

Sustav upravljanja i nadzora procesa skladištenja fluida

Mikulić, Matija

Master's thesis / Diplomski rad

2024

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:200:675593>

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

Download date / Datum preuzimanja: **2024-11-20**

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 Računarstvo

**SUSTAV UPRAVLJANJA I NADZORA PROCESA
SKLADIŠTENJA FLUIDA**

Diplomski rad

Matija Mikulić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju****Ocjena diplomskog rada na sveučilišnom diplomskom studiju**

Ime i prezime pristupnika:	Matija Mikulić
Studij, smjer:	Sveučilišni diplomski studij Računarstvo
Mat. br. pristupnika, god.	D1310R, 07.10.2022.
JMBAG:	0165082211
Mentor:	prof. dr. sc. Robert Cupec
Sumentor:	
Sumentor iz tvrtke:	Damir Bašić
Predsjednik Povjerenstva:	prof. dr. sc. Dražen Slišković
Član Povjerenstva 1:	prof. dr. sc. Robert Cupec
Član Povjerenstva 2:	izv. prof. dr. sc. Damir Filko
Naslov diplomskog rada:	Sustav upravljanja i nadzora procesa skladištenja fluida
Znanstvena grana diplomskog rada:	Procesno računarstvo (zn. polje računarstvo)
Zadatak diplomskog rada:	Provesti sintezu sustava upravljanja procesom skladištenja fluida u prostoru stanja s estimatorom varijabli stanja. Izraditi servisnu aplikaciju koja omogućuje dvosmjernu komunikaciju između računala (PC) i upravljačkog programa logičkog upravljača (PLC) radi prikupljanja i slanja podataka te njihovo pohranjivanje u bazu podataka. Izraditi aplikaciju za vizualizaciju cijelog sustava. Tema rezervirana za: Matija Mikulić Sumentor iz tvrtke: Damir Bašić
Datum ocjene pismenog dijela diplomskog rada od strane mentora:	09.09.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	24.09.2024.
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	24.09.2024.

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

Osijek, 24.09.2024.

Ime i prezime Pristupnika:

Matija Mikulić

Studij:

Sveučilišni diplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

D1310R, 07.10.2022.

Turnitin podudaranje [%]:

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Sustav upravljanja i nadzora procesa skladištenja fluida**

izrađen pod vodstvom mentora prof. dr. sc. Robert Cupec

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 pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak diplomskog rada.....	2
2. MOTIVACIJA ZA UPRAVLJANJA RAZINOM TEKUĆINE U SPREMNICIMA: PRIMJENA U INDUSTRIJI I POSTOJEĆE RJEŠENJE.....	3
2.1. Primjena u industriji.....	3
2.2. Pilot postrojenje skladištenja tekućine u spremnicima	4
3. MODELIRANJE TEHNIČKOG SUSTAVA	7
3.1. Matematički model sustava skladištenja tekućine.....	8
3.2. Linearizacija procesa.	13
3.2.1. Prikaz u prostoru stanja	13
3.2.2. Prijenosna funkcija.....	15
3.3. Eksperimentalno određivanje parametara procesa.....	16
3.4. Usporedba matematičkog modela sa stvarnim podacima.....	23
3.5. Analiza sustava.....	26
3.6. Analiza mjernog šuma	29
3.7. Procjena parametara modela rekurzivnom metodom najmanjih kvadrata	31
3.8. Sinteza regulatora	34
3.8.1. Sinteza regulatora u prostoru stanja.....	36
4. IZRADA APLIKACIJE ZA SUSTAV SKLADIŠTENJA TEKUĆINE U SPREMNICIMA	40
4.1. Definiranje podataka za razmjenu između PLC-a i servera.....	41
4.1.1. Kružni spremnik.....	43
4.1.2. Podatkovni spremnik.....	44
4.2. Baza podataka	45
4.3. Razvoj komunikacijskog sustava	46
4.3.1. Siemens S7.Net Plus biblioteka	47
4.3.2. Posrednik poruka	48
4.4. Opis i funkcionalnost aplikacijskog programskog sučelja	51
4.5. Izrada Web Aplikacije.....	52

4.5.1. Izgled web aplikacije.....	53
5. RAD APLIKACIJE I REZULTATI REGULACIJE	60
5.1. Prikaz osnovnih funkcionalnosti web aplikacije.....	60
5.2. Rezultati regulacije razine tekućine u spremniku	66
5.3. Rezultati rada rekurzivne metode najmanjih kvadrata.....	71
6. ZAKLJUČAK.....	76
LITERATURA	77
SAŽETAK.....	78
ABSTRACT	79

1. UVOD

U suvremenom svijetu automatizacije, programabilni logički upravljači (engl. *programmable logic controller*, PLC) zauzimaju ključnu ulogu u upravljanju industrijskim procesima. Oni omogućuju automatizaciju složenih industrijskih operacija, povećavaju proizvodnu efikasnost i smanjuju troškove proizvodnje. PLC-ovi pružaju pouzdan i prilagodljiv sustav sposoban za nadzor i regulaciju širokog spektra industrijskih procesa, od proizvodnih linija do elektrana. Cilj ovakvih sustava je minimizirati ljudske intervencije, povećati brzinu izvođenja i sigurnost svih operacija u okviru proizvodnog procesa. [1] Osim optimizacije procesa, automatizacija značajno doprinosi zaštiti zdravlja radnika, eliminirajući potrebu za njihovim izlaganjem opasnim uvjetima rada.

Za složene sustave javlja se potreba i za odgovarajućim sučeljima za nadzor i upravljanje. Riječ je o takozvanim HMI/SCADA sučeljima koja, uz nadzor, operaterima omogućavaju interakciju s PLC-ovima putem grafičkih sučelja. Od izrazite je važnosti da navedena sučelja budu jednostavna i intuitivna kako bi operateri mogli brzo donositi odluke na temelju podataka u stvarnom vremenu. Razvojem tehnologija, javljaju se brži i jači procesori te grafičke kartice koje omogućavaju složene prikaze i simulacije. To je omogućilo razvoj složenijih aplikacija koje mogu obavljati zadatke koje zahtijevaju visoke računalne performanse.

U ovom radu implementirana je aplikacija koja omogućava operateru vizualizaciju sustava skladištenja tekućine u spremnicima laboratorijskog pilot postrojenja. Osim vizualizacije cijelog sustava, aplikacija nudi i određene simulacije rada postrojenja i upravljačkih algoritama na temelju razvijenog modela sustava. Sustav je modeliran u prostoru stanja, pri čemu su parametri modela dobiveni eksperimentalnom analizom. Za upravljanje sustavom korišten je linearni kvadratni regulator (engl. *linear quadratic regulator*, LQR) koji koristi informacije mjernih članova za upravljanje razinom tekućine u spremniku. Također, implementirana je i rekurzivna metoda najmanjih kvadrata (engl. *recursive least squares method*, RLS) koja omogućava identifikaciju parametara sustava u stvarnom vremenu na temelju podataka koji pristižu s postrojenja. Ta metoda omogućuje automatsko prilagođavanje parametara regulatora, čime je osigurana optimalna regulacija razine tekućine u postrojenju.

S obzirom da je predmet upravljanja razina tekućine u spremniku, u drugom poglavlju rada dan je osvrt na takav proces, primjeri iz industrije te slična postojeća rješenja.

U trećem poglavlju rada objašnjena je svrha izrade matematičkog modela tehničkih procesa. Detaljno je opisan proces laboratorijskog postrojenja te je razvijen njegov matematički model. Na temelju razvijenog modela, izvedena je analiza sustava te je objašnjen princip sinteze regulatora.

U četvrtom je poglavlju opisana aplikacija za vizualizaciju i simulaciju postrojenja skladištenja tekućine u spremnicima.

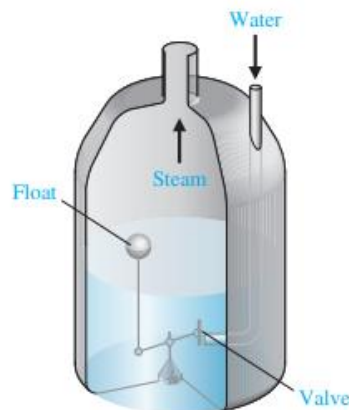
U petom poglavlju prikazan je rad aplikacije i rezultati upravljačkih algoritama.

1.1. Zadatak diplomskog rada

Provesti sintezu sustava upravljanja procesom skladištenja fluida u prostoru stanja. Izraditi servisnu aplikaciju koja omogućuje dvosmjernu komunikaciju između računala (PC) i upravljačkog programa logičkog upravljača (PLC) radi prikupljanja i slanja podataka te njihovo pohranjivanje u bazu podataka. Izraditi aplikaciju za vizualizaciju cijelog sustava.

2. MOTIVACIJA ZA UPRAVLJANJA RAZINOM TEKUĆINE U SPREMNICIMA: PRIMJENA U INDUSTRIJI I POSTOJEĆE RJEŠENJE

Proces automatizacije razine tekućine u spremniku bilježi svoje početke još iz 1765. godine kada je Ruski izumitelj *I. Polzunov* konstruirao prvi regulator za upravljanje razinom vode u kotlu parnog stroja, u sustavu s povratnom vezom. Princip regulacije razine tekućine prikazan je na slici 2.1. Sustav se sastoji od plovka koji se podiže i spušta s razinom vode. Taj plovak je povezan s polugom te se podizanjem plovka postiže mehanizam otvaranja i zatvaranja ventila. Ta akcija regulira dotok vode u spremniku. Kada je dosegnuta određena visina plovka, plovak zatvara ventil kako bi zaustavio dotok vode. [2]



Slika 2.1. Regulator razine tekućine

Izvor: Dorf, R.C., & Bishop, R.H. (2016). *Modern Control Systems* (13th ed.). Pearson

Navedeni sustav bio je revolucionaran jer je omogućio automatsko održavanje konzistentne razine vode, smanjujući potrebu za ručnim nadzorom. Danas se automatizacija razine tekućine koristi u mnogim industrijama. Moderni sustavi koriste naprednije senzore i kontrolne tehnologije za precizno upravljanje razinom tekućine u spremnicima.

2.1. Primjena u industriji

U industrijama u kojima postoje spremnici za skladištenje različitih fluida, vrlo je važno osigurati točno i pouzdano upravljanje njihovom razinom kako ne bi došlo do prelijevanja jer ono može uzrokovati značajnu štetu okolišu, nepotrebne troškove čišćenja i gubitak vrijednih resursa. To je

posebno kritično u industrijama kao što su farmaceutska industrija, proizvodnja energije, naftna i plinska industrija te čelična industrija.

U farmaceutskoj industriji vrlo je važna regulacija razine vode kako bi se osigurala kontinuirana opskrba i održali sterilni uvjeti u procesu proizvodnje. Regulatori razine vode osiguravaju da proizvodnja teče kontinuirano, što je ključno za sigurnost i učinkovitost proizvodnje lijekova.

U proizvodnji energije, voda se koristi za hlađenje u elektranama. Važno je da regulatori održavaju stalnu opskrbu vodom kako bi se spriječilo pregrijavanje i osigurao siguran rad elektrane.

U naftnoj i plinskoj industriji, voda se koristi u procesima bušenja i frakcioniranja. Senzori za mjerenje razine vode ugrađeni su u spremnike kako bi se osigurala stabilna opskrba tekućine za ove procese. Osim što omogućuju neprekidnost procesa, ovi senzori također omogućuju precizno praćenje i automatsku regulaciju razine u spremnicima, što smanjuje otpad i optimizira operativne troškove.

U čeličnoj industriji voda se koristi za hlađenje, čišćenje i transport materijala u proizvodnim pogonima. Regulatori razine vode su neophodni kako bi se osigurala optimalna razina u spremnicima, omogućavajući tako neprekidan proces proizvodnje čelika bez gubitka vode. [3]

2.2. Pilot postrojenje skladištenja tekućine u spremnicima

U suradnji s tvrtkom *Danieli Systec*, izrađeno je laboratorijsko postrojenje (slika 2.2.) za skladištenje tekućine.

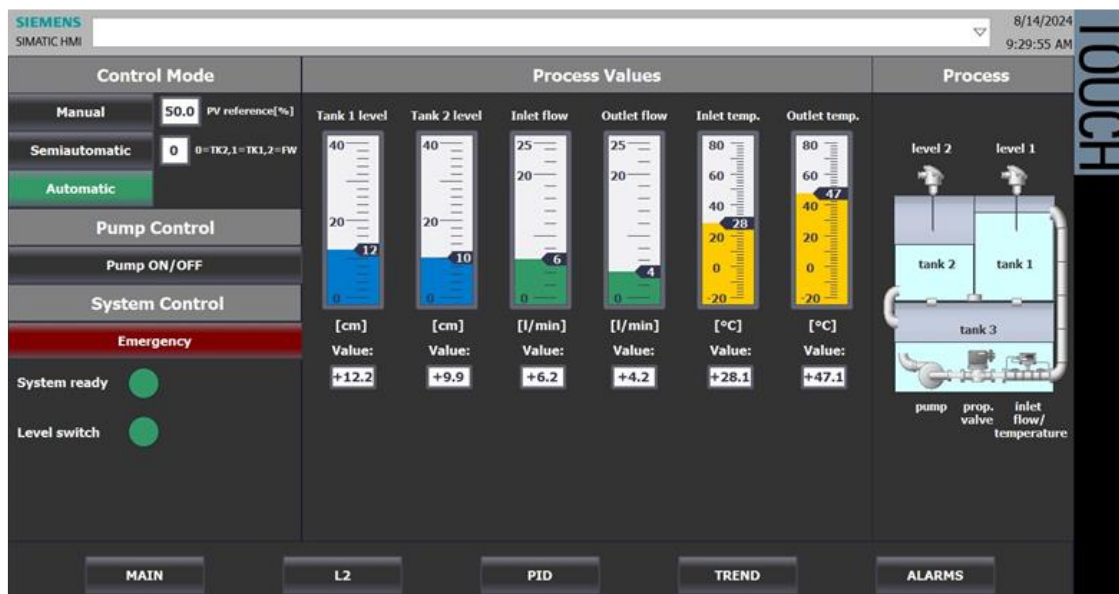


Slika 2.2. Pilot postrojenje za sustav skladištenja tekućine

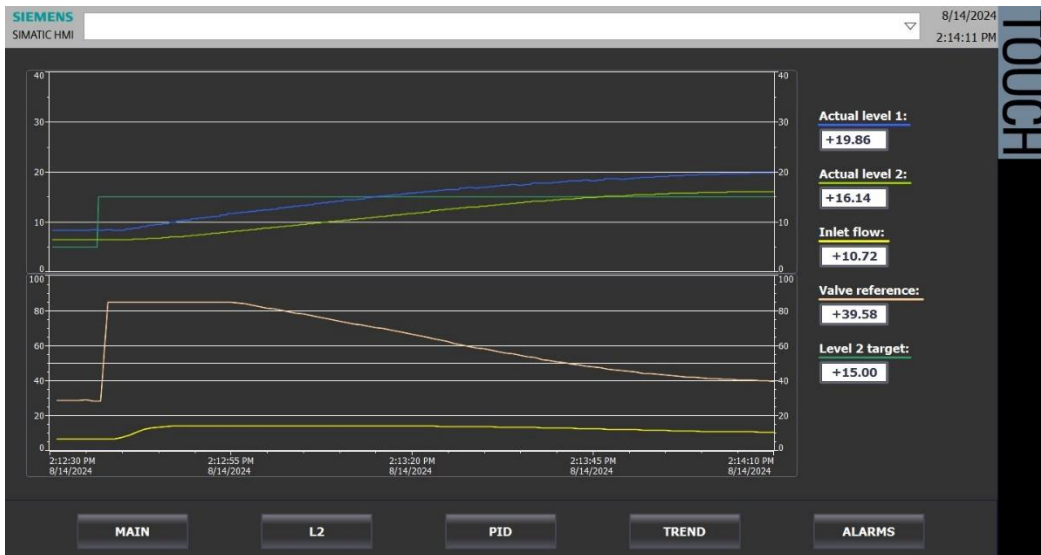
Za taj je sustav korišten *Siemensov* PLC 1500 na kojem je implementiran upravljački program. Također, postoji i HMI sučelje koje pruža operateru jednostavne funkcionalnosti poput vizualizacije svih signala sustava u stvarnom vremenu i odabir vrste regulacije. Slike 2.3., 2.4. i 2.5. prikazuju izgled sučelja i njegove funkcionalnosti. Međutim, HMI ima svoja ograničenja u pogledu:

- detaljne analize sustava (kakvoća regulacije),
- simulacije cijelog procesa,
- trajnog arhiviranja podataka,
- složenijeg prilagođavanja upravljačkih algoritama.

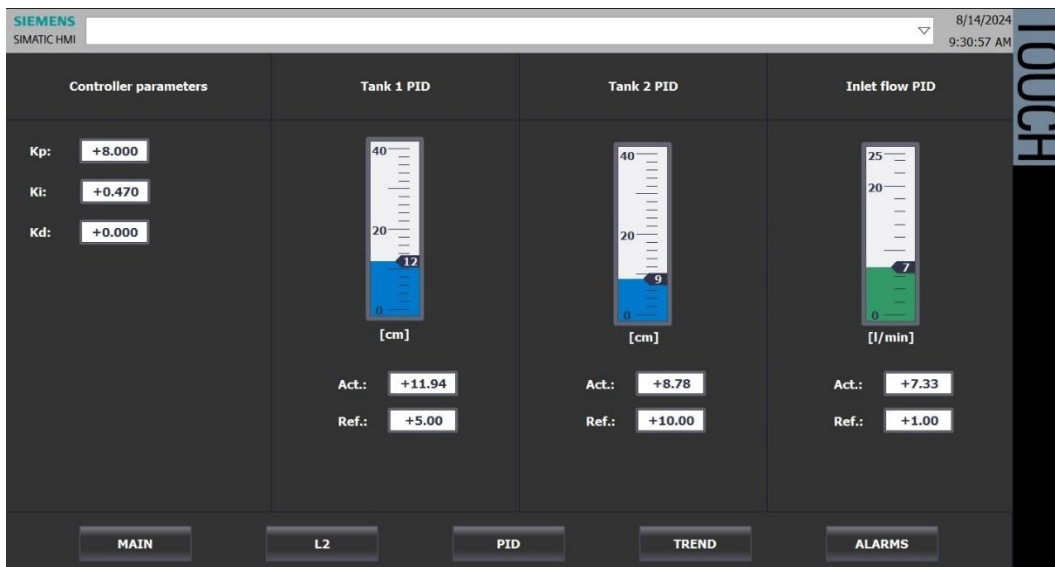
Zbog tih je ograničenja u ovom radu napravljena aplikacija koja pruža te dodatne funkcionalnosti, a sam opis i izgled te aplikacije nalazi se u četvrtom poglavlju.



Slika 2.3. Sučelje koje prikazuje trenutno stanje sustava



Slika 2.4. Sučelje koje prikazuje sve procesne veličine



Slika 2.5. Sučelje koje prikazuje aktivnu regulaciju

3. MODELIRANJE TEHNIČKOG SUSTAVA

U suvremenom svijetu, automatizacija igra ključnu ulogu u optimizaciji tehničkih sustava. Njen primarni cilj je unaprijediti funkcionalnost tih sustava, smanjiti potrebu za ljudskom intervencijom i podići razinu efikasnosti u proizvodnim procesima. Tehnički sustavi koji se automatiziraju su po svojoj prirodi složeni. Sastoje se od različitih dijelova koja su međusobno povezana u prostoru i vremenu na složen način, djelujući u skladu s raznim fizikalnim i kemijskim zakonima. Automatizacija uvodi novu dimenziju u ove sustave, transformirajući ih u autonomne entitete sposobne za samostalno funkcioniranje. [4]

Bitan korak u izgradnji takvih automatiziranih sustava jest izgradnja njihovog modela. Između ostalog, modeliranje omogućuje:

- analizu unutarnjeg vladanja složenih procesa,
- simulaciju vladanja, dijagnostiku i predviđanje,
- projektiranje sustava upravljanja,
- adaptivno upravljanje,
- projektiranje virtualnog senzora.

U ovom je radu naglasak na analizi, sintezi, simulaciji i predikciji budućih stanja procesa.

U tehničkim sustavima najčešći su matematički modeli koji se sastoje od jedne ili više matematičkih jednadžbi (diferencijalne jednadžbe, prikaz u prostoru stanja, prijenosne funkcije, karakteristike i sl.). Često je riječ o diferencijalnim jednadžbama koje iskazuju promjenu različitih relevantnih procesnih veličina u vremenu i prostoru. Također, matematički modeli izražavaju prirodnu povezanost između tih veličina. Do tih funkcionalnih povezanosti moguće je doći:

- teorijskom analizom
- eksperimentalnom analizom
- kombinacijom teorijske i eksperimentalne analize

Teorijska analiza podrazumijeva dobro poznavanje fizikalno-kemijskih zakona i postavljanje jednadžbi ravnoteže. Navedenom analizom ostvaruje se uvid u unutarnje odnose procesa. S druge strane, potrebna je visoka razina znanja i razumijevanja procesa te su često dobiveni matematički modeli presloženi za daljnje primjene.

Eksperimentalna analiza omogućava stvaranje modela sustava na osnovu stvarnih mjerenja, čime se izbjegava potreba za dubokim teorijskim znanjem o svim fizikalnim i kemijskim aspektima procesa. Ova analiza omogućava lakše određivanje parametara modela u odnosu na teorijsku

analizu koja zahtijeva složene matematičke izvode. Međutim, eksperimentalna analiza daje ograničeno razumijevanje jer ne pruža uvid u fizikalne principe koji upravljaju procesom.

Treći pristup povezuje teorijsku i eksperimentalnu analizu. Kroz teorijsku analizu moguće je odrediti strukturu modela (red diferencijalne jednačbe, nelinearne ovisnosti i sl.), a kroz eksperimentalnu analizu određuju se parametri modela. Upravo je ovakav pristup korišten za opis tehničkog sustava obrađenog u kasnijim dijelovima ovog poglavlja.

Do mjernih podataka moguće je doći kroz:

- provedbu eksperimenata,
- preuzimanjem pogonskih podataka.

Eksperimentima je omogućena izgradnja modela u kontroliranim uvjetima i željenim radnim točkama. Moguće je pobuditi proces na njegovim karakterističnim frekvencijama, uz minimalne utjecaje vanjskih čimbenika i smetnji, što rezultira raznolikim i informativnim skupom podataka za modeliranje.

S druge strane, provođenje eksperimenta može biti financijski i vremenski skupo te se stoga modeliranje često provodi pomoću pogonskih podataka snimljenih kroz dulje vremensko razdoblje za vrijeme normalnog rada postrojenja. Ti podatci su obično manje informativni i u njima se nalazi mnoštvo šuma što iziskuje posebnu potrebu za njihovom predobradom u fazi modeliranja.

U ovom radu provedeni su eksperimenti za određivanje početnih vrijednosti parametara modela procesa, a potom su korišteni podatci, za vrijeme normalnog rada postrojenja, za prilagodbu tih parametara u slučaju promjene uvjeta ili radne točke.

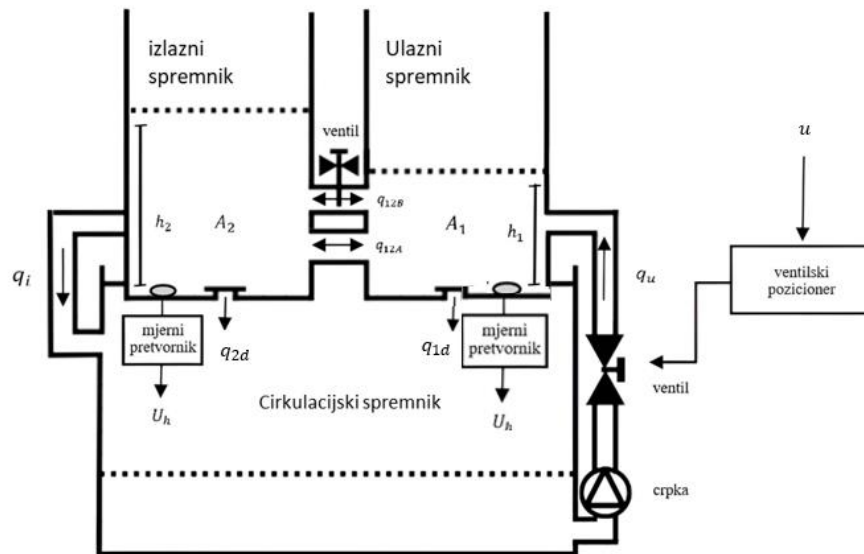
U nastavku slijedi prikaz i opis pilot postrojenja te postavljanje jednačbi koje ga opisuju.

3.1. Matematički model sustava skladištenja tekućine

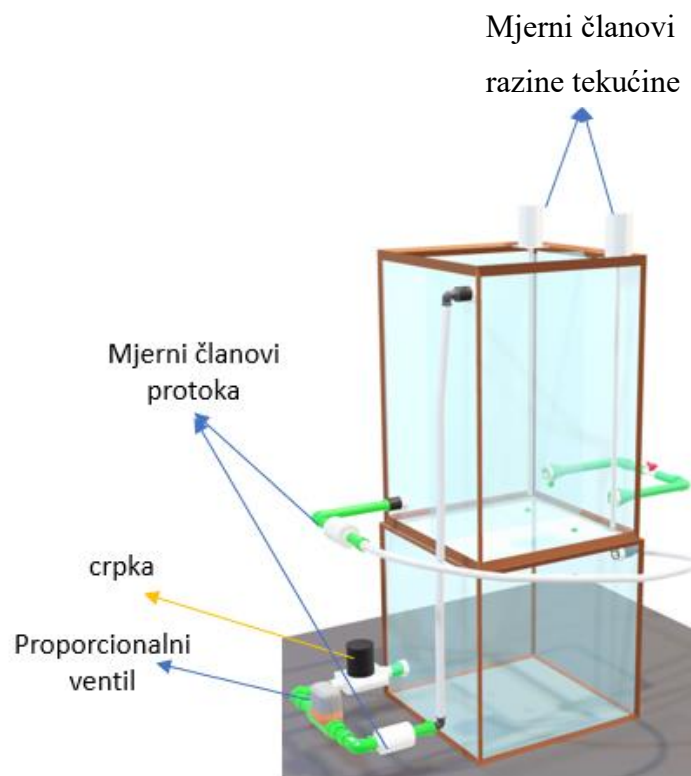
Za bolje razumijevanje sustava i radi lakše izrade matematičkog modela, napravljen je shematski prikaz (slika 3.1.) i 3D model pilot postrojenja (Slika 3.2.). Glavni elementi sustava su:

- Mjerni članovi:
 - dva mjerna člana za mjerenje razine tekućine u spremnicima,
 - dva mjerna člana za mjerenje protoka i temperature tekućine,
 - mjerni član koji detektira nisku razinu tekućine u cirkulacijskom spremniku,
- Izvršni članovi:
 - ventilski pozicioner,

- crpka,
- Ostalo:
 - mehanički ventil između prvog i drugog spremnika.



Slika 3.1. Shema pilot postrojenja



Slika 3.2. 3D model postrojenja

U nastavku slijedi razrada matematičkog modela navedenog postrojenja.

Crpka radi u binarnom načinu rada (isključeno/uključeno) s konstantnom brzinom, a protok tekućine u postrojenju se regulira putem ventilskog pozicionera. Drugim riječima, protok i razina tekućine u spremnicima upravljani su promjenom otvorenosti ventila.

Ovisnost ulaznog toka o otvorenosti ventila aproksimirana je kao PT_1 član¹. Mjerni članovi brzo prate promjene protoka, a kašnjenje senzora protoka je zanemarivo. Diferencijalna jednačba tada glasi:

$$q_u(t) + T_v \frac{dq_u}{dt} = K_v x_v \quad (3-1)$$

gdje je:

T_v – vremenska konstanta ventila,

K_v – konstanta proporcionalnosti ventila,

x_v [%] – otvorenost ventila.

Dinamička svojstva pozicionera su zanemariva u odnosu na dinamiku procesa uskladištenja tekućine, pa se otvorenost ventila o električnoj struji na ulazu pozicionera može opisati sljedećim izrazom:

$$x_v = K_m \cdot u(t) \quad (3-2)$$

gdje je:

K_m – konstanta proporcionalnosti ventilskog pozicionera,

u [mA] – ulazni signal koji se dovodi na ventilski pozicioner (električna struja).

Diferencijalna jednačba koja opisuje promjenu volumena V [m^3] tekućine u spremniku može se općenito izraziti kao:

$$\frac{dV}{dt} = \sum_i \text{ulazni protok}_i - \sum_j \text{izlazni protok}_j \quad (3-3)$$

¹ PT_1 član predstavlja linearni dinamički sustav koji se opisuje linearnom diferencijalnom jednačbom 1. reda.

Budući da ulazni i izlazni spremnik imaju jednaku i nepromjenjivu površinu baze, jednadžbe promjene volumena tekućine u spremnicima glase:

$$A \frac{dh_1}{dt} = q_u - q_{12,A} - q_{12,B} - q_{1d} \quad (3-4)$$

$$A \frac{dh_2}{dt} = q_{12,A} + q_{12,B} - q_i - q_{2d} \quad (3-5)$$

pri čemu je:

$A [m^2]$ – površina baze ulaznog i izlaznog spremnika ,

$h_1 [m]$ – razina tekućina u ulaznom spremniku,

$q_u [m^3/s]$ – ulazni protok,

$q_{12,A}, q_{12,B} [m^3/s]$ – protoci između ulaznog i izlaznog spremnika,

$q_{1d} [m^3/s]$ – protok između ulaznog i cirkulacijskog spremnika,

$h_2 [m]$ – razina tekućina u izlaznom spremniku,

$q_i [m^3/s]$ – izlazni protok,

$q_{2d} [cm^3/s]$ – protok između ulaznog i cirkulacijskog spremnika.

Ulazni i izlazni spremnici su otvoreni i imaju male otvore na dnu kroz koje teče tekućina. Koristeći *Bernoullijevu* jednadžbu, moguće je izvesti izraze za brzinu i protok tekućine:

$$p_1 + \rho gh_1 + \frac{1}{2} \rho v_1^2 = p_2 + \rho gh_2 + \frac{1}{2} \rho v_2^2 \quad (3-6)$$

gdje su:

$p_1, p_2 [Pa]$ – hidrostatski tlakovi tekućine pri vrhu i otvoru spremnika,

$p_a [Pa]$ – atmosferski tlak,

$g [m/s^2]$ – konstanta gravitacijskog ubrzanja,

$h_1, h_2 [m]$ – visina tekućine pri vrhu i otvoru spremnika,

$v_1, v_2 [m/s]$ – brzine tekućine pri vrhu i dnu spremnika.

Pretpostavlja se da je brzina tekućine na vrhu spremnika zanemariva, a tlak na vrhu i na dnu spremnika jednak atmosferskom tlakom. Ova pretpostavka dovodi do pojednostavljene jednadžbe [5]:

$$p_a + \rho gh_1 = p_a + \rho gh_2 + \frac{1}{2}\rho v_2^2. \quad (3-7)$$

Iz jednadžbe (3-7) može se izraziti *Torricellijev* zakon istjecanja tekućine kroz mali otvor. Jednadžbe (3-8) - (3-12) primjenjuju taj zakon za prethodno definirane protoke u jednadžbama (3-4) i (3-5)

$$q_i = K_i \sqrt{2g(h_2 - X)}, \quad (3-8)$$

$$q_{12,A} = K_{12,A} \sqrt{2g(h_1 - h_2)}, \quad (3-9)$$

$$q_{12,B} = K_{12,B} \sqrt{2g(h_1 - h_2)}, \quad (3-10)$$

$$q_{1d} = K_{1d} \sqrt{2gh_1}, \quad (3-11)$$

$$q_{2d} = K_{2d} \sqrt{2gh_2}, \quad (3-12)$$

gdje su:

$K_i, K_{12,A}, K_{12,B}, K_{1d}, K_{2d}$ – konstante istjecanja,

X – visina na kojoj se nalazi cijev izlaznog protoka od mjernog člana.

Konačno, Uvrštavanjem jednadžbi (3-8) - (3-12) u jednadžbe (3-4) i (3-5), dobivaju se sljedeći izrazi:

$$A \frac{dh_1}{dt} = q_u - (K_{12,A} + K_{12,B}) \sqrt{2g(h_1 - h_2)} - K_{1d} \sqrt{2gh_1} \quad (3-13)$$

te

$$A \frac{dh_2}{dt} = (K_{12,A} + K_{12,B}) \sqrt{2g(h_1 - h_2)} - K_i \sqrt{2g(h_2 - x)} - K_{2d} \sqrt{2gh_2}. \quad (3-14)$$

Jednadžbe (3-1), (3-2), (3-13) i (3-14) opisuju dinamičko vladanje procesa skladištenja tekućine. Očigledno je da su jednadžbe, izuzev (3-1) i (3-2), nelinearne stoga je u idućem potpoglavlju objašnjeno kako je moguće pristupiti rješavanju ovih jednadžbi.

3.2. Linearizacija procesa.

U području upravljanja složenim sustavima, linearizacija predstavlja korisnu metodu aproksimacije nelinearnog sustava jednostavnijim linearnim modelom. To se postiže putem aproksimacije nelinearnih funkcija *Taylorovim* redom u blizini odabrane radne točke, obično početnog stanja sustava.

Linearizacija nelinearnih sustava nudi brojne prednosti u pogledu analize i dizajna upravljačkih sustava te simulacije na računalu. Međutim, njezina točnost je ograničena na područje oko radne točke, a zanemarivanje nelinearnih učinaka može dovesti do lošije aproksimacije stvarnog sustava.

3.2.1. Prikaz u prostoru stanja

Slijedi postupak linearizacije matematičkog modela opisanog u poglavlju 3.1.

Neka je zadana radna točka $h_2(t = 0) = h_{20}$, odnosno početna razina tekućine u izlaznom spremniku. Preostale radne točke moguće je dobiti iz sljedećeg sustava jednadžbi:

$$0 = (K_{12,A} + K_{12,B})\sqrt{2g(h_{10} - h_{20})} - K_i\sqrt{2g(h_{20} - x)} - K_{2d}\sqrt{2gh_{20}}, \quad (3-15)$$

$$0 = q_{u0} - (K_{12,A} + K_{12,B})\sqrt{2g(h_{10} - h_{20})} - K_{1d}\sqrt{2gh_{10}} \quad (3-16)$$

$$x_{v0} = \frac{q_{u0}}{K_v}, \quad (3-17)$$

$$u_0 = \frac{x_{v0}}{K_m}. \quad (3-18)$$

Razvijanjem nelinearnih jednadžbi *Taylorovim* redom u okolini radne točke, dobivene su linearne aproksimacije predstavljene sljedećim sustavom jednadžbi:

$$A \frac{d\Delta h_1}{dt} = \Delta q_u - \left(\frac{(K_{12,A} + K_{12,B})g}{\sqrt{2g(h_{10} - h_{20})}} + \frac{K_{1d} * g}{\sqrt{2gh_{10}}} \right) \Delta h_1 + \left(\frac{(K_{12,A} + K_{12,B})g}{\sqrt{2g(h_{10} - h_{20})}} \right) \Delta h_2, \quad (3-19)$$

$$A \frac{d\Delta h_2}{dt} = \left(\frac{(K_{12,A} + K_{12,B})g}{\sqrt{2g(h_{10} - h_{20})}} \right) \Delta h_1 - \left(\frac{(K_{12,A} + K_{12,B})g}{\sqrt{2g(h_{10} - h_{20})}} + \frac{K_i * g}{\sqrt{2g(h_{20} - x)}} + \frac{K_{2d} * g}{\sqrt{2gh_{20}}} \right) \Delta h_2. \quad (3-20)$$

$$T_v \frac{d\Delta q_u}{dt} = -\Delta q_u + K_v \Delta x_v. \quad (3-21)$$

Uvedene su sljedeće supstitucije:

$$C_1 = \frac{(K_{12,A} + K_{12,B}) \cdot g}{\sqrt{2g(h_{10} - h_{20})}}, \quad (3-22)$$

$$C_2 = \frac{K_{1d} \cdot g}{\sqrt{2gh_{10}}}, \quad (3-23)$$

$$C_3 = \frac{K_i \cdot g}{\sqrt{2g(h_{20} - x)}} + \frac{K_{2d} \cdot g}{\sqrt{2gh_{20}}}. \quad (3-24)$$

Uvrštavanjem jednadžbi (3-22) – (3-24) u jednadžbe (3-18) i (3-19) dobiven je sljedeći sustav jednadžbi:

$$A \frac{d\Delta h_1}{dt} = \Delta q_u - (C_1 + C_2) \Delta h_1 + (C_1) \Delta h_2, \quad (3-25)$$

$$A \frac{d\Delta h_2}{dt} = (C_1) \Delta h_1 - (C_1 + C_3) \Delta h_2, \quad (3-26)$$

$$\Delta q_u + T_v \frac{d\Delta q_u}{dt} = K_v \Delta x_v. \quad (3-27)$$

Ovako dobiven linearni model predstavlja sustav u prostoru varijabli stanja koji se može predstaviti u matričnom obliku. Taj oblik se sastoji od dva temeljna dijela:

- **Jednadžba stanja:** Ove jednadžbe su u obliku sustava diferencijalnih jednadžbi prvog reda te opisuju kako se stanja sustava mijenjaju u vremenu. Drugim riječima, sadrže sve relevantne informacije o trenutnom stanju sustava. Jednadžbe stanja zapisane su kao:

$$\frac{dx}{dt}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (3-28)$$

gdje je:

$\mathbf{x}(t)$ - vektor stanja u vremenu t ,

A - matrica koja opisuje unutarnju dinamiku sustava,

B - matrica koja opisuje utjecaj ulaznih signala $\mathbf{u}(t)$ na promjenu stanja,

$\mathbf{u}(t)$ - vektor ulaznih signala

- **Jednadžba izlaza:** Ova jednadžba definira kako izlazi sustava ovise o varijablama stanja. Često je slučaj da su izlazi sustava zapravo neke od tih varijabli stanja. Ta jednadžba se također zapisuje u matričnom obliku kao:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (3-29)$$

gdje je:

$\mathbf{y}(t)$ - vektor izlaza sustava,

C - matrica koja opisuje kako se vektor stanja preslikava u izlazni vektor,

D - matrica koja opisuje utjecaj ulaza na izlazni vektor.

Ako se uzme u obzir da je izlaz sustava razina tekućine u drugom spremniku, tada se na temelju jednadžbi (3-25) – (3-27), zapis sustava u matričnom obliku može formulirati kao:

$$\begin{bmatrix} \Delta q_u' \\ \Delta h_1' \\ \Delta h_2' \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_v} & 0 & 0 \\ \frac{1}{A} & -\frac{(C_1 + C_2)}{A} & \frac{C_1}{A} \\ 0 & \frac{C_1}{A} & -\frac{(C_1 + C_3)}{A} \end{bmatrix} \cdot \begin{bmatrix} \Delta q_u \\ \Delta h_1 \\ \Delta h_2 \end{bmatrix} + \begin{bmatrix} \frac{K_v}{T_v} \\ 0 \\ 0 \end{bmatrix} \cdot \Delta x_v \quad (3-30)$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta q_u \\ \Delta h_1 \\ \Delta h_2 \end{bmatrix} \quad (3-31)$$

3.2.2. Prijenosna funkcija

Osim prikazivanja matematičkog modela u vremenskoj domeni, isti se može prikazati i u *Laplaceovoj* domeni. Primjenom *Laplaceove* transformacije na sustav u prostoru varijabli stanja

dobiva se prijenosna matrica $\mathbf{G}(s)$. Elementi te matrice su prijenosne funkcije, odnosno element $G_{i,j}$ u i -tom redu i j -tom stupcu te matrice predstavlja prijenosnu funkciju koja opisuje ovisnost i -tog izlaza o j -tom ulazu. [6]

Prednost ovakvog zapisa leži u tome što preoblikuje diferencijalne jednadžbe u algebarske jednadžbe, što olakšava analizu i rješavanje sustava.

Ako se uzme u obzir da je izlaz sustava razina tekućine u izlaznom spremniku, a ulazni signal otvorenost ventila, tada postoji samo jedna prijenosna funkcija i ona glasi:

$$G(s) = \frac{H_2(s)}{X_v(s)} = \frac{C_1 K_v}{T_v A^2 s^3 + (T_v A(2C_1 + C_2 + C_3) + A^2)s^2 + (T_v(C_1 C_2 + C_1 C_3 + C_2 C_3) + A(2C_1 + C_2 + C_3))s + (C_1 C_2 + C_1 C_3 + C_2 C_3)} \quad (3-32)$$

3.3. Eksperimentalno određivanje parametara procesa

U prethodnim potpoglavljima razrađena je struktura matematičkog modela postrojenja skladištenja tekućine. Ta je struktura dobivena iz fizikalnih zakona (*Bernoullijev zakon* i zakoni istjecanja tekućine) te iz pretpostavki o procesu (proporcionalnost ventilskeg pozicionera, linearna promjena protoka u odnosu na otvorenost ventila). Idući je korak određivanje parametara modela.

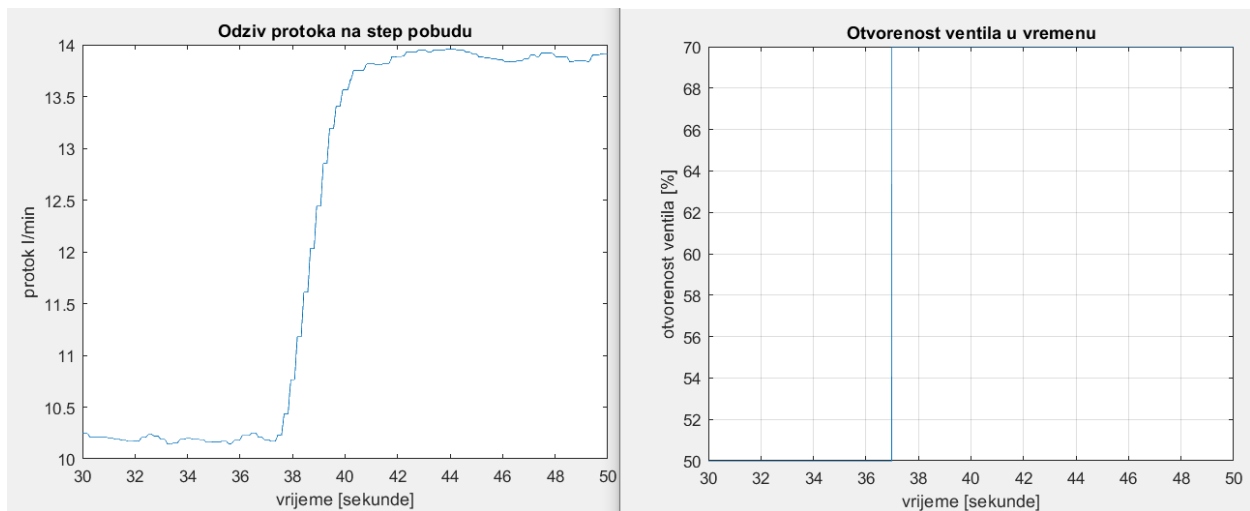
Parametre koje je potrebno odrediti su konstante istjecanja (K_i , $K_{12,A}$, $K_{12,B}$, K_{1d} , K_{2d}), konstante PT_1 člana (K_v , T_v) te konstanta ventilskeg pozicionera (K_m). Do tih parametara teško je doći teorijskom analizom jer ovise o konstrukciji ventila, geometriji i svojstvima cijevi te otvora kroz koje tekućina istječe. Zbog toga se ti parametri određuju na temelju podataka prikupljenih eksperimentima. U nastavku je objašnjeno kako doći do pojedinih parametara.

Ukoliko se na ventilskeg pozicioner dovede struja od 4 mA tada je otvorenost ventilskeg pozicionera 0%, a ako se dovede struja od 20 mA tada je otvorenost 100%. Iz tih je podataka moguće odrediti konstantu proporcionalnosti K_m rješavanjem sljedeće jednadžbe:

$$100 = K_m(20-4). \quad (3-33)$$

Dakle, konstanta proporcionalnosti K_m iznosi 6.25 %/mA.

Za određivanje ovisnosti ulaznog protoka o otvorenosti ventila, snimljen je odziv protoka na skokovitu pobudu otvorenosti ventila (prijelazna funkcija). Slika 3.1. prikazuje taj odziv kada se otvorenost ventila promijeni sa 50% na 70%.



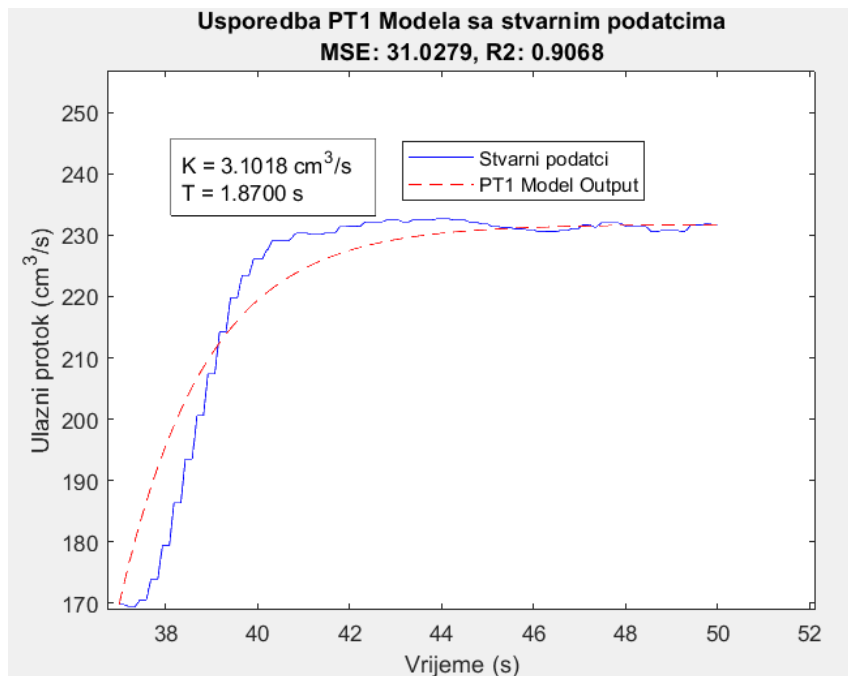
Slika 3.1. Odziv ulaznog protoka na skokovitu pobudu otvorenosti ventila

Dakle, iz grafa je vidljivo da je taj odnos moguće aproksimirati kao PT_1 član koji se sastoji od dviju konstanti (T_v i K_v). Vremensku konstantu ventila T_v moguće je aproksimirati određivanjem vremena potrebnog da prijelazna funkcija postigne 63% konačne vrijednosti. Ova metoda se temelji na teoriji eksponencijalnog odziva sustava prvog reda na skokovitu pobudu. Razlog za korištenje vrijednosti 63% leži u matematičkoj definiciji eksponencijalnog rasta. [7]

Koeficijent proporcionalnosti K_v se računa kao omjer promjene izlazne veličine i promjene ulazne veličine:

$$K_v = \frac{q_{u,konačno} - q_{u,početno}}{x_{v,konačno} - x_{v,početno}} \quad (3-34)$$

Slika 3.2. prikazuje usporedbu dobivenog modela sa stvarnim podacima postrojenja te metriku koja opisuje točnost aproksimacije (najmanje kvadratno odstupanje i koeficijent determinacije).



Slika 3.2. Usporedba PT1 modela sa stvarnim podacima

PT_1 član opisuje linearnu dinamiku. Drugim riječima, Ako se ulazni signal promijeni i rezultira određenom promjenom u izlaznom signalu, prema linearnoj dinamici PT_1 člana, ista proporcionalna promjena očekuje se i pri drugim promjenama ulaznog signala. U stvarnim sustavima to često nije slučaj što znači da postoji određena nelinearnost (drugačije vrijednosti K_v , T_v u različitim radnim točkama). Tablica 3.1. prikazuje kako se mijenjaju vrijednosti pojačanja i vremenske konstante proporcionalnosti za različite pobude. Uočava se kako su ti parametri promjenjivi i ovise o radnim točkama sustava. Te promjene nisu značajne kada se otvorenost ventila mijenja u rasponu od 0 do 70%. Kada se otvorenost ventila promijenila sa 70% na 90% uočeno je kako je parametar K_v značajno manji, što ukazuje na zasićenje sustava.

Tablica 3.1. Vrijednost K_v , T_v za različite pobude

Početna otvorenost ventila	Konačna otvorenost ventila	T_v	K_v
50%	70%	1.87 s	3.1018 cm ³ /s
40%	60%	1.74 s	3.0519 cm ³ /s
30%	50%	1.44 s	2.7731 cm ³ /s
70%	90%	1.72 s	0.8247 cm ³ /s

Konstante istjecanja određuju se provođenjem zasebnih eksperimenata u kojima se tekućina propušta kroz jednu cijev dok je protok kroz ostale cijevi onemogućen. U nastavku slijedi primjer određivanja konstante istjecanja tekućine kroz izlaznu cijev (između izlaznog i cirkulacijskog spremnika).

Ukoliko se onemoguće svi protoci osim izlaznog protoka q_i , tada jednačba (3-14) postaje:

$$\frac{dh_2}{dt} = -\frac{K_i}{A}\sqrt{2gh_2(t)} = -\frac{K_i\sqrt{2g}}{A}\sqrt{h_2} = -K\sqrt{h_2(t)} \quad (3-35)$$

Rješenje ove diferencijalne jednačbe uz $h_2(0) = h_{20}$ glasi:

$$\hat{h}_2(t) = \left(h_{20}^{\frac{1}{2}} - \frac{Kt}{2}\right)^2 \quad (3-36)$$

Jednačba (3-36) predstavlja model. Ako se prikupe stvarne, izmjerene vrijednosti razine tekućine u spremniku u nekom vremenskom razdoblju, tada je moguće odrediti nepoznati parametar K pomoću kriterijske funkcije. Najčešća kriterijska funkcija za određivanje nepoznatih parametara modela jest najmanje kvadratno odstupanje koje za dani problem glasi:

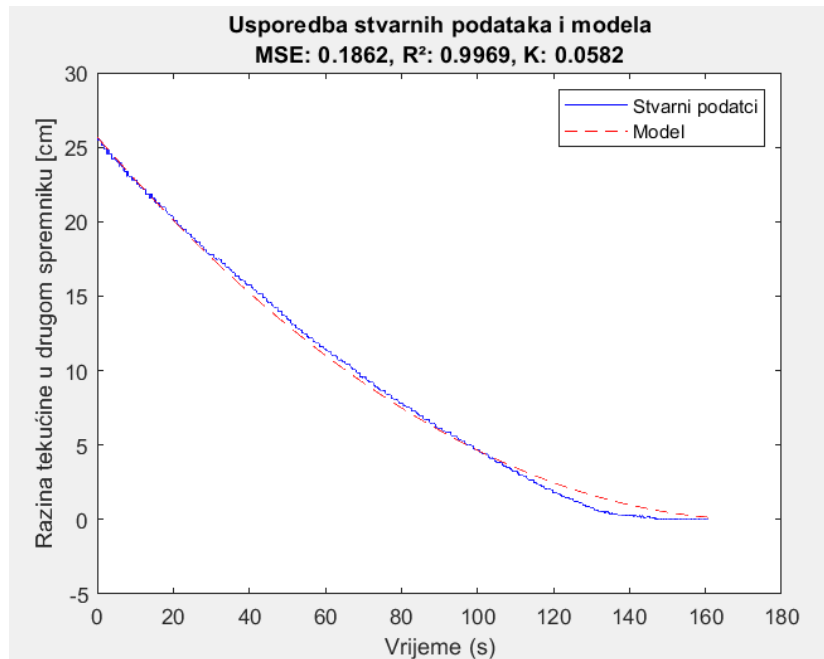
$$E(K) = \sum_i (h_{2,i} - \hat{h}_{2,i})^2 = \sum_i \left(h_{2,i} - \left(h_{20}^{\frac{1}{2}} - \frac{Kt_i}{2}\right)^2\right)^2 \quad (3-37)$$

Cilj je minimizirati kriterijsku funkciju $E(K)$, odnosno pronaći takav parametar K za koji će zadana funkcija postići minimum. Dakle, potrebno je pronaći prvu derivaciju funkcije $E(K)$ i izjednačiti je s 0, odnosno:

$$\frac{dE}{dK}(K) = 0. \quad (3-38)$$

Slika 3.3. prikazuje usporedbu stvarnih mjerenja razine tekućine u drugom spremniku s modelom

za izračunatu vrijednost parametra $K_1 = \frac{K_i\sqrt{2g}}{A} = 0.0582$.



Slika 3.3. Usporedba modela sa stvarnim vrijednostima razine tekućine (omogućen protok q_i)

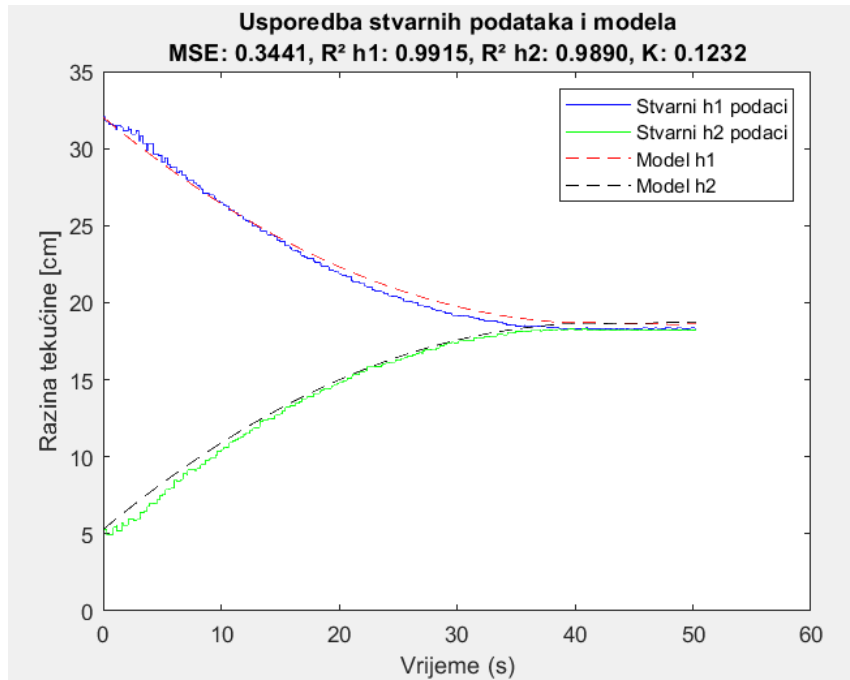
Na isti je način moguće odrediti parametar K (konstantu istjecanja) za preostale protoke. Slika 3.4. prikazuje razinu tekućine u ulaznom (prvom) i izlaznom (drugom) spremniku tijekom vremena. Dobivena vrijednost parametra K_2 koja opisuje ovisnost protoka $q_{12,B}$ o promjeni razine tekućine iznosi $K_2 = \frac{K_{12,B} \sqrt{2g}}{A} = 0.1232$.

Slika 3.5. također prikazuje razinu tekućine u tim spremnicima te dobivenu vrijednost parametra K_3 za protok $q_{12,A}$ koja iznosi $K_3 = \frac{K_{12,A} \sqrt{2g}}{A} = 0.0702$.

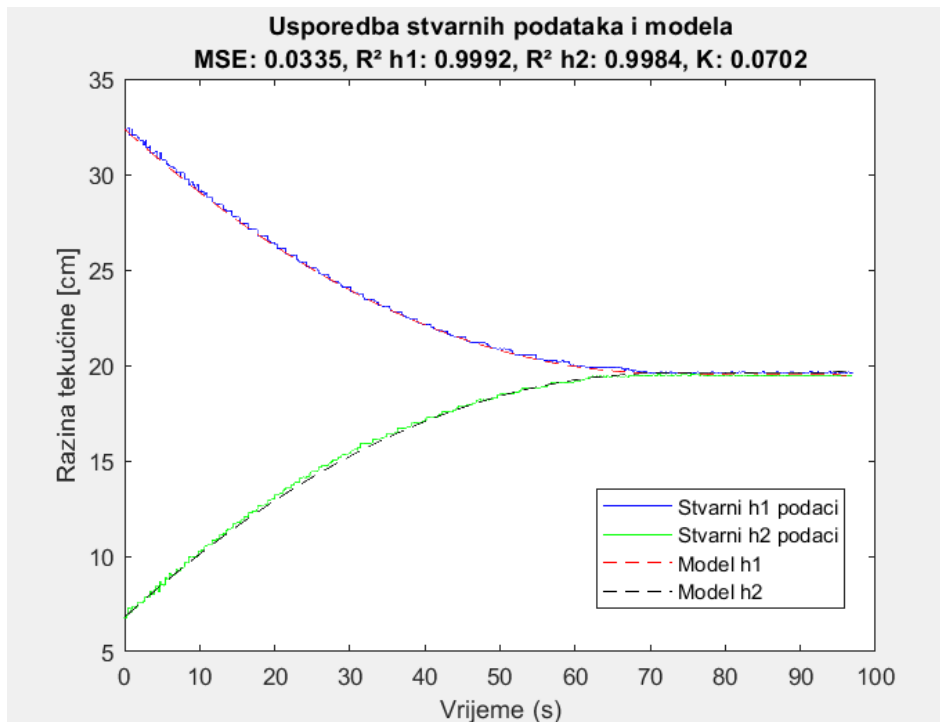
Osim što postoji protok između ulaznog i izlaznog spremnika, postoji protok između ulaznog i cirkulacijskog spremnika te između izlaznog i cirkulacijskog spremnika. Za navedene protoke izveden je jedan eksperiment jer su spremnici jednakih dimenzija s jednakim otvorima prema cirkulacijskom (donjem) spremniku. Dobivena vrijednost konstante (slika 3.6.) iznosi $K_4 = \frac{K_{1d} \sqrt{2g}}{A} = \frac{K_{2d} \sqrt{2g}}{A} = 0.0203$.

Također je proveden eksperiment u kojem su mehanički ventil i otvor između ulaznog i izlaznog spremnika omogućili protoke $q_{12,A}$ i $q_{12,B}$. Rezultati tih mjerenja i usporedba s modelom vidljiva je na slici 3.7. Može se primijetiti kako ta izračunata konstanta $K_5 = 0.1895$ približno odgovara zbroju konstanti K_2 i K_3 (konstante istjecanja pojedinačnih protoka). Pogreška je vrlo mala i iznosi

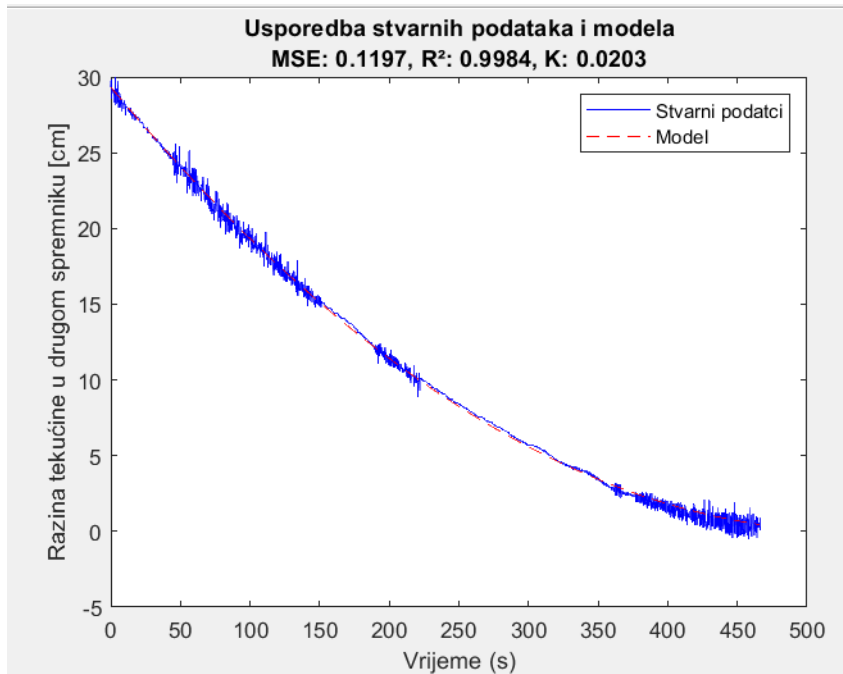
0.0039. Ovaj rezultat potvrđuje da se ukupni utjecaj dvaju istovremenih protoka može uspješno modelirati zbrojem pojedinačnih utjecaja tih protoka.



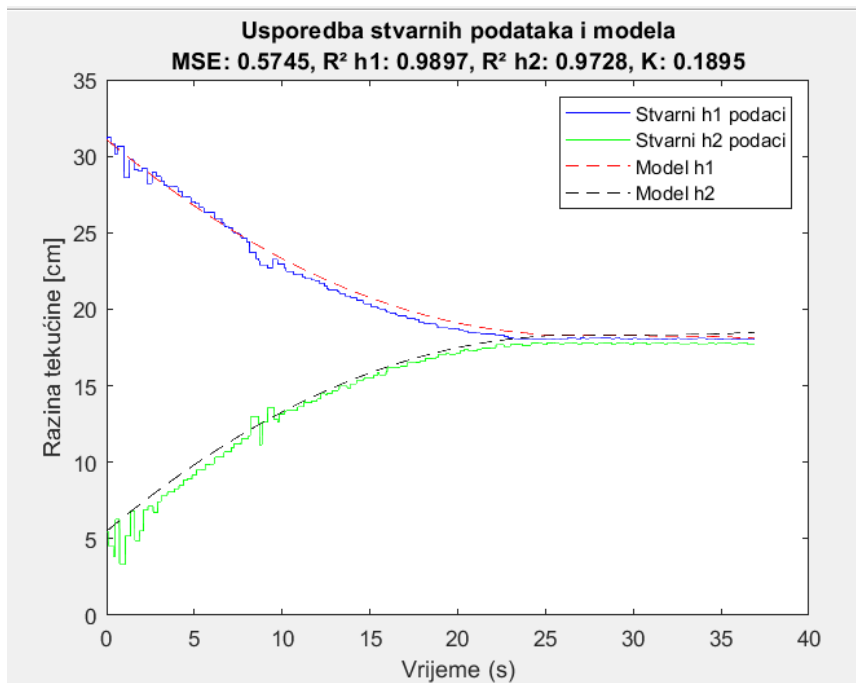
Slika 3.4. Usporedba modela sa stvarnim vrijednostima razina tekućina (omogućen protok $q_{12,B}$)



Slika 3.5. Usporedba modela sa stvarnim vrijednostima razina tekućina (omogućen protok $q_{12,A}$)



Slika 3.6. Usporedba modela sa stvarnim vrijednostima (omogućen protok q_{1d})



Slika 3.7. Usporedba modela sa stvarnim vrijednostima (omogućeni protoci $q_{12,A}$ i $q_{12,B}$)

Konačno, tablica 3.2. prikazuje sve parametre procesa.

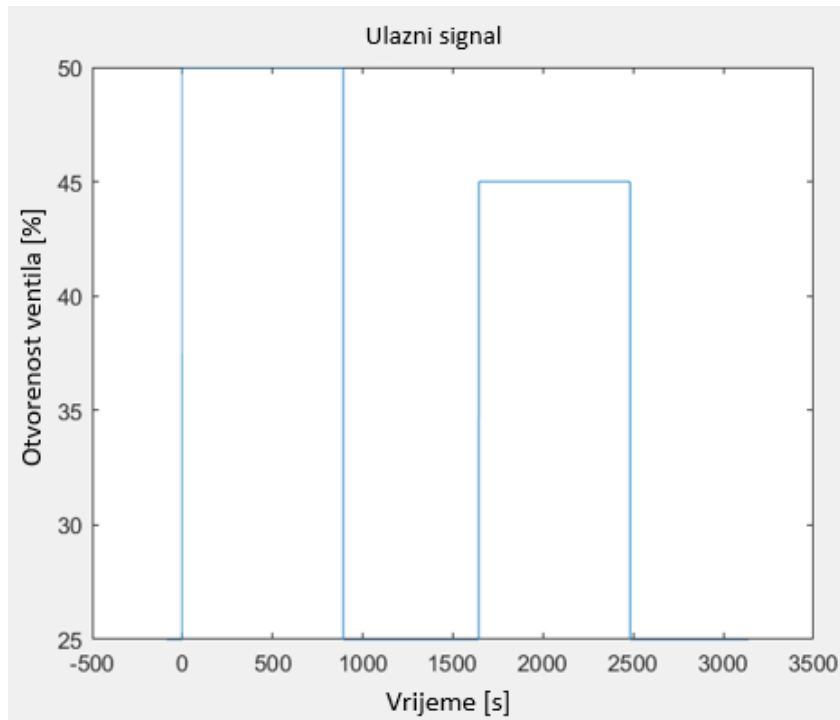
Tablica 3.2. Parametri procesa skladištenja tekućine

Konstanta	Oznaka	Vrijednost	Mjerna jedinica
Površina baze spremnika	$A_1 = A_2$	450	cm^2
Konstanta istjecanja Između ulaznog i cirkulacijskog spremnika	K_{1d}	0.20623	cm^2
Konstanta istjecanja Između ulaznog i izlaznog spremnika	$K_{12,A}$	0.71318	cm^2
Konstanta istjecanja Između ulaznog i izlaznog spremnika	$K_{12,B}$	1.25162	cm^2
Konstanta istjecanja Između izlaznog i cirkulacijskog spremnika	K_i	0.59128	cm^2
Konstanta istjecanja Između izlaznog i cirkulacijskog spremnika	K_{2d}	0.20623	cm^2
Vremenska konstanta ventila	T_v	1.74	s
Konstanta proporcionalnosti ventila	K_v	3.0519	cm^3/s
Konstanta proporcionalnosti ventilskog pozicionera	K_m	6.25	%/mA
Konstanta gravitacijskog ubrzanja	g	981	cm/s^2

3.4. Usporedba matematičkog modela sa stvarnim podacima

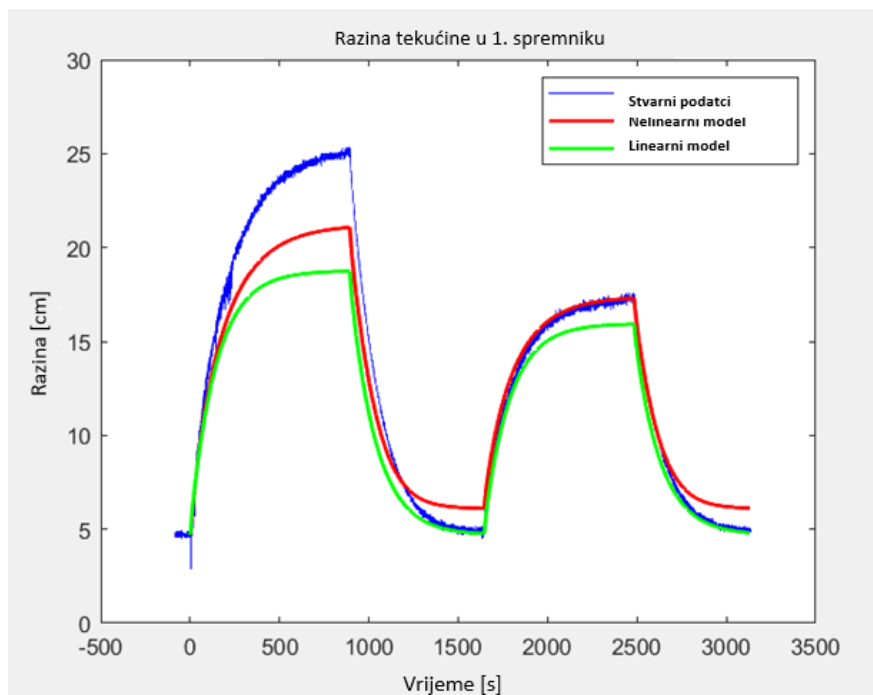
Nakon što su određeni svi parametri matematičkog modela, važno je provjeriti koliko model dobro opisuje stvarni proces. Usporedba između matematičkog modela i stvarnih (testnih) podataka ključna je za procjenu točnosti i pouzdanosti modela. Precizno usklađivanje modela s testnim podacima pokazuje da model adekvatno opisuje sustav, što omogućuje njegovu daljnju primjenu za analizu i sintezu regulatora.

Provedeno je nekoliko skokovitih promjena otvorenosti ventila, prikazanih na slici 3.8., pri čemu su dobiveni odzivi svih varijabli stanja. Slike 3.9. – 3.11. prikazuju usporedbu stvarnih mjerenja s nelinearnim modelom i modelom lineariziranim u radnoj točki koja predstavlja početne vrijednosti varijabli stanja.

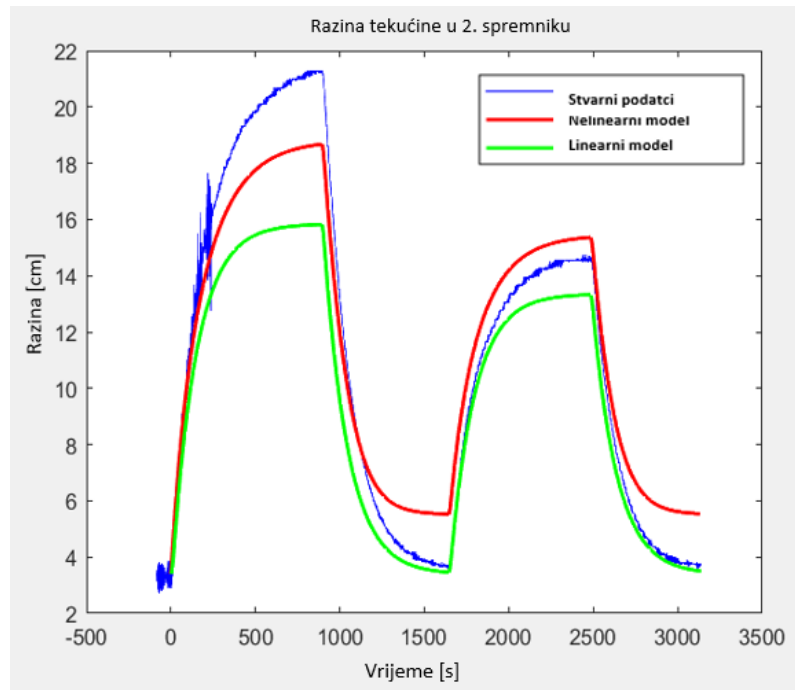


Slika 3.8. Skokovita pobuda – otvorenost ventila

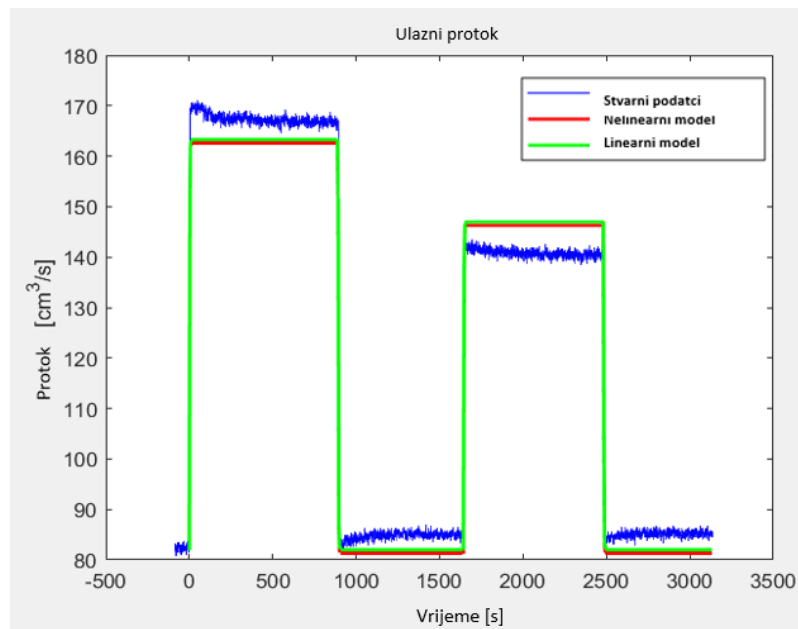
Slika 3.11.



Slika 3.9. Razina tekućine u prvom spremniku i usporedba s matematičkim modelom



Slika 3.10. Razina tekućine u drugom spremniku i usporedba s matematičkim modelom



Slika 3.11. Ulazni protok i usporedba s matematičkim modelom

Iz grafova je moguće zaključiti kako nelinearni model dobro prati stvarne podatke, iako postoje određena odstupanja u stacionarnom stanju što se može pripisati prisutnosti nelinearnih učinaka koje model ne može u potpunosti obuhvatiti. Linearni model također pokazuje blisku korelaciju sa stvarnim podacima, ali u stacionarnom stanju pokazuje veću pogrešku u odnosu na nelinearni

model. To je zbog udaljavanja od radne točke u kojoj je model lineariziran. Unatoč tome, linearni model dobro prati brzinu procesa (dinamiku).

Ova odstupanja moguće je kvantificirati pomoću mjera srednje kvadratne pogreške (engl. *mean squared error*, MSE) i koeficijenta determinacije (engl. *coefficient of determination*, R^2). U tablici 3.3. prikazani su rezultati ovih mjera za oba modela. Rezultati pokazuju kako nelinearni model ima malu srednju kvadratnu pogrešku i veliki iznos koeficijenta determinacije za razinu tekućine u oba spremnika, što ukazuje na dobru aproksimaciju stvarnog procesa. Za ulazni protok, oba modela pokazuju slične vrijednosti MSE i R^2 .

Očekivano je da nelinearni model bolje prati stvarne podatke zbog svoje sposobnosti da obuhvati složenije dinamičke karakteristike sustava. Linearni model, iako jednostavniji, pokazuje zadovoljavajuće rezultate, posebno u dinamičkom dijelu procesa, ali veće pogreške u stacionarnom stanju ukazuju na ograničenja koja proizlaze iz linearnog pristupa modeliranju. Unatoč tim razlikama, oba modela pokazuju vrlo dobre rezultate u praćenju stvarnih podataka, čime potvrđuju svoju korisnost u daljnjoj analizi.

Tablica 3.3. Procjena točnosti modela: MSE i R^2 metrike

Model	MSE $h_1 [cm^2]$	R^2 h_1	MSE $h_2 [cm^2]$	R^2 h_2	MSE $q_u [cm^6/s^2]$	R^2 q_u
Nelinearni model	3.3534	0.9275	2.2532	0.9363	26.406	0.9796
Linearni model	7.7134	0.8333	6.0555	0.8288	24.747	0.9809

3.5. Analiza sustava

Nakon što je uspostavljeno da linearni model dobro opisuje stvarni proces, ključno je provesti detaljnu analizu modela prije nego što se pristupi sintezi upravljačkog algoritma. Analiza sustava u prostoru stanja omogućava duboko razumijevanje dinamike sustava, njegove stabilnosti, upravljivosti i osmotrivosti.

Neka je matematički model lineariziran u radnoj točki definiranoj u tablici 3.4.

Tablica 3.4. Radna točka

Radna točka	$h_1 [cm^2]$	$h_2 [cm^2]$	$q_u [cm^3/s]$
Iznos	4.90	3.4	84.52

Dobiven je linearni matematički model opisan jednadžbama (3-39) i (3-40).

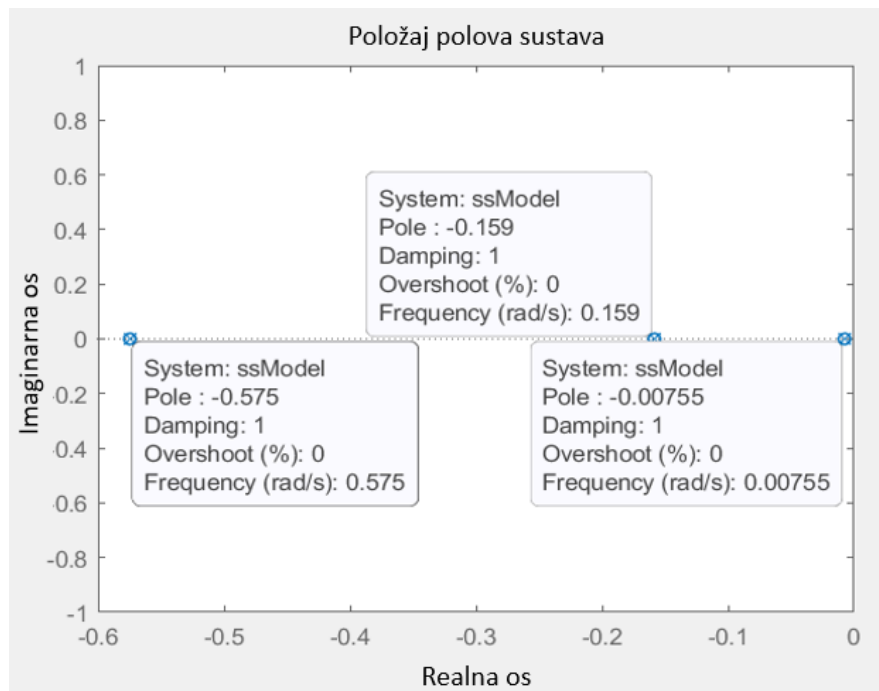
$$\begin{bmatrix} \Delta q_u' \\ \Delta h_1' \\ \Delta h_2' \end{bmatrix} = \begin{bmatrix} -0.5737 & 0 & 0 \\ 0.0022 & -0.0879 & 0.0848 \\ 0 & 0.0848 & -0.0958 \end{bmatrix} \cdot \begin{bmatrix} \Delta q_u \\ \Delta h_1 \\ \Delta h_2 \end{bmatrix} + \begin{bmatrix} 1.8689 \\ 0 \\ 0 \end{bmatrix} \cdot \Delta x_v \quad (3-39)$$

$$y = [0 \quad 0 \quad 1] \cdot \begin{bmatrix} \Delta q_u \\ \Delta h_1 \\ \Delta h_2 \end{bmatrix} \quad (3-40)$$

Od posebne je važnosti matrica **A** koja opisuje dinamičko vladanje sustava. Iz te matrice moguće je doći do zaključka o stabilnosti procesa provjerom svojstvenih vrijednosti. Dakle, Sustav je asimptotski stabilan ako sve svojstvene vrijednosti matrice **A** imaju negativne realne vrijednosti. Te vrijednosti se određuju rješavanjem sljedeće karakteristične jednadžbe:

$$\det(\lambda I - A) = 0 \quad (3-41)$$

Pri čemu λ predstavlja svojstvene vrijednosti matrice **A**. Slika 3.12. prikazuje dobivene izračunate svojstvene vrijednosti (polove) i njihove položaje u kompleksnoj ravnini. Svi polovi imaju negativne realne vrijednosti što znači da je sustav stabilan.

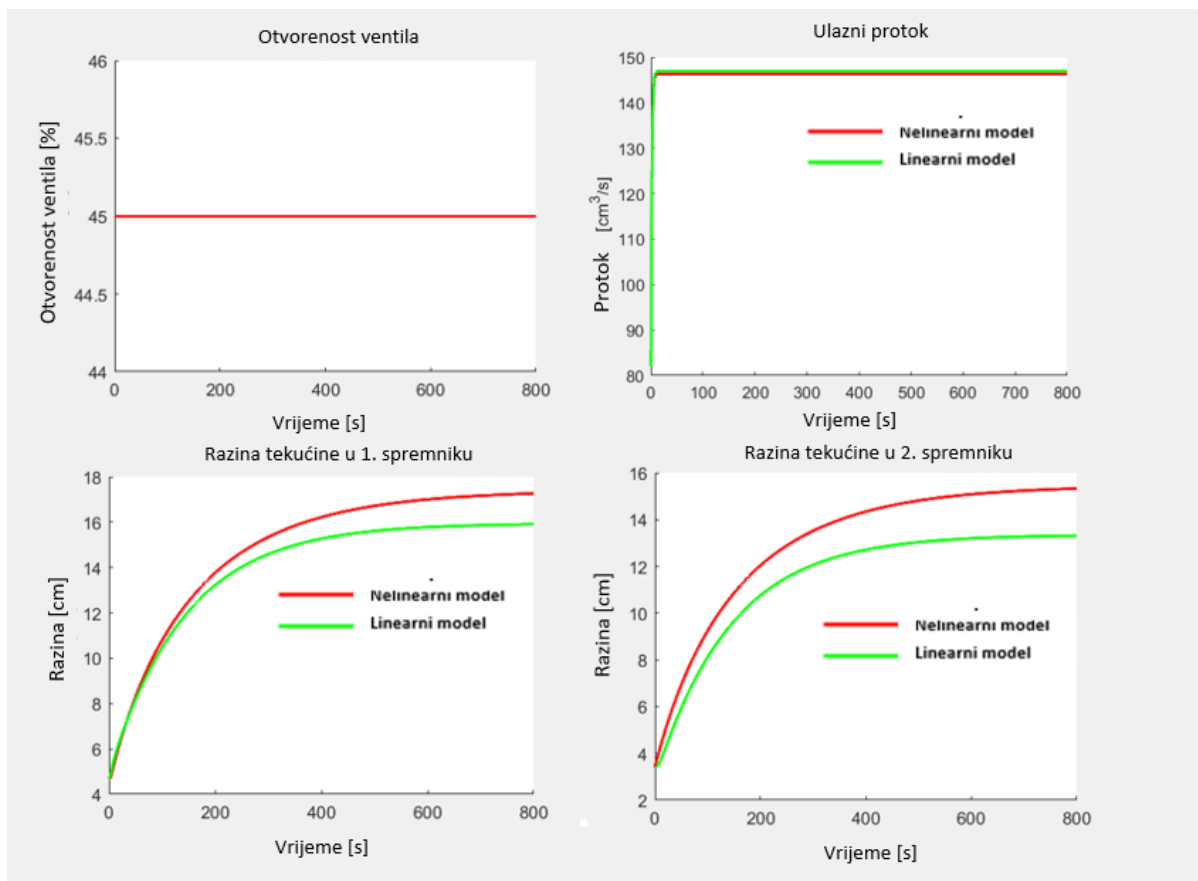


Slika 3.12. Prikaz polova procesa u kompleksnoj ravnini

Uočljivo je i da sustav ima jedan pol koji je izrazito dominantan u odnosu na preostala dva. To se može bolje primijetiti ukoliko se iz tih polova izračunaju vremenske konstante sustava. Vremenske konstante sustava izračunavaju se kao negativna recipročna vrijednost svojstvenih vrijednosti matrice **A**. One se mogu definirati kao parametri koji opisuju brzinu promjene sustava na ulazni signal. Dobivene su sljedeće vremenske konstante:

$$T_1 = 1.74 \text{ s}, T_2 = 143.7440 \text{ s}, T_3 = 5.6572 \text{ s}$$

Uočljivo je kako je jedna vremenska konstanta (143.744 s) izrazito velika i da će upravo ona najviše utjecati na dinamiku i brzinu procesa. To se može uočiti iz sljedećih grafičkih prikaza koji prikazuju odzive svih varijabli stanja na promjenu otvorenosti ventila (slika 3.13.).



Slika 3.13. Odzivi svih varijabli stanja te usporedba linearnog i nelinearnog modela

Uz stabilnost, važno je provjeriti upravljivost te osmotrivost sustava. Upravljivost sustava predstavlja sposobnost da se sustav dovede iz bilo kojeg početnog stanja u bilo koje željeno stanje u određenom i konačnom vremenskom intervalu, uz primjenu odgovarajućeg ulaznog signala. Za provjeru upravljivosti koristi se matrica upravljivosti **W**:

$$W = [B, AB, A^2B, \dots, A^{n-1}B] \quad (3-42)$$

Pri čemu n predstavlja broj varijabli stanja. Za definirani sustav (3-39) vrijedi da je matrica W punog ranga što znači da je sustav upravljiv.

Osmotrivost sustava, s druge strane, predstavlja sposobnost da se sva stanja mogu procijeniti na temelju dostupnih izlaznih mjerenja (u svakom trenutku). Dakle, ako je sustav osmotriv moguće je rekonstruirati sve varijable stanja što je ključno za upravljačke algoritme o kojima je nešto više rečeno u poglavlju 3.8. Matematička formulacija osmotrivosti sustava dana je sljedećim izrazom:

$$O = [C, CA, \dots, CA^{n-1}]. \quad (3-43)$$

S obzirom da za sustav (3-39) matrica O ima puni rang, može se zaključiti da je sustav osmotriv.

3.6. Analiza mjernog šuma

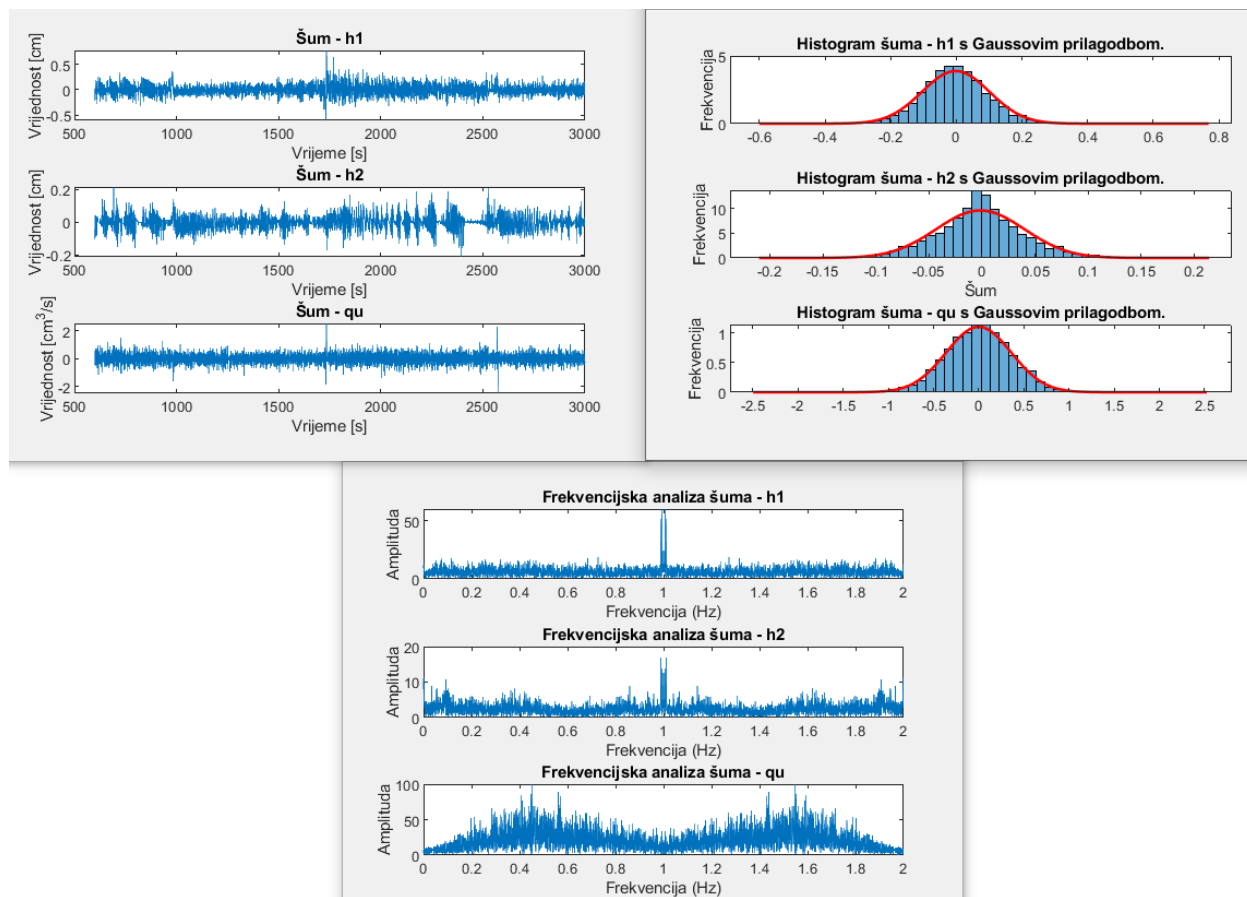
Analiza šuma ključan je korak u obradi podataka i modeliranju sustava. Šum je stohastička komponenta koja se superponira na korisni signal i koja je uzrokovana nesavršenošću mjernih članova. Precizno razumijevanje šuma omogućava bolje modeliranje sustava, poboljšanje mjernih tehnika, optimizaciju upravljačkih algoritama i algoritama za identifikaciju procesa.

Slika 3.14. prikazuje analizu šuma za svaku varijablu stanja. Na lijevoj strani slike prikazani su vremenski grafovi šuma za razinu tekućine u prvom spremniku h_1 , za razinu tekućine u drugom spremniku h_2 te za ulazni protok q_u . Šum u h_1 pokazuje relativno stabilne oscilacije oko nule s povremenim pojavljivanjem stršćih vrijednosti (engl. *peaks*). Šum u h_2 ima sličnu strukturu, ali amplitudu manjeg iznosa. Šum u q_u također oscilira oko nule.

Desna strana slike prikazuje histograme šuma. Histogrami prikazuju raspodjelu vrijednosti šuma, dok crvena linija predstavlja *Gaussovu* razdiobu prilagođenu tim podatcima. Vidljivo je da šumovi u svim varijablama dobro prate normalnu razdiobu. Za h_1 , histogram pokazuje simetričnu distribuciju oko nule s relativno malim rasponom. Šum u h_2 također pokazuje simetričnu distribuciju, ali s užim rasponom, dok šum u q_u ima širi raspon.

Donji grafovi prikazuju *Fourierovu* analizu šuma za sve tri varijable stanja. Grafovi prikazuju amplitude različitih frekvencija prisutnih u šumu. Frekvencijska analiza šuma u h_1 pokazuje dominantan vrh oko frekvencije od 1 Hz, dok su amplitude ostatka spektra relativno niske. U slučaju varijable stanja h_2 , analiza također otkriva nekoliko značajnih stršćih vrijednosti, uključujući jedan oko 0.8 Hz. Struktura šuma kod h_2 slična je onoj kod h_1 , ali s nešto drugačijim dominantnim frekvencijama. Frekvencijska analiza šuma u q_u pokazuje širi spektar frekvencija s

višim amplitudama u niskofrekvencijskom području. Postoji nekoliko dominantnih vrhova između 0.4 Hz i 1.6 Hz.



Slika 3.14. Analiza šuma u mjernim podacima

Izračunati omjeri signal-šum (SNR) za varijable h_1 , h_2 i q_u prikazani su u tablici 3.5. Veći iznosi SNR, konkretno u rasponu od 40 do 50 dB, ukazuju na vrlo dobar omjer signala i šuma. To znači da je korisni signal značajno jači od šuma.

Tablica 3.5. Omjer signal-šum (SNR)

Varijabla stanja	SNR (dB)
h_1	43.28
h_2	49.68
q_u	50.59

3.7. Procjena parametara modela rekurzivnom metodom najmanjih kvadrata

Parametri, zajedno sa strukturom modela, čine temeljni element za analizu i predikciju stanja sustava. Važno je da izgrađeni model vjerno opisuje dinamiku stvarnog sustava. Struktura modela pruža okvir koji definira kako ulazni signali utječu na izlaze sustava, dok parametri kvantificiraju taj odnos. Precizno određivanje parametara ključno je za postizanje visoke točnosti i pouzdanosti modela.

U poglavlju 3.3. opisan je postupak određivanja parametara procesa. Ovi parametri se zatim koriste za izgradnju linearnog sustava koji se, u kontekstu daljnje analize i obrade, pokazuje pogodnijim od nelinearnog sustava zbog manjih računalnih zahtjeva. S druge strane, linearni sustavi imaju ograničenu sposobnost aproksimacije stanja sustava udaljavanjem od radne točke u kojoj je proces lineariziran. Dodatno, linearni sustavi ne uzimaju u obzir moguće promjene parametara u vremenu. Kako bi se riješio taj problem, postoje određeni matematički alati i algoritmi koji omogućuju da model iterativno mijenja parametre na temelju dostupnih podataka. Jedan od takvih algoritama jest rekurzivna metoda najmanjih kvadrata (engl. *recursive least squares method*, RLS).

RLS metoda predstavlja algoritam koji omogućuje kontinuiranu identifikaciju procesa u stvarnom vremenu, prilagođavajući se stalnom pritjecanju novih podataka. Ova metoda kombinira prednosti jednostavnosti linearnih modela s potrebom za dinamičkom prilagodbom, omogućujući precizno praćenje i upravljanje sustavima u stvarnom vremenu. Drugim riječima, ova metoda predstavlja podlogu za određene adaptivne algoritme upravljanja. [4]

U opisanom procesu skladištenja tekućine od posebne je važnosti jednadžba stanja, konkretno matrice \mathbf{A} i \mathbf{B} , jer one definiraju dinamičko vladanje procesa. Ako se uzme u obzir da podatci pristižu kontinuirano i u jednakim vremenskim razmacima T , tada je moguće jednadžbu stanja (3-28) diskretizirati koristeći ZOH diskretizaciju kao što je to prikazano sljedećom jednadžbom:

$$\mathbf{x}(k) = \mathbf{A}_d \mathbf{x}(k-1) + \mathbf{B}_d \mathbf{u}(k-1) \quad (3-44)$$

pri čemu se matrice \mathbf{A}_d i \mathbf{B}_d dobivaju preko sljedećih izraza:

$$\mathbf{A}_d = e^{\mathbf{A}T}, \quad \mathbf{B}_d = \int_0^T (e^{\mathbf{A}t} \mathbf{B} dt) \quad (3-45)$$

Elementi tih matrica su parametri koje je potrebno ažurirati te je njih moguće zapisati u vektorskom obliku kao:

$$\Theta(\mathbf{k}-1) = [a_{11,d} \ a_{12,d} \ a_{13,d} \ a_{21,d} \ a_{22,d} \ a_{23,d} \ a_{31,d} \ a_{32,d} \ a_{33,d} : b_{11,d} \ b_{21,d} \ b_{31,d}] \quad (3-46)$$

Na temelju tog vektora parametara procesa preoblikuje se matrica prethodnih mjernih i upravljačkih signala:

$$\Phi(\mathbf{k}) = \begin{bmatrix} q_u(k-1) & h_1(k-1) & h_2(k-1) & 0 & 0 & 0 & 0 & 0 & 0 & u(k-1) & 0 & 0 \\ 0 & 0 & 0 & q_u(k-1) & h_1(k-1) & h_2(k-1) & 0 & 0 & 0 & 0 & u(k-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_u(k-1) & h_1(k-1) & h_2(k-1) & 0 & 0 & u(k-1) \end{bmatrix} \quad (3-47)$$

Matematički model se tada može formulirati sljedećim izrazom:

$$\hat{\mathbf{x}}(\mathbf{k}) = \Theta(\mathbf{k}-1) \Phi^T(\mathbf{k}) \quad (3-48)$$

Pogreška modela se računa na temelju stvarnih vrijednosti varijabli stanja, koje dolaze sa senzora, i njihovih estimiranih vrijednosti modelom:

$$\epsilon(\mathbf{k}) = \mathbf{x}(\mathbf{k}) - \Theta(\mathbf{k}-1) \Phi^T(\mathbf{k}) \quad (3-49)$$

Od posebne je važnosti *Kalmanova* matrica pojačanja koja služi za izračunavanje promjene parametara modela. Ta matrica se može dobiti na temelju matrice kovarijance:

$$\mathbf{K}(\mathbf{k}) = \frac{\mathbf{P}(k-1)\Phi^T(\mathbf{k})}{\mathbf{I} + \Phi(\mathbf{k})\mathbf{P}(k-1)\Phi^T(\mathbf{k})} \quad (3-50)$$

Matrica kovarijance se uobičajeno inicijalizira kao jedinična matrica pomnožena nekim velikim skalarom. Veća vrijednost skalara predstavlja veću nesigurnost u odabiru inicijalnih vrijednosti parametara sustava. Ta se matrica potom iterativno ažurira na sljedeći način:

$$\mathbf{P}(\mathbf{k}) = \mathbf{P}(k-1) - \mathbf{K}(\mathbf{k})\Phi(\mathbf{k})\mathbf{P}(k-1) \quad (3-51)$$

Nove vrijednosti parametara izračunavaju se na temelju prethodno definirane *Kalmanove* matrice pojačanja:

$$\Theta(\mathbf{k}) = \Theta(\mathbf{k}-1) + \epsilon(\mathbf{k})\mathbf{K}^T(\mathbf{k}) \quad (3-52)$$

U nastavku je prikazana primjena RLS algoritma. Za demonstraciju, uzima se linearni model iz poglavlja 3.5 i na njega se superponira šum analiziran u poglavlju 3.6. Ovaj postupak omogućuje simulaciju stvarnih uvjeta rada sustava, čime se pruža uvid u učinkovitost algoritma pod utjecajem šuma.

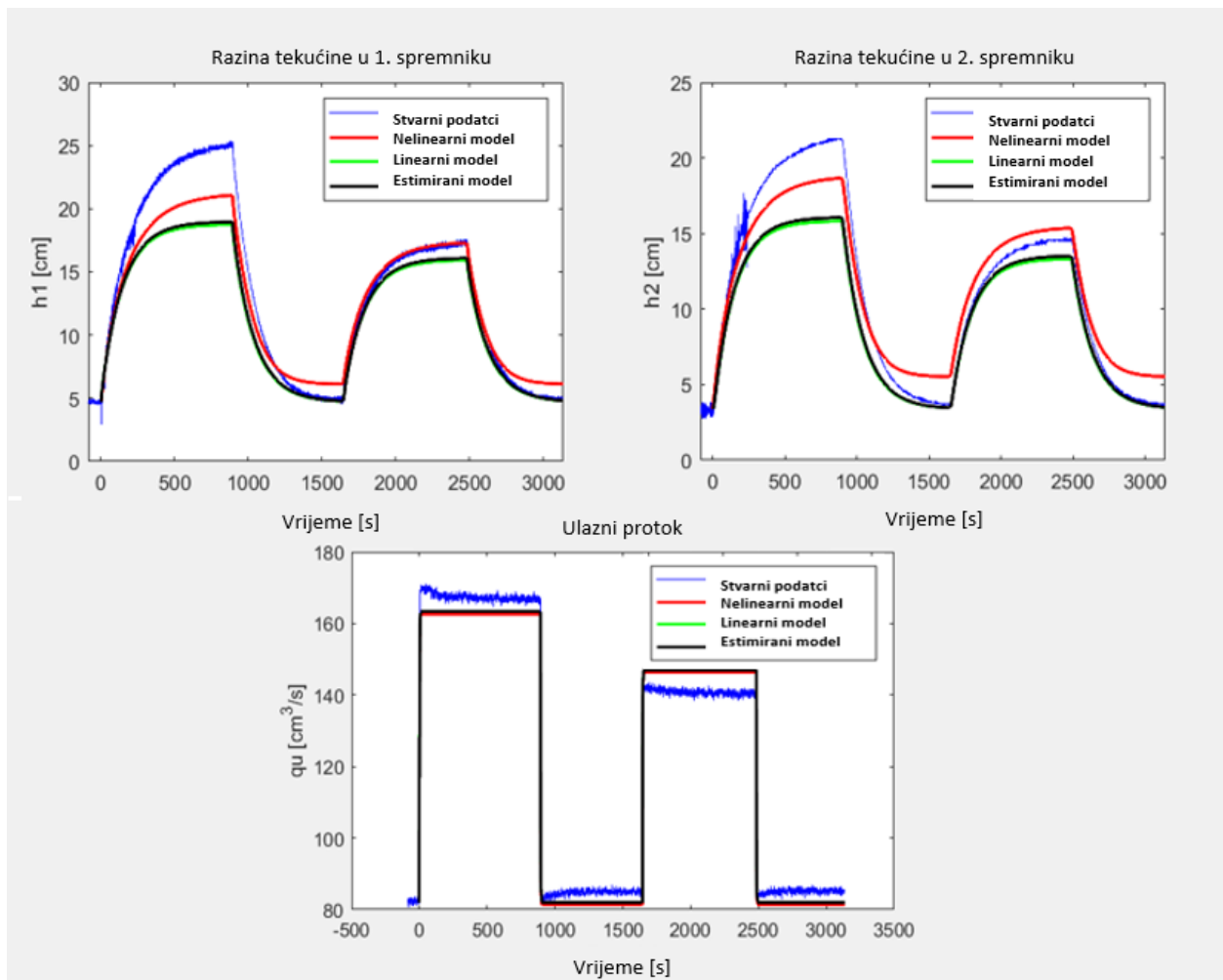
Kroz primjenu RLS algoritma, procjenjuju se vremenske konstante modela u stvarnom vremenu. Dobivene vremenske konstante uspoređuju se s vremenskim konstantama linearnog modela kako bi se ocijenila točnost i pouzdanost estimiranog modela.

Tablica 3.6. prikazuje usporedbu estimiranih vremenskih konstanti sustava i vremenskih konstanti polaznog linearnog modela. RLS algoritam pokazuje zadovoljavajuće rezultate u procjeni vremenskih konstanti sustava. Apsolutne pogreške su relativno male, što ukazuje na to da algoritam može učinkovito pratiti i predviđati dinamiku sustava čak i u prisutnosti šuma. Ova sposobnost čini RLS algoritam korisnim alatom za adaptivno upravljanje i identifikaciju procesa u stvarnim uvjetima.

Tablica 3.6. Usporedba estimiranog modela sa polaznim linearnim modelom

	Stvarna vremenska konstanta	Procijenjena vremenska konstanta	Apsolutna pogreška
T_1	1.7400	1.7828	0.0428
T_2	143.7440	145.6298	1.8858
T_3	5.6572	5.2369	0.4203

Slika 3.15. prikazuje usporedbu matematičkog modela (uključujući i estimirani model) sa stvarnim podacima. Iz grafova se može zaključiti kako estimirani matematički model dobro opisuje stvarni proces.



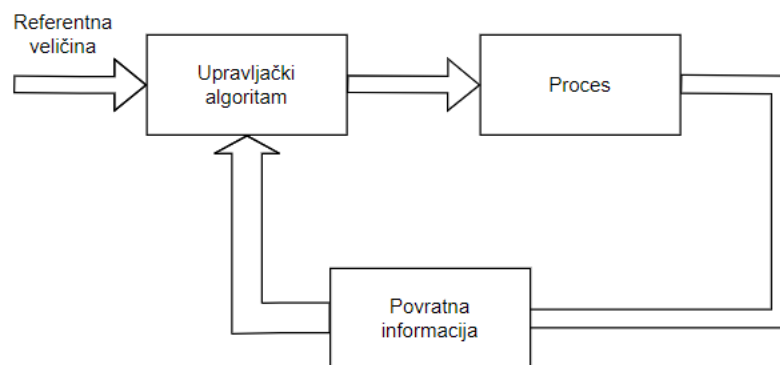
Slika 3.15. Usporedba matematičkog modela sa stvarnim podatcima

3.8. Sinteza regulatora

Upravljanje procesom podrazumijeva nadzor, kontrolu i održavanje željenih parametara (procesnih veličina) ili performansi procesa kroz primjenu regulatora. Regulatori automatski prilagođavaju ulaze ili rad procesa kako bi se postigla željena izlazna vrijednost ili performansa. Ovo je korak koji slijedi nakon analize sustava, dakle nakon što je utvrđena dinamika, stabilnost, upravljivost i osmotrivost tog sustava.

Matematički model predstavlja osnovu za sintezu regulatora. Međutim, važno je imati na umu da matematički model ne može u potpunosti obuhvatiti sve kompleksnosti stvarnog sustava. Stvarni sustavi često su podložni različitim vanjskim smetnjama, nepredviđenim događajima, nesavršenostima u sensorima ili aktuatorima te promjenama u okolini koje nisu u potpunosti modelirane.

U tu se svrhu koristi zatvoreni regulacijski krug (slika 3.16.) čime regulator dobiva povratnu informaciju o trenutnom stanju procesnih veličina. Ta informacija se koristi za kompenzaciju poremaćaja čime se osigurava održavanje određenih procesnih veličina na željenim vrijednostima. Također, ako je proces inherentno nestabilan, moguće ga je stabilizirati odabirom odgovarajućeg regulatora. Međutim, također postoji mogućnost da se stabilan proces destabilizira ukoliko se odabere neodgovarajući regulator. Dakle, važno je odabrati takav regulator koji će osigurati stabilnost sustava i kojim će se postići željena dinamika.



Slika 3.16. Zatvoreni regulacijski krug

Nakon što je odabran regulator, potrebno je utvrditi njegovo djelovanje na stvarni sustav. Evaluacijom stacionarne pogreške i dinamičkih pokazatelja kakvoće, moguće je analizirati performanse tog regulatora. Stacionarna pogreška predstavlja razliku između željene vrijednosti i stvarnog izlaza procesa u stacionarnom stanju.

Neki od dinamičkih pokazatelja kakvoće su: [8]

- Vrijeme porasta: Vrijeme potrebno da izlaz sustava poraste od 10% do 90% svoje konačne vrijednosti.
- Vrijeme ustaljenja: Vrijeme potrebno da izlaz sustava ostane unutar određenog postotka svoje konačne vrijednosti (obično 3% ili 5%)
- Maksimalno nadvišenje: Maksimalna vrijednost koju izlaz sustava dostiže iznad željene vrijednosti

U nastavku je opisana metoda određivanja parametara regulatora pomoću sinteze regulatora u prostoru stanja.

3.8.1. Sinteza regulatora u prostoru stanja

U prostoru varijabli stanja popularne su dvije metode za sintezu regulatora: metoda postavljanja polova (engl. *Pole placement*) te linearni kvadratni regulator (engl. *linear quadratic regulator*, LQR).

Metoda postavljanja polova odnosi se na postupak pozicioniranja polova zatvorenog regulacijskog kruga sustava kako bi se postigla željena dinamička svojstva. Polovi zatvorenog regulacijskog kruga određuju kako će sustav reagirati na promjene u ulazima i na smetnje te time izravno utječu na brzinu odziva, stabilnost i robusnost sustava. Međutim, postavljanje polova može biti izazovno zbog nedostatka smjernica gdje postaviti te polove (svojstvene vrijednosti matrice **A**).

Umjesto postavljanja svojstvenih vrijednosti matrice sustava na željene točke u kompleksnoj ravnini, LQR pristupa minimizaciji ukupnog troška, koji je definiran kao kvadratna funkcija stanja i upravljačkih signala, kako bi se odredila optimalna povratna veza stanja. Ovaj pristup koristi funkciju troška koja balansira između smanjenja odstupanja stanja sustava od željene referentne točke i intenziteta upravljačkih signala potrebnih za postizanje te referentne točke. Funkcija koju je potrebno minimizirati sljedećeg je oblika: [9]

$$J = \int_0^{\infty} (\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + u(t)^T \mathbf{R} u(t)) dt \quad (3-53)$$

gdje je:

$\mathbf{x}(t)$ – vektor varijabli stanja,

$u(t)$ – upravljački signal,

Q – matrica koja penalizira odstupanje stanja od željenog,

R – matrica koja penalizira korištenje upravljačkih signala .

Ukoliko je matematički model diskretiziran, tada jednadžba za kriterij optimizacije J poprima sljedeći oblik [10]:

$$J = \sum_{k=0}^{\infty} (\mathbf{x}[k]^T \mathbf{Q} \mathbf{x}[k] + \mathbf{u}[k]^T \mathbf{R} \mathbf{u}[k]) \quad (3-54)$$

Postavljanje matrice **Q** i **R** ključno je za određivanje performansi i ponašanje upravljačkog sustava. Matrica **Q** obično se postavlja kao dijagonalna matrica pri čemu elementi na dijagonali određuju težinu za odstupanje pojedinih stanja sustava od željenih vrijednosti. Veće vrijednosti podrazumijeva veću penalizaciju stanja što može utjecati na brži odziv, ali također može uvesti i veće oscilacije u sustav. Matrica **R** također se uobičajeno postavlja kao dijagonalna matrica pri

čemu svaka vrijednost na dijagonali određuje penalizaciju za korištenje odgovarajućeg signala. Veće vrijednosti rezultiraju korištenjem manjeg intenziteta i magnitude upravljačkih signala čime se postiže smanjena potrošnja, ali i sporiji odziv. Pri definiranju matrice \mathbf{R} za upravljačke signale važno je uzeti u obzir fizička ograničenja aktuatora, uključujući njihovu maksimalnu snagu i brzinu.

Minimizacija funkcije troška (3-53) povezana je s rješavanjem *Riccatijeve* jednadžbe koja kao rezultat omogućuje pronalaženje optimalne povratne veze stanja. *Riccatijeva* jednadžba dana je sljedećim izrazom:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0 \quad (3-55)$$

Pri čemu je

\mathbf{P} – matrica koja predstavlja rješenje te jednadžbe.

Ukoliko je riječ o matematičkom modelu u kontinuiranoj vremenskoj domeni tada se iz matrice \mathbf{P} određuje vektor optimalne povratne veze stanja \mathbf{K} sljedećim izrazom:

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}. \quad (3-56)$$

Međutim, jednadžba (3-55) vrijedi za kontinuirane sustav. Ukoliko je matematički model diskretiziran, tada je potrebno riješiti sljedeću jednadžbu:

$$\mathbf{P}_{j-1} = \mathbf{Q} + \mathbf{A}_d^T \mathbf{P}_j \mathbf{A}_d - \mathbf{A}_d^T \mathbf{P}_j \mathbf{B}_d (\mathbf{R} + \mathbf{B}_d^T \mathbf{P}_j \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P}_j \mathbf{A}_d \quad (3-57)$$

Za $j = N, \dots, 1$.

Iz jednadžbe (3-57) moguće je odrediti vektor optimalne povratne veze stanja \mathbf{K} sljedećim izrazom:

$$\mathbf{K}_j = (\mathbf{R} + \mathbf{B}_d^T \mathbf{P}_{j+1} \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P}_j \mathbf{A}_d. \quad (3-58)$$

Za $j = 0, \dots, N-1$.

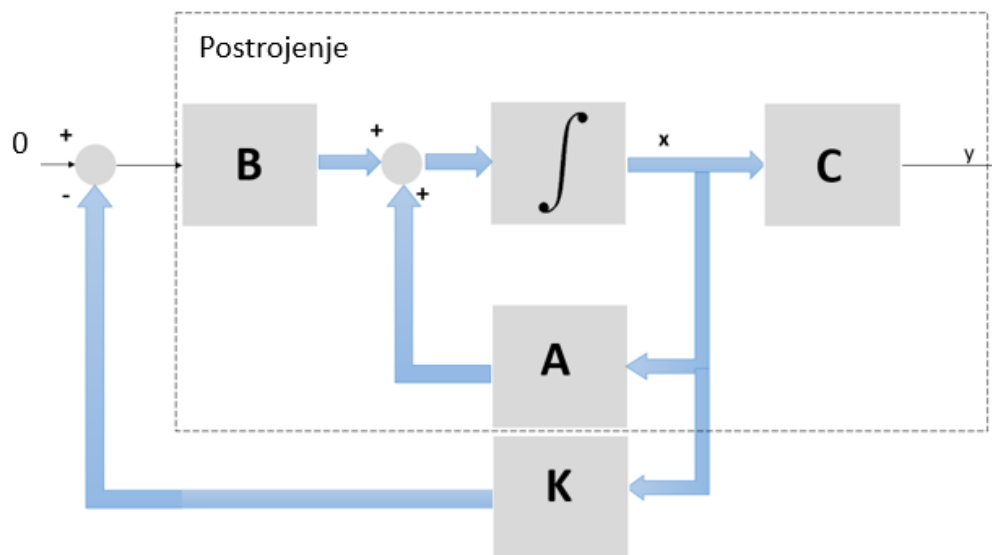
Konačno, slijedi upravljački zakon u kontinuiranoj i diskretnoj vremenskoj domeni:

$$\mathbf{u}(\mathbf{t}) = -\mathbf{K} \mathbf{x}(\mathbf{t}) \text{ ili } \mathbf{u}(\mathbf{k}) = -\mathbf{K} \mathbf{x}(\mathbf{k}), \quad (3-59)$$

Iz jednadžbe (3-59) vidljivo je kako je potrebno u svakom trenutku poznavati vrijednosti svih varijabli stanja za implementaciju upravljačkog zakona. Ukoliko neke varijable stanja nisu mjerljive, sustav mora biti osmotriv kako bi se iz matematičkog modela, na temelju izlaznih mjerenja, mogle procijeniti sve varijable stanja. *Kalmanov* filter i *Luenbergerov* estimator su dva

dobro poznata algoritma za procjenu stanja. U sustavu skladištenja tekućine definirane su tri varijable stanja: ulazni protok, razina tekućine u prvom spremniku i razina tekućine u drugom spremniku. Sve tri varijable stanja su izravno mjerljive te stoga ne postoji potreba za njihovom estimacijom. Drugim riječima, upravljački zakon iz jednadžbe (3-59), može se izravno implementirati na podacima dobivenim mjernim članovima.

Slika 3.17. prikazuje shemu zatvorenog regulacijskog kruga za dani upravljački zakon.



Slika 3.17. Zatvoreni regulacijski krug

Međutim, osnovni LQR upravljački zakon opisan jednadžbom (3-59) može samo stabilizirati sustav, ali ne može osigurati praćenje referentne vrijednosti izlaza. Također, na slici 3.17. vidljivo je kako ne postoji povratna veza izlaza. Kako bi se omogućilo praćenje referentne vrijednosti, potrebno je augmentirati sustav u prostoru stanja. Augmentacija podrazumijeva definiranje nove varijable stanja u već postojeći originalni sustav stanja. S obzirom da se želi postići praćenje referentne veličine, potrebno je dodati varijablu stanja koja predstavlja integral pogreške. Dodavanje te nove varijable stanja augmentacijom, predstavljeno je sljedećim izrazom:

$$\begin{aligned}
 \mathbf{A}_a &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \\
 \mathbf{B}_a &= \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \\
 \mathbf{C}_a &= \begin{bmatrix} \mathbf{C} \\ \mathbf{0} \end{bmatrix}
 \end{aligned}
 \tag{3-60}$$

pri čemu su

$\mathbf{A}_a, \mathbf{B}_a, \mathbf{C}_a$ – matrice augmentiranog prostora stanja.

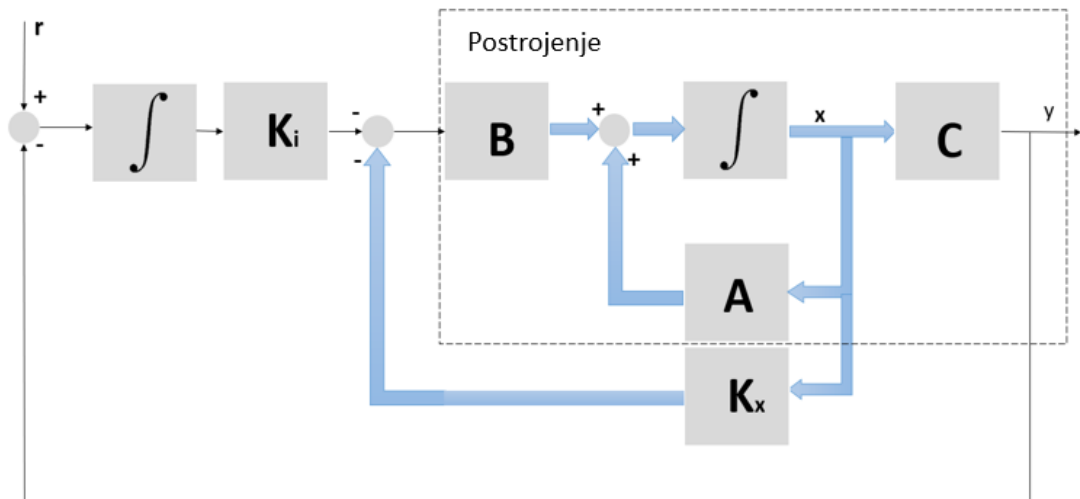
Dobiven je novi upravljački zakon koji omogućuje praćenje referentne veličine i minimizaciju pogreške u stacionarnom stanju, a on glasi:

$$\mathbf{u}(t) = -\mathbf{K}_x \mathbf{x}(t) - K_i \int_0^T e(t) dt. \quad (3-61)$$

U diskretnoj vremenskoj domeni vrijedi sljedeći izraz:

$$\mathbf{u}(k) = -\mathbf{K}_x \mathbf{x}(k) - K_i \sum_{i=0}^k e(i). \quad (3-62)$$

Implementacija upravljačkog zakona iz jednačbe (3-61), odnosno (3-62), prikazana je na slici 3.18. kao shema zatvorenog regulacijskog kruga.



Slika 3.18. Zatvoreni regulacijski krug s upravljačkim zakom

4. IZRADA APLIKACIJE ZA SUSTAV SKLADIŠTENJA TEKUĆINE U SPREMNICIMA

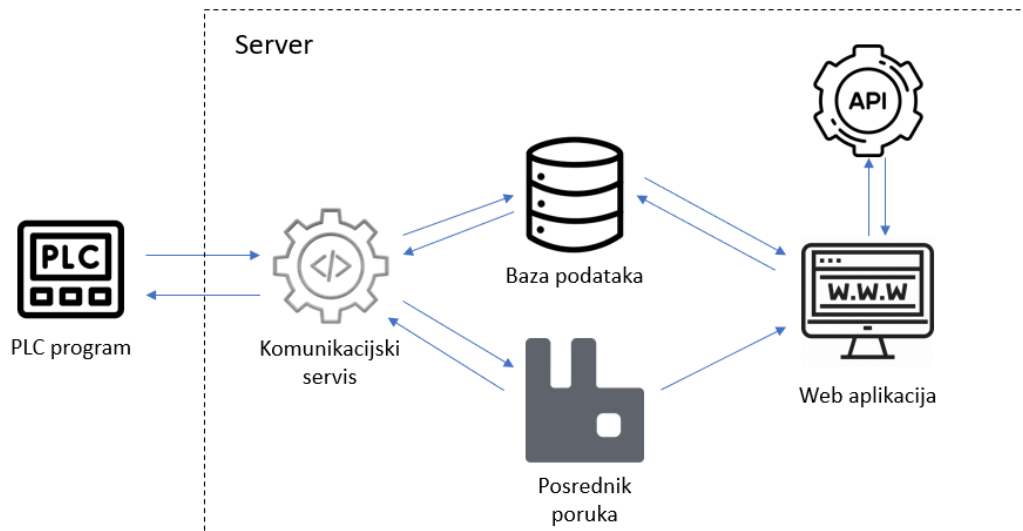
U ovom poglavlju opisan je razvoj programske podrške za sustav upravljanja, analize i nadzora skladištenja tekućine u laboratorijskom pilot postrojenju. Prije izrade programske podrške, potrebno je definirati osnovne specifikacije sustava. Definirani su sljedeći zahtjevi:

- **Prikaz podataka u stvarnom vremenu:** omogućavanje korisnicima da imaju stalni uvid u trenutno stanje procesnih veličina.
- **Pregled pilot postrojenja:** detaljan i ažuriran prikaz svih relevantnih informacija o postrojenju, uključujući rad senzora i regulatora.
- **Intuitivno korisničko sučelje:** razvoj jednostavnog i lako razumljivog sučelja koje omogućava brz i učinkovit pristup funkcijama analize podataka i sinteze sustava upravljanja.

Za razvoj aplikacije koja će ispuniti navedene zahtjeve, potrebno je dizajnirati arhitekturu programske podrške koja obuhvaća sljedeće komponente:

1. Komunikacijski servis koji omogućava razmjenu podataka između PLC-a (na kojem je implementirana niža razina upravljanja) i servera.
2. Baza podataka koja je dizajnirana za pohranu svih podataka PLC-a, uključujući podatke o razini tekućine, protocima i drugim relevantnim parametrima. Baza podataka osigurava trajnu pohranu i omogućava pristup podacima za analizu i izvještavanje.
3. Aplikacijsko programsko sučelje (engl. *Application programming interface*, API) koje implementira niz algoritama za analizu i sintezu regulacije sustava. Ovaj sloj omogućava kompleksne obrade podataka i pruža rezultate koji se koriste za regulaciju i nadzor sustava.
4. Web aplikacija koja omogućava korisnicima pristup sustavu putem intuitivnog grafičkog sučelja. Omogućava pregled podataka u stvarnom vremenu i upravljanje sustavom regulacije.

Slika 4.1. prikazuje glavne dijelove sustava.



Slika 4.1. Glavni dijelovi programske podrške

4.1. Definiranje podataka za razmjenu između PLC-a i servera

Prije izrade komunikacijskog sustava za razmjenu informacija između servera i PLC-a, potrebno je definirati poruke i podatke koje će se razmjenjivati. U tu je svrhu napravljena *LanDef* tablica unutar *MS Excel* programa koja definira:

- Izvor podataka i odredište (PLC ili server),
- Identifikacijsku oznaku bloka podataka na PLC-u,
- Ukupnu veličinu podatkovnog bloka,
- Strukturu poruke (zaglavlje, podatkovni dio, podnožje)

Zaglavlje i podnožje poruke osiguravaju ispravnu komunikaciju i pomažu pri identifikaciji novih poruka za obradu, dok podatkovni dio predstavlja stvarni sadržaj poruke. Za sve podatke definirana je adresa, oznaka, opis i tip varijable. Detaljnije objašnjenje strukture podataka nalazi se u poglavljima 4.1.1 i 4.1.2.

Primjer jedne poruke prikazana je na slici 4.2.

Message Source:		PLC							PORT			18.5	
Message Destination:		LEVEL 2 SERVER							EVN			193	
Address	Tag	Description	Fmt	Size	Min	Max	Unit	Remarks	N/A	Rev	Ver	ToC	
MESSAGE BUFFER HEADER												6	
DBW 0000		Change Counter	UIN							00	A	2	0
DBW 0002		Buffer Pointer	UIN		1	5				00	A	2	2
DBW 0004		Max Elements	UIN		5	5				00	A	2	4
MESSAGE DATA STRUCTURE												37	
DBD 0006		Valve position feedback	FP	5			%			00	A	4	6
DBD 0010		Inlet flow	FP				m3/s			00	A	4	10
DBD 0014		Liquid level in the input tank	FP				m			00	A	4	14
DBD 0018		Liquid level in the output tank	FP				m			00	A	4	18
DBD 0022		Outlet flow	FP				m3/s			00	A	4	22
0026		Datetime	DTL							00	A	12	26
DBB 0038		flag	BB							00	A	1	38
DBD 0039		sample	DI									4	39
CIRCULAR BUFFER FOOTER												2	
DBW 0191		Auxiliary Counter	UIN							00	A	2	191

Slika 4.2. Primjer LanDef poruke

Osim poruke prikazane na slici 4.2., koja uključuje informacije o procesu te vrijednosti mjernih i izvršnih članova, definirano je još nekoliko poruka potrebnih za praćenje sustava skladištenja tekućine u spremnicima u stvarnom vremenu. Svaka poruka ima naziv u formatu L1L2_imePoruke ili L2L1_imePoruke. Oznaka L1L2 označava smjer poruke od PLC-a (L1) prema serveru (L2), dok oznaka L2L1 označava smjer poruke od servera (L2) prema PLC-u (L1). Popis svih poruka prikazan je u tablici 4.1.

Tablica 4.1. Popis poruka i opis

Naziv	Opis
L2L1_WATCHDOG	Potvrđuje aktivnu vezu između PLC-a i servera
L1L2_PROCESS_DATA	Podaci mjernih i izvršnih članova koje PLC šalje serveru.
L2L1_SETPOINT	Postavke referentnih veličina koje server šalje PLC-u
L1L2_ALARMS	Informacije upozorenja o kritičnim razinama tekućine u spremnicima, o neispravnom radu crpke i ventilskog pozicionera
L1L2_CONTROLLER_PARAMS	Trenutno postavljeni parametri regulatora
L2L1_CONTROLLER_PARAMS	Poruka za postavljanje parametara regulatora
L2L1_REQUEST_COTNROL	Zahtjev za kontrolu (postavljanje parametara regulatora i referentne veličine)
L1L2_CONTROL_MODE	Trenutni način upravljanja

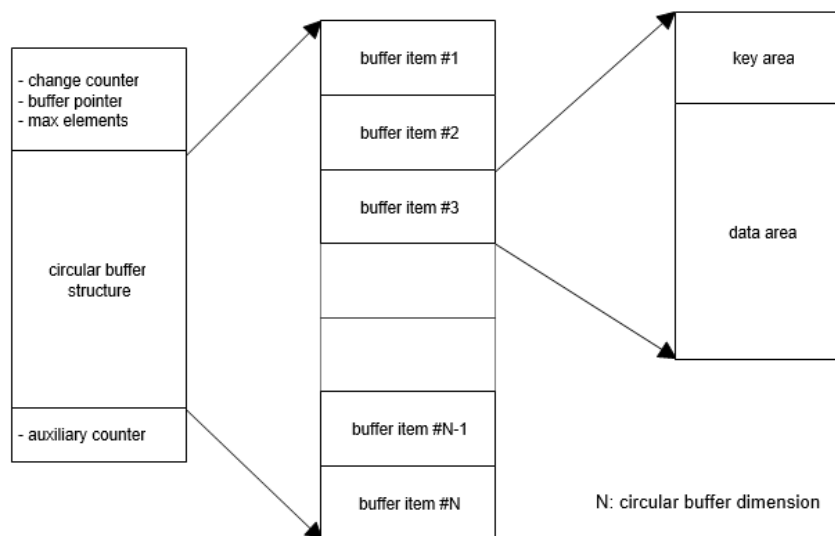
Izuzetno je važno osigurati neprekidnu komunikaciju između PLC-a i servera kako bi se spriječio gubitak podataka. Da bi se to ostvarilo definiran je jedinstven način razmjene podataka. Najčešće se koriste dva pristupa:

1. Kružni spremnik,
2. Podatkovni spremnik.

4.1.1. Kružni spremnik

Kružni spremnik je polje struktura koje se koristi za komunikaciju usmjerenu od PLC-a prema serveru. Njegova struktura (slika 4.3.) sastoji se od:

- Zaglavlja: Sadrži informacije o brojaču promjena (engl. *Change counter*), pokazivaču na spremnik (engl. *Buffer pointer*) i dimenziji spremnika,
- Podatkovnog dijela: Polje struktura, gdje svaka struktura sadrži identifikacijsku oznaku i podatke,
- Podnožja: Pomoćni brojač (engl. *Auxiliary counter*) za potvrdu promjene,



Slika 4.3. Struktura kružnog spremnika

Prvi korak u procesu upisivanja novog podatka je ažuriranje brojača promjene, koji se inkrementira svaki put kad se izvrši nova promjena u spremniku. Taj brojač inkrementira entitet (PLC ili server) koji upisuje nove podatke u strukturu. Nakon ažuriranja brojača promjena, nova struktura podataka, koja uključuje identifikacijsku oznaku i stvarne podatke, dodaje se u kružni spremnik na sljedeću slobodnu poziciju. Nakon toga, pokazivač na spremnik se ažurira kako bi ukazivao na

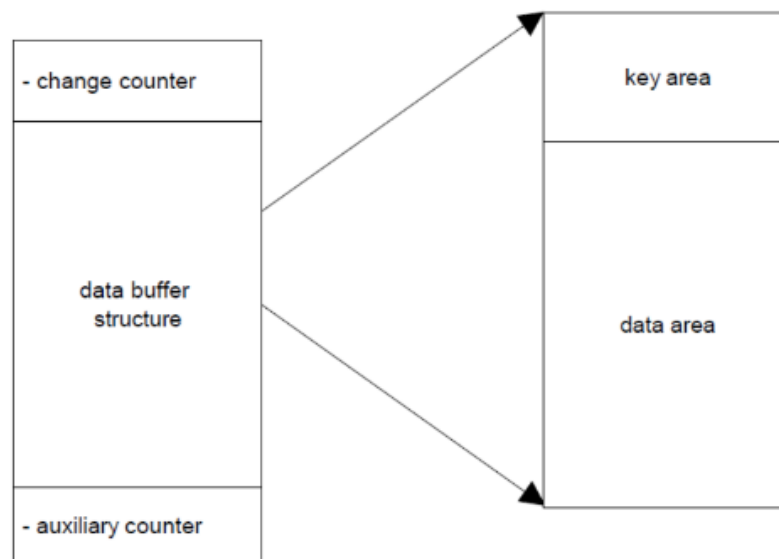
poziciju novododane strukture podataka, čime se osigurava praćenje posljednje ažurirane strukture. Pomoćni brojač se također ažurira tako da njegova vrijednost odgovara brojaču promjene, što omogućava servisnoj aplikaciji da potvrdi završetak procesa upisivanja podataka.

Korištenje brojača promjena i pomoćnog brojača značajno olakšava prepoznavanje promjena u kružnom spremniku. Komunikacijski servis može pratiti samo dvije vrijednosti umjesto cijele strukture podataka, što ubrzava proces i smanjuje mogućnost gubitka podataka. Ako aplikacija propusti očitati promjenu, novi podatci će se upisati na sljedeće slobodno mjesto u spremniku, dok će prethodni podatci ostati u memoriji. Ovo omogućuje naknadnu obradu podataka, čime se smanjuje rizik od gubitka važnih informacija.

4.1.2. Podatkovni spremnik

Podatkovni spremnik je polje struktura koje se koristi za komunikaciju usmjerenu od servera prema PLC-u. Struktura ovog spremnika (slika 4.4.) slična je kružnom spremniku, a sastoji se od:

- Zaglavlja: Sadrži informacije o brojaču promjena (engl. *Change counter*),
- Podatkovnog dijela: struktura podataka,
- Podnožja: Pomoćni brojač (engl. *Auxiliary counter*) za potvrdu promjene.



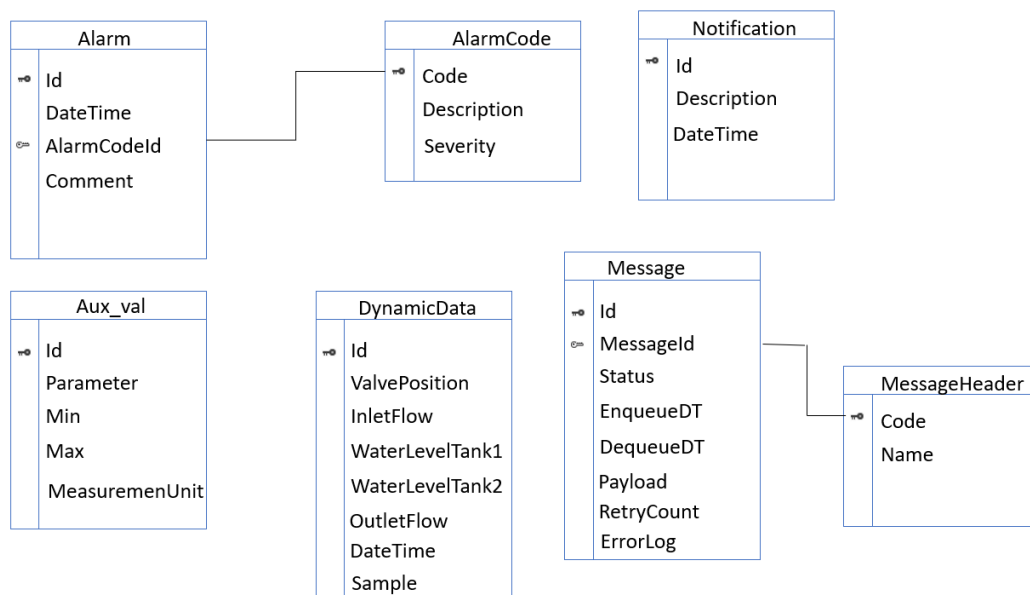
Slika 4.4. Struktura podatkovnog spremnika

Za prijenos informacija od servera prema PLC-u, za razliku od kružnog spremnika, dovoljna je jedna struktura umjesto cijelog polja struktura. Iako se koristi samo jedan element, postupak brojača promjena i pomoćnog brojača ostaje isti. U ovom scenariju, servisna aplikacija je

odgovorna za inkrementiranje brojača i ažuriranje podataka prema ranije navedenim pravilima. Ova pojednostavljenost proizlazi iz činjenice da brza izmjena podataka iz servisne aplikacije prema PLC-u često nije potrebna.

4.2. Baza podataka

Za potrebe trajnog arhiviranja pogonskih podataka, potrebno je izraditi bazu podataka. Baza podataka omogućava organizirano i sigurno pohranjivanje informacija te jednostavan pristup i manipulaciju podacima. U ovom je radu korištena SQL baza podataka (*engl. Structured Query Language*), a na slici 4.5. prikazan je dijagram te baze.



Slika 4.5. Dijagram baze podataka

Tablice *Alarm* i *AlarmCode* pohranjuju poruke upozorenja (L1L2_ALARMS). Moguća su upozorenja o neispravnom radu sustava (neispravan rad izvršnih članova i senzora) i kritične razine tekućina u spremnicima.

Tablica *DynamicData* sadrži informacije senzora i aktutora. Trajno pohranjivanje ovih podataka omogućava naknadnu indentifikaciju procesa i izradu matematičkog modela na temelju podataka.

Tablica *Message* i *MessageHeader* služe za komunikaciju između Web aplikacije i komunikacijskog servisa. Korištenjem ovih tablica omogućena je razmjena poruka između web aplikacije i komunikacijskog servera. Komunikacijski servis ciklički čita nove podatke pohranjene

u tablici *Message* i, na temelju tipa poruke, odlučuje u koji podatkovni blok PLC-a će zapisati podatke.

Tablica *Aux_val* sadrži informacije o minimalnoj i maksimalnoj dopuštenoj vrijednosti pojedinih parametara procesa (ograničenja senzora i aktuatora).

Tablica *Notification* služi za prikazivanje obavijesti u web aplikaciji prilikom promjene statusa postrojenja, gubitka veze s PLC-om i sl.

4.3. Razvoj komunikacijskog sustava

Nakon što su definirane sve poruke koje će se razmjenjivati između PLC-a i servera te nakon što je izgrađena baza podataka, sljedeći je korak stvoriti robustan sustav servisnih aplikacija. Ovaj sustav treba osigurati pouzdanu razmjenu podataka i optimalnu komunikaciju između svih komponenti sustava.

Sustav servisnih aplikacija mora biti dizajniran tako da osigura pravilan rad aplikacije u slučaju nepredviđenih okolnosti, kao što su prekid veze s bazom podataka ili prekid veze s PLC-om. Bez obzira na situaciju, aplikacija mora nastaviti s radom bez pogrešaka i prekida, čime se osigurava stabilnost i pouzdanost cijelog sustava.

Implementacija ovakvog sustava uključuje sljedeće:

- **Zapisivanje događaja i praćenje:** kontinuirano praćenje statusa aplikacija i bilježenje svih događaja, što omogućava brzu identifikaciju i rješavanje problema.
- **Automatski oporavak:** mehanizmi za automatski oporavak u slučaju prekida veze, koji omogućuju nastavak rada čim se veza ponovno uspostavi.

Servisne aplikacije razvijene su u C# programskom jeziku unutar *Visual Studio 2022* razvojnog okruženja. One nemaju grafičko korisničko sučelje, ali posjeduju ključne metode poput *start*, *stop*, i *onTimer* za upravljanje njihovim radom. Ove metode omogućuju pokretanje i zaustavljanje servisa te periodično izvršavanje zadataka putem brojača vremena.

Komunikacija s PLC-om realizirana je putem *Ethernet-a* koristeći metode PUT i GET za razmjenu podataka. Metoda GET omogućava dohvaćanje podataka iz PLC-a, dok metoda PUT omogućava slanje podataka PLC-u. Taj mehanizam razmjene podataka omogućava biblioteka *Siemens S7.Net plus* o kojoj je nešto više rečeno u poglavlju 4.3.1.

Za komunikaciju između različitih servisnih aplikacija korišten je posrednik poruke RabbitMQ koji je detaljnije objašnjen u poglavlju 4.3.2

4.3.1. Siemens S7.Net Plus biblioteka

S7.Net upravljački je program za PLC namijenjen isključivo za *Siemens* PLC-ove i koji podržava komunikaciju isključivo putem *Etherneta*. Za ostvarenje komunikacije s PLC-om potrebno je definirati IP adresu PLC-a te fizičku konfiguraciju (engl. *rack*) i poziciju modula unutar PLC sustava (engl. *slot*).[11]

S obzirom da za laboratorijsko postrojenje skladištenja tekućine u spremnicima, postoji samo jedan PLC (S7-1500), komunikaciju je moguće ostvariti kao što je prikazano na slici 4.6.

Linija Kôd

```
1:       String ip = 192.168.100.21; short rack = 0; short slot = 1;
2:       Plc plc = new Plc(CpuType.S71500, ip, rack, slot); //inicijalizacija
3:       plc.Open(); //uspostavljanje veze
4:       plc.Close(); //zatvaranje veze
```

Slika 4.6. Primjer ostvarenja veze s PLC-om

Za čitanje i pisanje u podatkovne blokove PLC-a dostupno je nekoliko metoda. Slika 4.7. prikazuje primjer koda za čitanje i pisanje u podatkovnom bloku PLC-a. Vrlo je važno da podatci koji se upisuju ili čitaju iz podatkovnih blokova budu definirani varijablama u istom redosljedu i istim tipom.

Linija Kôd

```
1:       public class ExampleData {
2:           public short integer1 { get; set; }
3:           public bool bool1 { get; set; }
4:           public bool bool2 { get; set; }
5:           public short integer2 { get; set; }
6:       }
7:       public void ReadFromDB(short DB, short offsetAddr) {
8:           ExampleData exampleData = new ExampleData();
9:           _plc.ReadClass(exampleData, DB, offsetAddr);
10:       }
11:       public void WriteToDB(short DB, short offsetAddr) {
12:           ExampleData exampleData = new ExampleData();
13:           _plc.WriteClass(exampleData, DB, offsetAddr);
14:       }
```

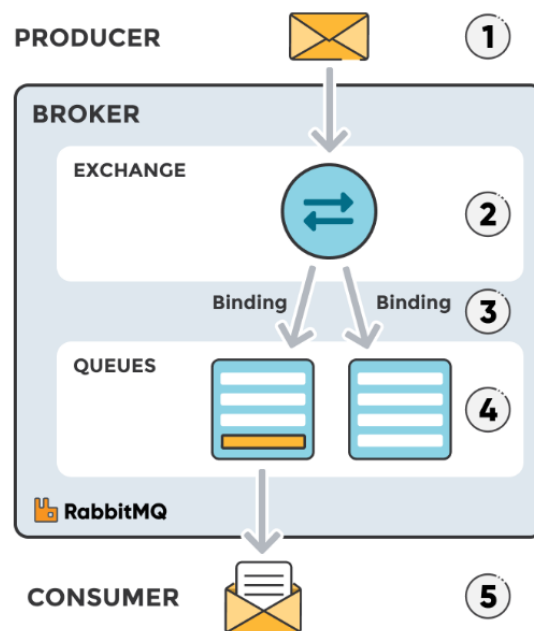
Slika 4.7. Primjer pisanja i čitanja iz podatkovnih blokova PLC-a

4.3.2. Posrednik poruka

RabbitMQ je softver za upravljanje redovima poruka, poznat i kao posrednik poruka ili upravitelj redova. Ukratko, to je softver gdje se definiraju redovi na koje se aplikacije povezuju kako bi poslale jednu ili više poruka. Poruka može sadržavati informacije o procesu ili zadatku koji treba započeti na drugoj aplikaciji, ili jednostavno tekstualnu poruku. Softver za upravljanje redovima pohranjuje poruke dok se aplikacija potrošač/primatelj (engl. *Consumer*) ne poveže i preuzme poruku s reda. Aplikacija primatelj tada obrađuje tu poruku. [12]

Pomoću posrednika poruka moguće je rasteretiti određene dijelove programske podrške na način da se pojedini zadatci delegiraju na preostale dijelove sustava. Ovo je posebno važno za sustave u kojima je potrebno obrađivati informacije u stvarnom vremenu.

Princip rada posrednika poruka prikazan je na slici 4.8.



Slika 4.8. Princip rada *RabbitMQ* posrednika poruka

Izvor: [Part 1: RabbitMQ for beginners - What is RabbitMQ? - CloudAMQP](#)

Osnovni tijek slanja i primanja poruke je sljedeći:

1. Proizvođač stvara poruku i šalje ju *RabbitMQ* brokeru.

2. Unutar *RabbitMQ*-a, poruka prvo dolazi do sustava razmjene (engl. *Exchange*). Sustav razmjene je komponenta koja odlučuje kako će se poruka usmjeriti do jednog ili više redova na temelju zadanih pravila.
3. Sustav razmjene koristi veze (engl. *Binding*) kako bi usmjerio poruke prema odgovarajućim redovima. Veze definiraju koji redovi će primiti određene poruke iz sustava razmjene.
4. Redovi su mjesta gdje poruke čekaju da budu preuzete od strane potrošača/primatelja. Poruke ostaju u redovima dok ih potrošači/primatelji ne preuzmu.
5. Potrošač/primatelj preuzima poruke iz reda i obrađuje ih prema svojim potrebama. Nakon obrade, poruka se može ukloniti iz reda.

Slika 4.9. prikazuje primjer kôda za slanje i primanje poruka putem *RabbitMQ*-a.

Linija Kôd

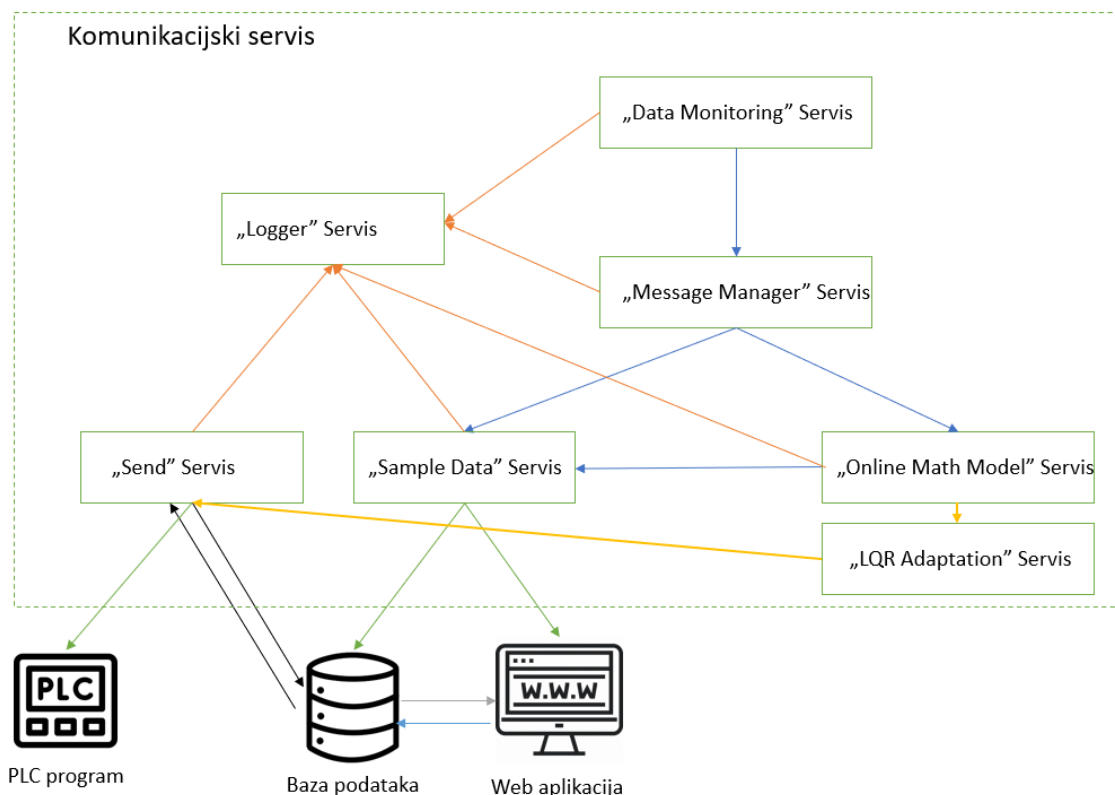
```

1      public void SendMessage(string routingKey, MessageBase message) {
2          var properties = _model.CreateBasicProperties();
3          properties.Persistent = true;
4          properties.Priority = message.Priority;
5          if (_routingKeyQueueMap.TryGetValue(routingKey, out var
queueName)) {
6              byte[] body =
Encoding.UTF8.GetBytes(JsonConvert.SerializeObject(message));
7              _model.BasicPublish(_exchangeName, routingKey, properties,
body);
8          } else {
9              // in case a routing key doesn't exist
10             }
11         }
12     public async Task ReadMessageFromQueueAsync(string queue,
Func<byte[], Task> onMessageReceived) {
13         var consumer = new AsyncEventingBasicConsumer(_model);
14         if (_connection is not null && _connection.IsOpen) {
15             consumer.Received += async (ch, ea) => {
16                 byte[] body = ea.Body.ToArray();
17                 await onMessageReceived(body);
18                 await Task.CompletedTask;
19                 _model.BasicAck(ea.DeliveryTag, false);
20             };
21             _model.BasicConsume(queue, false, consumer);
22         }
23     Await Task.CompletedTask;
24     }

```

Slika 4.9. Primjer slanja i čitanja poruka putem *RabbitMQ* sustava

Komunikacijski sustav za razmjenu podataka između servera i PLC-a sastoji se od nekoliko servisnih aplikacija. Korištenjem više servisnih aplikacija moguće je rasteretiti rad pojedinačnih komponenti. Delegacija zadataka vrši se preko već spomenutog sustava posrednika poruka RabbitMQ-a. Slika 4.10. prikazuje sve servisne aplikacije i smjerove komunikacije.



Slika 4.10. Pregled komunikacijskog servisa

Uloga je „*Data Monitoring*“ servisne aplikacije ciklički pratiti promjene u podatkovnim blokovima PLC-a. Drugim riječima, ovaj servis provjerava postoje li nove poruke te ukoliko postoje, šalje poruku „*Message Manager*“ servisu. „*Message Manager*“ servis zatim čita podatke iz kružnog spremnika. Ako se radi o podacima procesa (vrijednosti sa mjernih i izvršnih članova), prosljeđuje poruku s tim informacijama „*Online Math Model*“ servisu koji predviđa vrijednosti izlaza, varijabli stanja te ažurira parametre matematičkog modela. „*LQR Adaptation*“ adaptira parametre regulatora na temelju ažuriranog matematičkog modela.

„*Sample Data*“ servisna aplikacija prikuplja sve poruke te ih sprema u bazu podataka ili ih izravno prosljeđuje web aplikaciji.

„Send“ servisna aplikacija zadužena je za slanje podataka PLC-u, konkretno slanje zahtjeva za upravljanje sustavom skladištenja tekućine u spremnicima, ažuriranje parametara regulatora i postavljanje referentne veličine procesa.

Uloga „Logger“ servisne aplikacije je zapisivanje i pohranjivanje svih važnih događaja i podataka u sustavu. Bilježi promjene u podatkovnim blokovima PLC-a, greške i iznimke koje se pojave tijekom rada sustava i aktivnosti drugih servisnih aplikacija.

4.4. Opis i funkcionalnost aplikacijskog programskog sučelja

Ovo poglavlje opisuje aplikacijsko programsko sučelje (API) izgrađeno korištenjem *Flask*-a, mikro web framework-a za python, koji omogućava različite analize sustava upravljanja. API pruža niz krajnjih točaka (engl. *endpoints*) koje omogućuju korisnicima slanje zahtjeva za specifične analize sustava upravljanja kao što su krivulja mjesta korijena, *Bodeov* dijagram, odziv na step pobudu, izračun polova zatvorenog regulacijskog kruga, izračun LQR regulatora te dinamičkih pokazatelja kakvoće regulacije.

Web aplikacija koristi *HttpClient* za slanje HTTP zahtjev prema ovom API-ju kako bi dobila potrebne informacije za prikazivanje korisnicima. Kada korisnik na web sučelju inicira određenu analizu (npr. krivulja mjesta korijena ili odziv na step pobudu), web aplikacija šalje odgovarajući zahtjev API-ju. API obrađuje zahtjev, izvodi potrebne izračune i vraća rezultate natrag web aplikaciji. Ti rezultati se zatim prikazuju korisniku u prikladnom formatu.

Razlog izgradnje ovog API-ja jest omogućiti izvođenje složenih izračuna i analiza na serveru čime se rasterećuje klijent. Korištenjem ovog API-ja, web aplikacija može pružiti naprednu analizu i vizualizaciju podataka upravljačkog sustava, omogućujući korisnicima detaljnu i interaktivnu analizu performansi sustava.

Primjer pozivanja krajnje točke prikazan je na slici 4.11. i 4.12.

Linija **Kôd**

```
1        @app.route('/calculate', methods=['POST'])
2        def calculate():
3            data = request.get_json()
4            # For demonstration return some dummy data
5            result = {
6                'settling_time': 12.5,
7                'rise_time': 3.2,
```

```

8         'overshoot': 15.7,
9         'error': 0.02
10    }
11    return jsonify(result)

```

Slika 4.11. Primjer API-ja

Linija Kôd

```

1     private async Task<Data> CalculateAsync() {
2         using (var httpClient = new HttpClient()) {
3             var data = new {
4                 A = _mathematicalModel.A,
5                 B = _mathematicalModel.B,
6                 C = _mathematicalModel.C,
7                 D = _mathematicalModel.D,
8             };
9             var json = JsonConvert.SerializeObject(data);
10            var content = new StringContent(json, Encoding.UTF8,
11                "application/json");
12            var response = await
13                httpClient.PostAsync("http://127.0.0.1:5000/calculate", content);
14            if (response.IsSuccessStatusCode) {
15                var responseBody = await
16                response.Content.ReadAsStringAsync();
17                var body =
18                JsonConvert.DeserializeObject<Data>(responseBody);
19                return body;
20            }
21            else {
22                throw new Exception("Error: Unable to receive data!");
23            }
24        }
25    }

```

Slika 4.12. Primjer poziva API-ja na strani klijenta

4.5. Izrada Web Aplikacije

Za izradu korisničkog sučelja koristi se ASP.NET Core *framework*, koji omogućava stvaranje modernih i prilagodljivih web aplikacija. Ova platforma pruža infrastrukturu za rukovanje zahtjevima korisnika, upravljanje bazama podataka i implementaciju poslovne logike.

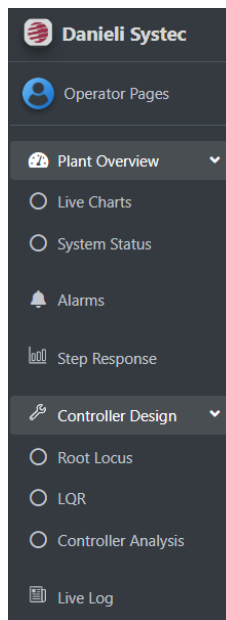
Za prezentacijski sloj web aplikacije koriste se HTML i CSS opisni jezici. HTML se koristi za strukturiranje sadržaja web stranice, definirajući elemente poput zaglavlja, slika i poveznica. CSS se koristi za uređivanje tih elemenata, omogućavajući prilagodbu izgleda web stranice prema željenim specifikacijama. Zahvaljujući CSS-u, web stranica ima dosljedan i vizualno privlačan izgled na različitim uređajima i rezolucijama ekrana.

JavaScript programski jezik ključan je za interaktivne elemente i vizualizaciju podataka. Uz pomoć *JavaScript* biblioteka poput *Chart.js* ili *three.js*, moguće je kreirati dinamične grafičke prikaze i 3D vizualizacije koje korisnicima prikazuju podatke na intuitivan i pregledan način.

Za programiranje na strani poslužitelja koristi se C# programski jezik. C# programski jezik odabran je zbog svoje efikasnosti i snažne podrške unutar .NET ekosustava.

4.5.1. Izgled web aplikacije

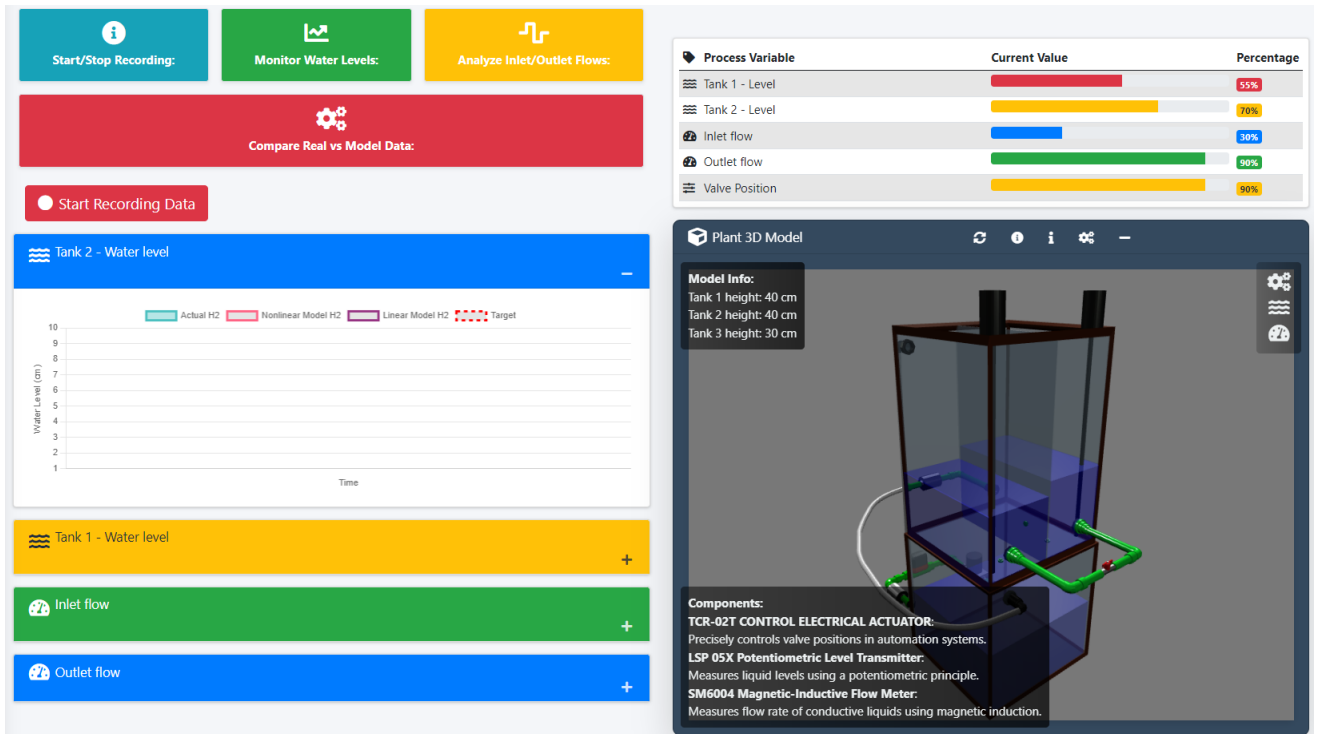
Slika 4.13. prikazuje izbornik stranice web aplikacije čije su funkcionalnosti objašnjene u nastavku,



Slika 4.13. Navigacijski izbornik web aplikacije

Odjeljak „*Plant Overview*“ sadrži stranice koje daju sveobuhvatan pregled stanja postrojenja. „*Live Charts*“ stranica (slika 4.14.) služi za prikaz svih relevantnih procesnih veličina u stvarnom vremenu te usporedbu s matematičkim modelom. Ova stranica omogućava korisnicima pregled različitih grafičkih prikaza poput razine tekućine u spremnicima, protoke i otvorenost ventila. Uz grafičke prikaze, stranica sadrži i 3D vizualizaciju postrojenja koje olakšava predodžbu o

trenutnom stanju sustava. Crveni gumb „Start Recording Data“ omogućava korisnicima snimanje podataka i spremanje u .csv format.



Model Data Comparison

Variable	Real Data	Model Data	Error
Tank 1: Water Level	0	0	0
Tank 2: Water Level	0	0	0
Inlet Flow	0	0	0
Outlet Flow	0	N/A	N/A

Real-time Data

Variable	Value	Percentage	Status
Tank 1 Level	0 cm	0%	Stable
Tank 2 Level	0 cm	0%	Stable
Inlet Flow	0 L/min	0%	Stable

Model Prediction Accuracy

Variable	Accuracy
Tank 1 Accuracy	90%
Tank 2 Accuracy	85%
Inlet Flow Accuracy	80%

Slika 4.14. Izgled „Live Charts“ stranice

Stranica „System Status“ (slika 4.15.) prikazuje ostvarenu vezu s PLC-om i status različitih komponenti sustava. Na vrhu stranice prikazan je dijagram veze između PLC-a i računala, zajedno

s informacijama o tipu PLC-a , procesorskoj jedinici, IP adresi, fizičkoj konfiguraciji i trenutnom statusu. Ispod dijagrama veze nalaze se „lampice“ koje prikazuju status kritičnih komponenti sustava.

The screenshot displays a web interface for monitoring a PLC system. It is divided into two main sections: 'PLC Connection' and 'Plant Status'.

PLC Connection Section:

- On the left, there is a diagram showing a PLC rack connected to a laptop via a network connection, with a 'Connected' status indicator below it.
- On the right, a table titled 'PLC Information' provides the following details:

PLC Information	
Type:	Siemens S7-Net
CPU-Type:	S7-1500
IP-Address:	192.168.131.171
Rack:	0
Slot:	1
Status:	Active

Plant Status Section:

- This section contains six status indicators, each with a colored icon and a label:
 - Pump Active:** Green icon with a water drop.
 - Level Switch:** Red icon with a toggle switch.
 - Valve Operating Range:** Green icon with a crosshair.
 - Tank 1 critical Level:** Red icon with wavy lines.
 - Tank 2 critical Level:** Red icon with wavy lines.
 - System Ready:** Green icon with a gear.

Slika 4.15. Izgled „System status“ stranice

„Alarms“ stranica (slika 4.16.) pruža pregled svih upozorenja postrojenja. Na vrhu stranice nalaze se polja za filtriranje alarma prema vremenskom periodu i tipu alarma. Na dnu stranice nalazi se histogram koji prikazuje učestalost pojavljivanja pojedinih alarma.

Filter Alarms

Date Time From: dd/mm/yyyy --:-- Date Time To: dd/mm/yyyy --:-- Alarm Code ID: Severity: All

Enable Filter Disable Filter Refresh

Alarm Id	Alarm Code	Log Date Time	Description	Comment	Severity	
1	1	03/06/2024 15:13:00	Tank 1 is fully empty (LowLow).	Tank 1 reached its minimum water level. Please refill immediately.	Critical	Edit
2	2	03/06/2024 19:41:00	Tank 1 is almost empty (Low).	Tank 1 is running low on water. Consider refilling soon.	Warning	Edit
3	3	03/06/2024 22:41:09	Tank 1 water level exceeding safe limits (HighHigh)	Water level in Tank 1 is dangerously high. Check for overflow.	Critical	Edit
4	4	03/06/2024 22:41:09	The water level in tank 1 is approaching a critical point (High)	Water level in Tank 1 is rising rapidly. Monitor closely.	High	Edit
5	5	03/06/2024 22:41:09	Tank 2 is fully empty (LowLow).	Tank 2 is completely empty. Fill it up as soon as possible.	Critical	Edit
6	6	03/06/2024 22:41:09	Tank 2 is almost empty (Low).	Tank 2 is almost empty. Refill to avoid running out of water.	Warning	Edit
7	7	03/06/2024 22:41:09	Tank 2 water level exceeding safe limits (HighHigh)	Water level in Tank 2 has exceeded safe limits. Investigate immediately.	Critical	Edit
8	8	03/06/2024 22:41:09	The water level in tank 2 is approaching a critical point (High)	Water level in Tank 2 is rising rapidly. Monitor closely.	High	Edit
9	9	03/06/2024 22:41:09	The water level in bottom tank is critically low	Water level in the bottom tank is critically low. Immediate action required.	Critical	Edit
10	10	03/06/2024 22:41:09	Inlet flow fault!	There is a fault in the inlet flow. Investigate and rectify.	Critical	Edit

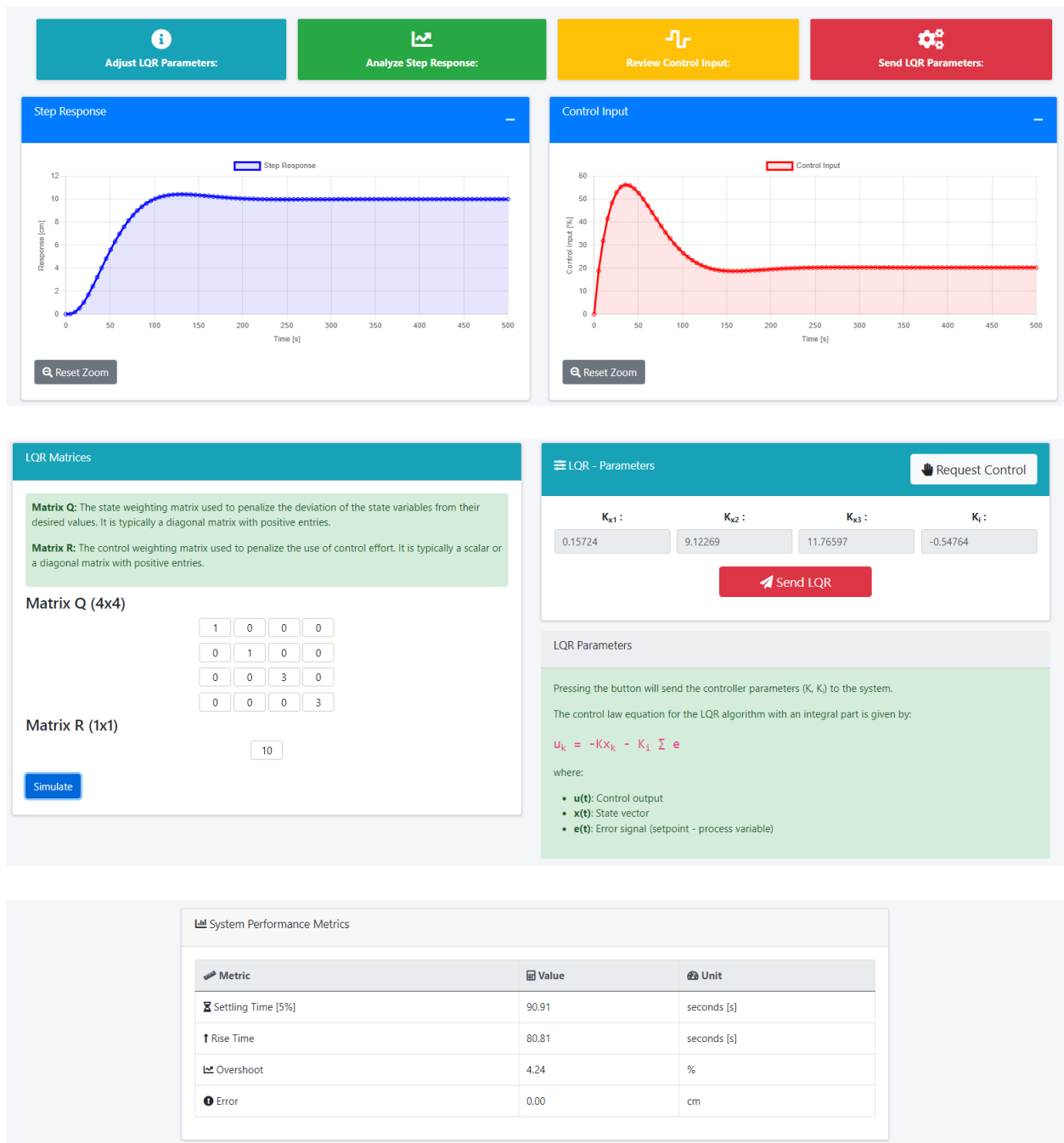
Previous 1 2 3 Next



Slika 4.16. Izgled „Alarms“ stranice

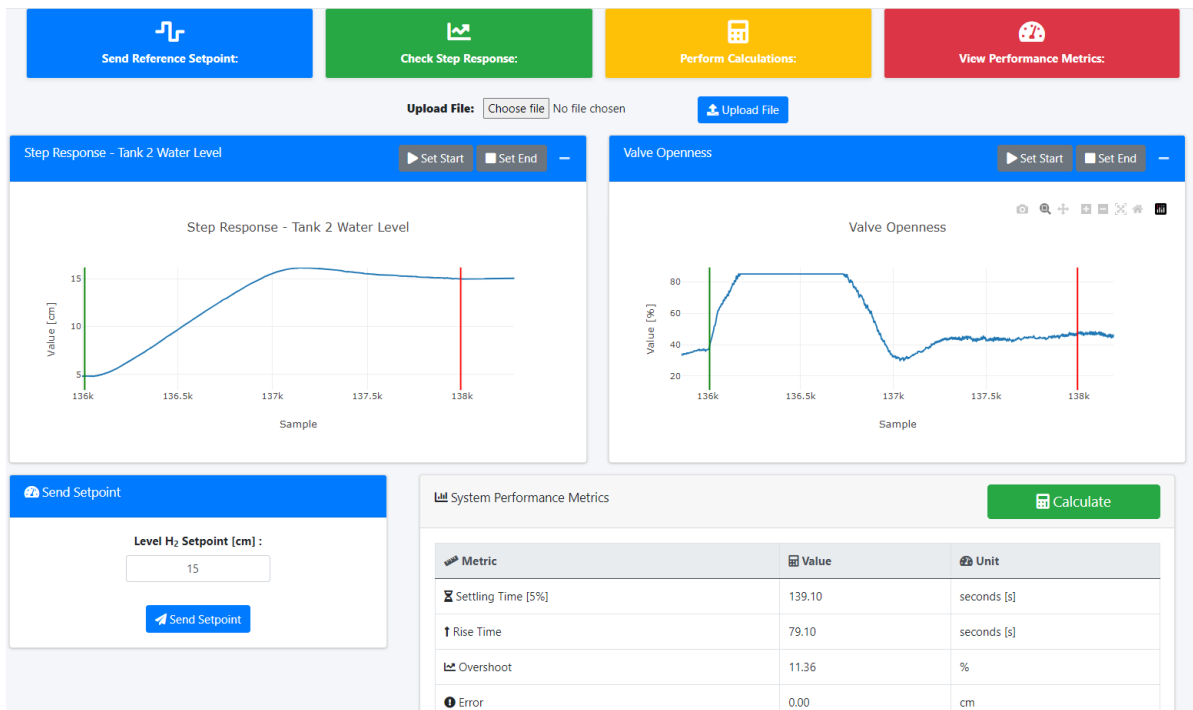
Odjeljak „*Controller Design*“ omogućava korisniku sintezu regulatora i analizu sustava. Stranica „*Linear Quadratic Regulator*“ (slika 4.17.) pruža korisnicima konfiguriranje LQR matrica sa sintezu upravljačkog zakona, simulaciju odziva sustava i upravljačkog signala. Ti grafički prikazi omogućavaju korisnicima vizualno procjenjivanje ponašanja sustava. Pri dnu stranice nalazi se tablica koja opisuje kakvoću regulacije.

LQR parametri izračunavaju se na temelju diskretnog matematičkog modela, odnosno korištena je implementacija diskretnog LQR-a objašnjenog u poglavlju 3.8.1. Te je parametre potom moguće poslati PLC-u pritiskom na „*Send LQR*“ dugme



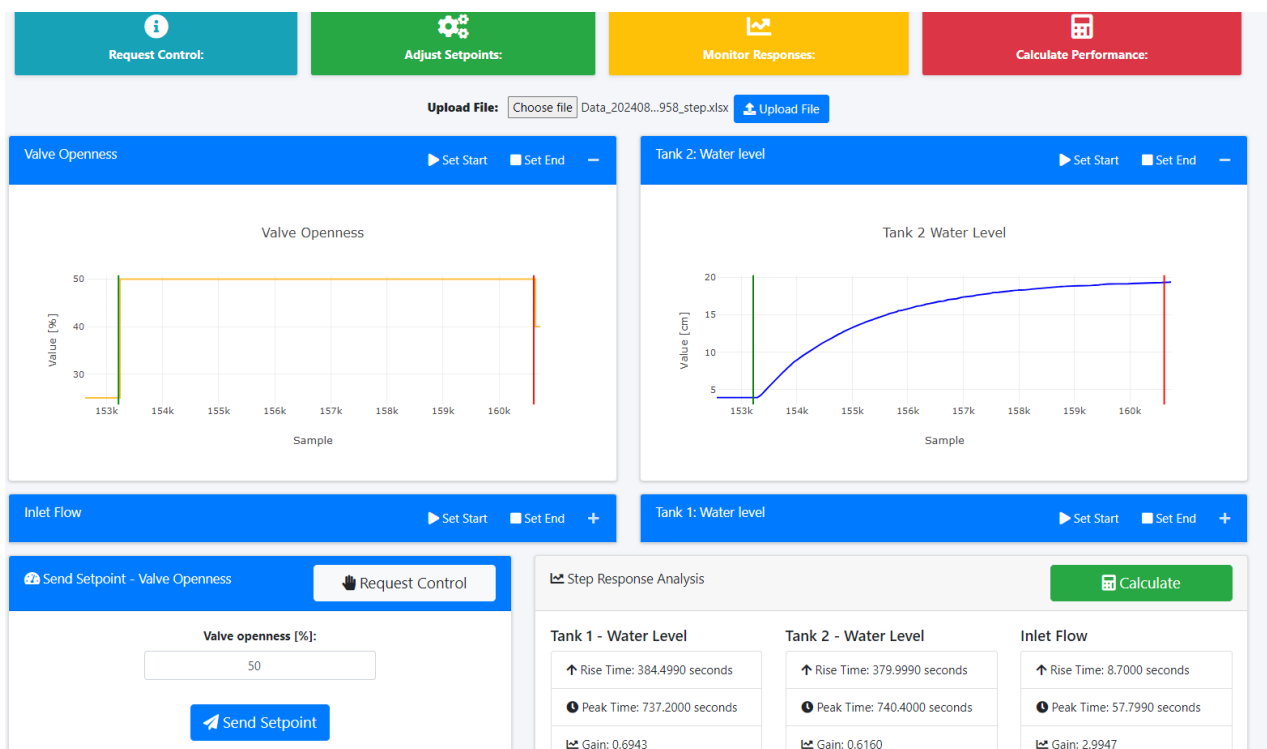
Slika 4.17. Sučelje za LQR

Stranica „Controller Analysis“ (slika 4.18.) omogućava testiranje regulacije na stvarnom postrojenju. Nudi sučelje za slanje referentne veličine i prikaz odziva izlazne veličine (razina tekućine u drugom spremniku). Ova stranica također uključuje tablicu s metrikama performansi sustava, omogućujući usporedbu rezultata stvarnog postrojenja s rezultatima matematičkog modela.



Slika 4.18. Izgled „Controller Analysis“ stranice

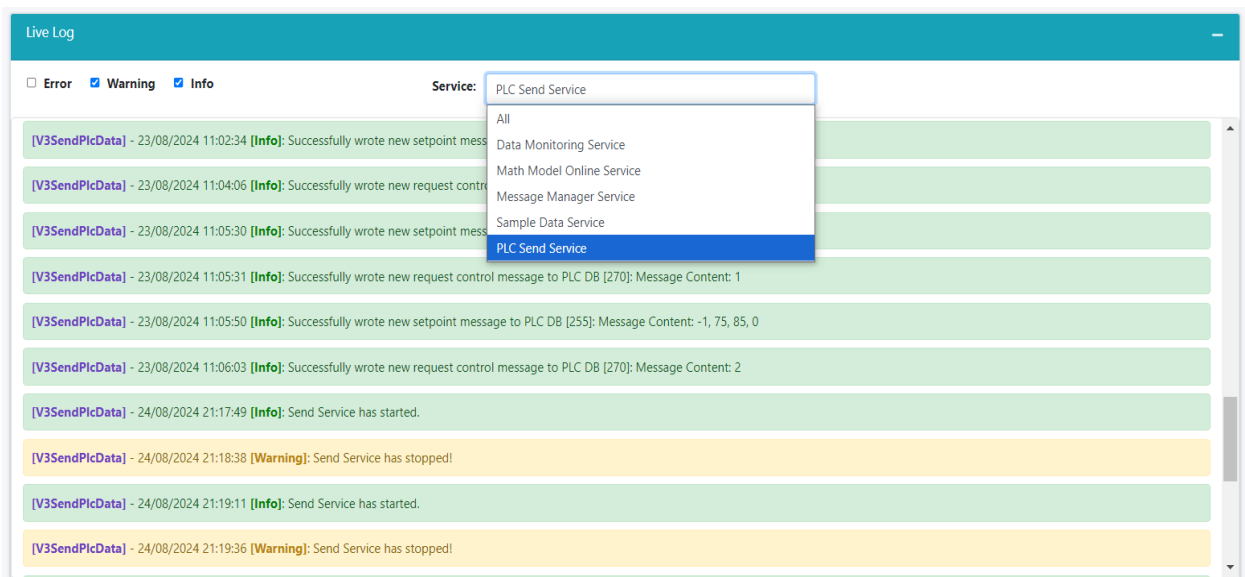
Stranica „Step Response“, prikazana na slici 4.19., omogućuje korisnicima postavljanje otvorenosti ventila kako bi se analizirala pobuda procesnih veličina na skokovitu promjenu ulaza. Pri dnu stranice se nalazi tablica koja izračunava pojačanje procesa, vrijeme porasta te vrijeme maksimalnog nadvišenja.



Slika 4.19. Izgled „Step response“ stranice

Stranica "Live Log" omogućava pregled rada komunikacijskih servisa, uključujući njihovo pokretanje i zaustavljanje, te evidentiranje eventualnih pogrešaka. Na ovaj način moguće je utvrditi mjesto i razlog nastanka pogreške što pomaže u brzom dijagnosticiranju i otklanjanju problema te osigurava nesmetan rad sustava. Na vrhu stranice nalazi se filter koji omogućava filtriranje zapisa prema različitim kategorijama kao što su „info“, „error“ i „warning“. Također, moguće je filtrirati prikaz za pojedinačni servis, omogućujući fokusirano praćenje specifičnih događaja i stanja tog servisa.

Slika 4.20. prikazuje izgled te „Live Log“ stranice.



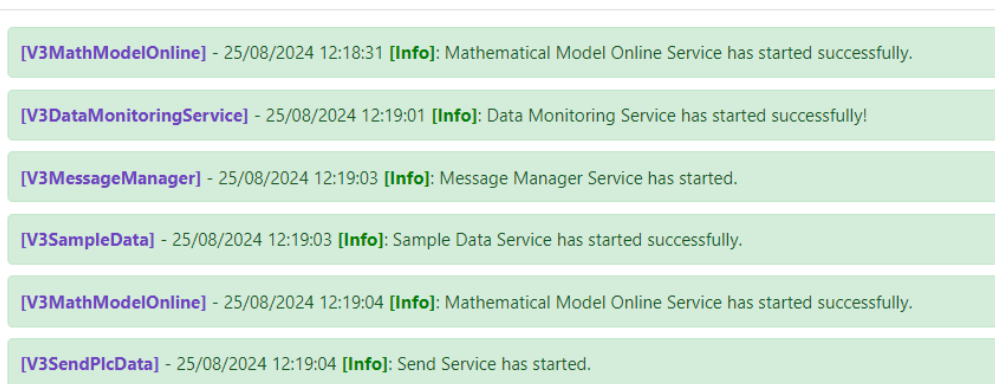
Slika 4.20. Izgled „Live Log“ stranice

5. RAD APLIKACIJE I REZULTATI REGULACIJE

Nakon razvoja programske podrške, opisane u prethodnom poglavlju, potrebno je provesti testiranje rada aplikacije i njenih funkcionalnosti. Ovaj proces uključuje provjeru prikaza podataka u stvarnom vremenu, uspješnosti ostvarivanja veze s PLC-om, prijem statusnih podataka s PLC-a te dobivanje obavijesti nakon pokretanja određenih akcija. Također, slijedi demonstracija rada LQR i RLS algoritama, s posebnim naglaskom na kvalitetu regulacije i efikasnost postignutih rezultata.

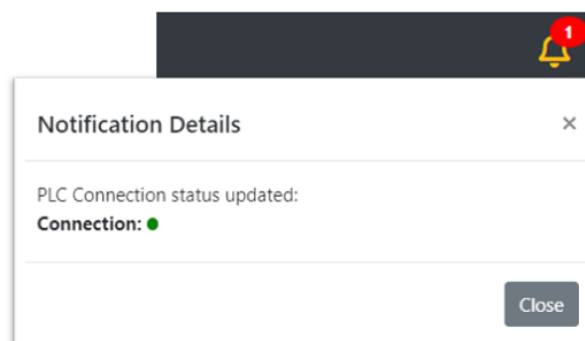
5.1. Prikaz osnovnih funkcionalnosti web aplikacije

Prilikom otvaranja web aplikacije, najprije je potrebno provjeriti rade li ispravno svi servisi opisani u poglavlju 4.3. Kao što je već rečeno, prikaz statusa komunikacijskih servisa moguće je provjeriti unutar „Live Log“ stranice. Slika 5.1. prikazuje kako su svi servisi ispravno pokrenuti.



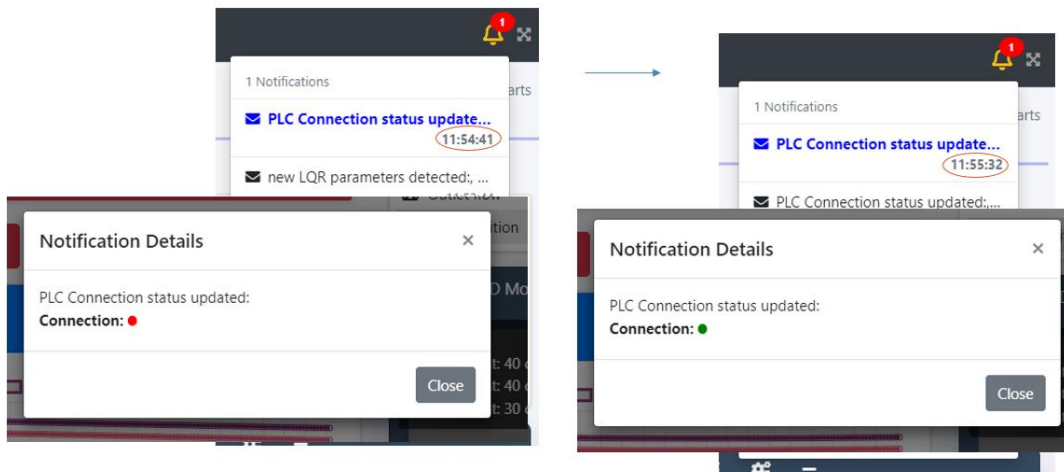
Slika 5.1. Statusne poruke komunikacijskog servisa

Nakon što su svi servisi pokrenuti, na aplikaciji bi trebala doći obavijest o uspješnom povezivanju s PLC-om. Obavijesti se prikazuju u gornjem desnom kutu web aplikacije (slika 5.2.).



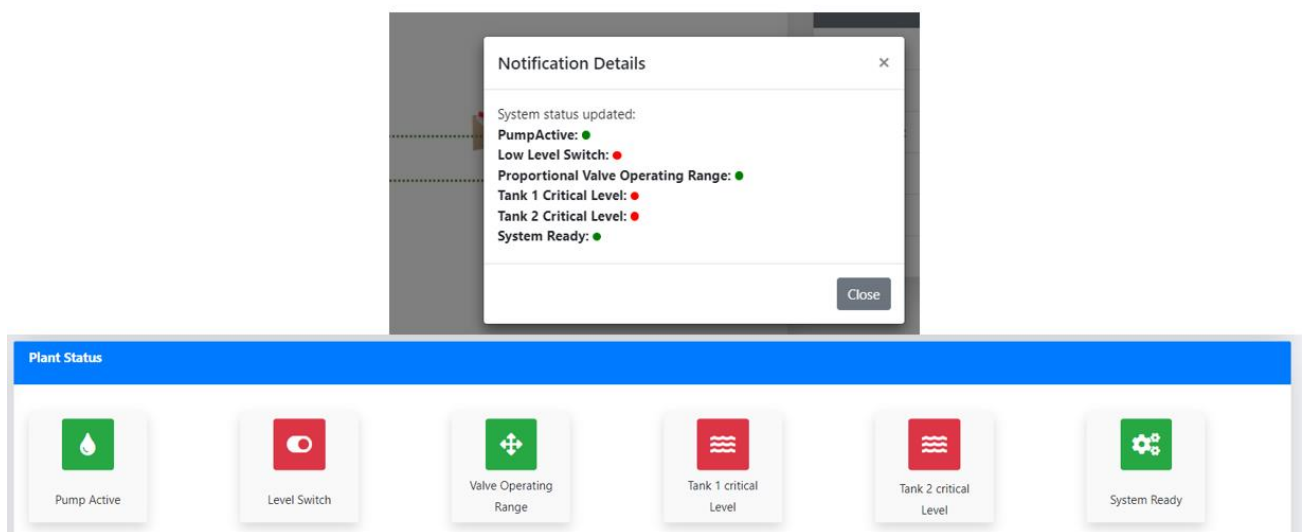
Slika 5.2. Prikaz obavijesti o uspostavi veze s PLC-om

Ukoliko dođe do gubitka veze s PLC-om, ubrzo potom dolazi odgovarajuća obavijest. Nakon gubitka veze, komunikacijski servisi automatski pokušavaju uspostaviti novu vezu, osiguravajući kontinuitet rada sustava. Slika 5.3. prikazuje nove obavijesti o gubitku veze i o automatskom oporavku.



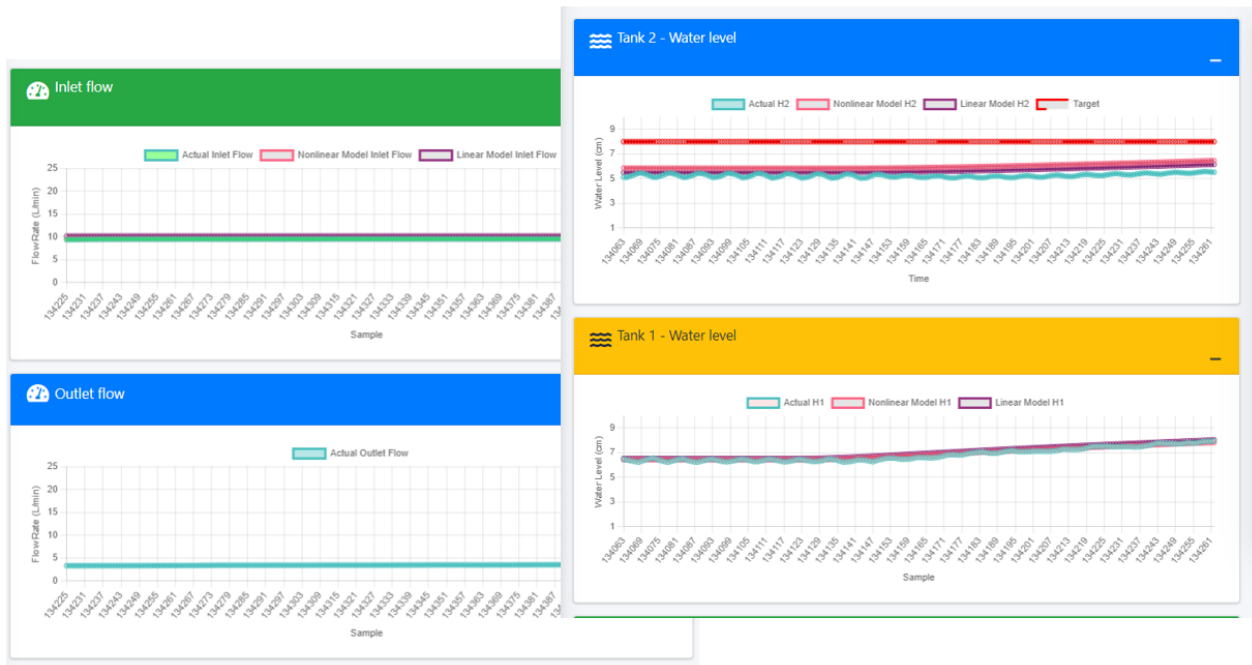
Slika 5.3. Prikaz obavijesti o gubitku veze i automatskom oporavku

Osim obavijesti o uspostavljenoj vezi s PLC-om, korisničko sučelje prikazuje i detaljne informacije o trenutnom stanju laboratorijskog postrojenja, uključujući rad crpke, ventilskog pozicionera i o kritičnim razinama tekućine u spremnicima (slika 5.4.).



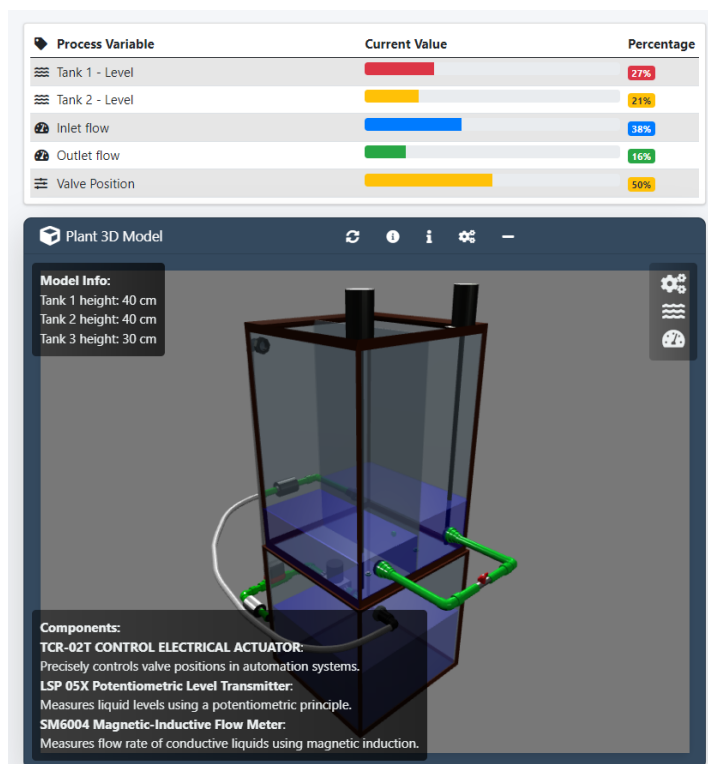
Slika 5.4. Status postrojenja

Ukoliko je sustav spreman (uključena crpka, otvoren ventilski pozicioner, nisu dostignute kritične razine u spremnicima), očekuje se pristizanje vrijednosti sa senzora, odnosno procesne veličine. Slika 5.5. prikazuje grafički i dinamičan prikaz procesnih veličina u stvarnom vremenu. Osim što se prikazuju vrijednosti sa senzora, također se prikazuju vrijednosti dobivene linearnim i nelinearnim matematičkim modelom koje proračunava dedikirani matematički servis.

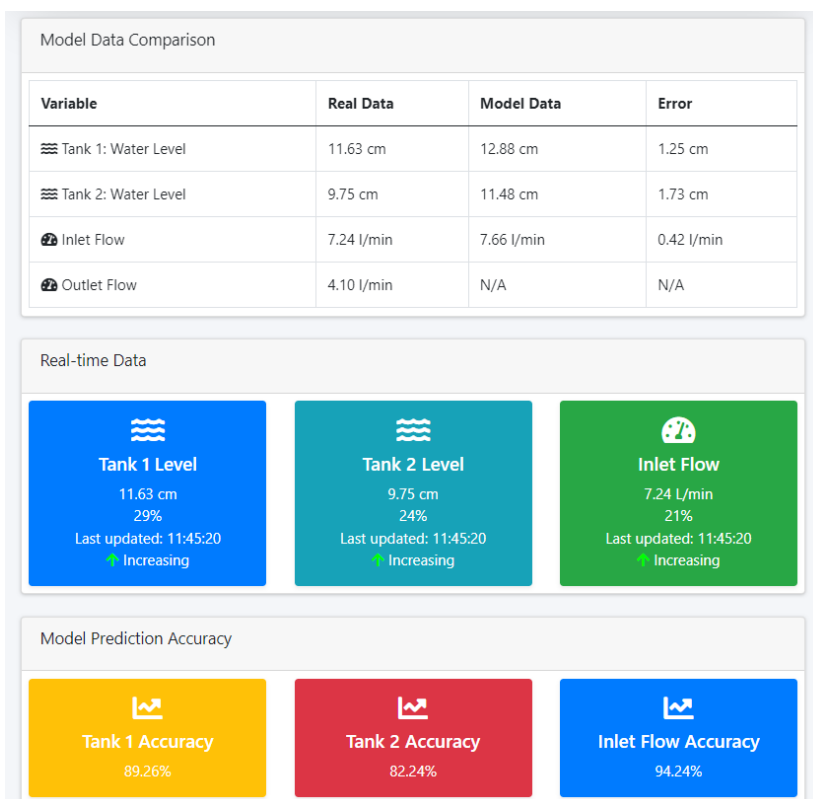


Slika 5.5. Grafički prikaz procesnih veličina

Osim grafičkog prikaza, ponuđena je i 3D vizualizacija postrojenja, prikazanog slikom 5.6., koja dinamički prikazuje razine tekućine u spremnicima. Uz to, moguće je i pratiti koliko je odstupanje matematičkog modela od stvarnih podataka te informacije o porastu ili smanjivanju pojedinih procesnih veličina što je prikazano na slici 5.7.

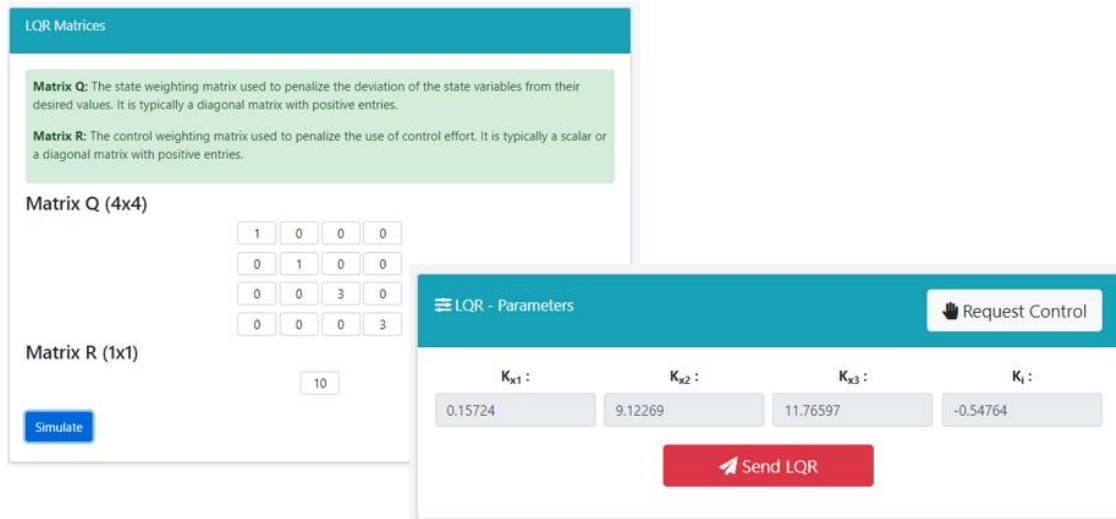


Slika 5.6. Vizualizacija 3D postrojenja

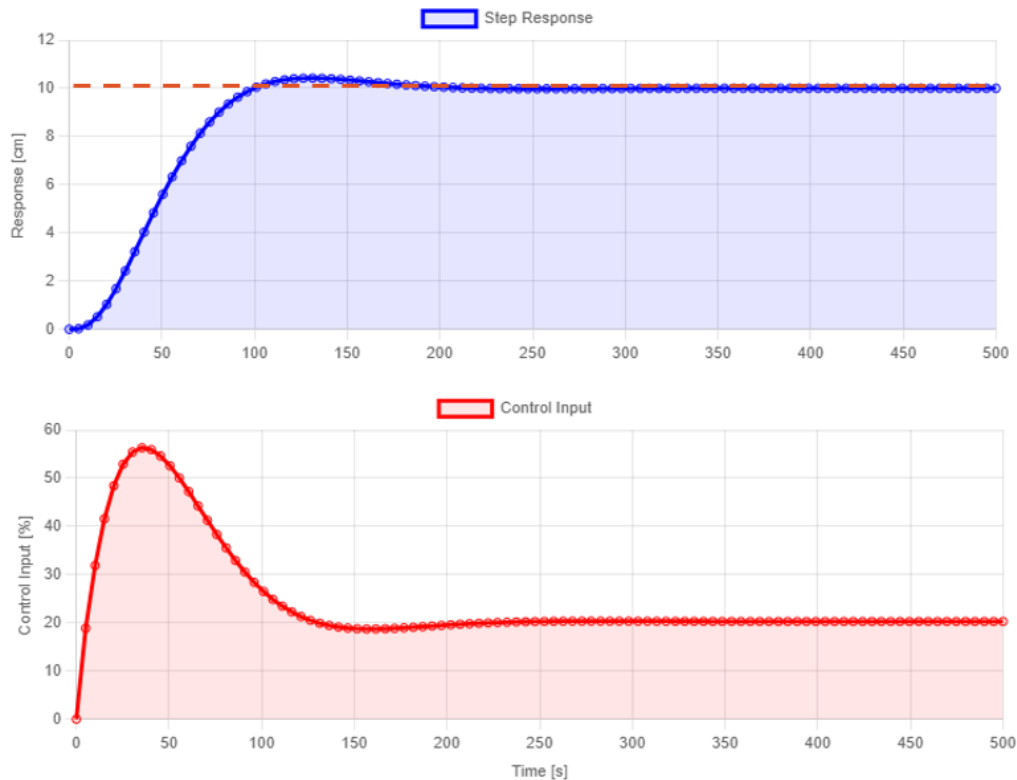


Slika 5.7. Usporedba stvarnih podataka sa matematičkim modelom

Ukoliko se želi regulirati razina tekućine u drugom spremniku, moguće je ručno konfigurirati parametre diskretnog LQR-a. Na stranici „LQR“ moguće je izračunati vrijednosti regulatora postavljanjem odgovarajućih vrijednosti matrice \mathbf{Q} i \mathbf{R} (uloga ovih matrica objašnjena je u poglavlju 3.8.1) kao što je prikazano na slici 5.8. Koeficijente regulatora proračunava API te uz njih još vraća i odziv regulirane veličine te promjenu ulaza. Odzivi su prikazani na slici 5.9.

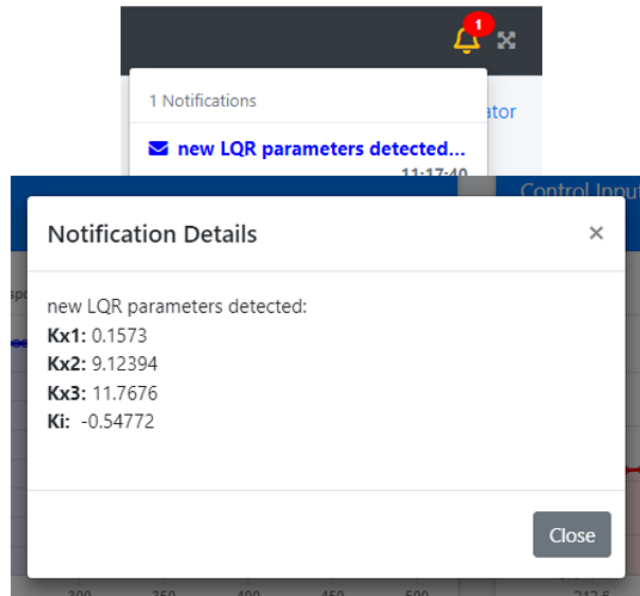


Slika 5.8. Sučelje za konfiguriranje parametara LQR regulatora



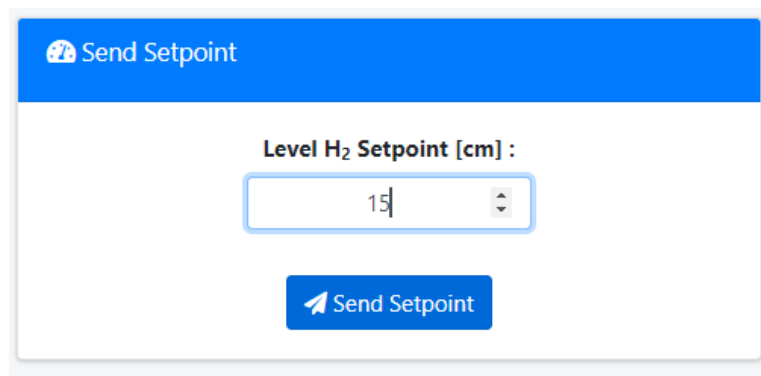
Slika 5.9. Odziv regulirane veličine i promjena ulazne veličine

Ukoliko je korisnik zadovoljan sa regulatorom, istog je moguće konfigurirati na PLC-u pritiskom na dugme „Send LQR“. Ukoliko je PLC primio te parametre, na web aplikaciji stiže nova obavijest koja sadrži informaciju o postavljanim parametrima (slika 5.10.).



Slika 5.10. Obavijest o novim parametrima LQR-a

Idući je korak slanje željene vrijednosti razine tekućine u drugom spremniku pritiskom na dugme „Send setpoint“ prikazanog na slici 5.11..

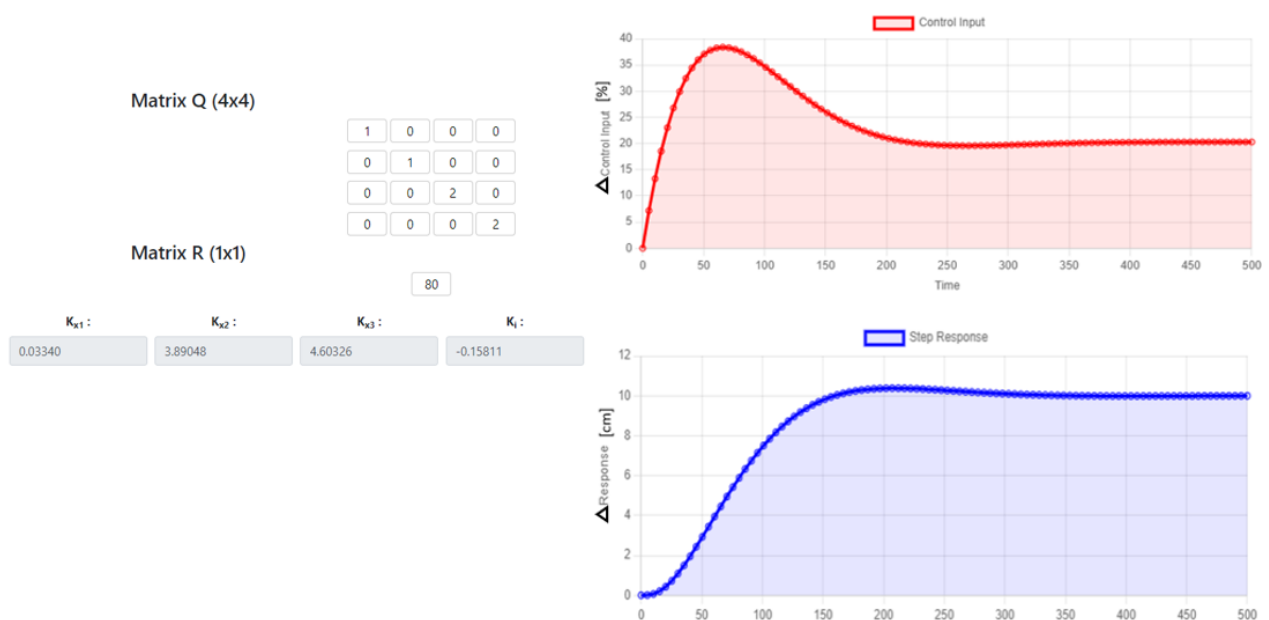


Slika 5.11. Postavljanje željene vrijednosti razine tekućine u drugom spremniku

5.2. Rezultati regulacije razine tekućine u spremniku

Nakon što su izračunati parametri diskretnog LQR-a (vrijeme diskretizacije je u skladu s postavljenim vremenom ciklusa izvođenja LQR-a na PLC-u), potrebno je testirati rad upravljačkog zakona na stvarnom postrojenju. U nastavku slijedi usporedba statičkih (pogreška u stacionarnom stanju) i dinamičkih (vrijeme porasta, vrijeme ustaljivanja, maksimalno nadvišenje) pokazatelja kakvoće regulacije između simuliranih i stvarnih podataka.

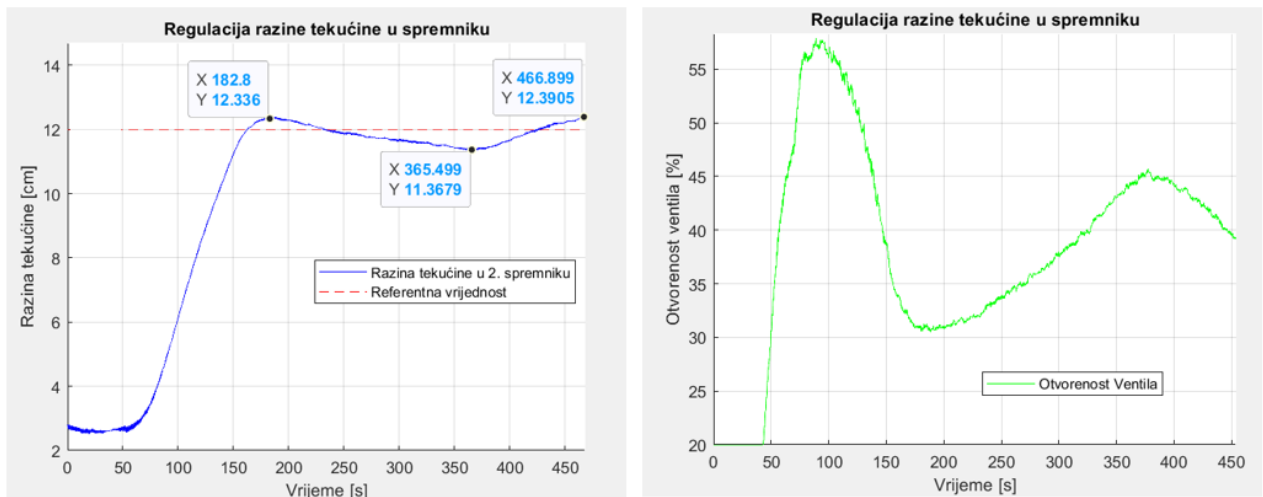
Slika 5.12. prikazuje dobiveni simulacijski odziv regulatora i regulirane veličine na temelju matematičkog modela lineariziranog u predviđenom radnom području.



Slika 5.12. Postavljanje LQR parametara i odziv simulacije

Rezultati dobiveni od laboratorijskog postrojenja su prikazani na slici 5.13., dok Tablica 5.1. prikazuje pokazatelje kakvoće regulacije i njihovu usporedbu s onima iz simulacije. Ti pokazatelji kakvoće regulacije slični su dobivenim iz simulacije. Maksimalno nadvišenje i vrijeme porasta su vrlo slični dok je vrijeme ustaljivanja bitno drugačije. Odstupanja u vremenu ustaljivanja su očekivana s obzirom na to da simulacija ne može u potpunosti obuhvatiti sve kompleksnosti stvarnog sustava, uključujući ograničenja aktuatora i senzora, vanjske smetnje i mjerne šumove.

U simulaciji je zabilježena maksimalna promjena otvorenosti ventila od 37%, dok je na stvarnom sustavu izmjerena promjena od 37.65%. Ovo predstavlja relativno malo odstupanje između simuliranih i stvarnih rezultata.

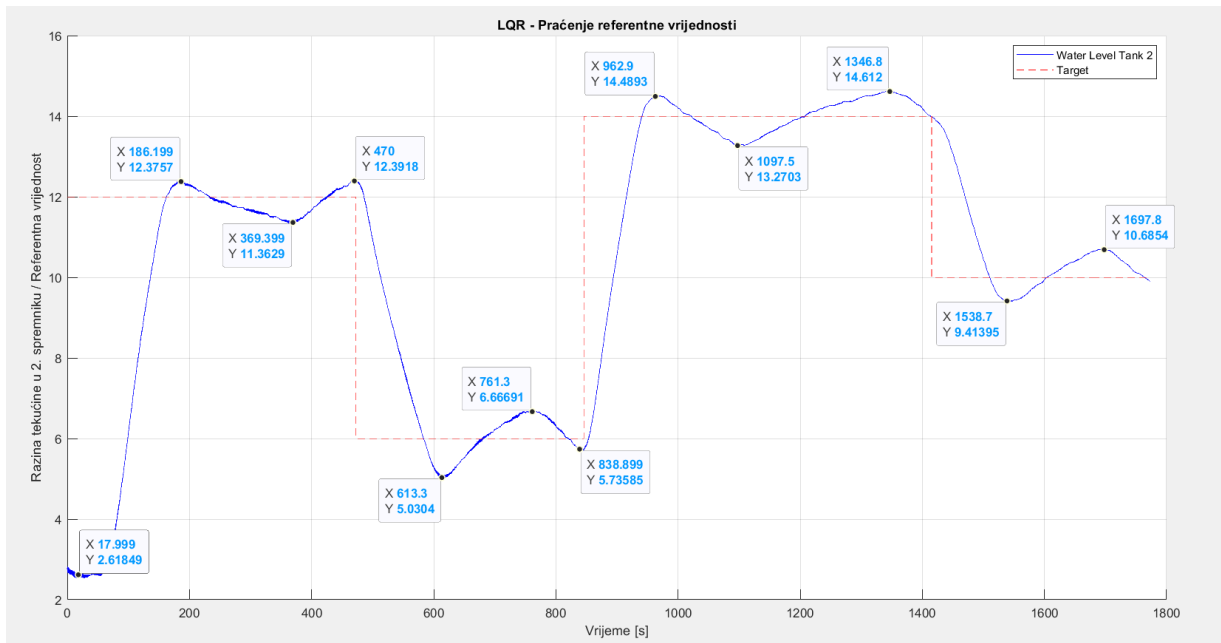


Slika 5.13. Regulacija razine tekućine u drugom spremniku i otvorenost ventila

Tablica 5.1. Usporedba pokazatelja kakvoće regulacije

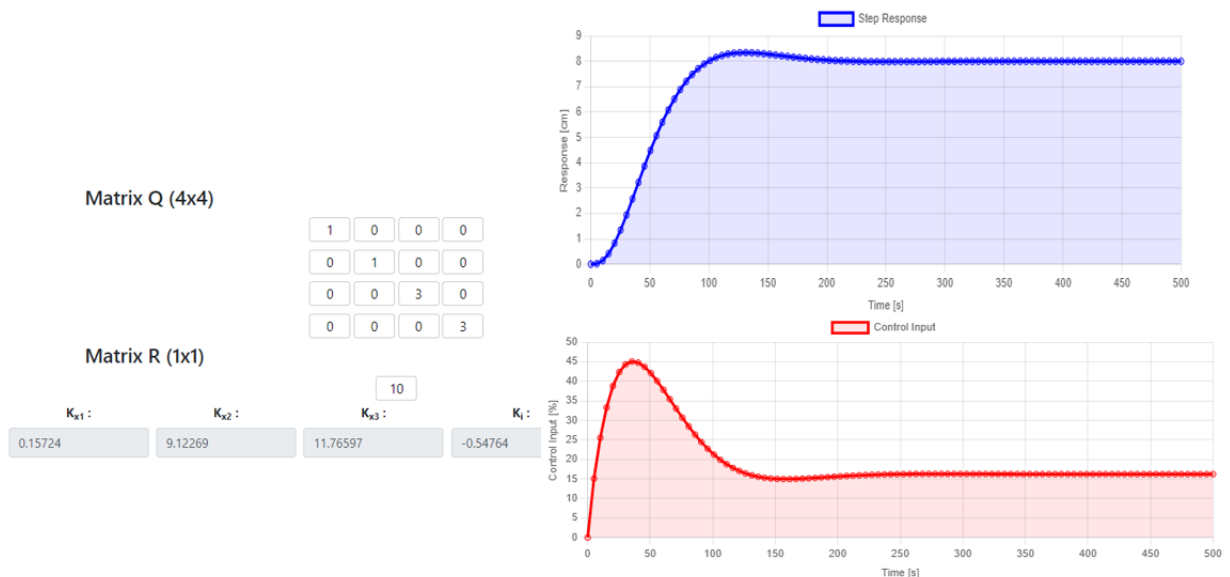
Pokazatelj kakvoće	Stvarna vrijednost	Vrijednost sa simulacije	Mjerna jedinica
Vrijeme ustaljivanja [5%]	335.70	141.41	s
Vrijeme porasta [10% - 90%]	102.40	131.31	s
Maksimalno nadvišenje	4.03%	3.81%	-
Pogreška u stacionarnom stanju	< -0.64, + 0.4 >	0.00	cm

Slika 5.14. prikazuje slijedno vođenje razine tekućine u spremniku. Vidljivo je kako razina tekućine oscilira oko referentne vrijednosti unutar $\pm 0,6$ cm. Ova odstupanja su posljedica inertnosti sustava, odnosno izlaz procesa ne može trenutno reagirati na promjene u ulaznom signalu. Potrebno je vrijeme da se promjene u upravljačkom signalu prenesu kroz sustav i manifestiraju u promjeni razine tekućine. Drugim riječima, postoji transportno kašnjenje. Uz to postoji i ograničenje aktuatora u pogledu preciznog upravljanja dotokom tekućine (rezolucija solenoida ventila).



Slika 5.14. Slijedno vođenje

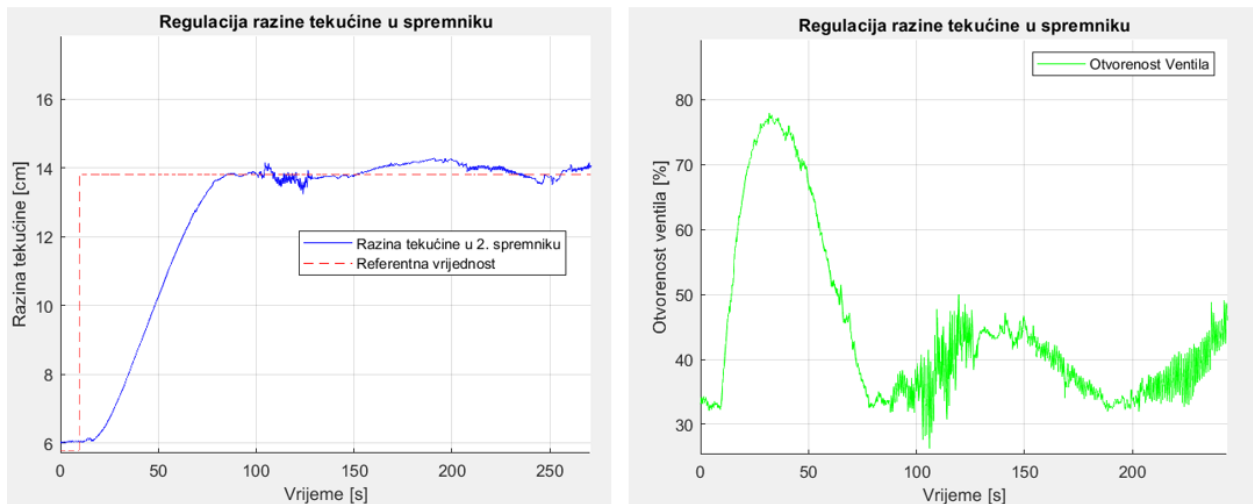
Ukoliko se želi postići kraće vrijeme porasta, potrebno je smanjiti penalizaciju ulaznog signala predefeniranjem matrice \mathbf{R} . Slika 5.15. prikazuje nove vrijednosti LQR-a te simulacijski odziv regulatora i regulirane veličine.



Slika 5.15. Postavljanje LQR parametara i odziv simulacije

Rezultati dobiveni od laboratorijskog postrojenja su prikazani na slici 5.16., dok Tablica 5.2. prikazuje pokazatelje kakvoće regulacije i njihovu usporedbu s onima iz simulacije.

Simulacijski podatci su izračunali najvišu promjenu ulaznog signala (otvorenosti ventila) u iznosu od 45%, dok je na stvarnom sustavu ona iznosila 45.14%. Na stvarnim podacima se također mogu uočiti značajne promjene otvorenosti ventila koje izgledaju gotovo kao šum. Ove promjene su rezultat premalog vremena diskretizacije LQR upravljačkog zakona na PLC-u, zbog čega sustav nastoji neprekidno kompenzirati čak i najmanja odstupanja od željene vrijednosti.

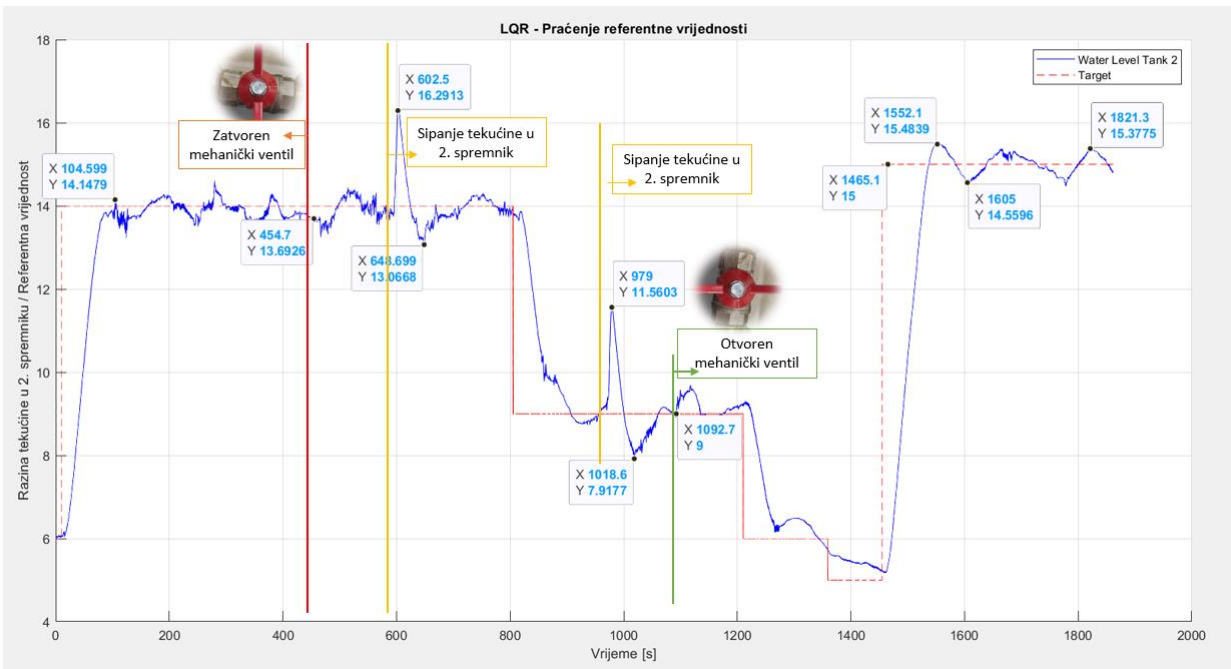


Slika 5.16. Postavljanje LQR parametara i odziv simulacije

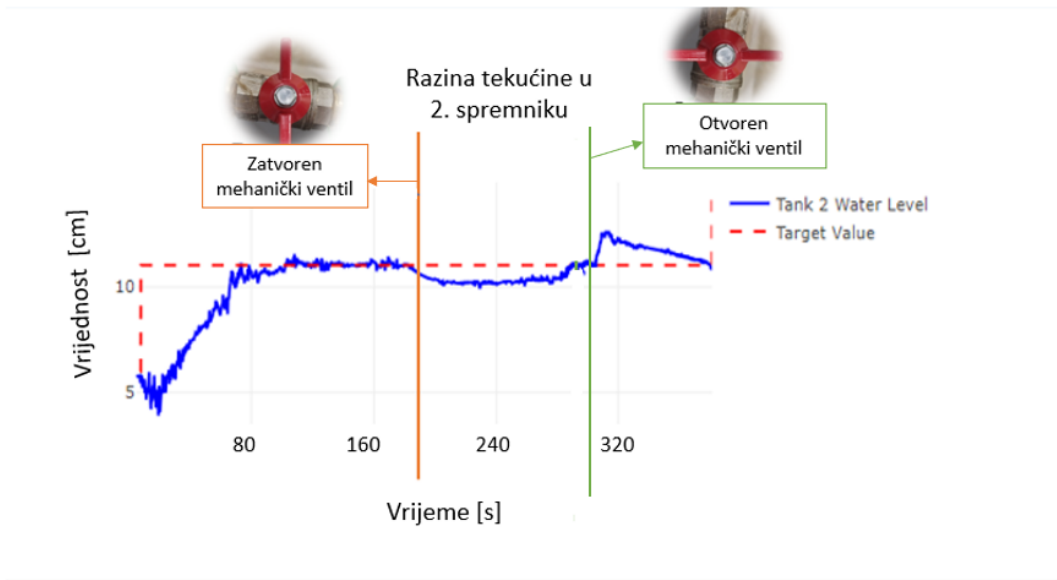
Tablica 5.2. Usporedba pokazatelja kakvoće regulacije

Pokazatelj kakvoće	Stvarna vrijednost	Vrijednost sa simulacije	Mjerna jedinica
Vrijeme ustaljivanja [5%]	117.60	90.91	s
Vrijeme porasta [10% - 90%]	60.80	80.81	s
Maksimalno nadvišenje	3.93%	4.24%	-
Pogreška u stacionarnom stanju	< -0.7, +0.7 >	0.00	cm

Vrlo je važno promotriti i ponašanje regulatora pod utjecajem različitih smetnji. U sustav su uvedene dvije vrste smetnji (otvaranje/zatvaranje mehaničkog ventila između prvog i drugog spremnika te dodavanje tekućine u drugi spremnik) što je prikazano na slici 5.17. i 5.18. Nakon zatvaranja i otvaranja mehaničkog ventila, dolazi do blage promjene razine tekućine, ali regulator uspijeva kompenzirati tu smetnju. Tijekom dodavanja tekućine dolazi do naglog povećanja razine u spremniku, što uzrokuje smanjenje otvorenosti ventila. Kao rezultat, razina tekućine kratkotrajno opada ispod referentne vrijednosti, ali se ubrzo vraća na željenu vrijednost.



Slika 5.17. Regulacija pod utjecajem smetnji

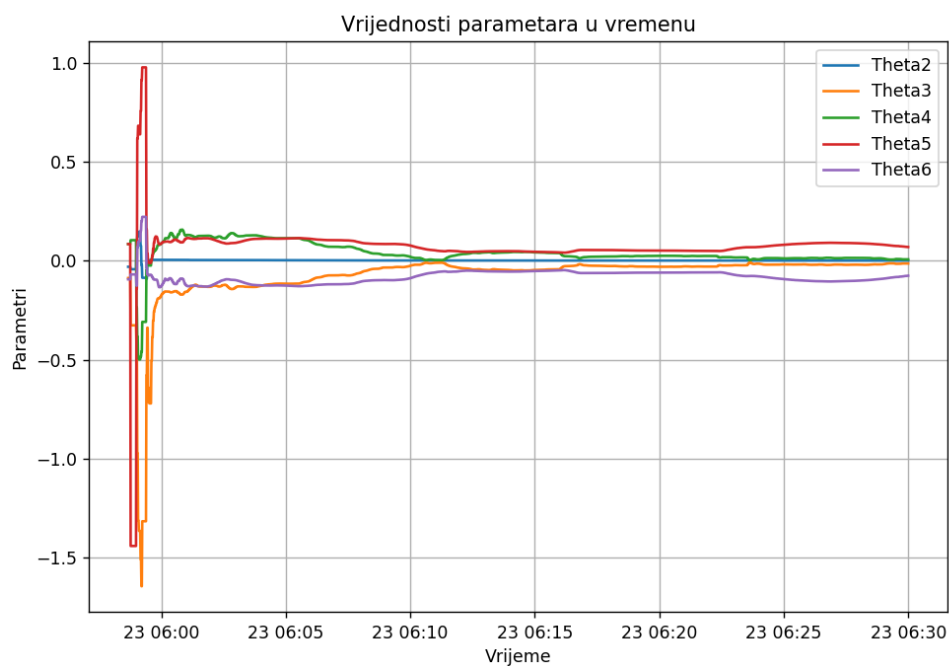


Slika 5.18. Regulacija pod utjecajem smetnji

5.3. Rezultati rada rekurzivne metode najmanjih kvadrata

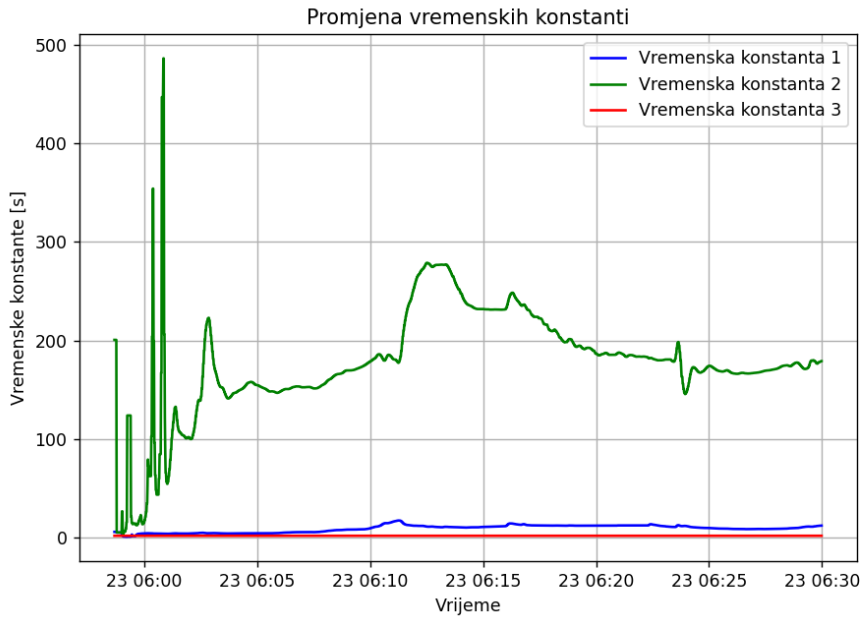
U poglavlju 5.2 razmotreno je konfiguriranje parametara regulacije putem sučelja na web aplikaciji. Uz to, omogućeno je i automatsko izračunavanje parametara regulatora na temelju matematičkog modela procesa kojeg ažurira „*MathModelOnline*“ servis koristeći rekurzivnu metodu najmanjih kvadrata.

Rad rekurzivne metode najmanjih kvadrata uočava se na slici 5.19. koja prikazuje kako su se mijenjali parametri matematičkog modela. Uočava se kako su se ti parametri nakon kratkog vremena stabilizirali.



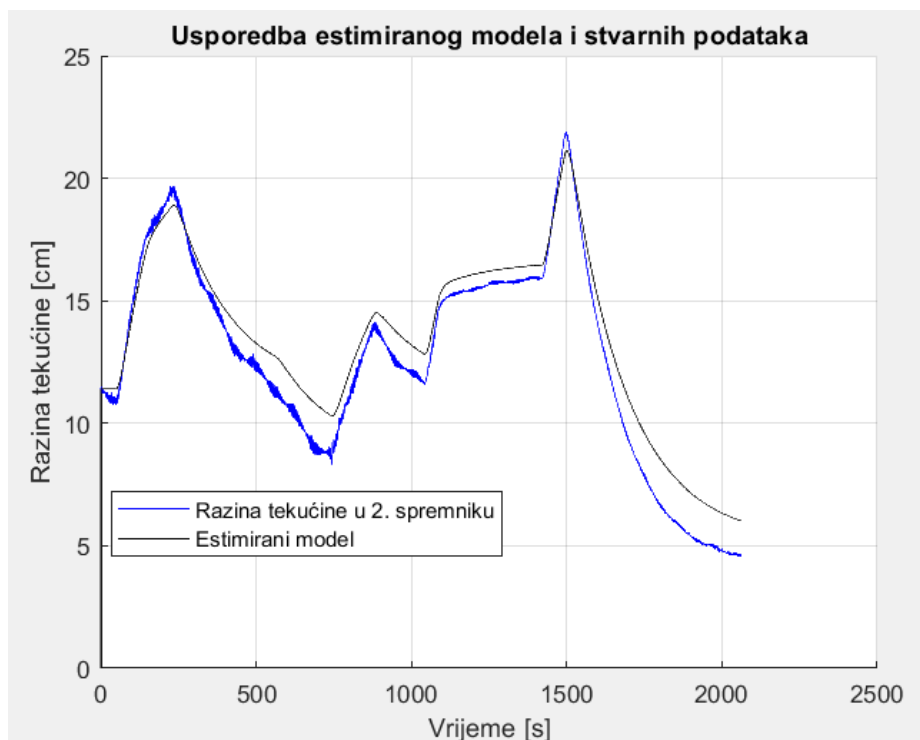
Slika 5.19. Promjena parametara matematičkog modela

Kako bi se dobio intuitivniji uvid u dinamiku sustava, na slici 5.20. su prikazane vremenske konstante. Te vrijednosti vremenskih konstanti u većoj mjeri se poklapaju s vremenskim konstantama linearnog modela dobivenog na temelju provedenih eksperimenata.

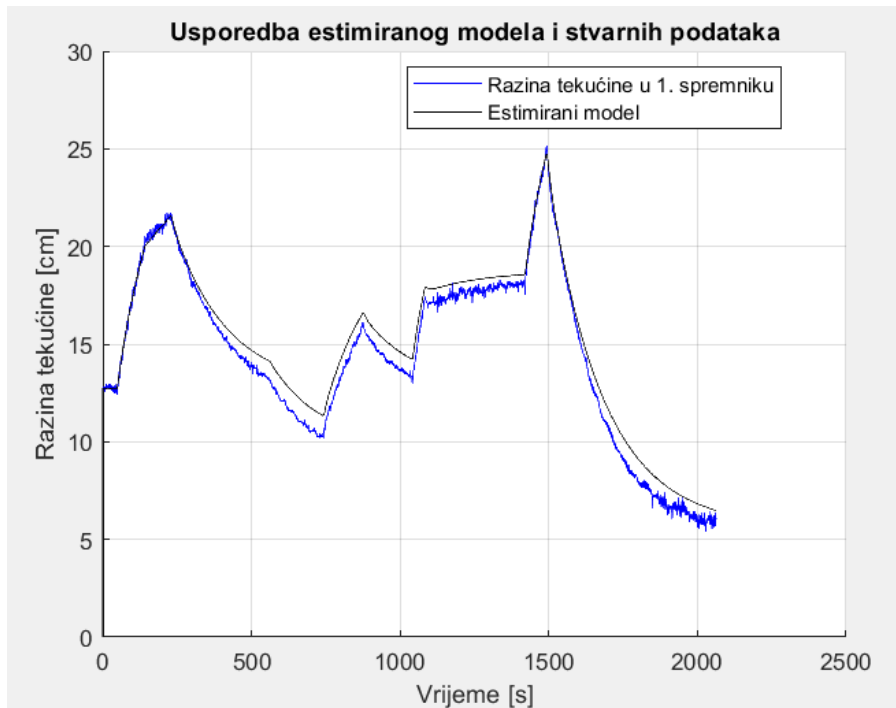


Slika 5.20. Promjena vremenskih konstanti

Za validaciju ispravnosti dobivenog matematičkog modela, potrebno je ispitati podudaranje s testnim podacima. Slike 5.21. i 5.22. prikazuju usporedbu dobivenog matematičkog modela i stvarnih podataka. Može se uočiti kako estimirani model dobro aproksimira stvarni proces.

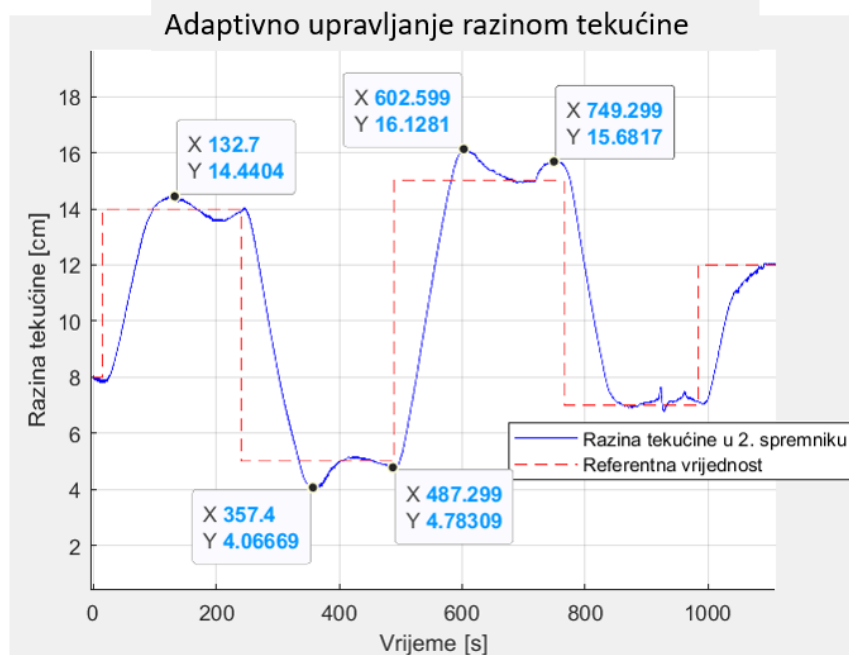


Slika 5.21. Usporedba modela sa stvarnim podacima – razina tekućine u drugom spremniku



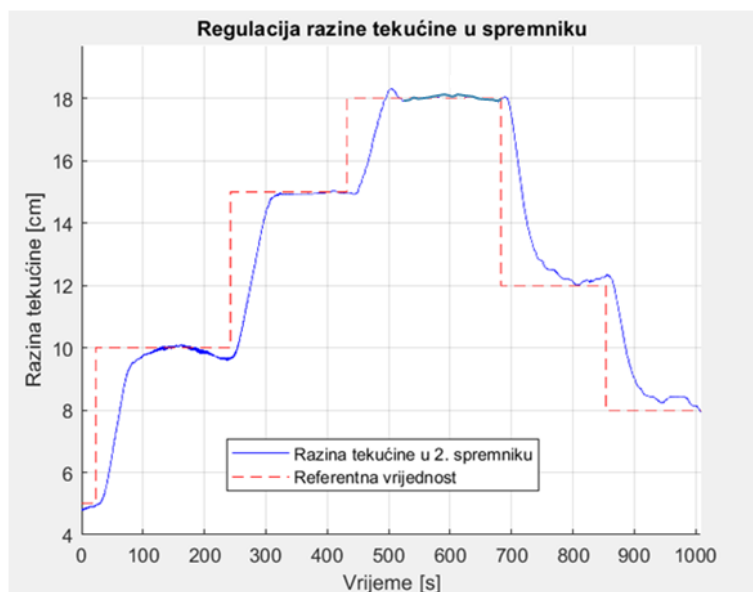
Slika 5.22. Usporedba modela sa stvarnim podacima – razina tekućine u prvom spremniku

Svrha je kontinuirane identifikacije modela, prilagoditi parametre regulatora. Slika 5.23. prikazuje regulaciju razine tekućine u drugom spremniku za vrijeme kontinuiranog ažuriranja parametara LQR-a.

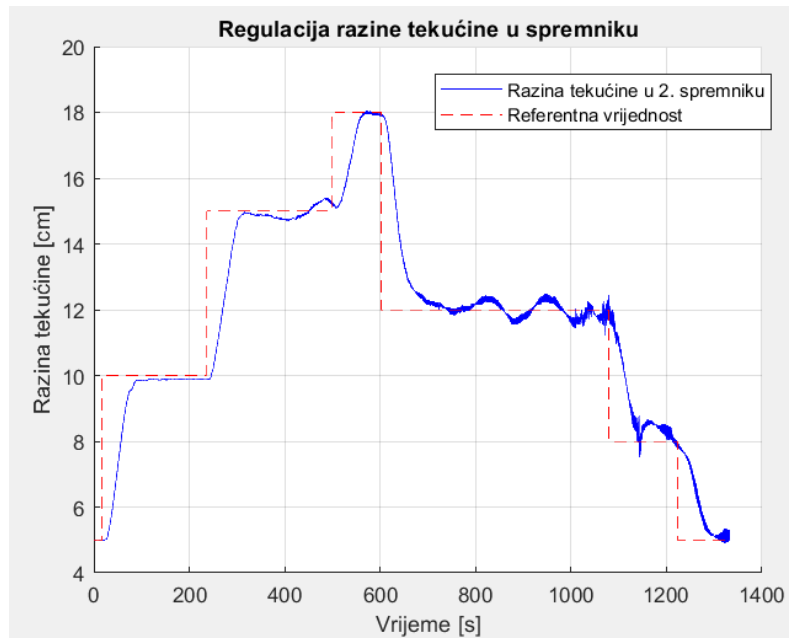


Slika 5.23. Regulacija razine tekućine u drugom spremniku tijekom adaptivnog upravljanja

Ova metoda omogućila je podešavanje parametara linearnog modela u stvarnom vremenu, čime je omogućeno automatsko konfiguriranje parametara LQR-a. S druge strane, pokazalo se da je matematički model dobiven eksperimentalnom analizom bio sasvim dovoljan za uspješno reguliranje sustava. Proces skladištenja tekućine u spremnicima ne pokazuje značajnu nelinearnost, a parametri sustava se ne mijenjaju u velikoj mjeri. Zbog toga kontinuirana identifikacija modela nije nužna, te je dovoljno koristiti iste parametre LQR-a za sve radne točke sustava. Slike 5.24. i 5.25. potvrđuju ovaj zaključak. Slika 5.24. prikazuje regulaciju razine tekućine u svim radnim točkama sustava s projektiranim regulatorom iz poglavlja 5.2, dok slika 5.25. prikazuje kako isti regulator funkcionira kada je zatvoren mehanički ventil između prvog i drugog spremnika te kada je onemogućen jedan od protoka između drugog i cirkulacijskog spremnika.



Slika 5.24. Regulacija razine tekućine u drugom spremniku



Slika 5.25. Regulacija razine tekućine u drugom spremniku (zatvoren mehanički ventil i protok)

6. ZAKLJUČAK

Aplikacija razvijena tijekom ovog rada uspješno je omogućila upravljanje i nadzor sustava skladištenja tekućine, pružajući korisnicima uvid u performanse postrojenja u stvarnom vremenu. Programsko rješenje omogućilo je praćenje ključnih procesnih veličina, statusa i alarma, čime je unaprijeđena operativna efikasnost i sigurnost sustava. Uz to, aplikacija je korisnicima omogućila napredne mogućnosti upravljanja i vizualizacije, što je doprinijelo boljoj kontroli i razumijevanju sustava.

Što se tiče regulacije, postignuti su zadovoljavajući rezultati u pogledu dinamičkih i statičkih pokazatelja kakvoće regulacije. Primijenjena rekurzivna metoda najmanjih kvadrata (RLS) omogućila je kontinuiranu identifikaciju modela i prilagodbu parametara regulatora u stvarnom vremenu. Međutim, analiza je pokazala da je primjena RLS metode bila suvišna u ovom specifičnom slučaju, budući da parametri sustava nisu pokazivali značajne promjene. Nadalje, za različite radne točke, isti LQR regulator se pokazao adekvatnim i učinkovitim, bez potrebe za prilagodbom. Sustav je već funkcionirao stabilno i predvidljivo s postojećim parametrima.

Unatoč tome, primjena RLS metode demonstrirala je fleksibilnost sustava i njegovu sposobnost prilagodbe promjenjivim uvjetima rada, posebno kroz mogućnost istovremenog pronalaženja svih parametara modela, čime je omogućeno i automatsko podešavanje regulatora. Ovo može biti od velike koristi u složenijim sustavima s izraženijim promjenama parametara i nelinearnostima.

Na kraju, identificirane su mogućnosti za daljnja poboljšanja sustava. S obzirom na dinamiku procesa, visoka frekvencija uzorkovanja u PLC programu nije potrebna. Smanjenje frekvencije omogućilo bi učinkovit rad sustava bez ugrožavanja preciznosti regulacije. Nadalje, Implementacija virtualnih senzora osigurala bi kontinuitet u radu regulatora u slučaju kvara fizičkih senzora, što je posebno važno za ispravan rad LQR upravljačkog zakona. Ova poboljšanja mogla bi dodatno osnažiti funkcionalnost sustava i osigurati njegovu dugoročnu pouzdanost u radnim uvjetima.

Zaključno, aplikacija je uspješno ispunila postavljene ciljeve, pružajući stabilno i precizno upravljanje procesom skladištenja tekućine na temelju izgrađenog matematičkog modela, uz mogućnosti daljnjeg unaprijeđenja i prilagodbe sustava.

LITERATURA

- [1] „Use of PLC in Automation“. Pristupljeno: 09. lipanj 2024. [Na internetu]. Dostupno na: <https://www.plctable.com/use-of-plc-in-automation/>
- [2] Dorf, R.C., & Bishop, R.H., *Modern Control Systems*, 13. izd. 2017.
- [3] „Importance of water level control in various industries“, *Vacker*, str. 1–3, svi. 2023.
- [4] D. Slišković i R. Grbić, *Modeliranje temeljeno na podacima*. nastavni materijali, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, 2017.
- [5] S. Pleslić, „Pripreme za predavanja iz Fizike 1“. nastavni materijali, Fakultet elektrotehnike i računarstva. Pristupljeno: 12. svibanj 2024. [Na internetu]. Dostupno na: https://www.fer.unizg.hr/_download/repository/predavanja13-2014.pdf
- [6] R. Cupec, „Upravljanje procesima“. nastavni materijali, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek.
- [7] K. Ogata, *Modern Control Engineering*, 5. izd. Pearson, 2010.
- [8] N. Perić, *Automatsko upravljanje*. Zagreb: Skriptarnica, FER, 1998.
- [9] M. R.M., *Control and Dynamical Systems*. California Institute of Technology, 2006.
- [10] C. Mahulea, „Optimization and Optimal Control Lecture 4: Optimal Control: Discrete-Time LQR“. University of Zaragoza, Spain, 01. kolovoz 2008. Pristupljeno: 20. srpanj 2024. [Na internetu]. Dostupno na: <https://webdiis.unizar.es/~cmahulea/course/lecture4.pdf>
- [11] „S7.Net documentation“. 01. siječanj 2010. Pristupljeno: 10. veljača 2024. [Na internetu]. Dostupno na: <https://github.com/S7NetPlus/s7netplus>
- [12] L. Johansson, „What is RabbitMQ“. Pristupljeno: 12. lipanj 2024. [Na internetu]. Dostupno na: <https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html>

SAŽETAK

U ovom diplomskom radu obrađena je tema sustava upravljanja i nadzora procesa skladištenja tekućeg fluida, s posebnim fokusom na implementaciju aplikacije koja korisnicima omogućava vizualizaciju i simulaciju tog sustava. U radu je detaljno objašnjena potreba za modeliranjem tehničkog sustava, razrađen je matematički model za dani proces i eksperimentalno su određeni njegovi parametri. Dobiveni model uspoređen je sa stvarnim podacima kako bi se potvrdila njegova valjanost i analizirala njegova mogućnost točne aproksimacije stvarnog procesa. Nadalje, obuhvaćena je analiza modela i opis metoda za sintezu regulatora, uz korištenje RLS algoritma za prilagodbu modela u stvarnom vremenu. Također, opisana je programska podrška koja uključuje komunikacijske servise, bazu podataka, aplikacijsko programsko sučelje i web aplikaciju. Ova podrška omogućava dvosmjernu komunikaciju između PLC-a i servera, prikupljanje, slanje i pohranu podataka. Glavne funkcionalnosti web aplikacije obuhvaćaju simulaciju rada postrojenja i sintezu upravljačkih algoritama.

Ključne riječi: Identifikacija modela, linearni kvadratni regulator, rekurzivna metoda najmanjih kvadrata, upravljanje i nadzor skladištenja fluida, vizualizacija i simulacija sustava

ABSTRACT

Title: Control and supervision system for a fluid storage process

In this thesis, the topic of fluid storage process management and control system is addressed, with a special focus on the implementation of an application that allows users to visualize and simulate the system. The need for modeling the technical system is thoroughly explained, and a mathematical model for the given process is developed with its parameters determined experimentally. The obtained model is compared with real data to validate its accuracy and analyse its capability to accurately approximate the real process. Furthermore, the analysis of the model and the description of the methods for the synthesis of the controllers are included, with the use of the RLS algorithm for real-time model adjustment. Additionally, software support is described, which includes communication services, database, application programming interface and web application. This software enables bidirectional communication between the PLC and the server, data collection, transmission and storage. The main functionalities of the web application include simulating plant operation and the synthesis of the control algorithms.

Keywords: Model identification, linear quadratic regulator, recursive least squares method, fluid storage supervision and control, system visualization and simulation.