model te se prema prikazima na slikama 4.2 te 4.4 može vidjeti da linearizirani model zadovoljavajuće slijedi referentnu trajektoriju prilikom čega se sila potiska (upravljачka varijabla  $u_1$ ) može provjeriti kao optimalna zbog variranja vrijednosti oko umnoška mase i gravitacijskog ubrzanja drona (slika 4.3). Primjenom upravljanja na nelinearizirani model, nelinearizirani model ipak malo lošije prati zadanu referentnu trajektoriju. što je vidljivo prema slici 4.5, no kako su zanemareni vanjski poremećaji poput udara vjetra i slično, slijeđenje se može smatrati zadovoljavajućim.



Slika 4.5: Uvećan prikaz dijela prve trajektorije.

## 4.2 Druga trajektorija

Kao druga referentna trajektorija zadaje se prostorna spirala

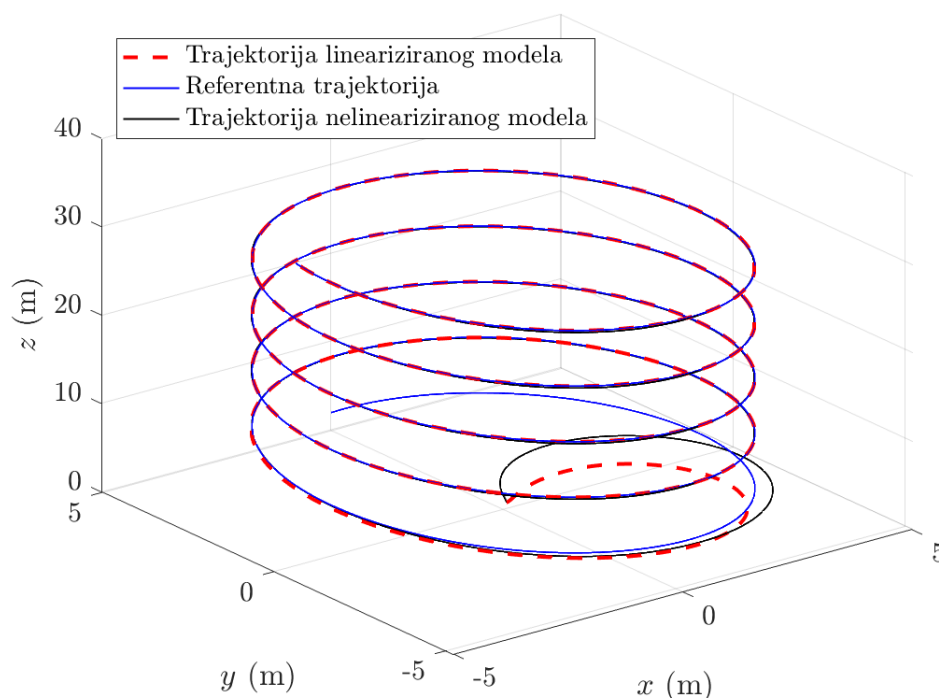
$$\begin{cases} x_{ref} = 4\sin(t), \\ y_{ref} = 5\cos(t), \\ z_{ref} = 2 + t, \\ \psi_{ref} = 3\sin(t), \end{cases} \quad (4.5)$$

Težinske matrice  $\mathbf{Q}$  i  $\mathbf{R}$  su postavljene na iste vrijednosti kao za prvu trajektoriju. Početni uvjeti varijabli stanja drona su zadani kao

$$\mathbf{x}(0) = [0.25, 0, 0.25, 0, 0, 0, 0, 0, 0, 0, 0]^T, \quad (4.6)$$

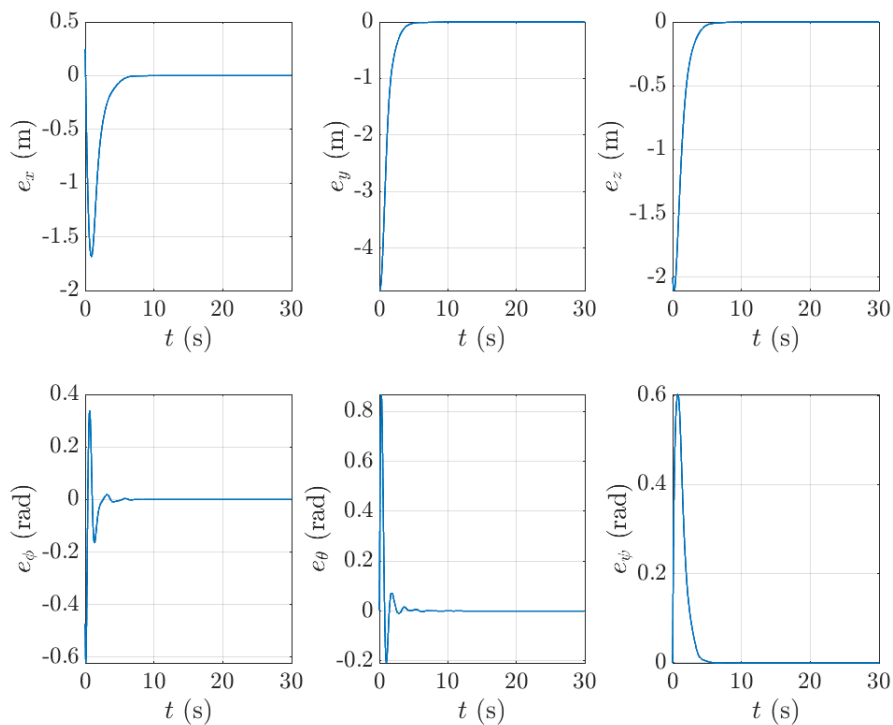
te simulacija traje 30 sekundi.

Do simulacijskih rezultata se dolazi istim putem kako je objašnjeno za prvu trajektoriju te se čitatelja, ako postoji zanimanje za *Matlab* kodove, ponovno upućuje na Prilog A.

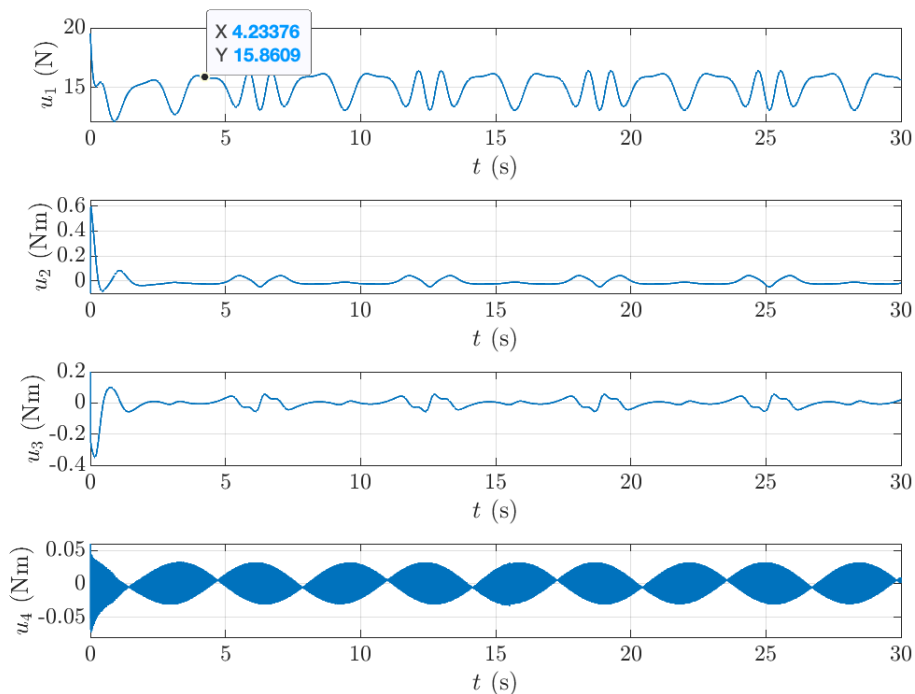


Slika 4.6: Prikaz slijeđenja druge trajektorije.

Slikom 4.6 prikazano je slijeđenje referentne trajektorije za linearizirani i nelinearizirani model. Prema slikama 4.7 te 4.9 ponovno je vidljivo da linearizirani model zadovoljavajuće slijedi zadanu trajektoriju. Upravljačka varijabla  $u_1$  se u ovom slučaju (slika 4.8) mijenja brže u odnosu na onu iz simulacijskih rezultata prve trajektorije, što je razlog kompliciranije trajektorije i dodatne vremenske ovisnosti  $\psi_{ref}$ . Ostale upravljačke varijable su kao i prije ne prevelikih iznosa (slika 4.8), no to je očekivano s obzirom na to da je dron malih dimenzija. Kao i prije, linearizirani model zanemarivo bolje prati referentnu trajektoriju od nelineariziranog, što je vidljivo u niže danom uvećanom prikazu dijela trajektorije u  $x - y$  koordinatnom sustavu (slika 4.10).

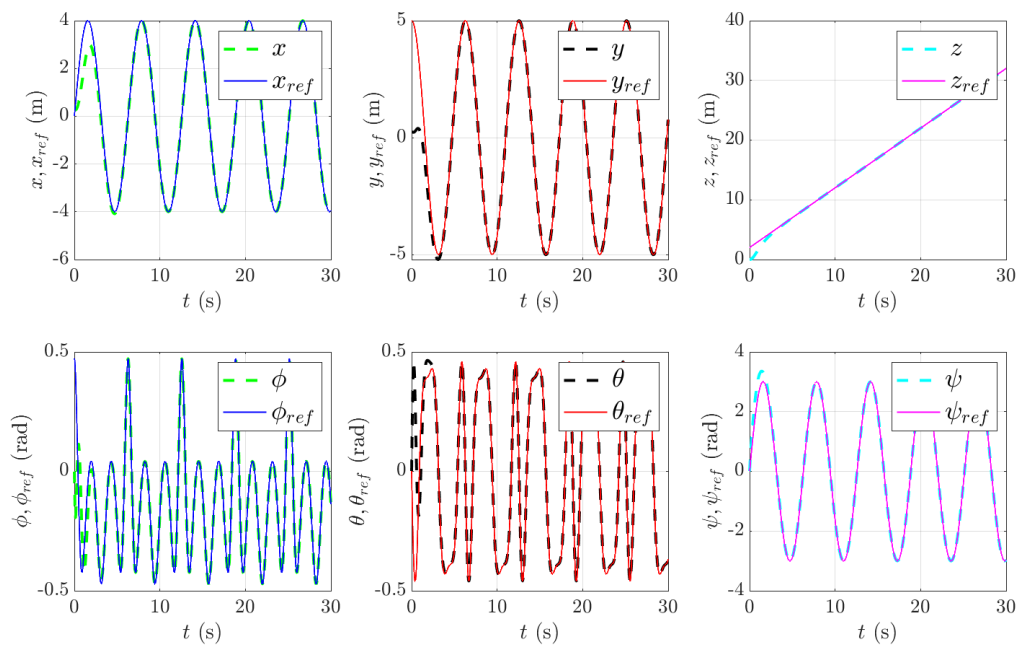


Slika 4.7: Pogreške slijeđenja referentnih položaja i orijentacija za linearizirani model (druga trajektorija).

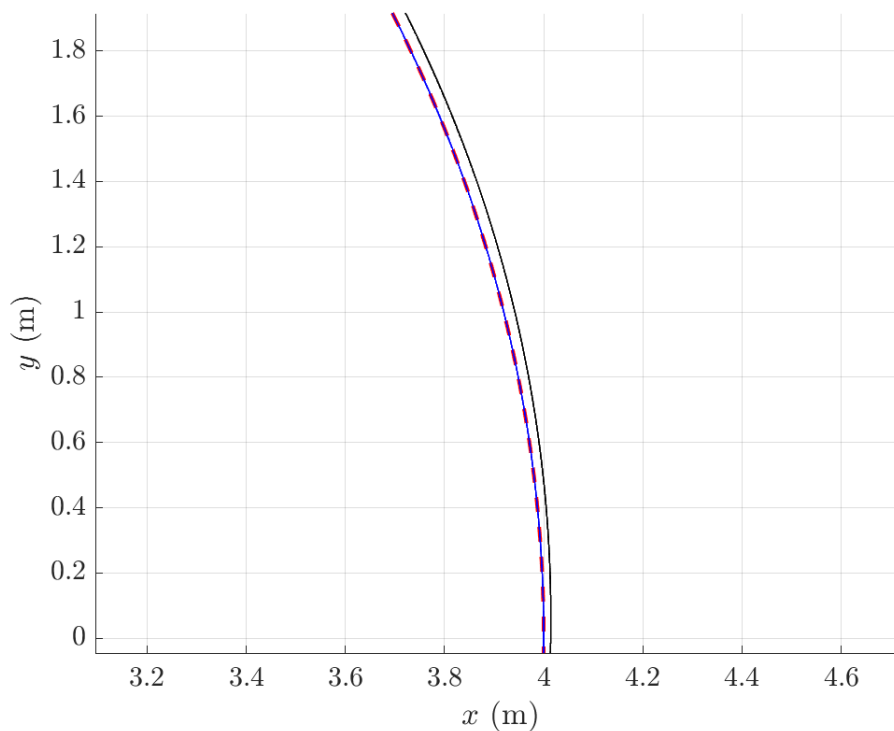


Slika 4.8: Iznosi upravljačkih varijabli lineariziranog modela za slijeđenje druge trajektorije.





Slika 4.9: Usporedba položaja i orijentacija lineariziranog modela s referentnima (druga trajektorija).



Slika 4.10: Uvećan prikaz dijela druge trajektorije.

## 5 Zaključak

Cilj ovog rada bila je implementacija vremenski promjenjivog linearnog kvadratičnog regulatora na četverorotorsku bespilotnu letjelicu u vidu slijeđenja nekoliko referentnih trajektorija. Objašnjenjem osnovnih pojmova vezanih uz optimalno upravljanje poput Hamilton-Jacobi-Bellmanove jednačbe te Riccatijeve jednačbe te kasnijim objašnjenjem linearnog kvadratičnog regulatora za vremenski invarijantne sustave, uslijedilo je definiranje vremenski promjenjivog linearnog kvadratičnog regulatora s primjenom na vremenski promjenjive sustave prilikom čega se objasnila i nužna tematika linearizacije nelinearnih sustava oko referentne trajektorije te prikaz istih u koordinatama pogreške u prostoru stanja.

Izvod potpunog matematičkog modela bespilotne letjelice bez vanjskih poremećaja, koji je ujedno nelinearan, popratila je linearizacija oko referentne trajektorije kojom su se izveli opći izrazi vremenski promjenjivih matrica  $\mathbf{A}(t)$  i  $\mathbf{B}(t)$  te definirale varijable stanja kao i upravljačke varijable. Takvim je definiranjem uslijedila simulacija lineariziranog sustava zapisanog u obliku koordinata pogreške, uz čiji se dodatni raspis direktno računali vektori stanja i upravljački vektori. Nelinearan model letjelice također je podvrgnut simulaciji za trajektorije zadane lineariziranom modelu te su se usporedila i razmotrila praćenja referentne trajektorije oba modela uz prikaz upravljačkih varijabli i pogreške slijeđenja za linearizirani model, koji je shodno bio glavna tematika ovog rada.

Kao daljnji korak razrade ovog rada preporučuje se primjena ovakve metode upravljanja na fizičkom modelu letjelice te razrada i usporedba drugih adaptivnih metoda poput modelskog prediktivnog upravljanja te *backstepping* metode u odnosu na vremenski promjenjiv linearni kvadratični regulator prilikom čega se u obzir uzimaju vanjski poremećaji.

# Literatura

- [1] F. L. Lewis, D. Vrabie, and V.L. Syrmos. *Optimal Control*. John Wiley & Sons, Hoboken, New Jersey, 2012.
- [2] E.D. Kirk. *Optimal control theory: an introduction*. Courier Corporation, North Chelmsford, Massachusetts, 2004.
- [3] J.P. How. *Principles Of Optimal Control: HJB Equation (šifra 16.323)*. Massachusetts Institute of Technology, Massachusetts, 2008. Dostupno na: <https://ocw.mit.edu/courses/16-323-principles-of-optimal-control-spring-2008/resources/lec4/>, [06-08-2024.].
- [4] R. Tedrake. *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (šifra 6.832)*. Massachusetts Institute of Technology, Massachusetts, 2023. Dostupno na: <https://underactuated.csail.mit.edu>, [15-07-2024.].
- [5] A. Laub. A schur method for solving algebraic riccati equations. *IEEE Transactions on automatic control*, 24(6):913–921, 1979.
- [6] B. Datta. *Numerical methods for linear control systems*. Elsevier, 2004.
- [7] D. Čosić. Runge kutta metode za numeričko rješavanje diferencijalnih jednadžbi. Bachelor's thesis, Osijek: Sveučilište u Osijeku, Fakultet primijenjene matematike i informatike, 2018.
- [8] F. Sabatino. Quadrotor control: modeling, nonlinear control design, and simulation. Master's thesis, Stockholm: KTH Royal Institute of Technology, 2015.
- [9] W.J. Yeagley. Optimal control for stabilization of quadrotor vehicle trajectories. Master's thesis, San Diego: San Diego State University, 2019.
- [10] I. González-Hernández, S. Salazar, R. Lozano, and O. Ramírez-Ayala. Real-time improvement of a trajectory-tracking control based on super-twisting algorithm for a quadrotor aircraft. *Drones*, 6(2):36, 2022.
- [11] Robotic Systems Lab. *Robot Dynamics Lecture Notes*. ETH Zürich, Zürich, 2017. Dostupno na: [https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2017/RD\\_HS2017script.pdf](https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2017/RD_HS2017script.pdf), [08-08-2024.].

- [12] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.
- [13] Y. Mutoh and L. Awatsu. Simple controller for trajectory tracking control of quadrotor unmanned aerial vehicle. In *2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM)*, pages 213–218. IEEE, 2016.
- [14] C.W. Chang and J.K. Shiau. Quadrotor formation strategies based on distributed consensus and model predictive controls. *Applied Sciences*, 8(11):2246, 2018.

# A *Matlab* programski kodovi

## A.1 Funkcije

```
1  %% Matlab funkcija koja racuna matricu pojacanja kroz
   vremenske korake definiranog vremenskog intervala te kao
   izlaz vraca i matricu pojacanja kao funkciju vremena (obje
   matrice se mogu koristiti za rjesavanje sustava, no
   pristup nije jednak)
2
3  function K_matrica = dobivanje_pojacanja(A_ulaz, B_ulaz,
   Q_ulaz, R_ulaz, t0, tf, dt)
4
5  vremenski_koraci = (tf-t0)/dt;
6
7  [n,m] = size(B_ulaz(t0));
8
9  K_matrica = zeros(m,n)
10
11 for i = 1:vremenski_koraci
12
13     A_t = A_ulaz(i);
14     B_t = B_ulaz(i);
15     Q_t = Q_ulaz(i);
16     R_t = R_ulaz(i);
17
18     [K, ~, ~] = lqr(A_t, B_t, Q_t, R_t);
19
20     K_matrica(:,:,i) = K;
21
22 end
23
24 end
```

```
1  %% Funkcija za rjesavanje linariziranog sustava
2
```

```
3 function dx = linearnisustav(t, x, ref, xrefdot, At, Bt, Kt)
4
5 Att = At(t);
6 Btt = Bt(t);
7 Ktt = Kt(t);
8 xref = ref(t); % prijasnje definirana referentna trajektorija
9 xdref = xrefdot(t); % prijasnje definirana derivacija
   referentne trajektorije
10 dx = xdref + (Att-Btt*Ktt)*x - (Att-Btt*Ktt)*xref;
11
12 end
```

```
1 %% Funkcija za rjesavanje nelinearnog sustava sustava
2
3 function xdot = nelinsustav(t, x, refx, At, Bt, Kt, refu)
4
5 masa = 1.5;
6 grav = 9.80665;
7 Ixx = 0.022;
8 Iyy = 0.022;
9 Izz = 0.0018;
10 KA = (Iy - Iz) / Ix;
11 KB = (Iz - Ix) / Iy;
12 KC = (Ix - Iy) / Iz;
13
14 Att = At(t);
15 Btt = Bt(t);
16 Ktt = Kt(t);
17 xref = refx(t);
18 uref = refu(t);
19
20 xdot = zeros(12,1);
21
22 ut = -Ktt*(x-xref) + uref;
23
24 u1 = ut(1);
25 u2 = ut(2);
```

```

26 u3 = ut(3);
27 u4 = ut(4);
28
29 xdot(1) = x(2);
30 xdot(2) = (u1 / masa) * (cos(x(11)) * sin(x(9)) * cos(x(7)) +
    sin(x(11)) * sin(x(7)));
31 xdot(3) = x(4);
32 xdot(4) = (u1 / masa) * (sin(x(11)) * sin(x(9)) * cos(x(7)) -
    cos(x(11)) * sin(x(7)));
33 xdot(5) = x(6);
34 xdot(6) = -grav + (u1 / masa) * (cos(x(9)) * cos(x(7)));
35 xdot(7) = x(8);
36 xdot(8) = x(10) * x(12) * KA + (u2) / Ixx;
37 xdot(9) = x(10);
38 xdot(10) = x(8) * x(12) * KB + (u3) / Iyy;
39 xdot(11) = x(12);
40 xdot(12) = x(8) * x(10) * KC + (u4) / Izz;
41
42 end

```

## A.2 Definiranje parametara i varijabli

```

1 %% Upotreba Symbolic paketa za potpuno definiranje referentne
    trajektorije (ovaj racun se odvija u internetskoj verziji
    Matlaba te se vrijednosti ponovno definiraju na Matlabu
    instaliranom na racunalu)
2
3 % Definiranje simbolickih varijabli
4 clc;
5 clearvars;
6 close all;
7 syms m g Ixx Iyy Izz x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 t
8
9 % Definiranje parametara
10 g = 9.80665;
11 Ixx = 0.022;

```

```
12 Iyy = 0.022;
13 Izz = 0.0018;
14 m = 1.5;
15
16 % Definiranje pocetnih varijabli referentne trajektorije
17 x_ref = ...;
18 y_ref = ...;
19 z_ref = ...;
20 psi_ref = ...;
21
22 % Racunanje potrebnih derivacija
23 ddx_ref = diff(x_ref, t, 2);
24 ddy_ref = diff(y_ref, t, 2);
25 ddz_ref = diff(z_ref, t, 2);
26 ddpsi_ref = diff(psi_ref, t, 2);
27
28 theta_ref = atan((ddx_ref*cos(psi_ref) + ddy_ref*sin(psi_ref)
    )/(g+ddz_ref));
29 phi_ref = atan((cos(theta_ref) * (ddx_ref*sin(psi_ref) -
    ddy_ref*cos(psi_ref)))/(g + ddz_ref));
30 ddphi_ref = diff(phi_ref, t, 1);
31 ddtheta_ref = diff(theta_ref, t, 1);
32
33 dx_ref = diff(x_ref, t, 1);
34 dy_ref = diff(y_ref, t, 1);
35 dz_ref = diff(z_ref, t, 1);
36 dphi_ref = diff(phi_ref, t, 1);
37 dtheta_ref = diff(theta_ref, t, 1);
38 dpsi_ref = diff(psi_ref, t ,1);
39
40 % Definiranje varijabli stanja
41 x1 = x_ref;
42 x2 = dx_ref;
43 x3 = y_ref;
44 x4 = dy_ref;
45 x5 = z_ref;
46 x6 = dz_ref;
```



```
47 x7 = phi_ref;
48 x8 = dphi_ref;
49 x9 = theta_ref;
50 x10 = dtheta_ref;
51 x11 = psi_ref;
52 x12 = dpsi_ref;
53
54 % Derivacija varijabli stanja potrebnih za daljnji racun
55 dx6 = diff(x6, t, 1);
56 dx8 = diff(x8, t, 1);
57 dx10 = diff(x10, t, 1);
58 dx12 = diff(x12, t, 1);
59
60 % Definiranje referentnog upravljanja
61 u1_ref = (dx6 + g)*(m/cos(x9)*cos(x7));
62 u2_ref = (dx8 - x10*x12*KA)*Ixx;
63 u3_ref = (dx10 - x8*x12*KB)*Iyy;
64 u4_ref = (dx12 - x8*x10*KC)*Izz;
65 u_ref = [u1_ref; u2_ref; u3_ref; u4_ref];
66
67 % Prikaz simbolickih izraza varijabli stanja i upravljanja u
    obliku Matlab funkcije
68 x1f = matlabFunction(x1)
69 x2f = matlabFunction(x2)
70 x3f = matlabFunction(x3)
71 x4f = matlabFunction(x4)
72 x5f = matlabFunction(x5)
73 x6f = matlabFunction(x6)
74 x7f = matlabFunction(x7)
75 x8f = matlabFunction(x8)
76 x9f = matlabFunction(x9)
77 x10f = matlabFunction(x10)
78 x11f = matlabFunction(x11)
79 x12f = matlabFunction(x12)
80
81 u1f = matlabFunction(u1_ref)
82 u2f = matlabFunction(u2_ref)
```

```
83 u3f = matlabFunction(u3_ref)
84 u4f = matlabFunction(u4_ref)
85 uf = matlabFunction(u_ref)
86
87 % Derivacija referentne trajektorije i prikaz u obliku Matlab
    funkcije
88 x1d = diff(x1, 1);
89 x2d = diff(x2, 1);
90 x3d = diff(x3, 1);
91 x4d = diff(x4, 1);
92 x5d = diff(x5, 1);
93 x6d = diff(x6, 1);
94 x7d = diff(x7, 1);
95 x8d = diff(x8, 1);
96 x9d = diff(x9, 1);
97 x10d = diff(x10, 1);
98 x11d = diff(x11, 1);
99 x12d = diff(x12, 1);
100 rdot = [x1d; x2d; x3d; x4d; x5d; x6d; x7d; x8d; x9d; x10d;
    x11d; x12d];
101
102 rdotf = matlabFunction(rdot);
```

```
1 %% Matlab kod za definiranje potrebnih matrica i definiranje
    ponovno definiranje izraza i parametara prethodno
    dobivenih u Symbolic paketu
2
3 clc;
4 clearvars;
5 close all;
6
7 g = 9.80665;
8 Ixx = 0.022;
9 Iyy = 0.022;
10 Izz = 0.0018;
11 m = 1.5;
12
```

```

13 % Pocetno i krajnje vrijeme simulacije
14 t0 = 0;
15 tf = 30;
16 dt = ... % na primjer 0.1
17 % Definiranje varijabli stanja, upravljackih varijabli te
    derivacije referentne trajektorije dobivenih putem
    Symbolic paketa
18 x1 = @(t) ...;
19 x2 = @(t) ...;
20 x3 = @(t) ...;
21 x4 = @(t) ...;
22 x5 = @(t) ...;
23 x6 = @(t) ...;
24 x7 = @(t) ...;
25 x8 = @(t) ...;
26 x9 = @(t) ...;
27 x10 = @(t) ...;
28 x11 = @(t) ...;
29 x12 = @(t) ...;
30
31 r = @(t) [x1(t); x2(t); x3(t); x4(t); x5(t); x6(t); x7(t); x8(
    t); x9(t); x10(t); x11(t); x12(t)]; % referentna
    trajektorija
32 rdot = @(t) [...]; % derivacija referentne trajektorija
33
34 u1_ref = @(t) ...;
35 u2_ref = @(t) ...;
36 u3_ref = @(t) ...;
37 u4_ref = @(t) ...;
38
39 u_ref = @(t) [u1_ref(t); u2_ref(t); u3_ref(t); u4_ref(t)];
40
41 % Matrice A i B
42 A = @(t) [[0 1 0 0 0 0 0 0 0 0 0 0]; [0 0 0 0 0 0 (u1_ref(t)
    ./m) .* (cos(x7(t)).*sin(x11(t)) - cos(x11(t)).*sin(x7(t))
    .*sin(x9(t))) 0 (u1_ref(t)./m) .* (cos(x7(t)).*cos(x9(t))
    .*cos(x11(t))) 0 (u1_ref(t)./m) .* (cos(x11(t)).*sin(x7(t)

```

```

) - cos(x7(t)).*sin(x9(t)).*sin(x11(t))) 0];
43 [0 0 0 1 0 0 0 0 0 0 0]; [0 0 0 0 0 0 -(u1_ref(t)./m)
.* (cos(x7(t)).*cos(x11(t)) + sin(x7(t)).*sin(x9(t)).*
sin(x11(t))) 0 (u1_ref(t)./m) .* (cos(x7(t)).*cos(x9(t)
)).*sin(x11(t))) 0 (u1_ref(t)./m) .* (sin(x7(t)).*sin(
x11(t)) + cos(x7(t)).*cos(x11(t)).*sin(x9(t))) 0];
44 [0 0 0 0 0 1 0 0 0 0 0]; [0 0 0 0 0 0 -(u1_ref(t)./m)
.* cos(x9(t)).*sin(x7(t)) 0 -(u1_ref(t)./m) .* cos(x7(
t)).*sin(x9(t)) 0 0 0]; [0 0 0 0 0 0 0 1 0 0 0];
45 [0 0 0 0 0 0 0 0 0 x12(t) .* ( (Iyy - Izz) / Ixx ) 0 x10(
t) .* ( (Iyy - Izz)/ Ixx )]; [0 0 0 0 0 0 0 0 0 1 0
0];
46 [0 0 0 0 0 0 0 x12(t) .* ( (Izz - Ixx) / Iyy ) 0 0 0 x8(t)
) .* ( (Izz - Ixx)/ Iyy )]; [0 0 0 0 0 0 0 0 0 0 0 1];
47 [0 0 0 0 0 0 0 x10(t) .* ( (Ixx - Iyy) / Izz ) 0 x8(t)
.* ( (Ixx - Iyy)/ Izz ) 0 0]];
48
49 B = @ (t) [[0 0 0 0; (1/m) * (sin(x7(t)) .* sin(x11(t)) + cos(
x7(t)) .* cos(x11(t)) .* sin(x9(t))) 0 0 0]; [0 0 0 0];
[-(1/m) .* (cos(x11(t)) .* sin(x7(t)) - cos(x7(t)) .* sin(
x9(t)) * sin(x11(t))) 0 0 0]; [0 0 0 0];
50 [(1/m) * cos(x7(t)) .* cos(x9(t)) 0 0 0]; [0 0 0 0]; [0
1/Ixx 0 0]; [0 0 0 0]; [0 0 1/Iyy 0]; [0 0 0 0]; [0 0
0 1/Izz]];
51
52 [N,M] = size(B(t0));
53
54 % Definiranje tezijskih matrica, za racun je potrebno da su
definirane kao funkcije vremena, iako u ovome radu nisu (
ali mogu biti)
55 Q = @ (t) eye(N); % pomnoziti zeljenim brojem ili definirati
rucno
56 R = @ (t) eye(M); % pomnoziti zeljenim brojem ili definirati
rucno
57
58 % Pocetni uvjeti
59 x0 = [...];

```

```

60
61 % Izracun matrice pojacanja
62 K_matrica= dobivanje_pojacanja(A ,B, Q, R, t0, tf, dt)
63
64 % Definiranje prethodno dobivene matrice pojacanja K_matrica
    kao funkciju koju ce naredba ode45 interpolirati kroz
    vrijeme simulacije za koje ista izvrsava numericku zadacu
65 K_nova = reshape(K_matrica, [], size(K_matrica, 3));
66
67 koraci = (tf-t0)/dt;
68 vrijeme = linspace(1, tf, koraci);
69
70 K_interpolacija= @(t) reshape(interp1(vrijeme, K_nova', t, "
    linear", "extrap")', size(K_nova, 1), size(K_nova, 2));

```

### A.3 Izvršavanje simulacija i grafički prikaz simulacijskih rezultata

```

1 %% Pozivanje funkcija za rjesavanje linearnog i nelinearnog
    sustava
2 [t_linearno, x_linearno] = ode45(@(t,x) linearnisustav(t,x ,
    r, rdot, A, B, K_interpolacija), [t0 tf], x0);
3
4 [t_nelinearno, x_nelinearno] = ode45(@(t,x) nelinsustav(t, x,
    r, A, B, K_interpolacija, u_ref), [t0 tf], x0);

```

```

1 %% Graficki prikaz simulacijskih rezultata (odnose se na
    linearni sustav, dok je nelinearni sustav samo prikazan
    kao trajektorija)
2
3 % Definiranje referentne trajektorije u obliku pogodnom za
    graficki prikaz
4 refe_trajekt = zeros(length(t_linearno),N);
5
6 for i = 1:length(t_linearno)
7     refe_trajekt(i, :) = r(t_linearno(i))';

```

```
8 end
9
10 % Podesavanje tipa grafickog prikaza
11 set(groot, 'defaulttextinterpreter', 'latex');
12 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
13 set(groot, 'defaultLegendInterpreter', 'latex');
14
15 set(gca, 'YTickLabel', '--');
16 set(groot, 'DefaultTextFontSize', 15);
17 set(groot, 'DefaultAxesFontSize', 15);
18
19 % Prikaz trajektorija
20 figure(1)
21 plot3(x_linearno(:,1), x_linearno(:,3), x_linearno(:,5), "r
    --", "LineWidth", 2)
22 hold on
23 plot3(refe_trajekt(:, 1), refe_trajekt(:,3), refe_trajekt(:,
    5), "b", "LineWidth", 1)
24 hold on
25 plot3(x_nelinearno(:,1), x_nelinearno(:,3), x_nelinearno(:,5)
    , "k", "LineWidth", 1)
26 grid on
27 legend("Trajektorija lineariziranog modela", "Referentna
    trajektorija", "Trajektorija nelineariziranog modela")
28 xlabel("$x$ (m)")
29 ylabel("$y$ (m)")
30 zlabel("$z$ (m)")
31
32
33 e_pogreska = x_linearno - refe_trajekt; % pogreska stanja
34
35 % Oznake e oznacavaju pogresku
36 % Prikaz e_x
37 figure(2)
38 subplot(2,3,1)
39 plot(t_linearno, e_pogreska(:,1), "LineWidth", 1)
40 xlabel("$t$ (s)")
```

```
41 ylabel("$e_x$ (m)")
42 grid on
43
44 % Prikaz e_y
45 subplot(2,3,2)
46 plot(t_linearno, e_pogreska(:,3), "LineWidth", 1)
47 xlabel("$t$ (s)")
48 ylabel("$e_y$ (m)")
49 grid on
50
51 % Prikaz e_z
52 subplot(2,3,3)
53 plot(t_linearno, e_pogreska(:,5), "LineWidth", 1)
54 xlabel("$t$ (s)")
55 ylabel("$e_z$ (m)")
56 grid on
57
58 % Prikaz e_phi
59 subplot(2,3,4)
60 plot(t_linearno, e_pogreska(:,7), "LineWidth", 1);
61 xlabel("$t$ (s)");
62 ylabel("$e_{\phi}$ (rad)");
63 grid on
64
65 % Prikaz e_theta
66 subplot(2,3,5)
67 plot(t_linearno, e_pogreska(:,9), "LineWidth", 1);
68 xlabel("$t$ (s)");
69 ylabel("$e_{\theta}$ (rad)");
70 grid on
71
72 % Prikaz e_psi
73 subplot(2,3,6)
74 plot(t_linearno, e_pogreska(:,11), "LineWidth", 1);
75 xlabel("$t$ (s)");
76 ylabel("$e_{\psi}$ (rad)");
77 grid on
```

```
78
79 % Ukupno upravljanje u
80 u_ukupno = zeros(M,length(t_linearno));
81
82 for i = 1:length(t_linearno)
83
84     u_ukupno(:,i) = - K_interpolacija(t_linearno(i))*
85         e_pogreska(i,:)' + u_ref(t_linearno(i));
86
87 end
88
89 % Prikaz u1
90 figure(3)
91 subplot(4,1,1)
92 plot(t_linearno, u_ukupno(1,:), "LineWidth", 1)
93 xlabel("$t$ (s)")
94 ylabel("$u_1$ (N)")
95 grid on
96
97 % Prikaz u2
98 subplot(4,1,2)
99 plot(t_linearno, u_ukupno(2,:), "LineWidth", 1)
100 xlabel("$t$ (s)")
101 ylabel("$u_2$ (Nm)")
102 grid on
103 axis([0 30 -0.1 0.4])
104
105 % Prikaz u3
106 subplot(4,1,3)
107 plot(t_linearno, u_ukupno(3,:), "LineWidth", 1)
108 xlabel("$t$ (s)")
109 ylabel("$u_3$ (Nm)")
110 grid on
111 axis([0 30 -0.7 0.1])
112
113 % Prikaz u4
114 subplot(4,1,4)
```



```
114 plot(t_linearno, u_ukupno(4,:), "LineWidth", 1)
115 xlabel("$t$ (s)")
116 ylabel("$u_4$ (Nm)")
117 grid on
118 axis([0 30 -0.06 0.02])
119
120 % Prikaz x i x_ref
121 figure(4)
122 subplot(2,3,1)
123 plot(t_linearno, x_linearno(:,1), "g--", "LineWidth", 2)
124 hold on
125 plot(t_linearno, refe_trajekt(:,1), "b", "LineWidth", 1)
126 grid on
127 xlabel("$t$ (s)")
128 ylabel("$x, x_{ref}$ (m)")
129 legend("$x$", "$x_{ref}$")
130
131 % Prikaz y i y_ref
132 subplot(2,3,2)
133 plot(t_linearno, x_linearno(:,3), "k--", "LineWidth", 2)
134 hold on
135 plot(t_linearno, refe_trajekt(:,3), "r", "LineWidth", 1)
136 grid on
137 xlabel("$t$ (s)")
138 ylabel("$y, y_{ref}$ (m)")
139 legend("$y$", "$y_{ref}$")
140
141 % Prikaz z i z_ref
142 subplot(2,3,3)
143 plot(t_linearno, x_linearno(:,5), "c--", "LineWidth", 2)
144 hold on
145 plot(t_linearno, refe_trajekt(:,5), "m", "LineWidth", 1)
146 grid on
147 xlabel("$t$ (s)")
148 ylabel("$z, z_{ref}$ (m)")
149 legend("$z$", "$z_{ref}$")
150
```

```
151 % Prikaz phi i phi_ref
152 subplot(2,3,4)
153 plot(t_linearno, x_linearno(:,7), "g--", "LineWidth", 2)
154 hold on
155 plot(t_linearno, refe_trajekt(:,7), "b", "LineWidth", 1)
156 grid on
157 xlabel("$t$ (s)")
158 ylabel("$\phi, \phi_{ref}$ (rad)")
159 legend("$\phi$", "$\phi_{ref}$")
160
161 % Prikaz theta i theta_ef
162 subplot(2,3,5)
163 plot(t_linearno, x_linearno(:,9), "k--", "LineWidth", 2)
164 hold on
165 plot(t_linearno, refe_trajekt(:,9), "r", "LineWidth", 1)
166 grid on
167 xlabel("$t$ (s)")
168 ylabel("$\theta, \theta_{ref}$ (rad)")
169 legend("$\theta$", "$\theta_{ref}$")
170
171 % Prikaz psi i psi_ref
172 subplot(2,3,6)
173 plot(t_linearno, x_linearno(:,11), "c--", "LineWidth", 2)
174 hold on
175 plot(t_linearno, refe_trajekt(:,11), "m", "LineWidth", 1)
176 grid on
177 xlabel("$t$ (s)")
178 ylabel("$\psi, \psi_{ref}$ (rad)")
179 legend("$\psi$", "$\psi_{ref}$")
```