

Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Departman za računarstvo i automatiku

Odsek za automatiku, geomatiku i upravljanje sistemima



PROJEKTNA DOKUMENTACIJA

# PREDMETNI PROJEKAT

## Upravljački algoritmi u realnom vremenu

Tema projekta: **Upravljanje serijskim robotskim dvozglobnim manipulatorom**

Profesor: dr Željko Kanović  
Asistenti: Gluhović Mihailo  
Golić Anastasija

Autori: Šećerov Nemanja  
Demirović Emina  
Grković Dušan  
Tokić Ilija

Novi Sad, jul 2024.

# 1. O PROJEKTU

## 1.1 Motivacija

Motivacija za izradu ovog predmetnog projekta nastala je jer je odabrana tema prepoznata kao sve češći, prisutniji i izazovniji problem u različitim industrijskim, medicinskim i istraživačkim aplikacijama, uzimajući u obzir široku primenu preciznog i efikasnog upravljanja robotskom rukom.

Robotske ruke su ključne komponente u automatizovanim sistemima jer omogućavaju obavljanje kompleksnih zadataka sa visokim nivoom tačnosti i ponovljivosti. Međutim, razvoj pouzdanih algoritama za kontrolu pokreta robotske ruke je izazovan zbog nelinearnosti sistema i potreba za brzim i skoro real-time reakcijama, pa je i to jedna od tema obrađenih pri izradi kako samog projekta, tako i projektne dokumentacije.

Specifikacija uključuje sve faze projektnog zadatka, i to u hronološkom rasporedu izrade, kako su ispod i nabrojane, a u nastavku svakako detaljnije objašnjene.

- **Izrada matematičkog modela robotske ruke:** Razvijanje preciznog matematičkog modela koji opisuje kinematiku i dinamiku robotske ruke uz identifikaciju i razmatranje svih relevantnih parametara, te formulaciju jednačina koje opisuju sistem, a koje zahtevaju adaptaciju u diskretan sistem radi jednostavnije implementacije.
- **Dizajn i implementacija kontrolera:** Razvijanje i testiranje PID i fuzzy kontrolera koji će omogućiti precizno upravljanje robotskom rukom, te poređenje performansi svakog od ovih kontrolera uz konvencionalan pristup drugih oblasti automatike.
- **Simulacija u LabVIEW-u:** Korišćenje LabVIEW okruženja za simulaciju modela robotske ruke i testiranje kontrolera u simuliranim uslovima pre „spuštanja“ na hardver.
- **Implementacija na sbRIO/cRIO:** Prenošnje simuliranih modela i kontrolera na National Instruments laboratorijski hardver radi testiranja i validacije performansi u stvarnim uslovima.
- **Razvoj korisničkog interfejsa:** Kreiranje intuitivnog i funkcionalnog korisničkog interfejsa koji omogućava lako upravljanje robotskom rukom, uključujući različite funkcionalnosti kao što su manualna kontrola putem HID uređaja ili kontrola preko Front Panel-a, te automatizovano posećivanje tačaka sa odabrane putanje i slično.

Za realizaciju ovog projekta korišćene su različite tehnologije i alati: prvenstveno **LabVIEW** za grafičko programiranje i pomenute hardverske real-time platforme i dodatni ulazni uređaji, a za matematičke proračune, podešavanje parametara, modelovanje, vizualizaciju i simulaciju **MATLAB** i **Simulink**.

## 1.2 Ciljevi projekta

CILJ	KAKO DO CILJA?
<b>Razvoj tačnog matematičkog modela</b>	Pomoću različitih matematičkih sredstava verno predstaviti dinamiku i kinematiku realnog sistema koja pokriva širok simulacioni i realizacioni opseg.
<b>Efikasna i stabilna kontrola pokreta robotske ruke</b>	Projektovati kontrolna sredstva konvencionalnih upravljačkih strategija sa akcentom na stabilnosti celokupnog sistema.
<b>Podešavanje parametara kontrolera i optimizacija rada sistema</b>	Eksperimentalno ili sistemski štimovati parametre i popravljati performanse (tačnost, preciznost ...) dok ne budu dostignute optimalne vrednosti.
<b>Simulacija rada modelovanog sistema</b>	Simulirati kako se upravlja modelom robotske ruke i to uz pomoć različitih ulaznih uređaja.
<b>Evaluacija performansi sistema u zavisnosti od pristupa</b>	Izvršiti uporednu analizu rada različitih kontrolera u različitim uslovima rada i identifikovati njihove prednosti i nedostatke, kao i ograničenja.
<b>Realizacija u realnom vremenu</b>	Implementirati razvijeno rešenje na hardver (sbRIO/cRIO), izvesti testove u realnim uslovima i verifikovati rad sistema.

## 1.3 Teorijska osnova projekta

Upravljanje uključuje primenu kontrolnih algoritama koji omogućavaju precizno pozicioniranje i manipulaciju delovima sistema. Robotske ruke su složeni mehanički sistemi sa više stepeni slobode, što znači da se njihovi pokreti mogu opisati nizom jednačina. Osnove upravljanja robotskom rukom uključuju sledeće komponente:

- **Kinematika** – proučava kako se kreće svaki segment robotske ruke u prostoru. Upotrebljeni su i principi direktne (računanje položaja krajnjeg efektora iz poznatih uglova zglobova) i inverzne kinematike (računanje potrebnih uglova zglobova za postizanje željenog položaja krajnjeg efektora).
- **Dinamika** - proučava sile i momente potrebne za postizanje željenih pokreta. Obuhvaćeno je kreiranje dinamičkih modela koji opisuju odnos između sila, odnosno momenata sila i kretanja robotske ruke.
- **Kontrolni algoritmi** za primenu različitih kontrolnih strategija kao što su PID kontroleri, adaptivni kontroleri fuzzy logika sa ciljem postizanja stabilne kontrole pokreta.

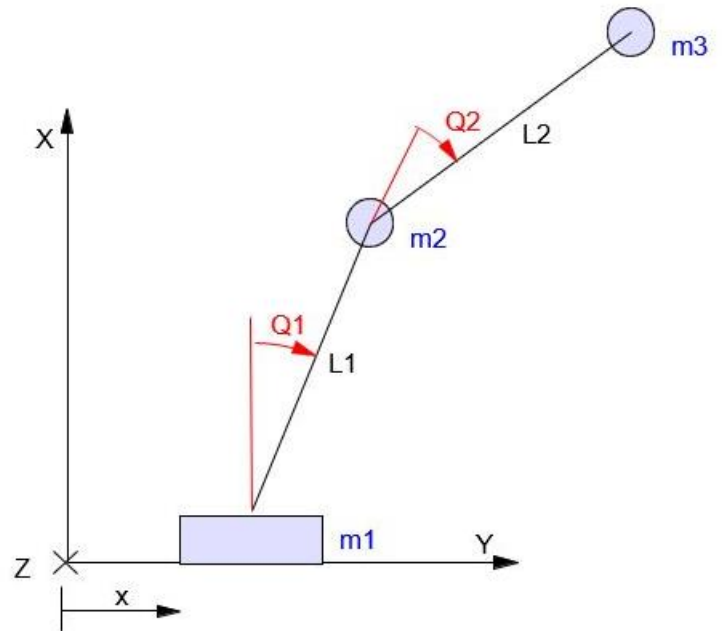
## 2. MATEMATIČKI MODEL SISTEMA

### 2.1 Formiranje matematičkog modela

Projektovani model robotske ruke se sastoji od kućišta i dva zglobna dela koji omogućavaju pokretljivost i efikasnost pomeraja.

Neka se u svakom od prelomnih delova nalazi centar masa  $m_1$ ,  $m_2$  i  $m_3$  i neka su udaljeni za određene dužine  $L_1$  i  $L_2$ . Odnos između komponenti sistema grafički je prikazan dijagramom (slika desno).

Važno je naglasiti da je sistem pozicioniran dvema dimenzijama i da je kretanje moguće sa dva stepena slobode.



Pozicija svakog centra mase predstavljena je kao funkcija translatorskog i ugaonih pomeraja, odnosno:

1:

$$y_1 = x$$

$$x_1 = 0$$

2:

$$y_2 = y_1 + \sin(\theta_1)$$

$$x_2 = x_1 + \cos(\theta_1)$$

3:

$$y_3 = y_2 + \sin(\theta_1 + \theta_2)$$

$$x_3 = x_2 + \cos(\theta_1 + \theta_2)$$

Ili jednostavnije:

$$y_1 = x$$

$$x_1 = 0$$

$$y_2 = x + s_1$$

$$x_2 = c_1$$

$$y_3 = x + s_1 + s_{12}$$

$$x_3 = c_1 + c_{12}$$

U nastavku su izračunati izvodi kako bi se došlo do izraza za kinematičku energiju.

$$\dot{y}_1 = \dot{x}$$

$$\dot{x}_1 = 0$$

$$\dot{y}_2 = \dot{x} + c_1 \dot{\theta}_1$$

$$\dot{x}_2 = -s_1 \dot{\theta}_1$$

$$\dot{y}_3 = \dot{x} + c_1 \dot{\theta}_1 + c_{12} (\dot{\theta}_1 + \dot{\theta}_2) \quad \dot{x}_3 = -s_1 \dot{\theta}_1 - s_{12} (\dot{\theta}_1 + \dot{\theta}_2)$$

$$KE = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2)$$

$$KE = \frac{1}{2} (\dot{x}^2) + \frac{1}{2} \left( (\dot{x} + c_1 \dot{\theta}_1)^2 + (-s_1 \dot{\theta}_1)^2 \right) + \frac{1}{2} \left( (\dot{x} + c_1 \dot{\theta}_1 + c_{12} (\dot{\theta}_1 + \dot{\theta}_2))^2 + (-s_1 \dot{\theta}_1 - s_{12} (\dot{\theta}_1 + \dot{\theta}_2))^2 \right)$$

Konačan izraz za kinetičku energiju:

$$KE = \frac{3}{2} \dot{x}^2 + \dot{\theta}_1^2 + \frac{1}{2} (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2c_1 \dot{x} \dot{\theta}_1 + c_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + c_{12} \dot{x} (\dot{\theta}_1 + \dot{\theta}_2)$$

I potencijalna energija je određena na sledeći način:

$$PE = m_1 g x_1 + m_2 g x_2 + m_3 g x_3$$

$$PE = g c_1 + g (c_1 + c_{12})$$

$$PE = 2g c_1 + g c_{12}$$

Takođe, formiran je i Lagranžijan koji je potreban za izračunavanje obrtnih momenata svakog od zglobova.

$$L = \frac{3}{2} \dot{x}^2 + \dot{\theta}_1^2 + \frac{1}{2} (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2c_1 \dot{x} \dot{\theta}_1 + c_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + c_{12} \dot{x} (\dot{\theta}_1 + \dot{\theta}_2) - 2g c_1 - g c_{12}$$

$$T_1 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1}$$

$$T_2 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2}$$

Konačni izrazi za obrtne momente zglobova:

$$\begin{aligned}T_1 &= 2\ddot{\theta}_1 + (\ddot{\theta}_1 + \ddot{\theta}_2) - 2s_1\dot{\theta}_1 + 2c_1\ddot{x} - s_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \\&\quad + c_2(\ddot{\theta}_1 + \ddot{\theta}_2) - s_2\dot{\theta}_1\dot{\theta}_2 + c_2\ddot{\theta}_1 - s_{12}\dot{x}(\dot{\theta}_1 + \dot{\theta}_2) + c_{12}\ddot{x} \\&\quad + 2s_1\dot{\theta}_1 + s_{12}\dot{x}(\dot{\theta}_1 + \dot{\theta}_2) - 2gs_1 - gs_{12} \\T_2 &= (\ddot{\theta}_1 + \ddot{\theta}_2) - s_2\dot{\theta}_1\dot{\theta}_2 + c_2\ddot{\theta}_1 - s_{12}\dot{x}(\dot{\theta}_1 + \dot{\theta}_2) + c_{12}\ddot{x} \\&\quad + s_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) + s_{12}\dot{x}(\dot{\theta}_1 + \dot{\theta}_2) - gs_{12}\end{aligned}$$

## 2.2 Linearizacija

Kako je evidentno da postoje nelinearnosti u okviru izvedene dinamike sistema, bilo je neophodno izvršiti linearizaciju modela kako bismo omogućili primenu linearnih upravljačkih tehnika, kao što su PID kontroleri ili druge tehnike analize i sinteze komponenata linearnog ponašanja.

Budući da nelinearnosti mogu da potiču iz različitih izvora kao što su nelinearna povratna sprega, trenje i druge sile otpora, nelinearne karakteristike aktuatora i slično, neophodno je razmotriti sva ograničenja linearizovanog modela kako bi se minimizovala moguća odstupanja stvarnog ponašanja od proračuna koji važe za radnu tačku.

Dakle, linearizacija je postupak kojim se nelinearni model aproksimira linearnim modelom u određenoj, tzv. radnoj tački rada (obično tačka u kojoj sistem funkcioniše u normalnim uslovima).

Ideja o linearizaciji primenom Gain – Scheduling algoritma je bila razmatrana, ali zbog postojanja velikog broja radnih tačaka nije implementirana, budući da kompleksnost izlazi iz domena ovog projekta.

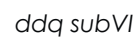
## 2.3 Diskretizacija

Diskretizacija je proces pretvaranja kontinualnog matematičkog modela u diskretni model.

U kontekstu upravljanja robotskom rukom, diskretizacija je ključna jer većina savremenih kontrolnih sistema koristi digitalne procesore za izvršavanje kontrolnih algoritama, a vrlo često je potrebno izvršiti i numeričku analizu ili implementaciju u realnom vremenu.

Diskretizacija je neophodna za prelazak sa teorijskog modela na praktičnu implementaciju, omogućavajući upotrebu digitalnih kontrolera i simulacija u razvoju složenih mehatroničkih robotskih sistema.

Pri izradi projektnog zadatka, izbegnuta je eksplicitna diskretizacija zbog nelinearne prirode modela i usled nedostatka iskustva i znanja u toj oblasti, već je primenjeno numeričko rešavanje diferencijalnih jednačina, i to pomoću Ojlerove metode. Na taj način je dobijen diskretni model sa izraženim numeričkim vrednostima, pa je u ostalim fazama razvoja nedostatak analitičkih karakteristika nadoknađen različitim simulacijama sa ciljem pronalaska vrednosti za parametre kao što je npr. vreme kašnjenja tako da dobijeni model bude što sličniji stvarnom sistemu i njegovom ponašanju.



### 3. UPRAVLJAČKI ALGORITMI I STRATEGIJE UPRAVLJANJA

#### 3.1 Konvencionalni pid kontroler

##### 3.1.1 Teorijska osnova PID-a

PID je jedan od najčešće korišćenih tipova kontrolera u automatskom upravljanju, zahvaljujući jednostavnoj implementaciji, razumljivosti i efikasnosti u širokom spektru primena. PID kontroler koristi tri različite komponente za generisanje kontrolnog signala: **proporcionalnu**, **integralnu** i **derivativnu** komponentu, pa je i to jedna od prednosti ove upravljačke tehnike, budući da nudi fleksibilnost pri podešavanju tri nezavisna parametra radi poboljšanja performansi.

**Proporcionalna komponenta (P)** je osnovni deo PID kontrolera i direktno je proporcionalna grešci  $e(t)$ , koja se definiše kao razlika između željene i trenutne vrednosti na izlazu. Proporcionalna komponenta pomaže da se smanji greška, ali sama po sebi može dovesti do stacionarne greške (steady-state error).

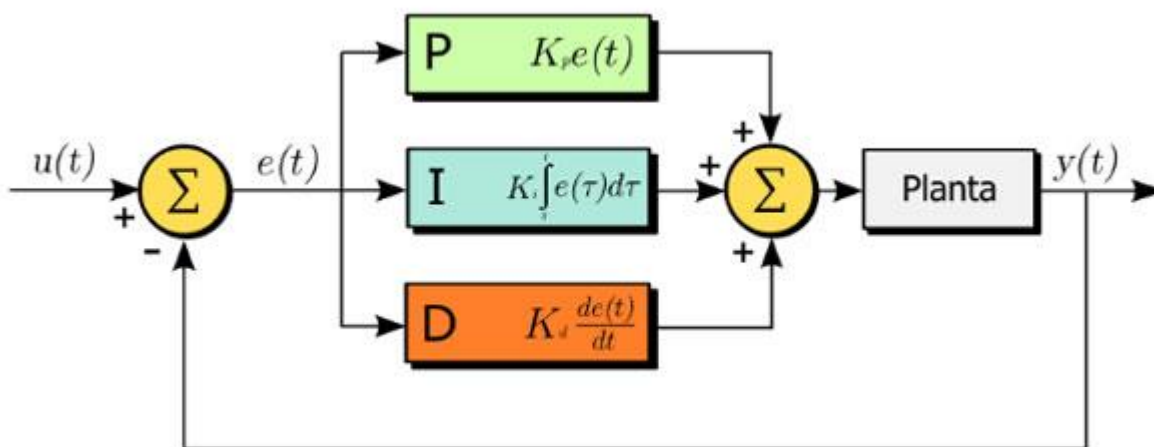
**Integralna komponenta (I)** računa sumu svih prethodnih grešaka tokom vremena i koristi se za eliminaciju stacionarne greške. Integralna komponenta povećava brzinu odgovora sistema na trajnu grešku, ali može uzrokovati prebačaj i nestabilnost ako nije pravilno podešena.

**Derivativna komponenta (D)** je proporcionalna brzini promene greške. Njena svrha je da predvidi buduće ponašanje greške i da ublaži efekat brzih promena, pa tako pomaže u smanjenju prebačaja i poboljšava stabilnost sistema, ali može biti osjetljiva na šum u merenju.

Kombinovanjem sve tri komponente dobijen je izraz za PID kontroler:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt}$$

PID regulator može biti predstavljen i blok šemom gde se jasno vidi jedinstvo ova tri kontrolna dejstva.

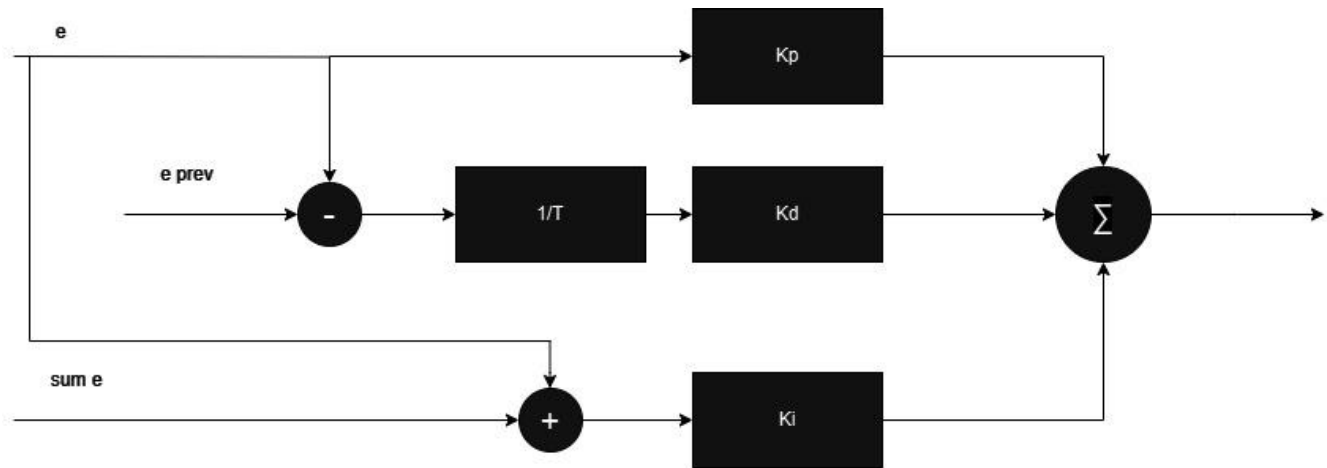


Blok šema PID kontrolera



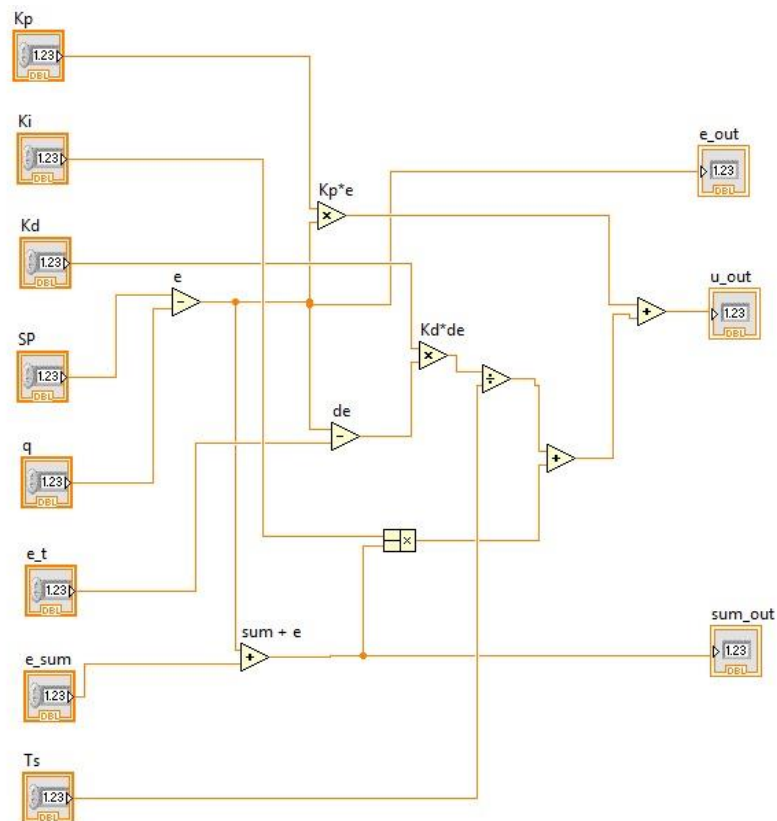
### 3.1.2 PID kontroler u modelu robotske ruke – implementacija

Proporcionalno-integralno-diferencijalni kontroler je implementiran radi efikasne kontrole robotske ruke. U konkretnom slučaju potrebna su dva PID kontrolera. Glavni cilj je da postoji stepen slobode kretanja svakog od zglobova, a da se pritom zaustavlja/pokreće na određenoj poziciji i u određenom vremenu.



*Diskretan PID kontroler*

Zatvoreni sistem upravljanja za robotsku ruku sa dva stepena slobode pruža uvid u modeliranje i upravljanje sistemom robotske ruke. Uprkos kašnjenju, zatvorena sprega omogućava sistemu da dinamički reaguje na odstupanja i održava željenu putanju kretanja robotske ruke.



*PID kontroler - LabView*

### 3.1.3 PID kontroler u modelu robotske ruke – eksperimentalno podešavanje parametara

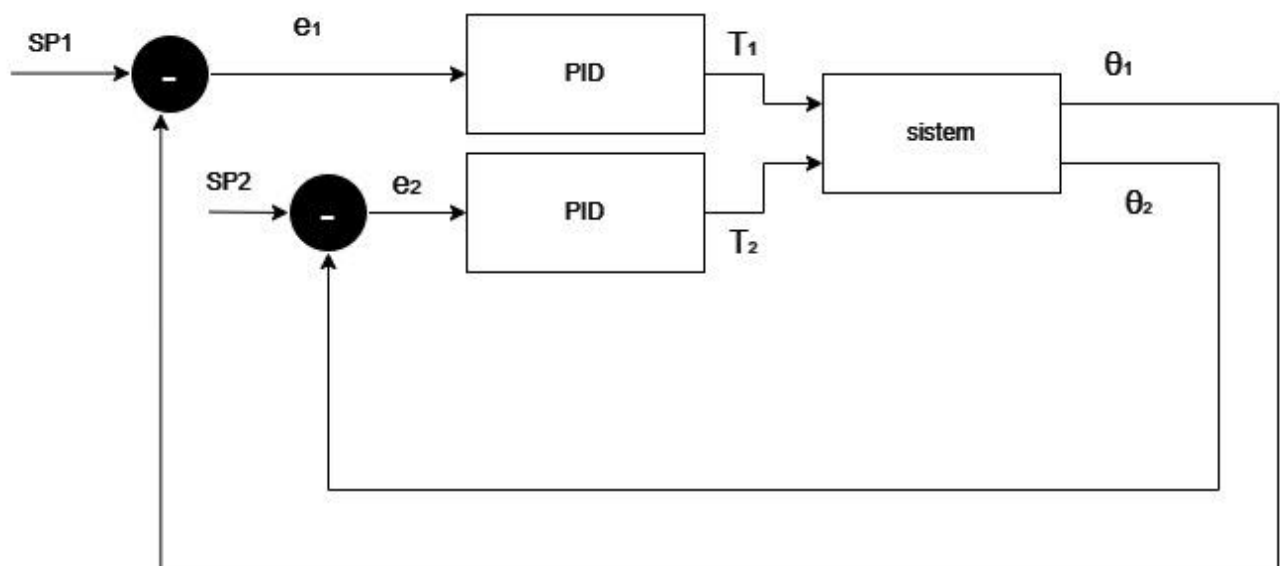
Nakon modelovanja PID kontrolera, bilo je neophodno podesiti njegove parametre, budući da je to ključni korak kojim se osiguravaju optimalne performanse sistema.

U okviru ovog projekta u fokusu je bilo štimanje parametara proporcionalne ( $K_p$ ), integralne ( $K_i$ ) i diferencijalne ( $K_d$ ) komponente kako bismo postigli stabilan i efikasan rad robotske ruke.

Ovaj proces obuhvata više koraka, a svakako podrazumeva i više mogućih metoda štimovanja od kojih su implementirane one koje su se pokazale kao najintuitivnije i najefikasnije.

Prvi pokušaj bilo je eksperimentalno podešavanje parametara. Kroz iterativan pristup postavljene su početne vrednosti za svaku komponentu i po konvenciji uvedene promene tako što je P dejstvo pojačavano dok sistem nije postao stabilan, a zatim D dejstvo da bi preskok bio eliminisan. Tako je kroz svaku iteraciju testiran odziv sistema i praćeni njegovi parametri kao što su prebačaj, vreme uspostavljanja, stabilnost i drugi.

Uspešno je uspostavljen stabilan rad PID-a, ali nijedna od isprobanih kombinacija parametara nije pokazala zadovoljavajuću tačnost. Zaključeno je da ovako naštimovan PID kontroler nikada neće raditi dovoljno precizno i sigurno, budući da je sistem nelinearan. Takođe, sistem robotske ruke je i **MIMO** (Multiple Input Multiple Output) arhitekture, što je dodatno otežavajuća okolnost jer PID nije sposoban da primi i pamti informacije o stanju druge promenljive koja utiče na odziv sistema. Nije bilo moguće da se ovaj problem reši samo modifikacijom parametara, već je bilo neophodno implementirati anti-windup tehnike poput clamping-a integratora ili povratne računice sa čim takođe nisu postignuti značajno bolji rezultati, pa je pažnja u daljoj izradi usmerena ka projektovanju kontrolera koji vršiti podešavanje i na taj način smanjiti grešku u ustaljenom stanju. Zbog toga smo uveli i fuzzy kontroler koji u zavisnosti od greške i promene greške menja parametre PID-a kako bi osigurali minimalnu grešku.



## 3.2 PID kontroler u modelu robotske ruke + fuzzy podešavanje parametara

Usled postojanja značajne greške u ustaljenom stanju u radu PID kontrolera kada se parametri podešavaju ručno, uveden je fuzzy kontroler koji dinamički podešava parametre. Ideja je da prema implementiranoj logici u zavisnosti od greške i od promene greške unosi promene parametara sa ciljem postizanja minimalne greške.

### 3.2.1 Princip fuzzy kontrole

Fuzzy kontrola se bazira na logici koja koristi "ako-onda" pravila slična ljudskom razmišljanju. Umesto korišćenja striktno definisanih ulaza i izlaza, fuzzy kontroler koristi skupove koji omogućavaju fleksibilnije i intuitivnije upravljanje sistemom.

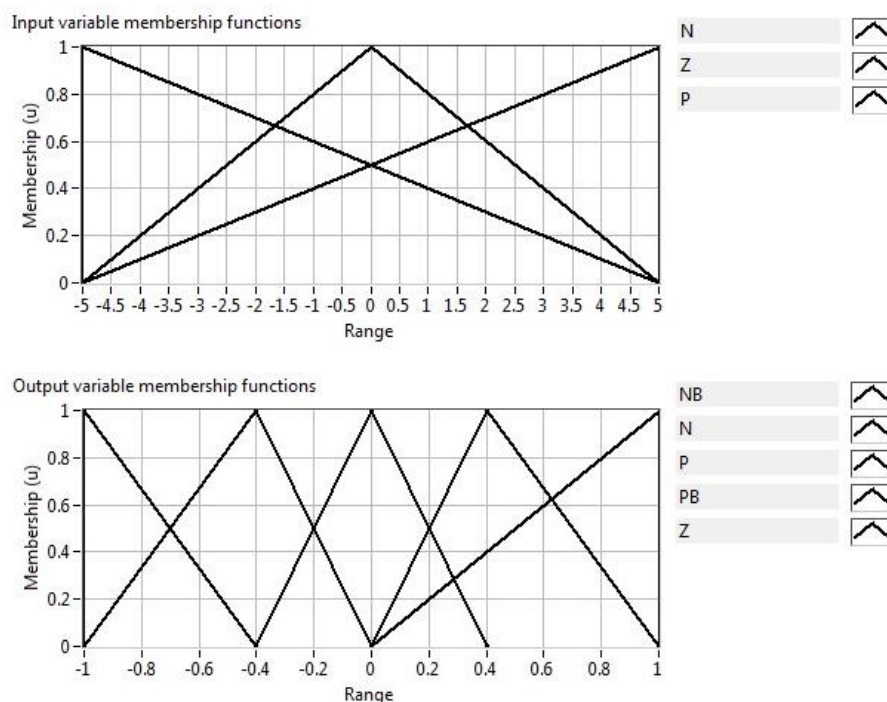
Proces dizajniranja fuzzy kontrolera uključuje više koraka, počev od definisanja ulaznih i izlaznih varijabli. U konkretnom slučaju, fuzzy kontroler će u zavisnosti od greške i od promene greške na ulazu modifikovati izlaz, odnosno menjati proporcionalnu komponentu  $K_p$ .

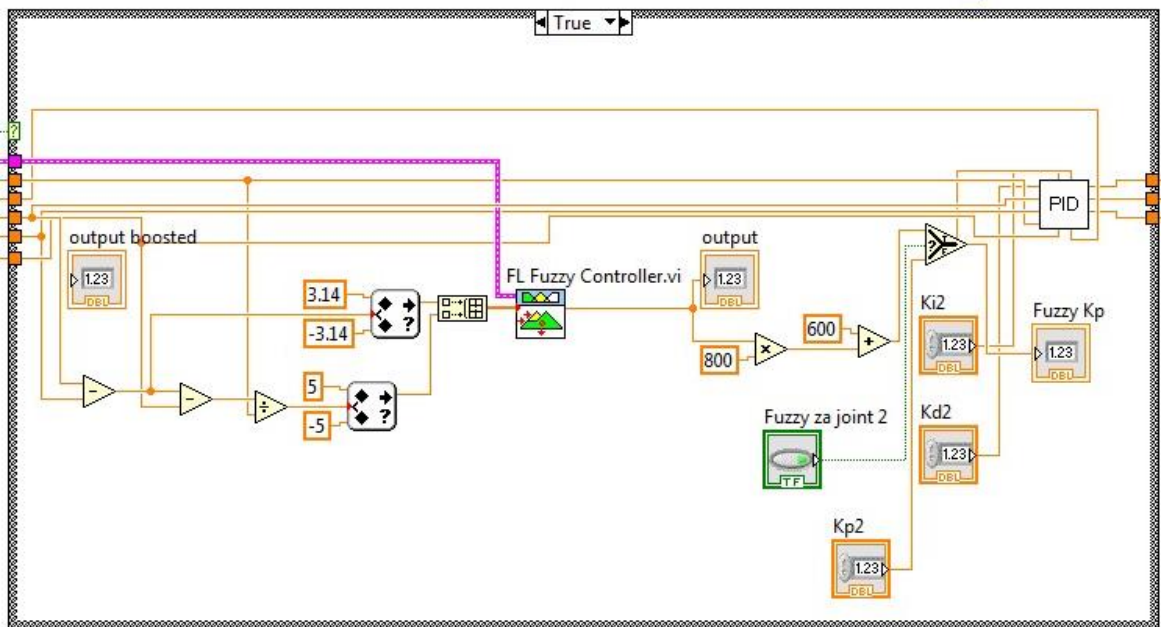
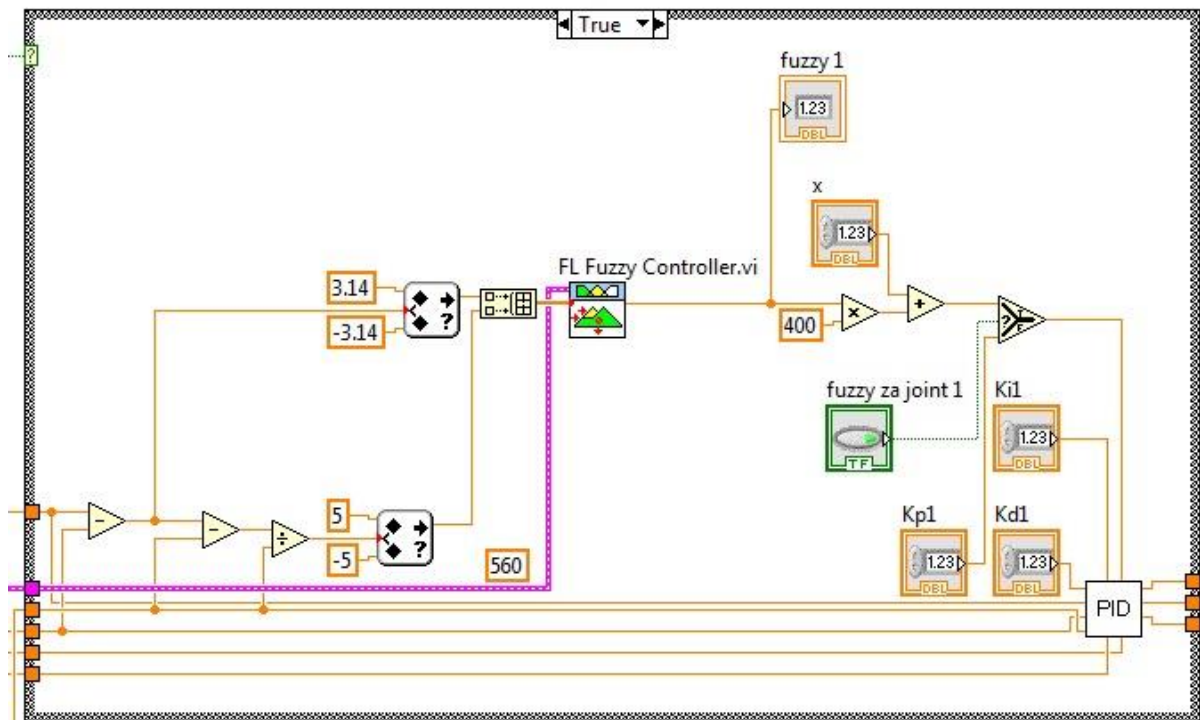
Sledeći korak je fuzzifikacija koja predstavlja konvertovanje stvarnih vrednosti ulaznih varijabli u neizrazite vrednosti koristeći funkcije pripadnosti. Na primer, greška je okarakterisana kao „mala“. Takođe, neophodno je i definisati skup pomenutih if-then pravila, kao što u implementiranom fuzzy-ju postoji, recimo, pravilo: *if e is P and de is P, then u is PB*.

Zatim se inferencijalnim mehanizmom primenjuju pravila na ulaze kako bi se dobili neizraziti izlazi koji se defuzzifikacijom konvertuju nazad u stvarne vrednosti na osnovu kojih će se vršiti dalje prilagođavanje parametara PID kontrolera.

### 3.2.2 Implementacija fuzzy kontrolera

Nakon što je fuzzy kontroler inicijalizovan, odnosno, nakon što je kreiran u MATLAB-u sa prethodno definisanim ulazima, izlazima i pravilima, izvršena je integracija sa PID kontrolerom u LabView-u. Zatim je svakako izvršeno i testiranje pomoću simulacija kako bi se proverila efikasnost fuzzy kontrolera u smanjenju greške i poboljšanju performansi sistema, a dodatna optimizacija je omogućena finim podešavanjem funkcija pripadnosti i pravila.





### 3.3 Sliding Mode Control

Sliding Mode Control ili upravljanje kliznim režimom je sve popularnija, a jedna od najrobustnijih i najefikasnijih metoda nelinearne kontrole posebno korisna za sisteme sa promenljivim parametrima ili neizvesnostima. SMC je pronašao veliku primenu u različitim oblastima kao što su robotika, aeronautika, automobilska industrija i energetska elektronika. U pitanju je tehnika povratne sprege koja koristi nelinearan zakon upravljanja kojim bi se stanje sistema dovelo do klizne površine na kojoj će njegova dinamika biti pojednostavljena i održalo na toj površini obično definisanoj hiperplanom u prostoru stanja.

Klizna površina se definiše pomoću prekidačke, odnosno kontinuirane funkcije koja menja znak kada stanje sistema pređe površinu. Zakon upravljanja je nelinearan jer se menja između dva režima rada u zavisnosti od znaka funkcije prekidača.

Postoje dva osnovna režima rada, i to klizni režim i režim dostizanja. U kliznom režimu se ulaže veliki napor upravljanja da održi sistem na kliznoj površini, dok se u režimu dostizanja primenjuje nulti napor upravljanja kako bi se sistem samo usmerio ka kliznoj površini.

Dizajniranjem funkcije prekidača i zakona upravljanja tako da stanje sistema ostane na kliznoj površini u svakom trenutku ispunjava se osnovni cilj SMC-a i osigurava da je sistem robustan na neizvesnosti, smetnje i nelinearnosti koje mogu uticati na njegovu dinamiku.

Podešavanje SMC-a uključuje odabir odgovarajuće klizne površine, kao i funkcije prekidača, zakona upravljanja i drugih parametara tako da performanse sistema zasigurno ispunjavaju specifikaciju.

### 3.3.1 SMC implementacija

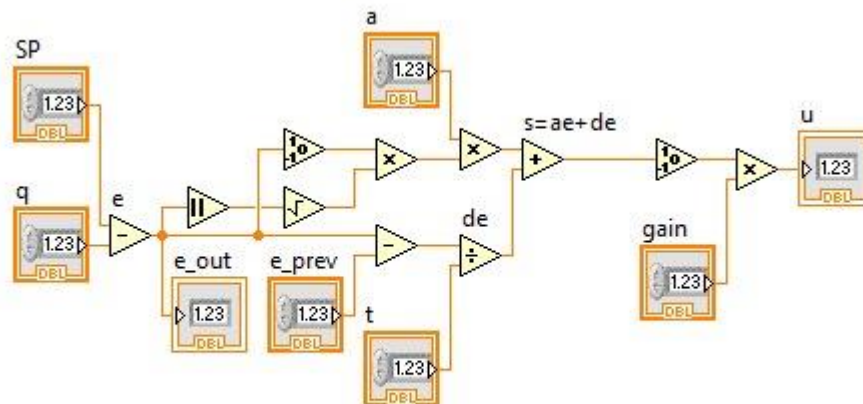
Kako linearnim režimima kontrole nisu postignuti željeni rezultati, implementirana je i nelinearna strategija upravljanja sa ciljem poboljšanja i radi mogućnosti poređenja različitih pristupa kontrole.

SMC je adekvatan izbor za konkretan slučaj, budući da je, osim što teorijski ima dobre performanse, pogodan za sisteme sa nesigurnostima, čemu ovaj tip sistema jeste podložan zbog međusobnog uticaja promena jednog zgloba na drugi.

Pri implementaciji su kao osnovni parametri korišćeni greška i izvod greške, dok je površina klizanja definisana pomoću korene funkcije. Ovaj nelinearni oblik funkcije izabran je jer je eksperimentalno utvrđeno da pruža bolje performanse u opisu našeg sistema.

Tokom implementacije, posebna pažnja je posvećena podešavanju parametara. Vrednost koeficijenta je prilagođena da optimizuje performanse sistema u smislu brzine odziva i stabilnosti. Maksimalna snaga kontrolnog signala je ograničena na  $\pm 100$ , kako bi se osigurala stabilnost, a sprečilo preopterećenje aktuatora.

Takođe je korišćena i povratna informacija iz sistema kako bi parametri bili dinamički prilagođavani u realnom vremenu, a robusnost kontrole ostvarena. Eksperimentalna validacija je pokazala značajno poboljšanje u preciznosti i stabilnosti upravljanja u poređenju sa prethodnim linearnim metodama.



### 3.4 Uporedna analiza performansi kontrolera

	konvencionalni PID	PID sa fuzzy	SMC
Prednost	Jednostavna implementacija	Efikasnost i pouzdanost u radu	Jednostavno i intuitivno podešavanje parametara
Prednost	Brže i manje zahtevno izvršavanje	Prilagodljivost i modularnost	Visok stepen robustnosti i otpornosti na greške
Mana	Loše performanse u radu sa nelinearnim sistemima	Kompleksna implementacija	Veliko opterećenje aktuatora i osetljivost na kašnjenje
Mana	Nemogućnost praćenja stanja drugih promenljivih u radu sa multivarijabilnim sistemima		Prisustvo oscilacija u ustaljenom stanju

## 4. NAČINI UPRAVLJANJA

### 4.1 Manuelna kontrola – Front Panel

Manuelna kontrola robotske ruke omogućava direktno upravljanje kretanjem robotske ruke pomoću klizača na front panelu. Ova opcija pruža intuitivan način za precizno podešavanje pozicije duž X i Y ose, omogućavajući jednostavno rukovanje i testiranje različitih položaja ruke.

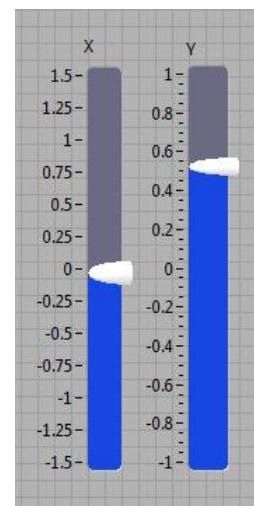
#### 4.1.1 Slide Controls - klizači

Na Front Panel-u se nalaze dva slajdera postavljena da kontrolišu kretanje robotske ruke po X i Y osi i to:

- **za X osu:** Pomeranjem slajdera se kontroliše horizontalno kretanje robotske ruke.
- **za Y osu:** Pomeranjem slajdera se kontroliše vertikalno kretanje robotske ruke.

Svaki pomeraj na slajderima se direktno prenosi na aktuatorske komponente robotske ruke, omogućavajući trenutnu promenu položaja. Ova kontrola je korisna jer ručno podešavanje omogućava brzo testiranje različitih pozicija kako bi se kalibracija sistema izvršila precizno. Osim toga, ovakav vid upravljanja je prilično jednostavan, jer je korisničko iskustvo čak i u eksperimentalnoj fazi dovoljno intuitivno.

Dakle, ovaj način kontrole je ključan za interaktivno i precizno upravljanje robotskom rukom, pružajući osnovu za dalje automatizovane i kompleksnije metode kontrole koje će biti opisane u narednim sekcijama.



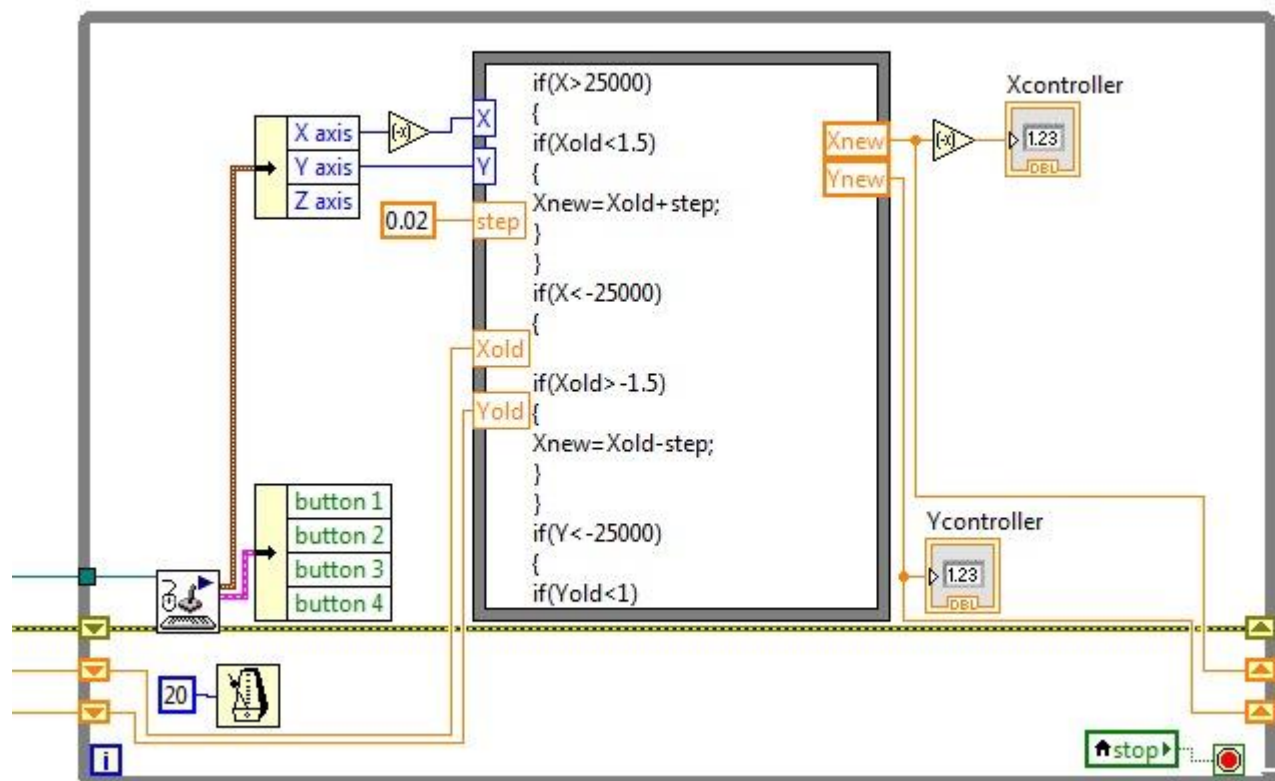
## 4.2 Manuelna kontrola – HID uređaj

Manuelna kontrola preko Human Interface Device (HID) uređaja, kao što je džojstik, omogućava intuitivno i efikasno upravljanje robotskom rukom. Korišćenjem HID uređaja, korisnici mogu lako i precizno manipulirati pozicijom robotske ruke, što povećava fleksibilnost i jednostavnost upravljanja u realnom vremenu.

### 4.2.1 Kontrola džojstikom za X i Y osu

Levi stik na HID uređaju koji je upotrebljen u izradi ovog projekta se koristi za upravljanje kretanjem robotske ruke po X i Y osi. Kao i obično, horizontalno pomicanje džojstika kontroliše kretanje po X osi, dok vertikalno pomicanje kontroliše kretanje po Y osi.

Ova metoda, poput prethodne, omogućava precizno upravljanje uz fino podešavanje pozicije robotske ruke, a samo korišćenje je možda čak i intuitivnije od slajdera, jer je upotreba džojstika jednostavna, a prirodna, poznata i popularna.



## 4.3 Kontrolisano kretanje po predodređenoj putanji

Dodatno je implementirana i mogućnost kontrolisanog kretanja po predodređenoj putanji kao ključna funkcionalnost za demonstraciju sposobnosti sistema da precizno prati zadate putanje realizovana pomoću automata stanja sa redovima unutar Producer-Consumer arhitekture (objašnjeno u nastavku).



Dakle, implementirana su tri dugmića za automatsko pokretanje robotske ruke duž prethodno definisanih putanja za koje su u konkretnom slučaju odabrana tri pokazna primera:

- **Dugme za kružnu putanju:** Pritiskom na ovo dugme, robotska ruka prati kružnu putanju.
- **Dugme za kvadratnu putanju:** Pritiskom na ovo dugme, robotska ruka prati kvadratnu putanju.
- **Dugme za trougaonu putanju:** Pritiskom na ovo dugme, robotska ruka prati trougaonu putanju.

Ideja je bila da i ovaj segment upravljanja bude implementiran na dva načina kao manuelna kontrola, odnosno, da bude moguće vršiti odabir putanje i preko dugmića na Front Panelu i preko dugmića na HID uređaju (zbog čega su i odabrani dati oblici putanja), ali zbog ograničenja u radu sa drajverom ostala je samo mogućnost prednjeg panela.

Dodatno je u okviru Python grafičkog korisničkog interfejsa realizovana i mogućnost kreiranja željene putanje kretanja robotske ruke sa odabranim brojem tačaka na njoj.

#### 4.4 Automat stanja sa redovima i Producer-Consumer arhitektura – implementacija

Implementacija automatskog sistema upravljanja se u ovoj fazi svodi na kreiranje sistema koji će omogućiti efikasno upravljanje sekvencama operacija koje robotska ruka treba da izvrši.

Automati stanja se koriste za upravljanje prelazima između različitih faza procesa, konkretno, kretanja robotske ruke.

Automati stanja su veoma efikasni za upravljanje složenim sekvencama pokreta, jer omogućavaju precizno praćenje putanje, budući da se stanjima automatima predstavljaju tačke koje robotska ruka treba da prati i pozicije koje mora da zauzme pri pomeranju. Osim toga, automati značajno doprinose fleksibilnosti i prilagodljivosti različitim zadacima, jer je lako i jednostavno izvršiti dodavanje, uklanjanje i modifikaciju stanja.

Dakle, svaka putanja je definisana kao niz stanja (tačaka) kroz koja robotska ruka mora proći. Automat sa redovima je tip konačnog automata koji se sastoji od skupa stanja povezanih prelazima i akcijama koje se pokreću na osnovu ulaznih signala ili internih događaja. U konkretnom slučaju će robotska ruka pratiti tačke po redosledu u kom se stanja nalaze u automatu, a to je, po definiciji reda, FIFO struktura.

Za realizaciju kontrolisanog kretanja, implementirana je Producer-Consumer arhitektura tako što:

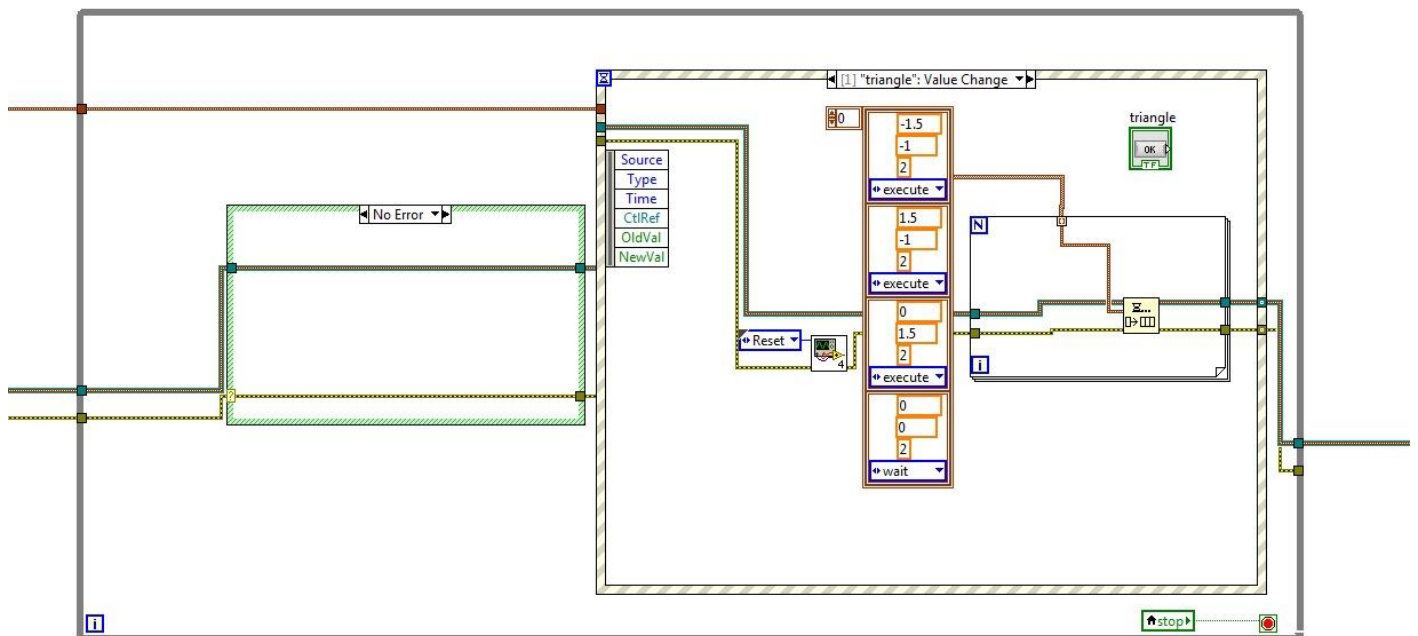
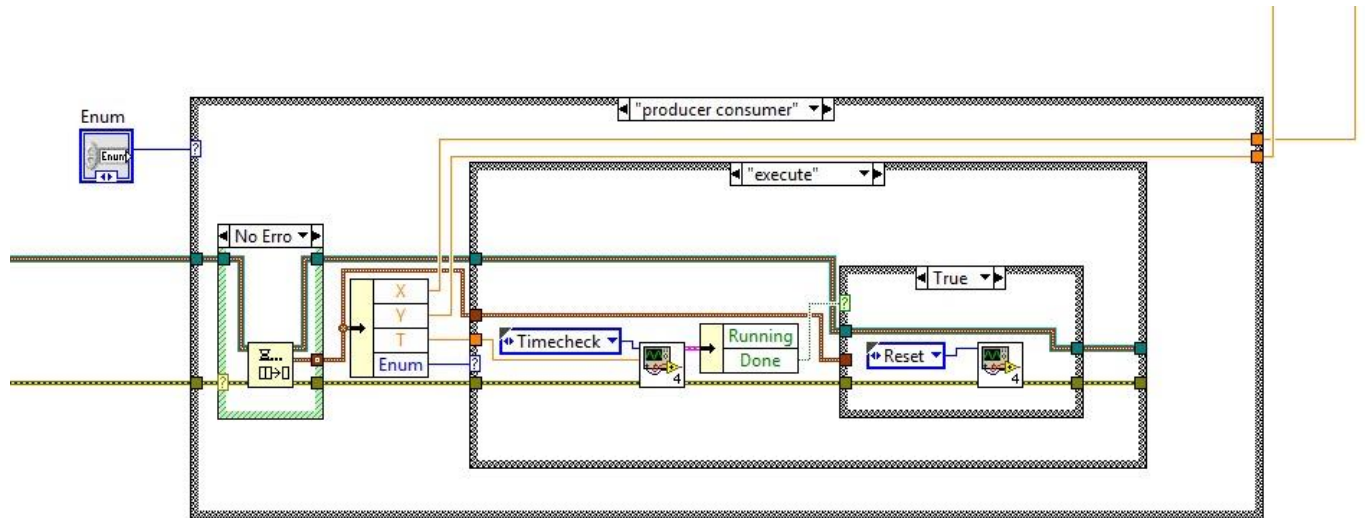
- **Producer** generiše naredbe za kretanje koje se stavljaju u red čekanja.
- **Consumer** preuzima naredbe iz reda čekanja i izvršava ih, pa se robotska ruka kreće prema zadatim koordinatama.

Ova arhitektura omogućava efikasno upravljanje resursima, jer Producer i Consumer mogu raditi paralelno čime se smanjuje vreme čekanja, a povećava efikasnost sistema. Takođe, ovakav sistem je vrlo robustan, jer ga P-C arhitektura štiti od većine grešaka kao što su privremeni prekidi u radu ili promene u putanji, pa će lako i brzo rešavati nepredviđene situacije.

Što se tiče predefinisanih putanja, odabrane su tri različite putanje sa ciljem demonstracije kontrolisanog kretanja, pri čemu su neke od njih implementirane numerički, a neke analitički.

Naime, kvadratna i trougaona putanja su zadate numeričkim hardcoded koordinatama, što je najjednostavniji pristup da se pokaže sposobnost robotske ruke da se pravolinijski kreće i da prati prave, odnosno, oštre uglove. Sa druge strane, kružnom putanjom se demonstrira da je robotska ruka sposobna da precizno prati zakrivljene linije, što je omogućeno analitičkim proračunom pomoću trigonometrijskih funkcija.





## 4.5 Computer Vision kontrola

Inspiracija za implementaciju upravljanja robotskom rukom pomoću prepoznavanja pokreta ljudske ruke povezana je sa gradivom iz drugih predmetnih oblasti, ali kao pozitivan primer povezivanja, budući da je upotrebom principa kompjuterskog vida, detekcije i praćenja objekata omogućen intuitivniji način interakcije sa robotom.

U ovom kontekstu, Computer Vision se koristi za identifikaciju i praćenje specifičnih tačaka na ruci korisnika, što omogućava precizno prenošenje pokreta na robotsku ruku. Ovaj pristup uz prirodnu kontrolu robota u realnom vremenu svakako samo poboljšava korisničko iskustvo.

Za komunikaciju između Python skripte koja obrađuje sliku koju dobija sa stream-a i LabVIEW softvera koji upravlja robotskom rukom odabran je UDP (User Datagram Protocol). Isprva je razmatran upotreba TCP, ali su se u radu, zbog veće složenosti samog protokola, javile tehničke poteškoće. Dakle, komunikacija se ostvaruje preko IP adrese i porta za prenos podataka, pri čemu se koordinate šalju kao 1D lista uređenih parova.

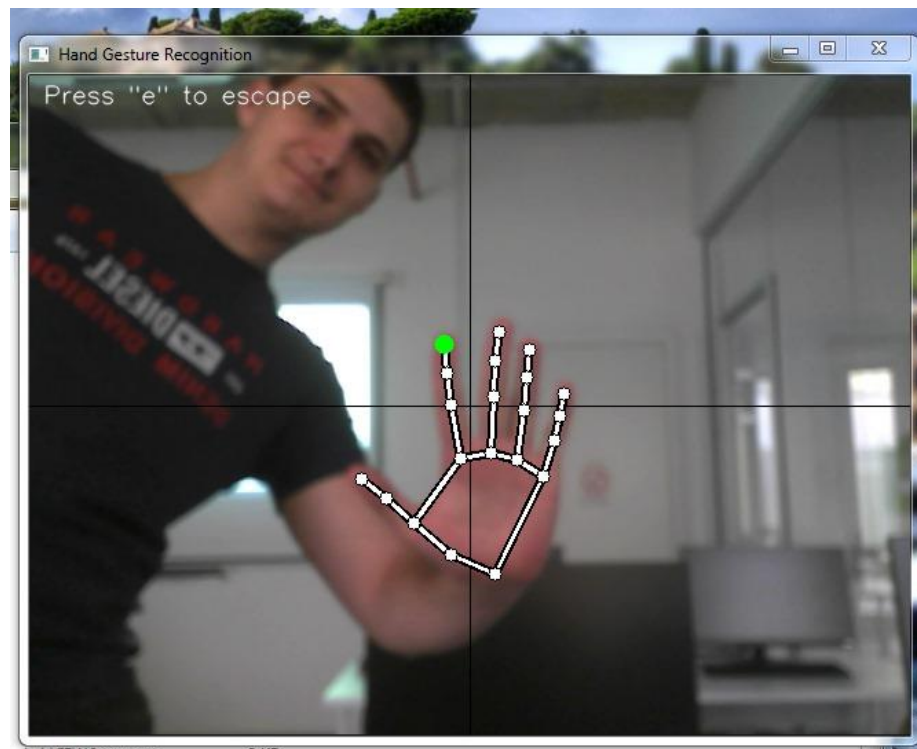
Takođe, napravljen je i adekvatan, a jednostavan grafički korisnički interfejs u Python aplikaciji.



#### 4.5.1 Hand recognition

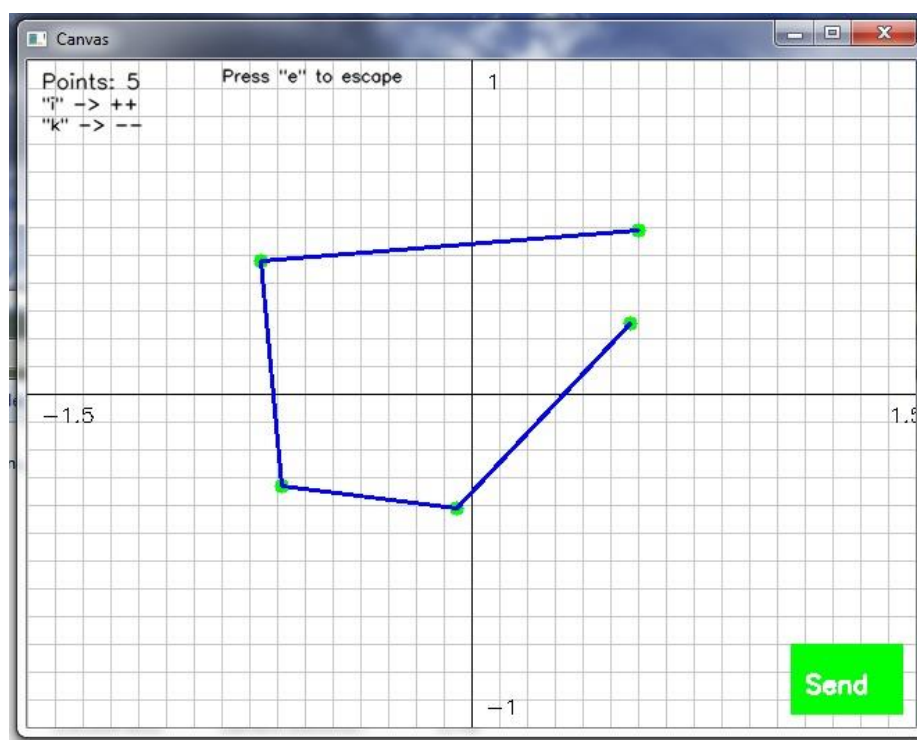
Pri izradi ovog dela projektnog zadatka korišćene su Python biblioteke OpenCV i MediaPipe za obradu slike i video prenosa. Implementirana je funkcionalnost prepoznavanja ruke pomoću prethodno obučenog modela koji prati koordinate dvadeset tačaka na šaci, od kojih se uzima samo vrh kažiprsta, pa se njegove koordinate normalizuju kako bi robotska ruka mogla da oponaša kretanje.

Centar kamere se dalje koristi kao početna tačka koordinatnog sistema, te se u odnosu na tu tačku referenciraju i koordinate kažiprsta, a samim tim i pozicije koje zauzima robotska ruka. Na taj način, kretanje robotske ruke sinhronizuje se sa pokretima ruke koja se detektuje i prati na kameri.



## 4.5.2 Kreiranje željene putanje – Drawing Mode

Na sličan način funkcioniše i obrada putanje zadate u Drawing Mode-u, budući da se odatle šalju koordinate koje se dalje obrađuju. U LabVIEW-u se vrši parsiranje, pa se dobijene odvojene vrednosti prosleđuju kao double x i y koordinate, a pozicije se stavljaju u red kao različita stanja kroz koja automat periodično prolazi.



## 5. VIZUALIZACIJA

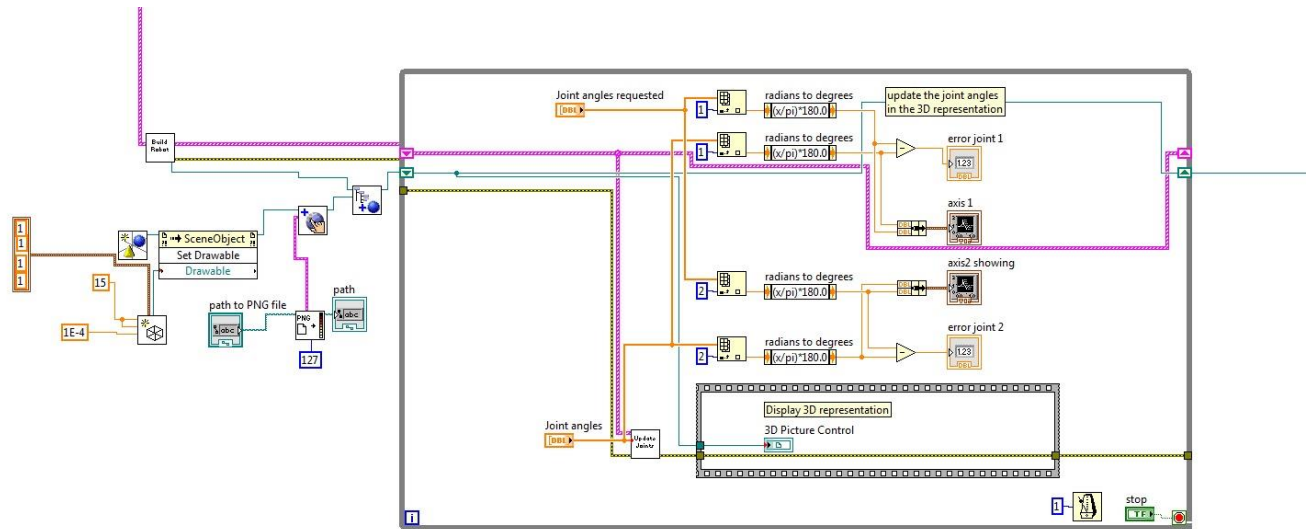
Vizualizacija sistema je ključni aspekt za razumevanje i efikasnu kontrolu kretanja robotske ruke. Konkretno, vizualizacija robotske ruke omogućava korisnicima da prate kretanje ruke u realnom vremenu kroz trodimenzionalni prikaz. Korišćenjem 3D grafike, prikazujemo trenutni položaj robotske ruke.

### 5.1 3D vizualizacija – implementacija

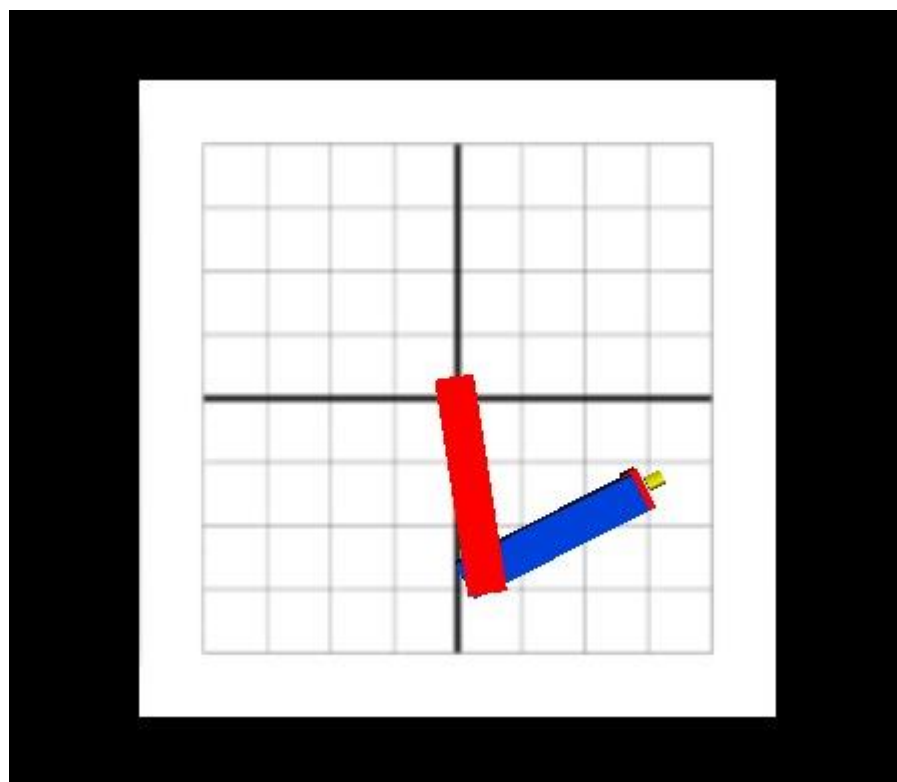
Trodimenzijski prikaz pruža više pogodnosti, jer značajno olakšava praćenje u razvojnim fazama i unapređuje korisničko iskustvo. Uz 3D prikaz je moguće pratiti kretanje robotske ruke iz različitih uglova i u realnom vremenu, i to sve uz realističan prikaz stvarnog izgleda i ponašanja. Na taj način je obezbeđeno intuitivno praćenje koje nudi i interaktivne opcije poput rotiranja, zumiranja, prevlačenja, pomeranja i td.

Implementacija 3D vizualizacije realizovana je uz pomoć naprednih grafičkih biblioteka iz LabVIEW-a koje su specijalizovane za kreiranje i prikazivanje trodimenzionalnih objekata kao što su zglobovi???, budući da poseduju funkcionalnosti za crtanje, transformaciju i animaciju 3D modela. Model robotske

ruke je svakako baziran na njenim stvarnim dimenzijama i karakteristikama, pa verno, iako veoma pojednostavljeno, reprodukuje njen izgled i ponašanje. Model se sastoji od više objekata grafičke 3D biblioteke koji su poređani u specifičnom redosledu, odnosno od cilindričnog kućišta preko zglobova do vrha robotske ruke (drugog zgloba).



Block Diagram



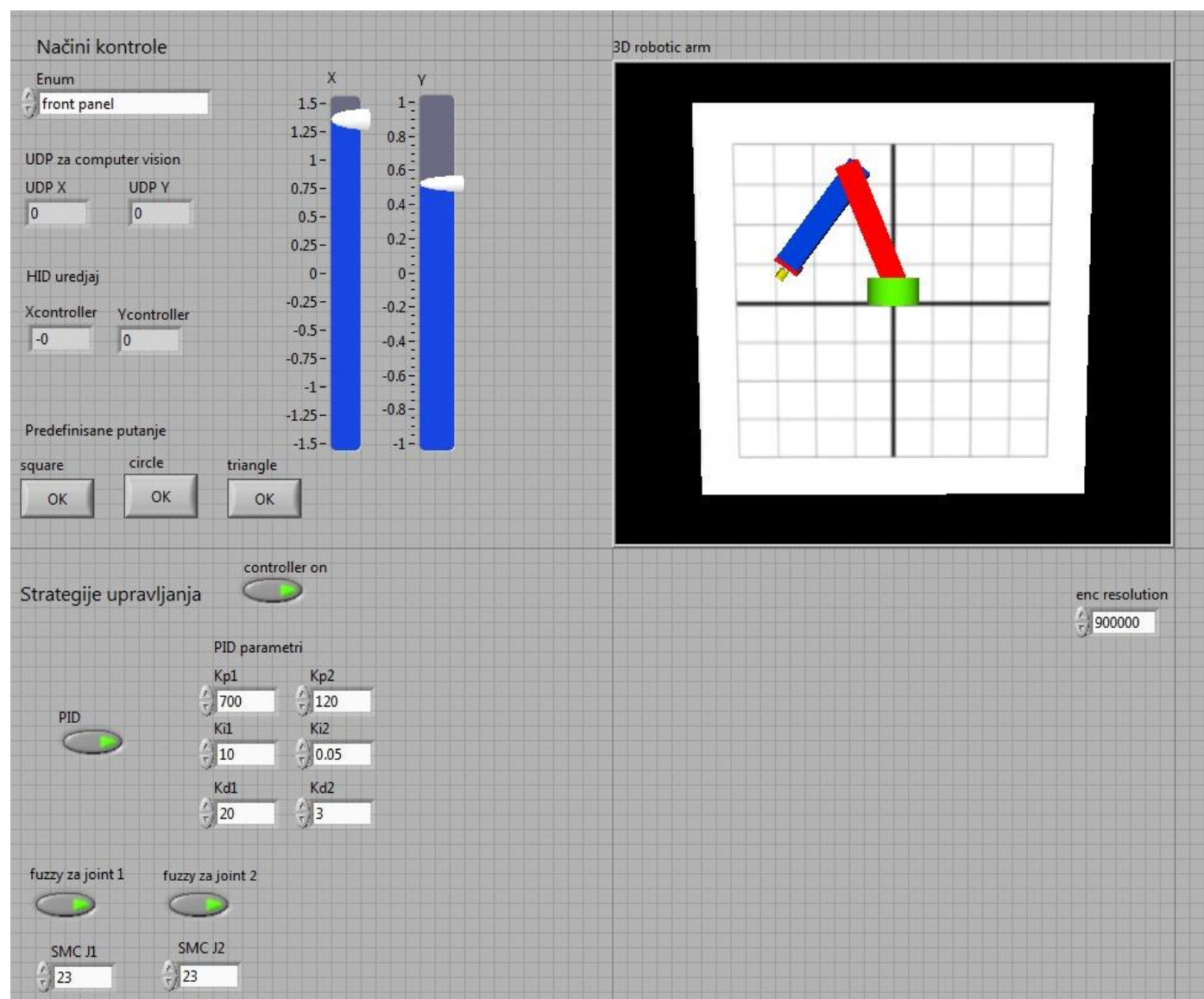
Front Panel

## 5.2 Korisnički interfejs

Korisnički interfejs (Graphic User Interface) dizajniran je sa fokusom na intuitivnost i jednostavnost korišćenja. Glavne karakteristike interfejsa uključuju interaktivne kontrole za upravljanje robotskom rukom, kao što su slajderi ili dugmad za odabir režima rada, putanje i td. Na taj način se preko interfejsa brzo i lako odabira manuelno ili automatsko upravljanje, pa onda u skladu sa tim vrši adekvatno upravljanje. Osim toga, na Front Panel-u se nalaze i indikatori za prikaz važnih informacija o trenutnom stanju robotske ruke, kao što je, recimo, trenutna pozicija, brzina, greška upravljanja, grafičke reprezentacije i drugi pokazatelji pomoću kojih se olakšano prati ponašanje sistema i blagovremeno reaguje na neželjene promene. GUI pruža vizuelne povratne informacije u realnom vremenu, što je posebno korisno za kontrolisanje i fino podešavanje parametara, a svakako poboljšava korisničko iskustvo i doprinosi razumljivosti.

Sa obzirom na to da je u pitanju predmetni projekat, postoji više pokaznih opcija kojima se uz pomoć prekidačkih kontrola demonstrira razlika između, npr. upravljačkih tehnika, upotrebljenih kontrolera, odabranog režima rada i slično.

Dakle, korisnički interfejs je realizovan kao kombinacija grafičkih elemenata, interaktivnih kontrola kako bi pružila sveobuhvatno i po mnogo čemu ilustrativno iskustvo upravljanja i upotrebe robotske ruke.



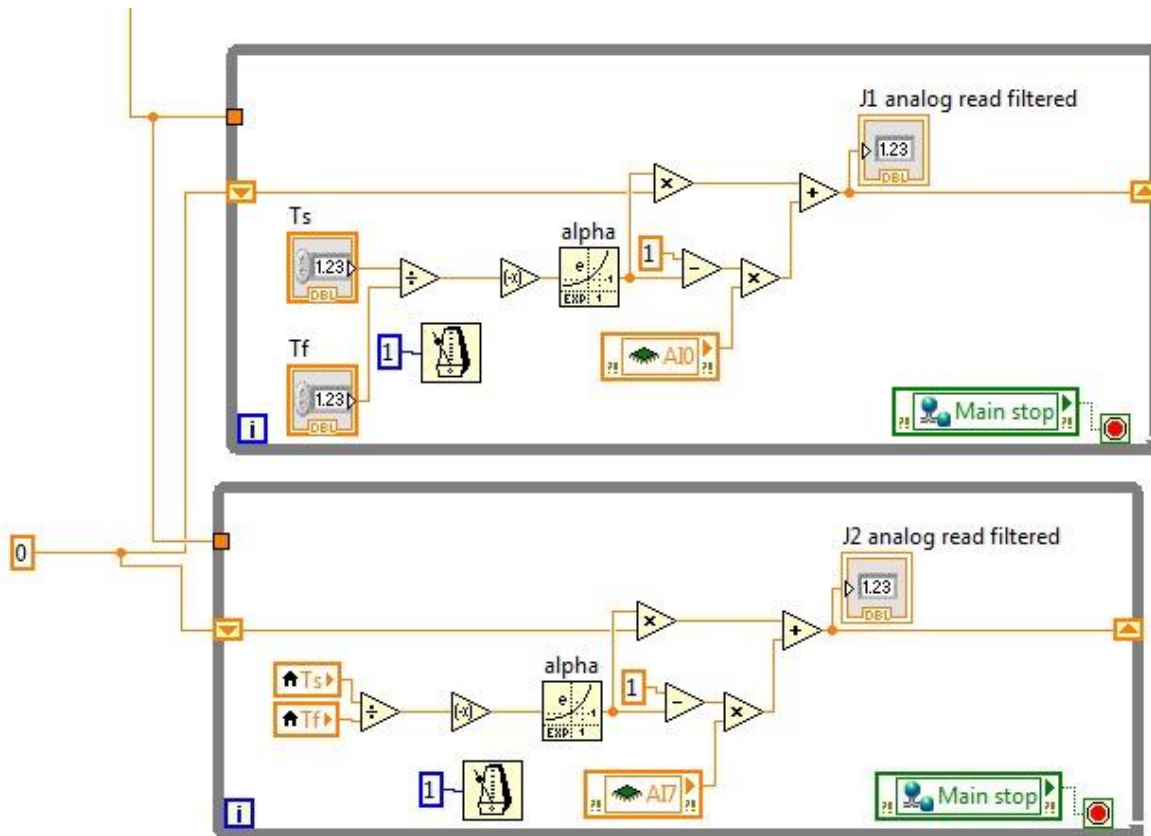
## 6. HARDWARE

Odlučeno je da se sistem evaluira na hardveru kako bi se dublje ispitala performanse i funkcionalnost u stvarnom okruženju, što je motivisano željom da se sistem testira u realnom vremenu i proverí efikasnost.

Za evaluaciju performansi sistema korišćene su cRIO i sbRIO platforme.

cRIO je odabran zbog svoje skalabilne arhitekture i pouzdanosti u integraciji kontrolnih sistema, dok sbRIO nudi visokoperformantan rad i nisko vreme kašnjenja, što je ključno za realno vremenske aplikacije.

Zbog pojave šuma na ulazima i izlazima tokom eksperimentalnih testiranja, bilo je neophodno implementirati filter prvog reda radi poboljšanja stabilnosti signala. Frekventni filter je dizajniran da filtrira visokofrekventne šumove i time poboljšava preciznost detekcije i kontrolu pokreta robotske, jer iako je uvedeno vremensko kašnjenje, dobijen je značajno "čistiji" signal. Takođe, šum dovodi nagle poremećaje, odnosno, promene pozicije bez promene stanja i konkretnih događaja u sistemu, pa je filtriranjem i to sprečeno.





## 7. ZAKLJUČAK

### 7.1 Analiza rešenja i postignutih rezultata

Sistem je testiran u različitim scenarijima kako bi se procenila njegova tačnost i pouzdanost. Testiranje je obuhvatilo sve strategije upravljanja i načine kontrole, različite brzine pokreta, kombinacije pozicija i dr.

Performanse sistema su uglavnom zadovoljavajuće, iako dosta promenljive u zavisnosti od toga koji kontroler je u upotrebi, o čemu je više reči bilo u delu za komparativnu analizu upravljanja.

Takođe, sistem vrlo nesigurne prirode je pokazao sposobnost stabilnog rada uz određenu robustnost u okvirima simulacije. Hardverska interpretacija je značajno manje efikasna i precizna, ali je i time ispunjen osnovni cilj, a to je pokazni i evaluativni aspekt.

Postignuto je i pozitivno korisničko iskustvo uz laku intuitivnu upotrebu za svaki od načina upravljanja, budući da i klizne kontrole i džojstik i pokreti ruke omogućavaju prilično prirodnu interakciju sa robotskim sistemom.

Zaključno, rezultati testiranja implementiranog rešenja ovog predmetnog projekta su bili zadovoljavajući, pogotovo sa obzirom na domen predmeta i nivo znanja u ovoj oblasti.

### 7.2 Moguća poboljšanja i modifikacije

Sveukupno, implementirani sistem je pokazao visok nivo efikasnosti i korisnosti.

Naravno, preporučljiv je dalji rad na poboljšanju i potencijalnim modifikacijama.

Ideja	Kako implementirati?
Proširenje funkcionalnosti robotske ruke	<ul style="list-style-type: none"><li>– Dodavanje hvataljke</li><li>– Simulacija hvatanja objekta i manipulacije njime</li></ul>
Poboljšanje diskretizacije	<ul style="list-style-type: none"><li>– Primena metode Runge-Kuta i slično</li></ul>
Analitički pristup rešenju	<ul style="list-style-type: none"><li>– Računanje PID parametara u više tačaka</li></ul>
Optimizacija rada HIL-a	<ul style="list-style-type: none"><li>– Model Predictive tehnika upravljanja</li></ul>
Adaptivno upravljanje	<ul style="list-style-type: none"><li>– Dinamičko podešavanje parametara u skladu sa promenama stanja sistema</li></ul>
Komunikacija sa eksternim modulima	<ul style="list-style-type: none"><li>– Preko LabView web servisa</li></ul>
Precizna fuzzy kontrola	<ul style="list-style-type: none"><li>– Kontroler koji uzima u obzir sva stanja sistema</li></ul>