

# Analiza i sinteza sustava upravljanja pozicijom loptice na gredi

---

**Mikulić, Matija**

**Undergraduate thesis / Završni rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:026777>

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

*Download date / Datum preuzimanja:* **2025-03-03**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni studij**

**ANALIZA I SINTEZA SUSTAVA UPRAVLJANJA  
POZICIJOM LOPTICE NA GREDE**

**Završni rad**

**Matija Mikulić**

**Osijek, 2022.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac ZIP - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 22.08.2022.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Matija Mikulić
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R 4401, 22.07.2019.
<b>OIB Pristupnika:</b>	62692317424
<b>Mentor:</b>	Izv. prof. dr. sc. Emmanuel Karlo Nyarko
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Analiza i sinteza sustava upravljanja pozicijom loptice na gredi
<b>Znanstvena grana rada:</b>	<b>Procesno računarstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	Potrebno je provesti analizu i sintezu sustava upravljanja pozicije loptice na gredi. Zadatak uključuje razvoj modela i simulaciju te vizualizaciju u MATLAB-u. (Tema rezervirana za: Matija Mikulić)
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	22.08.2022.
<b>Datum potvrde ocjene od strane Odbora:</b>	07.09.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 08.09.2022.

**Ime i prezime studenta:**

Matija Mikulić

**Studij:**

Preddiplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

R 4401, 22.07.2019.

**Turnitin podudaranje [%]:**

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Analiza i sinteza sustava upravljanja pozicijom loptice na gredi**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
1.1. Zadatak završnog rada .....	1
<b>2. PREGLED PODRUČJA TEME .....</b>	<b>2</b>
<b>3. OPIS PROCESA .....</b>	<b>3</b>
3.1. Matematički model .....	3
3.2. Linearizacija procesa .....	5
<b>4. ANALIZA STABILNOSTI I ODABIR REGULATORA .....</b>	<b>6</b>
4.1. Prijelazna funkcija .....	6
4.2. Krivulja mjesta korijena .....	8
4.3. Regulator s faznim prethodenjem .....	9
<b>5. IMPLEMENTACIJA KORISNIČKOG SUČELJA U MATLABU .....</b>	<b>11</b>
5.1. App Designer .....	11
5.2. Sučelje aplikacije .....	12
5.3. Simulink model .....	14
5.4. Kôd aplikacije .....	14
<b>6. SIMULACIJA I ANALIZA REZULTATA .....</b>	<b>23</b>
<b>7. ZAKLJUČAK .....</b>	<b>27</b>
<b>LITERATURA .....</b>	<b>28</b>
<b>SAŽETAK .....</b>	<b>29</b>
<b>ABSTRACT .....</b>	<b>30</b>

## 1. UVOD

Pojam automatskog upravljanja sve češće se javlja u suvremenoj industriji. Riječ je o jednoj temeljnoj disciplini zastupljenoj u tehničkim, ali i u netehničkim sustavima koje omogućava ostvariti određene ciljeve i rad bez prisutnosti čovjeka. Uvođenjem ovakvih sustava upravljanja utjecalo je znatno na povećanja produktivnosti u industriji, smanjenju potrošje energije, poboljšanju kvalitete proizvoda te brojne druge prednosti. No osim toga, brojni su sustavi vrlo složeni ili prebrzi da bi čovjek mogao njima uspješno upravljati, ili pak preopasni za zdravlje, iziskujući tako potrebu za automatizacijom tih sustava [1].

Premda proces obrađen u ovom završnom radu nije primjer industrijskog procesa, pokazano je kako je moguće napraviti sustav upravljanja za proces koji se može matematički opisati.

U drugom su poglavlju navedene i opisane metode korištene u radovima koji analiziraju i obrađuju problem upravljanja istog procesa.

U trećem je poglavlju opisan proces za kojeg je potrebno napraviti sustav upravljanja. Napravljen je matematički model te je proces lineariziran u okolini radne točke.

U četvrtom poglavlju slijedi analiza stabilnosti procesa grafoanalitičkom metodom krivulja mjesta korijena te prikaz prijelaznog odziva sustava. Objašnjen je regulator koji se koristi pri sintezi te izrazi koji će biti korisni pri njegovoj implementaciji.

U petom poglavlju slijedi implementacija zadatka u MATLAB programu. Napravljeno je korisničko sučelje korištenjem *App Designer* te je objašnjena sinteza regulatora.

U šestom poglavlju slijedi simulacija i analiza rezultata. Korištenje sučelja pokazano je kroz nekoliko primjera.

### 1.1. Zadatak završnog rada

Potrebno je provesti analizu i sintezu sustava upravljanja pozicije loptice na gredi. Zadatak uključuje razvoj modela i simulaciju te vizualizaciju u MATLAB-u.

## 2. PREGLED PODRUČJA TEME

U ovom su poglavlju razmatrane neke od metoda koje su korištene za sustav upravljanja pozicijom loptice na gredi. U radovima korišten je isti matematički model procesa i riješen isti zadatak kao i uvom radu, no uz nešto drugačije pristupe rješavanja problema. Zadatak je tih radova dizajnirati takav zatvoreni regulacijski krug kojim se postiže zadovoljavajuća dinamika sustava.

U radu [2] korištene su dvije metode za sintezu PID regulatora:

1. Zieglerova i Nicholsova metoda ruba stabilnosti
2. Metoda dijagrama koeficijenata (CDM)

Metoda ruba stabilnosti implementirana je na način da je najprije razmotreno pojačanje regulatora. Pojačanje je povećano dok sustav nije postigao granicu stabilnosti, odnosno do onog trenutka kada prijelazna funkcija ima trajne oscilacije. Iz prijelazne je funkcije potom očitani period oscilacija te su određeni parametri PID regulatora na temelju tog pojačanja i perioda oscilacije.

CDM algebarski je pristup za sintezu regulatora. Parametri dizajnirani tom metodom su indeksi stabilnosti te vremenske konstante. Indeksi stabilnosti određuju stabilnost sustava odnosno dinamičko vladanje sustava, dok vremenske konstante utječu na brzinu odziva sustava. Uz pomoć tih parametara određen je karakteristični polinom zatvorenog regulacijskog kruga te PID regulatora

Metodom dijagrama koeficijenata postignuta je bolja dinamika sustava. Vrijeme porasta, ustaljivanja te maksimalno nadvišenje dobiveni tom metodom bili su manjih iznosa od onih dobivenih metodom ruba stabilnosti.

U radu [3] korištena je aktivna regulacija odbijanja smetnji (ADRC). Tom je metodom omogućen brži odziv sustava te manje maksimalno nadvišenje. Kao rezultat dobiven je sustav s boljom dinamikom sustava u odnosu na zatvoreni regulacijski krug s PID regulatorom.

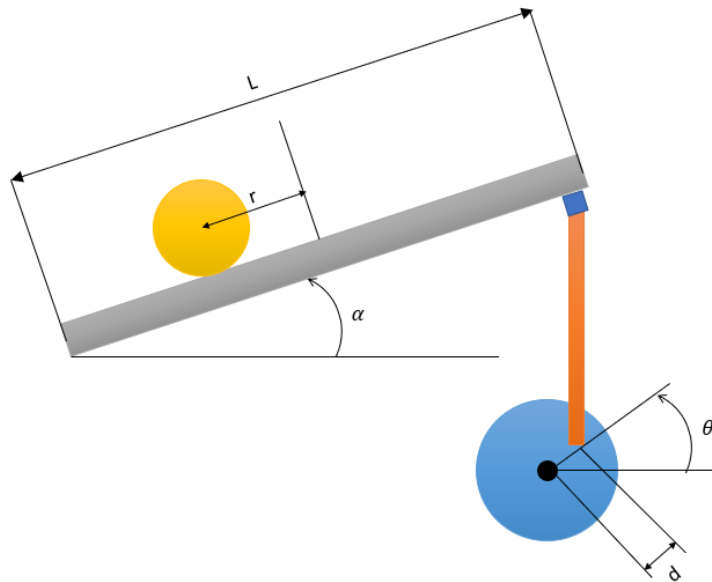
### 3. OPIS PROCESA

U ovom je poglavlju matematički opisan sustav loptice na gredi. Kako bi se izvršila analiza i sinteza sustava, najprije je izveden matematički model procesa. To je model koji se sastoji od jednažbi koje opisuju proces i opisuju vladanje procesa u statičkim i dinamičkim režimima.

Većina je procesa u prirodi nelinearna, pa tako i ovaj. Zbog nelinearnost, provedena je linearizacija procesa u okolini radne točke, odnosno aproksimacija nelinearnog sustava linearnim, u svrhu provođenja analize i sinteze.

#### 3.1. Matematički model

Model procesa prikazan je na slici 3.1. [4]. Slika 3.1. prikazuje lopticu koja se kotrlja na gredi. Njen položaj ovisi o kutu pod kojim se nalazi greda, a taj kut upravljani je krakom poluge koja je s jedne strane povezana s gredom, a s druge strane sa zupčanikom. Tablica 3.1. prikazuje sve varijable i konstante procesa.



Slika 3.1. Sustav loptice na gredi



Tablica 3.1. Parametri procesa

<b>Parametar</b>	<b>Simbol</b>	<b>Mjerna jedinica</b>
Masa loptice	m	kg
Polumjer loptice	R	m
Pomak kraka poluge	d	m
Gravitacijsko ubrzanje	g	$m/s^2$
Duljina grede	L	m
Moment inercije loptice	J	$kg \cdot m^2$
Pozicija loptice	r	m
Kut grede	$\alpha$	rad
Kut zupčanika	$\theta$	rad

Za određivanje jednadžbi procesa potrebno je koristiti fizikalne zakone. U ovom je procesu pretpostavljeno da je trenje između loptice i grede zanemarivo i da loptica ne može pasti sa grede. Sustav prikazuje jednodimenzionalno kretanje loptice i na nju djeluju sila gravitacije, translacijska sila, centrifugalna sila i inercija. Iz ovoga slijedi jednadžba ravnoteže:

$$0 = F_{translacije} + F_{rotacije} + F_x - F_{centrifugalna} \quad (3-1)$$

Pri čemu je sila translacije loptice:

$$F_{translacije} = m \cdot \frac{d^2r}{dt^2} \quad (3-2)$$

Sila uslijed rotacije je:

$$F_{rotacije} = \frac{J}{R^2} \cdot \frac{d^2r}{dt^2} \quad (3-3)$$

Komponenta sile teže je:

$$F_x = mg \cdot \sin \alpha \quad (3-4)$$

Centrifugalna sila je:

$$F_{centrifugalna} = mr \cdot \left(\frac{d\alpha}{dt}\right)^2 \quad (3-5)$$

Ako se uzme u obzir oblik loptice i njena ispunjenost, formula za moment inercije ispunjene sfere je [5]:

$$J = \frac{2mR^2}{5} \quad (3-6)$$

Uvrštavajući jednadžbe (3-2) do (3-5) u jednadžbu ravnoteže (3-1) dobije se konačni izraz i konačna jednadžba modela procesa:

$$0 = \left(\frac{J}{R^2} + m\right) \cdot \frac{d^2r}{dt^2} + mg \cdot \sin \alpha - mr \cdot \left(\frac{d\alpha}{dt}\right)^2 \quad (3-7)$$

### 3.2. Linearizacija procesa.

Diferencijalna jednadžba (3-7) nelinearna je jednadžba koju je potrebno linearizirati u svrhu analize i sinteze sustava. To se može postići ako se zanemari ovisnost druge derivacije kuta grede o pomaku loptice te uz pretpostavku da su promjene kuta grede male, tada vrijedi da je  $\sin \alpha \approx \alpha$ . Također se uzima u obzir da je omjer kuta grede i kuta zupčanika jednak omjeru pomaka kraka poluge i duljine grede, odnosno:

$$\alpha = \frac{d}{L} \theta \quad (3-8)$$

Uz gore navedeno, dobije se sljedeća linearna diferencijalna jednadžba 2. reda:

$$0 = \left(\frac{J}{R^2} + m\right) \cdot \frac{d^2r}{dt^2} + mg \cdot \frac{d}{L} \theta \quad (3-9)$$

Zadatak je upravljati pozicijom loptice na gredi mijenjajući kut grede, što znači da je izlazna veličina procesa pomak loptice, a ulazna veličina kut grede. Kako bi se dobio odnos izlazne i ulazne veličine, primijenjena je Laplaceova transformacija nad jednadžbom (3-9) čime je dobivena algebarska jednadžba sustava u kompleksnoj domeni. Omjer izlazne i ulazne veličine u takvoj domeni predstavlja prijenosnu funkciju sustava i ona ima sljedeći oblik:

$$G(s) = \frac{R(s)}{\theta(s)} = \frac{-mgd}{L\left(\frac{J}{R^2} + m\right)s^2} \quad [m/rad] \quad (3-10)$$

## 4. ANALIZA STABILNOSTI I ODABIR REGULATORA

Pri analizi nekog sustava važno je utvrditi njegovu stabilnost. Sustav može biti stabilan, nestabilan i na rubu stabilnosti. Neki linearno vremenski nepromjenjiv sustav asimptotski je stabilan ukoliko za njegovu težinsku funkciju vrijedi:

$$\lim_{t \rightarrow \infty} g(t) = 0 \quad (4-1)$$

Ukoliko limes (4-1) poprimi beskonačan iznos, sustav je stabilan, a ukoliko je jednak konstanti, tada je na granici stabilnosti [6].

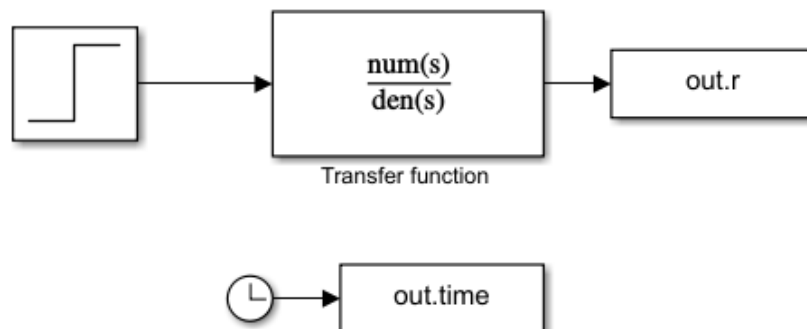
Postoji nekoliko metoda kojima se može odrediti stabilnost sustava. Te metode mogu biti analitičke i grafoanalitičke. U ovom je poglavlju prikazana grafoanalitička metoda krivulja mjesta korijena koja je ujedno korištena i za sintezu regulatora. Također je prikazan model procesa u simulinku te odziv na skokovitu pobudu.

### 4.1. Prijelazna funkcija

Prijelazna je funkcija odziv sustava na skokovitu pobudu. Skokovita je pobuda signal koji je matematički definiran na sljedeći način:

$$s(t) = \begin{cases} 1, & t > 0 \\ 0.5, & t = 0 \\ 0, & t < 0 \end{cases} \quad (4-2)$$

Za prikaz grafa prijelazne funkcije, napravljen je linearni model procesa u simulinku na temelju izvedene prijenosne funkcije (3-10). Model je prikazan na slici 4.1.

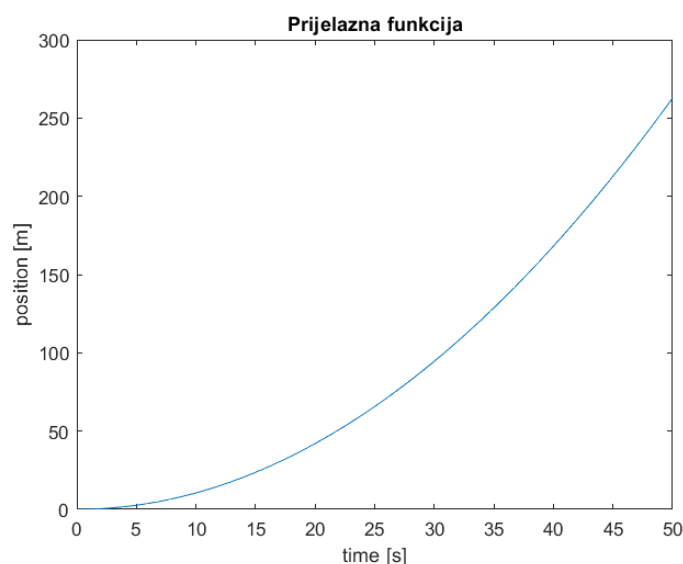


Slika 4.1. Simulink model linearnog sustava u otvorenom krugu

Kako bi se dobio odziv sustava, korištene su testne vrijednosti parametara procesa prikazane u tablici 4.1. Na temelju tih vrijednosti prikazana je prijelazna funkcija sustava na slici 4.2. te je primijećeno da pozicija loptice ide u beskonačno, odnosno ne postoji stacionarno stanje. Iz ovoga je utvrđena nestabilnost procesa te je za potrebe upravljanja sustava potrebno napraviti zatvoreni regulacijski krug za stabilizaciju procesa.

Tablica 4.1. Vrijednosti parametara prijenosne funkcije

Parametar	Simbol	Vrijednost
Masa loptice	m	0.11
Polumjer loptice	R	0.015
Pomak kraka poluge	d	0.03
Gravitacijsko ubrzanje	g	-9.81
Duljina grede	L	1
Moment inercije loptice	J	$9.9 \cdot 10^{-6}$



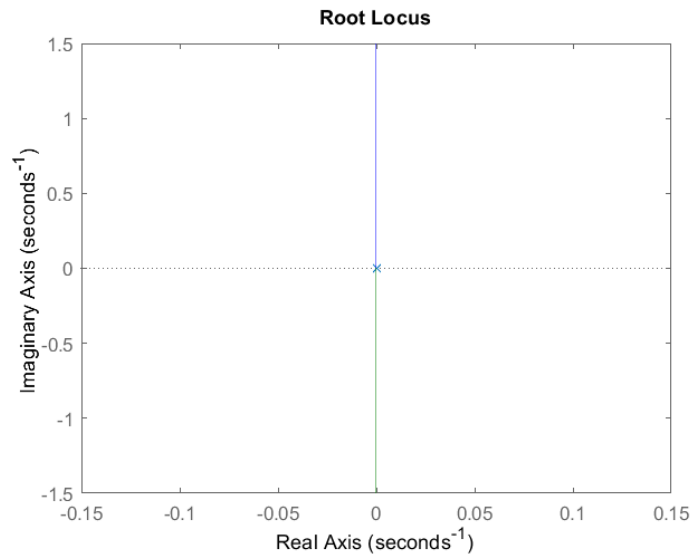
Slika 4.2. Odziv sustava na skokovitu pobudu

## 4.2. Krivulja mjesta korijena

Na temelju prijenosne funkcije procesa (3-10), zaključuje se da je riječ o sustavu drugog reda. Prijenosna funkcija sadrži dva pola koja se nalaze u ishodištu. S obzirom na postojanje višestrukosti pola, sustav je nestabilan. Dakle, bez obzira koje su vrijednosti parametara, sustav neće postati stabilan. Nestabilnost sustava uočena je u prethodnom potpoglavlju gdje je prikazana prijelazna funkcija procesa. Ona će imati isti oblik bez obzira na vrijednosti parametara procesa.

Krivulja mjesta korijena grafoanalitička je metoda koja se primjenjuje na prijenosnu funkciju otvorenog regulacijskog kruga. U ovom slučaju to je samo prijenosna funkcija procesa. Krivulja pokazuje kako se pomiču polovi zatvorenog regulacijskog kruga u kompleksnoj ravnini povećanjem vrijednosti pojačanja otvorenog regulacijskog kruga od nule do neizmjereno.

Slika 4.3. prikazuje krivulju mjesta korijena dobivenu iz prijenosne funkcije procesa uz testne vrijednosti parametara iz tablice 4.1. Uočeno je da sustav, bez obzira na vrijednosti pojačanja, neće postati stabilan jer se krivulja mjesta korijena nalazi na imaginarnoj osi. Dakle, potrebno je koristiti složeniji regulator za stabilizaciju procesa. Takav regulator će sadržavati dodatni pol i nulu kojom će se pomaknuti krivulja mjesta korijena tako da se polovi nalaze na lijevoj strani od imaginarne osi.



Slika 4.3. Krivulja mjesta korijena procesa

### 4.3. Regulator s faznim prethođenjem

U prošlom je potpoglavlju utvrđena nestabilnost procesa. Za stabilizaciju procesa potrebno je dodati regulator u zatvoreni regulacijski krug. Budući da se dva pola procesa nalaze u ishodištu, nije dovoljno samo pojačanje, već je potrebno da regulator sadrži nulu/e i pol/ove. U tu je svrhu odabran regulator s faznim prethođenjem koji ima sljedeći oblik:

$$G_R = K_R \frac{s - n_0}{s - p_0} \quad (4-2)$$

pri čemu  $K_R$  predstavlja pojačanje regulatora,  $n_0$  nulu regulatora te  $p_0$  pol regulatora. Nula regulatora s faznim prethođenjem uvijek je manja od pola. Također nula se nalazi bliže ishodištu koordinatnog sustava u kompleksnoj domeni od ostalih nula. S druge strane, dodani je pol nedominantan kako se ne bi dodatno usporio sustav te kako bi sustav bio otporniji na smetnje pri većim frekvencijama.

Ovaj regulator se može napisati i u drugačijem obliku:

$$G_R = K_R \frac{1 + Ts}{1 + \alpha Ts}, 0 < \alpha < 1 \quad (4-3)$$

Oblik (4-3) se koristi prilikom sinzeze proučavanjem frekvencijskih karakteristika, odnosno proučavanjem Bodeovog ili Nyquistovog grafa. No, kako se u ovom radu vrši sinteza regulatora pomoću krivulje mjesta korijena, izraz (4-2) povoljniji je za korištenje.

Regulator s faznim prethođenjem služi za stabilizaciju procesa na način da uz spomenute lokacije pola i nule regulatora pomakne krivulju mjesta korijena na lijevu stranu koordinatnog sustava ne mijenjajući asimptote krivulje. Osim stabilizacije, ovaj regulator omogućava i poboljšanje nekih karakteristika sustava te njihovo upravljanje. Regulator pozitivno utječe na prijelazni odziv smanjujući vrijeme porasta, vrijeme ustaljivanja i maksimalno nadvišenje koje predstavljaju neke od najvažniji karakteristika u sustavu upravljanja [7].

Već je spomenuto kako je proces upravljanja pozicijom loptice na gredi opisan diferencijalnom jednadžbom drugog reda. Postoje relacije koje su egzaktne samo za sustave drugog reda te za sustave višeg reda koji se mogu aproksimirati sustavom drugog reda, odnosno koji posjeduju dominantan par polova. Te relacije se uglavnom odnose na dinamičke pokazatelje kakvoće poput maksimalnog nadvišenja i vremena porasta.

Vrijeme porasta predstavlja vrijeme koje je potrebno prijelaznoj funkciji da poraste s 10% na 90% vrijednosti ustaljenog stanja, a njena je relacije oblika: [8]

$$t_R = \frac{\pi}{2\omega_n \cdot \sqrt{1 - \zeta^2}} \quad (4-4)$$

$\omega_n$  u jednadžbi (4-4) predstavlja prirodnu frekvenciju neprigušenih oscilacija, a  $\zeta$  koeficijent prigušenja.

Maksimalno nadvišenje predstavlja omjer razlike najveće postignute vrijednosti prijelazne funkcije i vrijednosti u ustaljenom stanju te razlike vrijednosti u ustaljenom stanju sa početnom vrijednosti. Izraz koji povezuje maksimalno nadvišenje i koeficijent prigušenja je: [6]

$$\Omega = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (4-5)$$

Izrazi (4-4) i (4-5) se primijenjuju pri sintezi regulatora. Zadatak je postaviti maksimalno dopušteno vrijeme porasta i maksimalno nadvišenja te s obzirom na njihove vrijednosti odrediti položaje pola i nule regulatora te odrediti pojačanje. Pri samoj implementaciji korisničkog sučelja u MATLAB-u objašnjen je princip rada.

## 5. IMPLEMENTACIJA KORISNIČKOG SUČELJA U MATLABU

Zadatak ovog završnog rada jest napraviti analizu i sintezu procesa upravljanja lopticom na gredi. U tu svrhu kreirana je aplikacija u MATLAB-u koja predstavlja sučelje u kojem se zadaju vrijednosti parametara procesa, postavljaju zahtjevi za sintezu regulatora, postavlja vrijednost referentne veličine, analizira i simulira proces. Za stvaranje aplikacija u MATLAB-u, moguće je koristiti dva razvojna okruženja, *GUIDE* i *App Designer*. U novijim verzijama MATLAB-a preporučeno je koristiti *App Designer*, dok će *GUIDE* biti potpuno uklonjen u novijim verzijama.

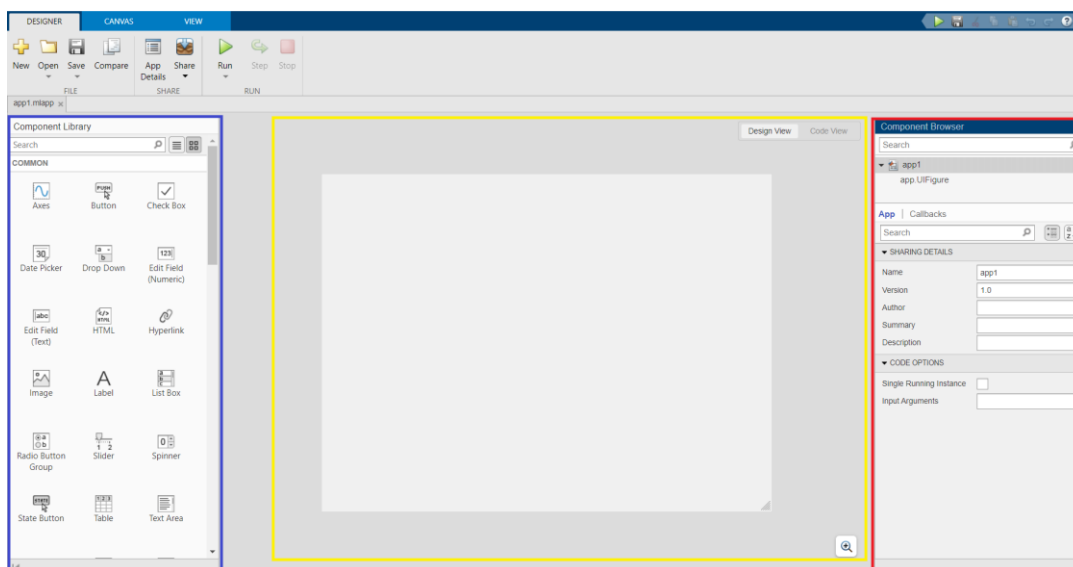
### 5.1. App Designer

Budući da je *App Designer* preporučeno okruženje za izradu aplikacija, ukratko je objašnjeno kako se koristi te u idućim potpoglavljima prikazano sučelje i kôd aplikacije za proces upravljanja pozicijom loptice na gredi.

*App Designer* zamjenjuje prethodno korišteni *GUIDE* nudeći brojne prednosti. Olakšava stvaranje sučelja aplikacija metodom povuci i ispusti te daje moderniji izgled sučelju. Metodom povuci i ispusti mogu se precizno pozicionirati komponente aplikacije te se automatski stvara objektno orijentirani kôd na temelju dizajna, bez potrebe za vlastitom implementacijom kao što je to bio slučaj kôd *GUIDE* okruženja. Nadalje, *App designer* nudi brojne nove funkcionalnosti poput bogatih uređivača svojstava, upravljanje metapodacima aplikacije, podršku za testiranje aplikacije te ugrađeni uređivač kôda. Također pruža brojne nove komponente i instrumente koji su bili prethodno nedostupni [9]

Na slici 5.1. prikazan je prozor *App Designera*. Plavim je pravokutnikom označena biblioteka komponenti koja sadrži sve dostupne komponente koje se mogu pozicionirati na prozoru aplikacije metodom povuci i ispusti. Žutim je pravokutnikom označen prozor aplikacije, a pritiskom na gumb *Design View* omogućen je pregled sučelja. Kôd aplikacije vidljiv je pritiskom na gumb *Code View*. Dio kôda automatski je generiran za inicijalizaciju komponenti aplikacije, za stvaranje i brisanje prozora aplikacije. Nadalje, crvenim je pravokutnikom prikazan pretraživač komponenti aplikacije, uređivač svojstava komponenti i stvoreni događaji. Osim događaja, moguće je definirati vlastita svojstva i funkcije.

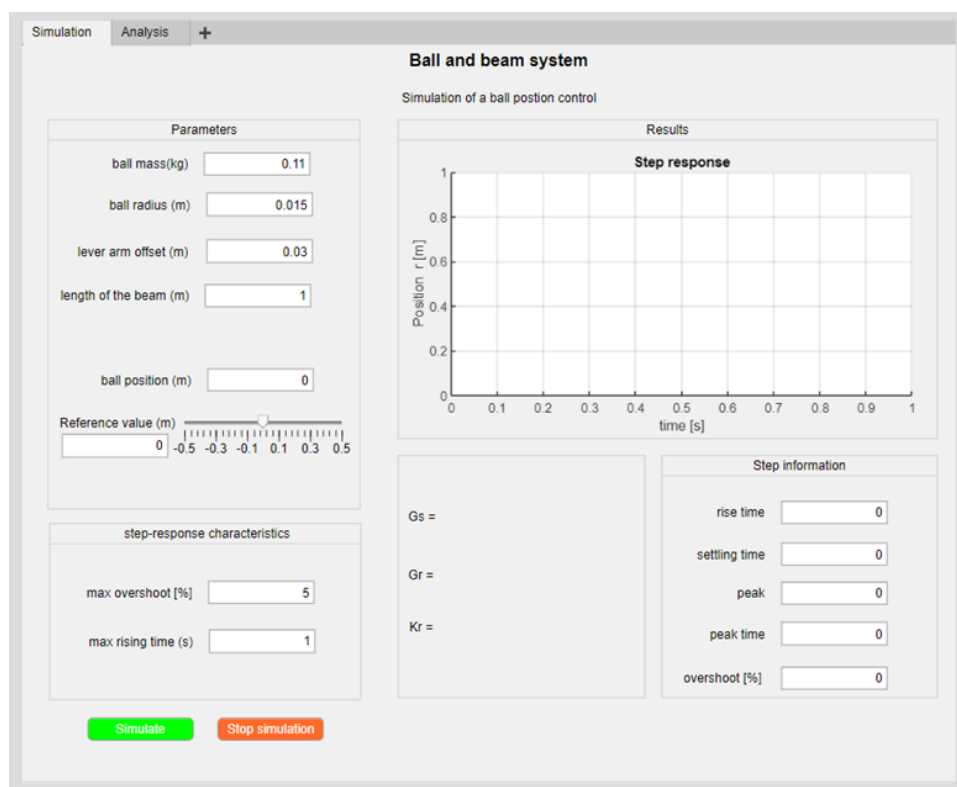




Slika 5.1. Izgled App Designera

## 5.2. Sučelje aplikacije

U nastavku opisana je izrada sučelja aplikacije koja omogućuje korisniku unos parametara procesa, zadavanje zahtjeva za regulator i analizu. Na slici 5.2. prikazano je sučelje koje sadrži dvije kartice.



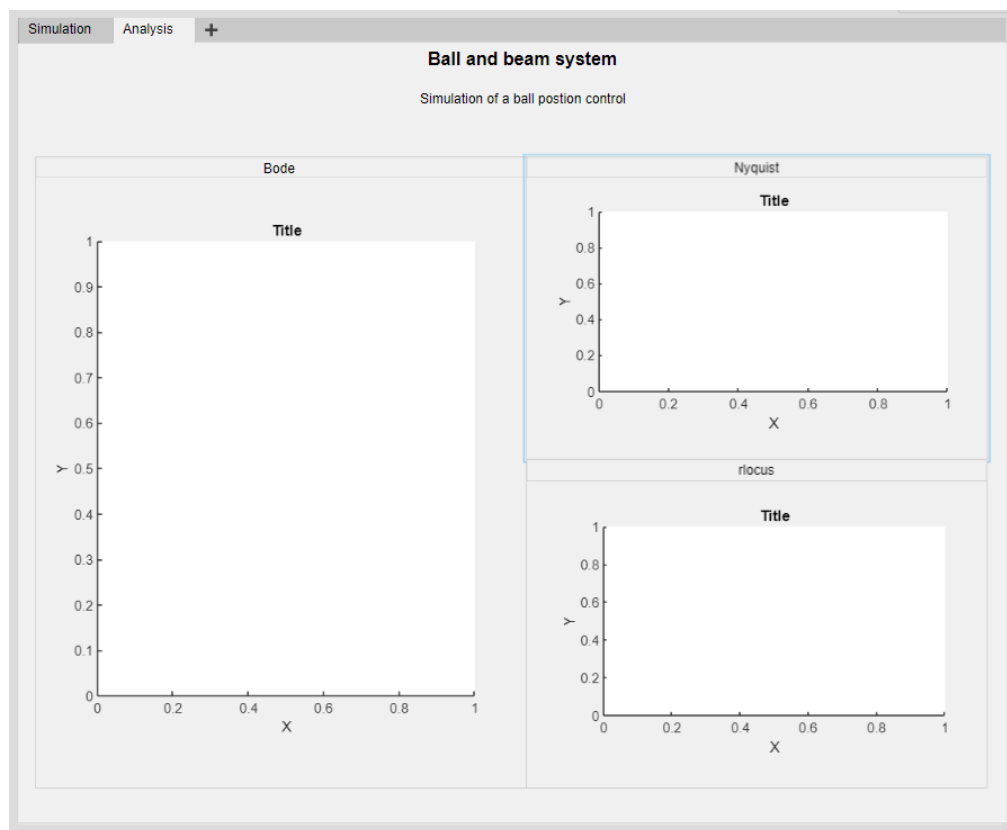
Slika 5.2. Sučelje aplikacije za simulaciju procesa

Kartica *Simulation* ima nekoliko skupina. U skupinu *Parameters* korisnik unosi u numerička polja vrijednosti parametara procesa, a potom preko numeričkog polja ili kliznika postavlja referentnu vrijednost pozicije loptice na gredi uzimajući u obzir da je referentna točka na sredini grede. S obzirom da je simulacija u stvarnom vremenu, ta referentna vrijednost se postavlja za vrijeme simulacije

Idući je korak, postavljanje zahtjeva za sintezu regulatora. Korisnik zadaje maksimalno dopušteno vrijeme porasta i maksimalno nadvišenje. Na temelju tih vrijednosti generiran je regulator s faznim prethodnjem koji će zadovoljiti te zahtjeve.

Klikom na dugme *Simulation* započinje simulacija u stvarnom vremenu. U pozadini se određuje prijenosna funkcija procesa, regulatora i pojačanje te ispisuje na sučelju. Klikom na dugme *Stop stimulation* završava simulacija te se u skupini *Step information* prikazuju informacije o dinamičkim pokazateljima kakvoće. Od dinamičkih pokazatelja kakvoće, najvažniji su maksimalno nadvišenje i vrijeme porasta. Iz tih vrijednosti se može zaključiti zadovoljava li regulator korisnikove zahtjeve.

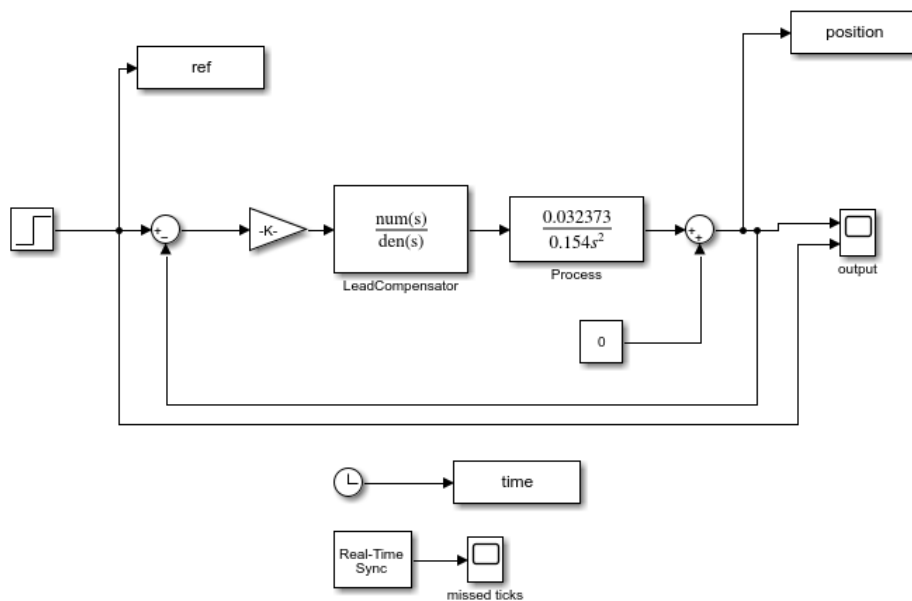
Kartica *Analysis* prikazana je na slici 5.3. te prikazuje tri grafa, odnosno Bodeov, Nyquistov i krivulju mjesta korijena.



Slika 5.3. Sučelje aplikacije za analizu sustava

### 5.3. Simulink model

Za potrebe simulacije, napravljen je model procesa u *Simulinku*. Slika 5.4. prikazuje zatvoreni regulacijski krug procesa upravljanja pozicijom loptice na gredi. Članovi polinoma prijenosnih funkcija te pojačanje mijenjaju vrijednosti na temelju zadanih vrijednosti parametara procesa te zahtjeva za sintezu regulatora. *Real-Time Sync* blok služi za simulaciju u stvarnom vremenu.



Slika 5.4. Simulink model procesa

### 5.4. Kôd aplikacije

U potpoglavlju 5.2. prikazano je sučelje aplikacije, a sada slijedi programsko rješenje problema. U nastavku slijedi prikaz kôda aplikacije te objašnjenje.

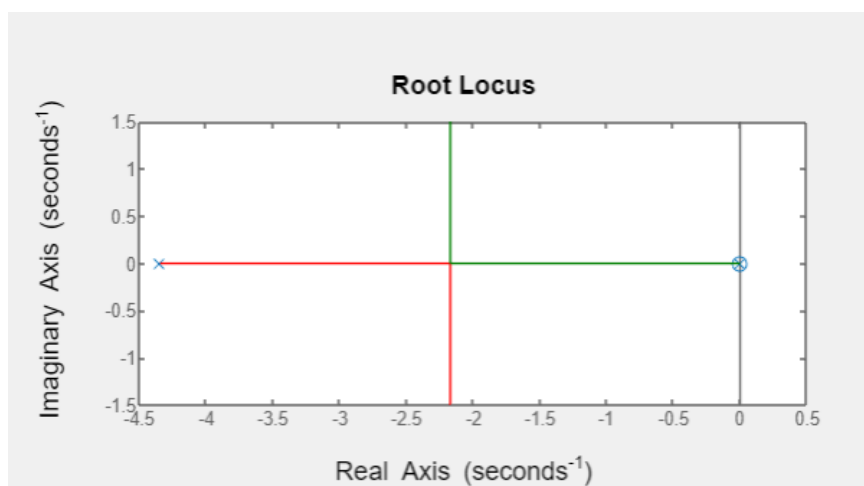
Na početku su definirana svojstva korištena u funkcijama. To su prijenosna funkcija regulatora, procesa i pojačanje. Njima se može pristupiti u funkcijama preko objekta *app*. Svojstva aplikacije prikazane su na slici 5.5.

## Linija Kod

```
1: properties (Access = private)
2:   Gr; %transfer function of a regulator
3:   Gs; % transfer function of a process
4:   Kr; %gain of a regulator
end
```

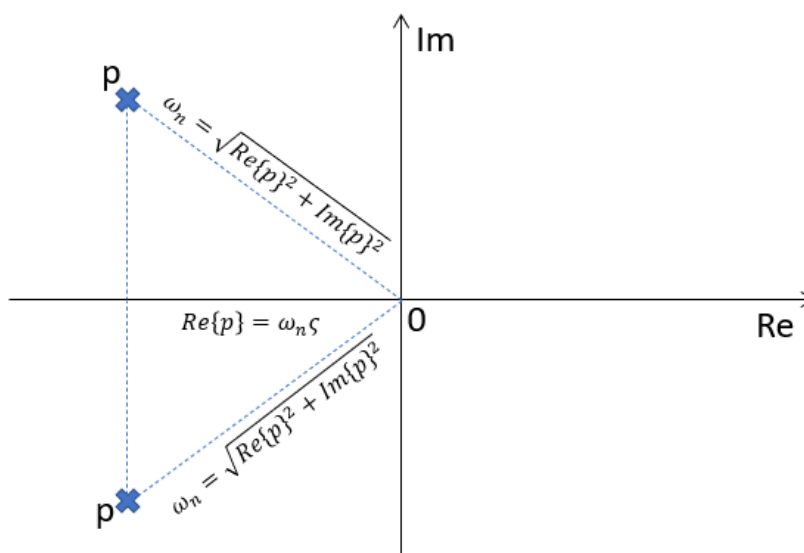
Slika 5.5. Definirana svojstva

Za stabilizaciju procesa odabran je regulator s faznim prethođenjem. On mora biti tako projektiran da bi zadovoljio tražene zahtjeve korisnika. Korisnik zadaje koliko smije biti maksimalno vrijeme porasta i maksimalno nadvišenje. Nula regulatora uvijek će biti pozicionirana uz ishodište kompleksne ravnine, odnosno bit će jednaka 0.001 kako bi poništila jedan od polova procesa u ishodištu. Nula se neće postavljati točno u ishodište jer bi to predstavljao čisti derivacijski član kojeg se ne može realizirati u stvarnim sustavima. Što se tiče pola regulatora, postavlja se na temelju prirodne frekvencije  $\omega_n$  iz izraza (4-4). U tu je svrhu najprije izračunat koeficijent prigušenja sustava iz izraza (4-5). Nakon što je izračunata frekvencija, pol se postavlja na dvostruko većoj vrijednosti. Na taj je način osigurano da vrijeme porasta nikada ne bude veće od zahtjevanog vremena porasta. To se postiglo na način da se tim izborom pola i nule regulatora pomakne krivulja mjesta korijena na lijevu stranu kompleksne ravnine bez promjene kuta asimptota krivulje i pozicioniranjem točke grananja na vrijednost jednako dobivenoj frekvenciji iz izraza (4-4). Pojačanje regulatora će pomicati polove zatvorenog regulacijskog kruga pri čemu će se dobiti frekvencija veća od izračunate, a time i manje vrijeme porasta. Za lakšu je vizualizaciju prikazana krivulja mjesta korijena pri čemu se može primijetiti položaj nule i pola regulatora, pomak krivulje te točka grananja (Slika 5.6.).



Slika 5.6. Krivulja mjesta korijena otvorenog regulacijskog kruga

Pojačanje regulatora određuje se na temelju krivulje mjesta korijena. Pojačanje regulatora utječe na maksimalno nadvišenje sustava, a kako pozicija polova zatvorenog regulacijskog kruga utječu na vrijednost pojačanja, potrebno je odrediti poziciju pola u svrhu dobivanja željenog maksimalnog nadvišenja. Prirodna frekvencija  $\omega_n$  predstavlja udaljenost pola od ishodišta, a koeficijent prigušenje sustava pomnožen sa tom frekvencijom predstavlja realni dio pola. Slika 5.7. prikazuje primjer kompleksnog konjugiranog pola u kompleksnoj ravnini, prirodnu frekvenciju  $\omega_n$  i koeficijent prigušenje sustava  $\zeta$ .



Slika 5.7. odnos pozicije pola u kompleksnoj ravnini s koeficijentom prigušenja i prirodnom frekvencijom sustava

Imajući na umu da je već izračunato prigušenje na temelju najvećeg dopuštenog maksimalnog nadvišenja, pronalazi se onaj pol koji će rezultirati prigušenjem većim od izračunatog jer većim prigušenjem se osigurava manje maksimalno nadvišenje. Ovo se postiže funkcijom *find* koja pronalazi prvi pol iz krivulje mjesta korijena koje zadovoljava taj uvjet. Funkcija koja pronalazi parametre regulatora prikazana je na slici 5.8.

## Linija Kod

```
5:     function calculateParameters(app,m,R,d,L,g,J,sigma,rt)
6:         %calculating damping based on the overshoot
7:         zeta=sqrt((log(sigma/100))^2/(pi^2+(log(sigma/100))^2));
8:         %calculating frequency based on the rising time (10% - 90%)
9:         omega=pi/(2*rt*sqrt(1-zeta^2));
10:        s = tf('s');
11:        app.Gs=-m*g*d/(L*(J/R^2 + m)*s^2);
12:        zo=0.001;
13:        po=omega*2;
14:        app.Gr=(s+zo)/(s+po);
15:        Go=app.Gr*app.Gs; %open loop transfer function
16:        [poles,gains]=rlocus(Go);
17:        %finding the index of the pole with damping > zeta
        index =
        find(abs(real(poles))./sqrt((real(poles)).^2+(imag(poles)).^2) >
        zeta & abs(real(poles))./sqrt((real(poles)).^2+(imag(poles)).^2) <
18:        (zeta+0.1), 1);
19:        pole=poles(index);
20:        sizeofArray=size(poles,1);
```

Slika 5.8. Funkcija koja određuje prijenosnu funkciju procesa i regulatora

Nakon što su uneseni parametri procesa i zahtjevi za sintezu regulatora, pritiskom na gumb *Simulate*, poziva se funkcija *SimulateButtonPushed* čiji je kod prikazan na slici 5.9. Ta funkcija prosljeđuje unešene parametre i zahtjeve za sintezu regulatora već spomenutoj funkciji *CalculateParameters*. Također poziva funkcije *displayInfo*, *calculateStepInfo*, *plotDiagrams*, *startSimulation* koje će redom biti objašnjene.

## ***Linija***    ***Kod***

```
21:     function SimulateButtonPushed(app, event)
22:         cla(app.UIAxes);
23:         m = app.ballmasskgEditField.Value; %mass
24:         R = app.ballradiusmEditField.Value; %radius
25:         d = app.leverarmoffsetmEditField.Value; %lever arm offset
26:         L = app.lengthofthebeammEditField.Value; %beam length
27:         g = -9.81;
28:         J=2*m*R*R/5; % moment of inertia for homogeneous filled sphere [kg
m * m]
29:         r0 = app.ballpositionmEditField.Value; %ball position
30:         rv = app.ReferencevaluemSlider.Value; %reference value
31:
32:         sigma = app.maxovershootEditField.Value; %max overshoot
33:         rt = app.maxrisingtimesEditField.Value; %max rising time
34:
35:         calculateParameters(app,m,R,d,L,g,J,sigma,rt);
36:         displayInfo(app)
37:         calculateStepInfo(app);
38:         plotDiagrams(app);
39:         startSimulation(app,r0,rv)
40:     end
```

Slika 5.9. Kôd funkcije koja započinje simulaciju

Funkcija `displayInfo`, prikazana na slici 5.10., prikazuje prijenosne funkcije procesa i regulatora te pojačanje na sučelju aplikacije pomoću simboličke varijable. Najprije je dobiven simbolički polinom iz funkcije *poly2sym*. Taj polinom je potom interpretiran funkcijom *latex*.

**Linija**    **Kod**

```
41:     function displayInfo(app)
42:         %display transfer functions and gain on a label
43:         [numGr,denGr]=tfdata(app.Gr);
44:         syms s
45:         Gr_sym=poly2sym(cell2mat(numGr),s)/poly2sym(cell2mat(denGr),s);
46:         Gr_text=latex(vpa(Gr_sym,4));
47:         app.GrTrLabel.Text=sprintf("$$%s$$",Gr_text);
48:         app.GrTrLabel.Interpreter='latex';
49:
50:         [numGs,denGs]=tfdata(app.Gs);
51:         syms s
52:         Gs_sym=poly2sym(cell2mat(numGs),s)/poly2sym(cell2mat(denGs),s);
53:         Gs_text=latex(vpa(Gs_sym,5));
54:         app.GsTrLabel.Text=sprintf("$$%s$$",Gs_text);
55:         app.GsTrLabel.Interpreter='latex';
56:
57:         app.KrvalueLabel.Text=num2str(app.Kr);
58:
59:     end
```

Slika 5.10. Kôd funkcije koja ispisuje prijenosne funkcije i pojačanje

Nadalje, funkcija `calculateStepInfo` daje informaciju o prijelaznom odzivu sustava, odnosno vrijeme porasta, vrijeme ustaljivanja i maksimalno nadvišenje. Za korisnika su važne informacije o maksimalnom nadvišenju i vremenu porasta te na temelju tih informacija, provjereno je podudaranje dobivenih rezultata sa korisnikovim zahtjevima. Kôd funkcije prikazan je na slici 5.11.

**Linija**    **Kod**

```
60:     function calculateStepInfo(app)
61:         Gx = feedback(app.Gr*app.Kr*app.Gs,1);
62:         info = stepinfo(Gx);
63:         app.risetimeEditField.Value = info.RiseTime;
64:         app.settlingtimeEditField.Value = info.SettlingTime;
65:         app.peakEditField.Value = info.Peak;
66:         app.peaktimeEditField.Value=info.PeakTime;
67:         app.overshootEditField.Value=info.Overshoot;
68:     end
```

Slika 5.11. Kôd funkcije koja daje informacije o prijelaznom odzivu



Funkcija `plotDiagrams`, prikazana na slici 5.12., služi za analizu sustava te prikazuje krivulju mjesta korijena, bodeov i nyquistov graf.

**Linija    Kod**

```
69:     function plotDiagrams(app)
70:         rlocus(app.rlocus, app.Gr*app.Gs);
71:         nyquist(app.nyquist, app.Kr*app.Gr*app.Gs);
72:         bode(app.bode, app.Kr*app.Gr*app.Gs);
73:     end
```

Slika 5.12. Kôd funkcije koja prikazuje krivulju mjesta korijena, bodeov i nyquistov graf

Funkcijom `startSimulation` učitava se simulink model prikazan na slici 5.4. Funkcija postavlja vrijednosti blokova zadanim vrijednostima parametara procesa i regulatora. Potom, otvara se scope blok koji prikazuje odziv sustava u stvarnom vremenu. Kôd funkcije prikazan je na slici 5.13.

**Linija    Kod**

```
74:     function startSimulation(app, r0, rv)
75:         [numGr, denGr] =tfdata(app.Gr);
76:         [numGs, denGs]=tfdata(app.Gs);
77:         numGr=cell2mat(numGr);
78:         denGr=cell2mat(denGr);
79:         numGs=cell2mat(numGs);
80:         denGs=cell2mat(denGs);
81:         %Loading model in the memory and setting the parameters
82:         load_system('ballAndBeamModel')
83:         set_param('ballAndBeamModel/Step', 'After', num2str(rv));
84:         set_param('ballAndBeamModel/Gain', 'Gain', num2str(app.Kr));
85:         set_param('ballAndBeamModel/LeadCompensator', 'Numerator',
86:         mat2str(numGr));
87:         set_param('ballAndBeamModel/LeadCompensator', 'Denominator',
88:         mat2str(denGr) );
89:         set_param('ballAndBeamModel/Process', 'Numerator',mat2str(numGs));
90:         set_param('ballAndBeamModel/Process', 'Denominator',
91:         mat2str(denGs) );
92:         set_param('ballAndBeamModel/Constant','Value', num2str(r0));
93:         set_param('ballAndBeamModel', 'SimulationCommand', 'start')
94:         %open the scope to view simulation
95:         open_system('ballAndBeamModel/output')
96:     end
```

Slika 5.13. Kôd funkcije koja učitava simulink model i započinje simulaciju

Omogućeno je mijenjanje referentne veličine tijekom simulacije funkcijama prikazane na slici 5.14. Kad god je promijenjena referentna veličina, ažuriraju se vrijednosti u *Simulink* modelu.

**Linija**    **Kod**

```
94:     function ReferencevalueSliderValueChanging(app, event)
95:         changingValue = event.Value;
96:         %updating simulink model parameters during simulation
97:         app.ReferencevalueEditField.Value = changingValue;
98:         set_param('ballAndBeamModel/Step', 'After',
99:                 num2str(changingValue));
100:     end
101:     function ReferencevalueEditFieldValueChanged(app, event)
102:         value = app.ReferencevalueEditField.Value;
103:         %setting ref value in the numeric field (insted of slider)
104:         %checking that the set value is between limits.
105:         if value < app.ReferencevaluemSlider.Limits(1)
106:             value = app.ReferencevaluemSlider.Limits(1);
107:             app.ReferencevalueEditField.Value = value;
108:         end
109:         if value > app.ReferencevaluemSlider.Limits(2)
110:             value = app.ReferencevaluemSlider.Limits(2);
111:             app.ReferencevalueEditField.Value = value;
112:         end
113:         %update slider and parameter in simulink model
114:         app.ReferencevaluemSlider.Value = value;
115:         set_param('ballAndBeamModel/Step', 'After', num2str(value));
116:     end
```

Slika 5.14. Kôd funkcija koje mijenjaju vrijednosti referentne veličine

Klikom na dugme *Stop simulation* završava simulacija te su podatci potrebni za prikaz rezultata na sučelju dohvaćeni pomoću funkcije *evalin*. Kôd funkcije za zaustavljanje simulacije prikazan je na slici 5.15.

***Linija***    ***Kod***

```
117: function StopsimulationButtonPushed(app, event)
118:     set_param('ballAndBeamModel', 'SimulationCommand', 'stop')
119:     %get structure position from simulink, from workspace
120:     position=evalin('base','position');
121:     ref=evalin('base','ref');
122:     %plot the result of simulation
123:     plot(app.UIAxes,position.time,position.signals.values);
124:     plot(app.UIAxes,ref.time,ref.signals.values);
125: end
```

Slika 5.15. Kôd funkcije za zaustavljanje simulacije

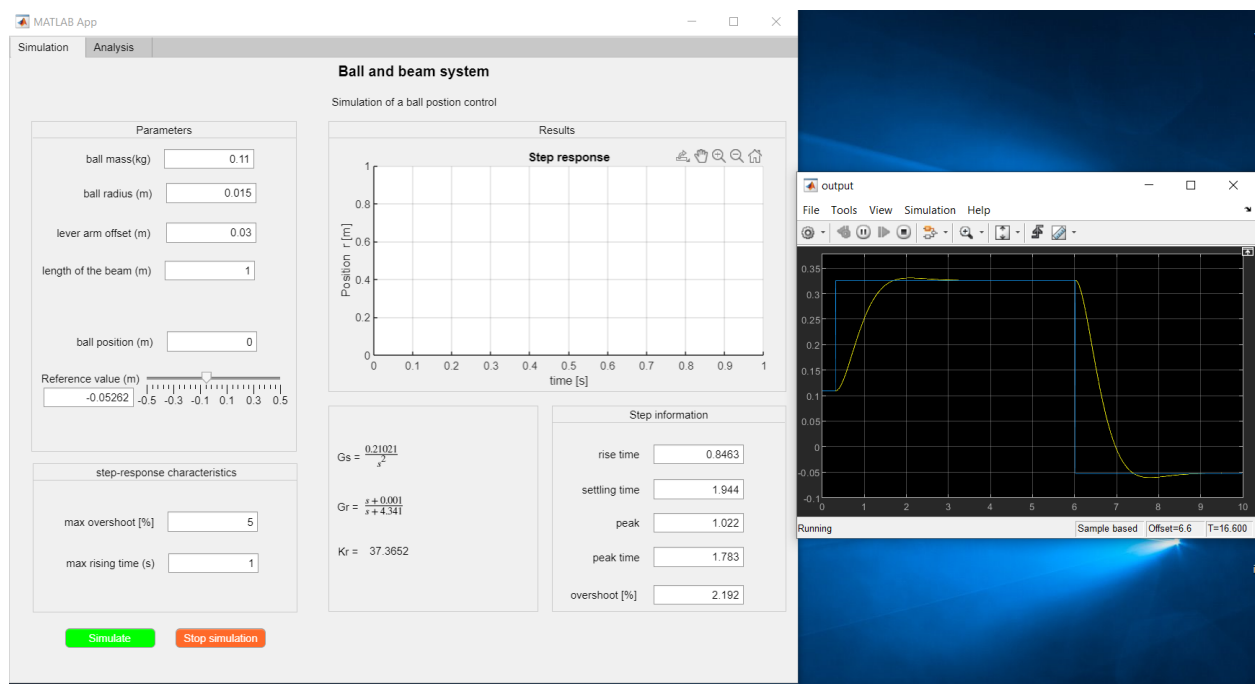
## 6. SIMULACIJA I ANALIZA REZULTATA

U ovom poglavlju prikazan je primjer korištenja aplikacije. Nakon što su unešeni parametri procesa i zahtjevi za sintezu regulatora, simulacija je pokrenuta i može se analizirati sustav. Tablica 6.1. prikazuje vrijednosti parametara procesa i tražene zahtjeve.

Tablica 6.1. Parametri procesa i regulator

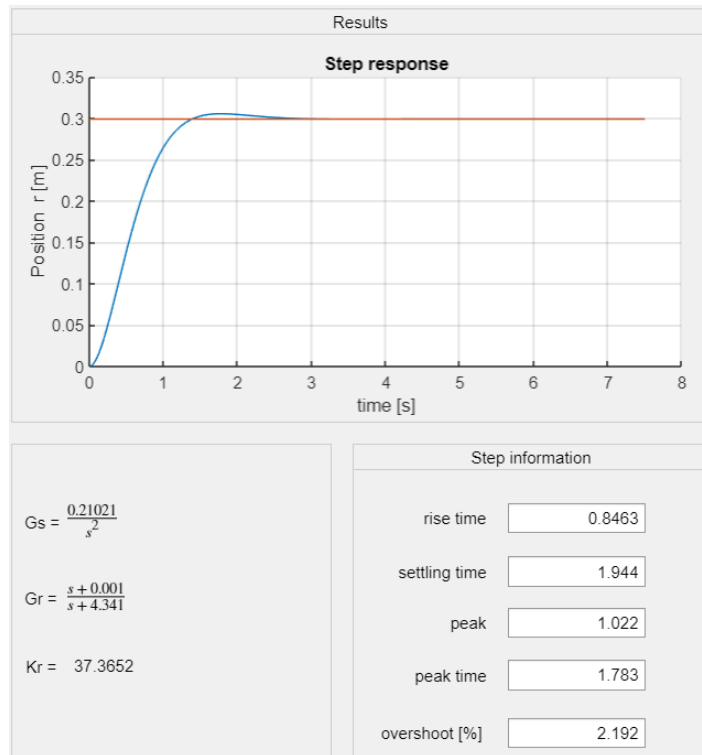
Parametar	Vrijednost	Mjerna jedinica
Masa loptice	0.11	kg
Polumjer loptice	0.015	m
Pomak kraka poluge	0.03	m
Duljina grede	1	m
Pozicija loptice	0	m
Maksimalno nadvišenje	<5	%
Vrijeme porasta	<1	s

Nakon pritiska na dugme *Start simulation*, zadaje se referentna pozicije loptice na gredi te prati odziv sustava u otvorenom *scope* bloku. Primjer korištenja sučelja prikazan je na slici 6.1.



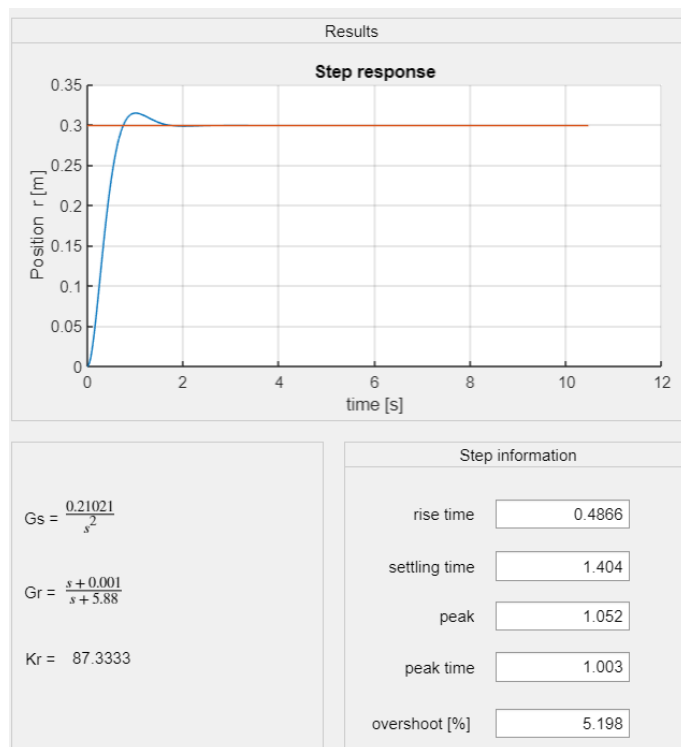
Slika 6.1. Simulacija upravljanja pozicijom loptice na gredi

Iz dobivenog prijelaznog odziva i dinamičkih pokazatelja kakvoće pod skupinom *Step information* (slika 6.2.) provjerava se ponašanja sustava. Dobiveno je vrijeme porasta 0.8463 sekundi, a maksimalno nadvišenje 2.192 % čime su zadovoljeni traženi zahtjevi za sintezu regulatora.



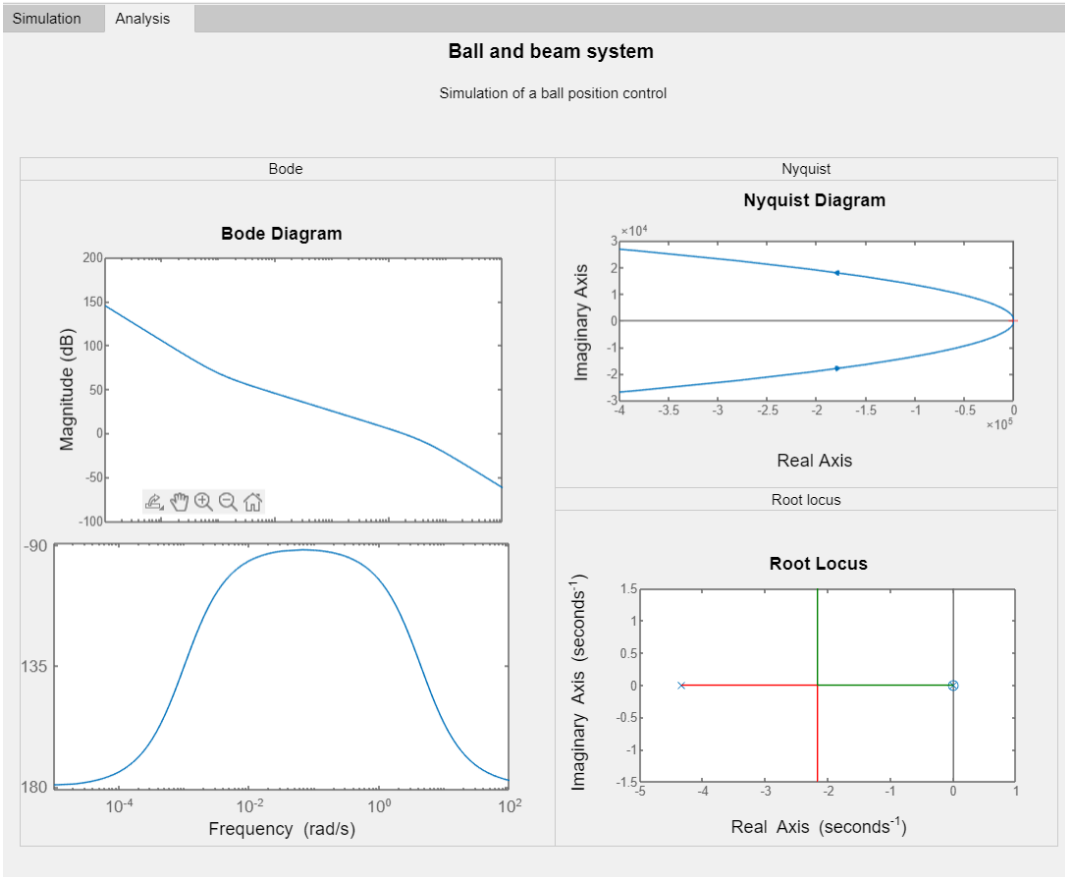
Slika 6.2. Simulacija upravljanja pozicijom loptice na gredi

Simulacija je ponovljena uz iste vrijednosti parametara procesa, no uz nešto drugačije zahtjeve. Zahtjev je da maksimalno nadvišenje bude manje od 7 %, a vrijeme porasta manje od 0.7 sekundi. Dobiveni su rezultati prikazani su na slici 6.4. te su zahtjevi korisnika ponovno zadovoljeni



Sl. 6.4. Simulacija upravljanja pozicijom loptice na gredi

Klikom na karticu *Analysis* korisnik može proučiti Bodeov, Nyquistov dijagram te krivulju mjesta korijena. Na slici 6.5. se može primjetiti utjecaj regulatora s faznim prethođenjem na sustav. Regulator će dovesti do uzdizanja fazno-frekvencijske karakteristike u određenom frekvencijskom području.



Sl. 6.5. Krivulje mjesta korijena, Nyquistov i Bode dijagram

## 7. ZAKLJUČAK

Cilj ovog rada bio je napraviti analizu i sintezu sustava upravljanja pozicijom loptice na gredi. Prije same analize i sinteze sustava, potrebno je bilo opisati proces matematičkim modelom. Iz matematičkog modela zaključena je nestabilnost procesa te se moglo zaključiti da je potrebno stabilizirati sustav zatvorenim regulacijskim krugom. U tu je svrhu odabran regulator s faznim prethođenjem. Zadavanjem određenih zahtjeva za sintezu takvog regulatora, postigla se tražena dinamika sustava. Na temelju maksimalnog nadvišenja i vremena porasta moglo se upravljati brzinom sustava što je bilo i testirano simulacijom odziva u korisničkom sučelju. Simulacijom se pokazalo da su traženi zahtjevi za upravljanje sustavom ispunjeni.



## LITERATURA

- [1] Z. Vrhovski, *Automatsko upravljanje*, 1. izd. Bjelovar: Croatiagraf d.o.o., 2013.
- [2] B. Meenakshipriya i K. Kalpana, „Modelling and Control of Ball and Beam System using Coefficient Diagram Method (CDM) based PID controller“, *Elsevier Sci. Direct - IFAC Pap.*, sv. 47, izd. 1, str. 620–626, ožujak 2014.
- [3] M. Ding, B. Liu, i L. Wang, „Position control for ball and beam system based on active disturbance rejection control“, *Inf. UK Ltd.*, sv. 7, izd. 1, str. 97–108, siječanj 2019.
- [4] D. Tilbury i B. Mesner, „Control Tutorials for MATLAB & Simulink“. Pristupljeno: 15. siječanj 2022. [Na internetu]. Dostupno na:  
<https://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SimulinkModeling>
- [5] „Hrvatska enciklopedija - moment inercije“. Leksikografski zavod Miroslav Krleža. Pristupljeno: 18. siječanj 2022. [Na internetu]. Dostupno na:  
<https://www.enciklopedija.hr/natuknica.aspx?ID=69680>
- [6] N. Perić, *Automatsko upravljanje*. Zagreb: Skriptarnica, FER, 1998.
- [7] „Lead Compensator“. ScienceDirect, 2018. Pristupljeno: 20. travanj 2022. [Na internetu]. Dostupno na: <https://www.sciencedirect.com/topics/engineering/lead-compensator>
- [8] W. Bolton, „Percentage Overshoot - Control Systems“. ScienceDirect, 2002. [Na internetu]. Dostupno na: <https://www.sciencedirect.com/topics/engineering/percentage-overshoot>
- [9] „MATLAB App Designer“. The MathWorks, Inc. Pristupljeno: 20. travanj 2022. [Na internetu]. Dostupno na: <https://www.mathworks.com/products/matlab/app-designer.html>

## SAŽETAK

Glavni je zadatak ovog rada analiza i sinteza sustava upravljanja korištenjem MATLAB programa. Najprije je proveden opis procesa koji uključuje određivanje matematičkog modela za zadani proces. Nakon što je bila utvrđena nelinearnost procesa izvršena je linearizacije istog. Potom je slijedila analiza procesa grafoanalitičkom metodom, odnosno krivuljom mjesta korijena. Analizom je utvrđena nestabilnost procesa, stoga je bilo potrebno projektirati takav regulator koji će ga stabilizirati. Za regulator odabran je regulator s faznim prethođenjem. Sinteza regulatora se provodi na temelju zahtjeva korisnika unešenih u korisničko sučelje izgrađenog u *App Designeru*. U konačnici, izvršena je simulacija kojom se pokazalo primjer korištenja sučelja te je kroz dva primjera pokazano kako su zadovoljena tražena dinamička vladanja.

Ključne riječi: analiza, App Designer, krivulja mjesta korijena, matematički model, MATLAB, regulator s faznim prethođenjem, simulacija, sinteza.

## **ABSTRACT**

Title: Analysis and Ball Position Control of a Ball and Beam System

The main task of this paper is the analysis and synthesis of a control system using MATLAB program. Firstly, a description of the process was made, which included determining a mathematical model for the given process. After nonlinearity of the process was determined, its linearization was performed. Then followed the analysis of the process using graphoanalytical method, ie the root locus diagram. The analysis confirmed the instability of the process, so it was necessary to design such a controller to stabilize it. A phase lead controller was selected. The design of the controller is performed based on the user requirements entered in the user interface built in the App Designer. Finally, a simulation was performed which showed an example of using the interface and showed through two examples how the dynamic requirements of the control system were met.

Key words: analysis, App Designer, root locus diagram, mathematical model, MATLAB, phase lead controller, simulation, synthesis