Projektiranje i upravljanje sustava kuglice na kotaču

Mustać, Jurica

Undergraduate thesis / Završni rad

2017

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:677742

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Jurica Mustać

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:
Prof. dr. sc. Željko Šitum, dipl. ing.

Student:
Jurica Mustać

Zagreb, 2017.

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

Zahvaljujem se mentoru, prof. dr. sc. Željku Šitumu, obitelji i kolegama na pruženoj podršci i savjetima koji su uvelike pomogli pri izradi ovog rada.

Jurica Mustać

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **JURICA MUSTAĆ** Mat. br.: 0035194969

Naslov rada na hrvatskom jeziku: **PROJEKTIRANJE I UPRAVLJANJE SUSTAVA KUGLICE NA KOTAČU**

Naslov rada na engleskom jeziku: **DESIGN AND CONTROL OF A BALL-ON-WHEEL SYSTEM**

Opis zadatka:

Balansirajući sustav kuglice na kotaču spada u skupinu laboratorijskih modela koji se mogu koristiti za eksperimentalnu provjeru naprednih algoritama regulacije i za objašnjenje teorije upravljanja multivarijabilnih, nelinearnih sustava koji su uz to inherentno nestabilni i podupravljani. Sustav se sastoji od kotača pogonjenog nekim aktuatorom, u ovom slučaju električnim motorom, kuglice koja se giba na kotaču, senzora položaja kuglice i upravljačke elektronike. Regulacijski zadatak svodi se na održavanje ravnotežnog položaja kuglice na gornjoj strani kotača i otklanjanje poremećaja tijekom procesa stabilizacije. Za uspješno rješenje zadatka potrebno je ostvarenje brzog procesiranja mjerenih signala, brzo izvršenje upravljačkog algoritma i kratko vrijeme reakcije izvršnih članova. Ovakvi laboratorijski sustavi prikladni su za edukaciju iz područja regulacije, mehatronike, senzorike, programiranja i dr.

U radu je potrebno:

- izvesti nelinearni i linearizirani matematički model sustava kuglice na kotaču u obliku prikladnom za simulaciju,
- razmotriti moguće načine regulacije sustava, načiniti sintezu regulatora i izvršiti simulaciju procesa, te odabrati najpovoljnije rješenje s obzirom na praktičnu izvedbu,
- izraditi nastavnu maketu procesa i ugraditi komponente mjernog i upravljačkog dijela sustava,
- razvijene regulacijske algoritme eksperimentalno provjeriti na laboratorijskom modelu.

Zadatak zadan:

30. studenog 2016.

Rok predaje rada:
1. rok: 24. veljače 2017.
2. rok (izvanredni): 28. lipnja 2017.
3. rok: 22. rujna 2017.

Predviđeni datumi obrane:
1. rok: 27.2. - 03.03. 2017.
2. rok (izvanredni): 30. 06. 2017.
3. rok: 25.9. - 29. 09. 2017.

Zadatak zadao:

Prof. dr. sc. Željko Šitum

v.d. predsjednika Povjerenstva:

Izv. prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ ..... **I**
POPIS SLIKA ..... **III**
POPIS OZNAKA ..... **IV**
SAŽETAK ..... **VI**
SUMMARY ..... **VII**
1. UVOD **1**
2. MODELIRANJE SUSTAVA **2**
2.1. EULER-LAGRANGEOVA METODA 2
2.2. PRIKAZ SUSTAVA 2
2.3. PODSUSTAVI 3
2.3.1. PODSUSTAV KOTAČA 4
2.3.2. PODSUSTAV KUGLICE 4
2.4. JEDNADŽBE SUSTAVA 6
2.5. PRIMJENA EULER-LAGRANGEOVE JEDNADŽBE 6
2.5.1. KUT ZAKRETA KOTAČA - VARIJABLA α 7
2.5.2. KUT POMAKA KUGLICE NA KOTAČU - VARIJABLA β 7
2.5.3. UVJET GIBANJA KUGLICE PO RUBU KOTAČA 8
2.6. LINEARIZACIJA SUSTAVA 9
2.7. SREĐIVANJE JEDNADŽBI 9
3. SIMULACIJA SUSTAVA **10**
3.1. PROSTOR STANJA 10
3.2. PREVOĐENJE SUSTAVA U PROSTOR STANJA 10
3.3. BLOK DIJAGRAM SUSTAVA 11
3.3.1. PARAMETRI SUSTAVA 12
3.3.2. POČETNI UVJETI SUSTAVA 12
4. PROJEKTIRANJE REGULATORA **13**
4.1. ODABIR REGULATORA 13
4.2. REGULATOR PO VARIJABLAMA STANJA 13
4.3. ODABIR TEŽINSKIH MATRICA Q I R 14
4.4. POJAČANJE REGULATORA 14
4.5. SIMULACIJA I ODZIV SUSTAVA 14
4.6. LQR REGULATOR U SIMULINK MODELU 15
5. IZBOR KOMPONENTI REALNOG SUSTAVA **18**

5.1. IZBOR AKTUATORA 18
5.2. KORAČNI MOTOR 18
5.3. PRORAČUN AKTUATORA 19
5.4. ODABIR UPRAVLJAČKOG SKLOPA 20
5.5. ODABIR SENZORA U SUSTAVU 21
5.5.1. ODABIR SENZORA UDALJENOSTI 22
5.6. ODABIR MIKROPROCESORA 23
6. KONSTRUKCIJA I UGRADNJA KOMPONENTI 25
6.1. MODELIRANJE I IZRADA KONSTRUKCIJE 25
6.2. KALIBRIRANJE SENZORA UDALJENOSTI 31
6.3. MOMENTNA KARAKTERISTIKA KORAČNOG MOTORA 33
7. IZVEDBA EKSPERIMENTA I REZULTATI 35
7.1. REGULACIJSKI ALGORITAM 35
7.1.1. DEKLARACIJA POTREBNIH VARIJABLI 35
7.1.2. MIJENJANJE TIMER-A 36
7.1.3. FUNKCIJA MJERENJA 38
7.1.4. SETUP I LOOP 39
7.2. REZULTATI EKSPERIMENTALNOG POSTAVA 40
8. ZAKLJUČAK 41
LITERATURA 42
PRILOZI 44
1. CD-R 44
2. MATLAB KOD 44

POPIS SLIKA

Slika 1.1. Izrađeni sustav kuglice na kotaču 1
Slika 2.1. Skica sustava kuglice na kotaču 3
Slika 2.2. Brzina rotacije kuglice 5
Slika 2.3. Uvjet gibanja kuglice 8
Slika 3.1. Blok dijagram sustava 11
Slika 3.2. Odziv sustava 12
Slika 4.1. Odziv sustava uz LQR regulator 15
Slika 4.2. Blok dijagram sustava s regulatorom po varijablama stanja 15
Slika 4.3. Odziv reguliranog sustava u Simulink-u 16
Slika 4.4. Kontaktna sila između kuglice i kotača 17
Slika 5.1. Koračni motor ST6018L3008 [7] 19
Slika 5.2. Upravljački sklop THB6064AH JP-6445A [9] 21
Slika 5.3. Infracrveni senzor udaljenost GP2Y0A21YK0F [11] 22
Slika 5.4. Shema električnog sklopa senzora [12] 23
Slika 5.5. Mikrokontroler Arduino Nano [14] 24
Slika 6.1. Okvirno modelirana konstrukcija 25
Slika 6.2. Metalni dio konstrukcije 26
Slika 6.3. Modelirani kotač 27
Slika 6.4. Svi dijelovi konstrukcije 27
Slika 6.5. Izrezani aluminijski profili 28
Slika 6.6. Izrađeni dio konstrukcije za motor 28
Slika 6.7. Spoj između motora i kotača 29
Slika 6.8. Modelirani dio za spoj senzora 30
Slika 6.9. Spoj senzora udaljenosti na konstrukciju 30
Slika 6.10. Gotova konstrukcija 31
Slika 6.11. Kalibriranje senzora 32
Slika 6.12. Karakteristika senzora po uputama proizvođača [12] 32
Slika 6.13. Generirana karakteristika senzora 33
Slika 6.14. Momentna karakteristika odabranog koračnog motora [17] 34
Slika 6.15. Generirana krivulja momentne karakteristike 34
Slika 7.1. Deklarirane varijable 35
Slika 7.2. Mijenjanje vrijednosti Timer-a 37
Slika 7.3. Funkcija mjerenje() 38
Slika 7.4. Digitalni regulator 39

POPIS OZNAKA

Oznaka	Jedinica	Opis
A	-	matrica koeficijenata sustava,
B	-	matrica ulaza,
C	-	matrica izlaza,
$C_{k,l}$	-	točke dodira između kuglice i kotača,
D	-	matrica direktnog preslikavanja ulaza,
e_i	-	maksimalno dozvoljena odstupanja varijabli,
g	m/s^2	gravitacijsko ubrzanje,
I_j	kgm^2	moment inercije *j*-te mase,
K	-	vektor pojačanja,
K_j	J	Kinetička energija *j*-te mase,
M	kg	masa kotača,
m	kg	masa kuglice,
N	N	reakcijska sila između kuglice i kotača,
n	-	broj varijabli stanja,
$O_{k,l}$	-	središta koordinatnih sustava kuglice i kotača,
P_j	J	Potencijalna energija *j*-te mase,
p	-	broj ulaza,
Q	-	težinska matrica,
q	-	broj izlaza,
q_i	m ili rad	*i*-ta upravljana koordinata,
$\dot{q}_i$	m/s ili rad/s	brzine *i*-te upravljane koordinate,
R	-	težinska matrica,
R	m	radijus kotača,
r	m	radijus kuglice,
$T_{i,j}$	N ili Nm	sila ili moment *i*-te koordinate za pokretanje *j*-te mase,
u	-	vektor ulaza,
v_j	m/s	brzina *j*-te mase,

$\mathbf{X_0}$	-	vektor početnih uvjeta,
$\mathbf{x}$	-	vektor varijabli stanja,
$\dot{\mathbf{x}}$	-	vektor derivacija varijabli stanja,
x,y	-	koordinatne osi,
$\mathbf{y}$	-	vektor izlaza,
α	rad	kut zakreta kotača,
β	rad	kut pomaka kuglice na kotaču,
φ	rad	kut zakreta kuglice,
$\dot{\alpha},\dot{\beta},\dot{\varphi}$	rad/s	kutne brzine,
$\ddot{\alpha},\ddot{\beta}$	rad/s^2	kutne akceleracije,

SAŽETAK

Kroz završni rad opisana je ideja i izvedba sustava kuglice na kotaču. Najprije je izveden dinamički model sustava primjenom odgovarajućih jednadžbi i izvršena njegova pretvorba u prostor stanja. Provedena je sinteza LQR regulatora te izvršena i objašnjena simulacija sustava u Simulink programskom okruženju. Slijedi prikaz konstruiranja same konstrukcije te tijek njene izrade. Objašnjavaju se komponente upravljačkog i mjernog dijela sustava. Komentirane su njihove karakteristike i prikazana njihova aplikacija na samom sustavu. Prikazan je razvijeni regulacijski algoritam koji je napisan u Arduino IDE programskom okruženju. Regulacijski algoritmi testirani su i na samoj izrađenoj maketi te su komentirani dobiveni rezultati.

Svrha ovog rada bila je izraditi nastavnu maketu koja objašnjava načine regulacije nestabilnih sustava te iskoristiti i primijeniti što više znanja i vještina stečenih do sada na fakultetu. Kroz rad su korišteni razni programski paketi i alati za obradu te je realizirani sustav ogledni primjer mehatroničkog sustava; fuzija strojarstva, elektronike i računarstva.

Ključne riječi: mehatronika, kuglica na kotaču, lqr regulator, podupravljani sustav, nastavna maketa

SUMMARY

In the following thesis, the idea and implementation of the Ball and wheel system is discussed. A dynamical model of the system is created with the application of the appropriate equations, and it's transformation into the state space representation is shown. Synthesis of the LQR regulator is conducted and the simulation of the system is executed and explained in the Simulink enviroment. Furthermore, the design of the construction and the course of it's buildup is shown. Components of the control and measurement part of the system are introduced too. Their properties are explained and their application on the system is shown . The developed control algorithms in the Arduino IDE is shown too. Control algorithms are tested on the actual system and the results are commented.

The point of this thesis was to make a teaching model which explains control methods of the unstable systems and to use as many knowledge and skills aquired during the undergraduate studies. Throughout the thesis, many software packages and tools are used so the resulting system is a prime example of a mechatronic system; fusion of the mechanical, electrical and software engineering.

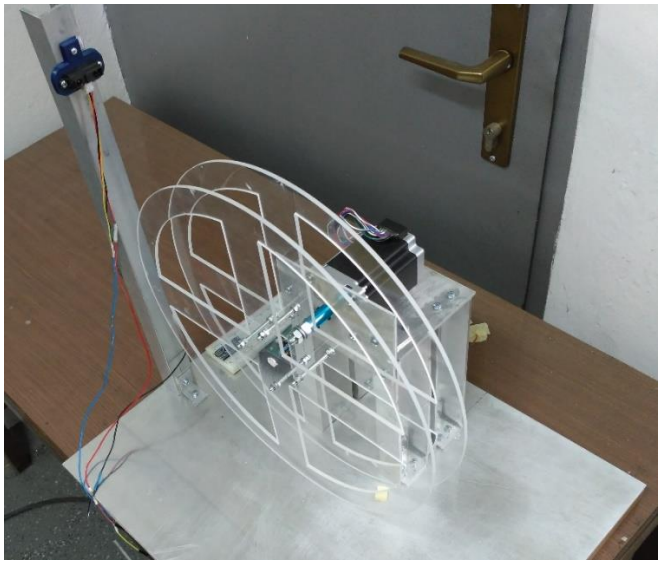
Keywords: mechatronics, Ball and wheel, lqr regulator, underactuated systems, teaching model

1. UVOD

Sustav kuglice na kotaču spada u inicijalno nestabilne i podupravljane sustave. Takvi sustavi prikladni su za ispitivanje raznih regulacijskih algoritama i pogodni su za edukaciju jer na upadljivoj nestabilnosti sustava prikazuju mogućnosti i svrhu regulacije.

Sustav se sastoji od kuglice, kotača, te upravljačkih i mjernih članova. Kotač je u ovoj izvedbi napravljen od dvije vodenim mlazom rezane kružne ploče pleksiglasa i po njegovom rubu se giba kuglica. Svrha ovog sustava je da se kuglica održava na vrhu kotača iako je njena stalna tendencija da pada uslijed djelovanja gravitacije. To se ostvaruje stalnim zakretanjem kotača u odgovarajućem smjeru te se zbog toga kuglica održava u ravnotežnom položaju. Rotacija kotača se ostvaruje koračnim motorom. Implementiran je direktni prijenos, tj. kotač i vratilo motora su spojeni spojkom koja se steže uvrtnim vijcima. Aktuator se pogoni sa upravljačkim sklopom koji signale sa mikrokontrolera pretvara u koračno gibanje motora i taj sklop je napajan sa posebnim napajanjem za koračni motor. U ravnini sa vrhom kotača postavljen je infracrveni senzor udaljenosti čija je svrha da mjeri poziciju kuglice na kotaču, tj. njenu udaljenost od senzora. Regulacijski kod se nalazi u mikrokontroleru koji se programira serijskom vezom preko računala. Radi boljeg predočavanja samog sustava, obično se izvodi njegov dinamički model, provede sinteza regulatora i izvrše simulacije. Tako je napravljeno i u ovom slučaju pa je tako izvedena dinamika sustava pomoću Euler-Lagrange-ovih jednadžbi. Dobivene su diferencijalne jednadžbe koje se prebacuju u prostor stanja. Prostor stanja je pogodan za odabrani LQR regulator. Primjenom LQR regulatora dobije se vektor pojačanja za svaku od varijabli stanja te se mogu onda provesti simulacije reguliranog sustava i procijeniti ispravnost odabranog regulatora i izračunatih pojačanja. Tada se može preći na eksperimentiranje na stvarnom modelu. To je obično iterativni postupak jer je stvarni sustav vrlo različit od izvedenog dinamičkog modela u kojem nisu uzete u obzir sve varijable i koji je većinom pojednostavljen. Tako je dobivanje ispravnih pojačanja regulatora uslijed kojih će doći do uspješnog upravljanja sustavom iterativni postupak koji može uključivati i ponovni izvod modela.

Izrađeni sustav je prikazan na Slika 1.1.



Slika 1.1. Izrađeni sustav kuglice na kotaču

2. MODELIRANJE SUSTAVA

2.1. EULER-LAGRANGEOVA METODA

Dinamika sustava Kuglice na kotaču modelirati će se pomoću Euler-Lagrangeove metode. Euler-Lagrangeova metoda je metoda za modeliranje dinamičkih sustava koja se temelji na energetskom principu, tj. pripadajuća Lagrangeova funkcija ima mjernu jedinicu energije. Lagrangeova mehanika ne unosi nikakve promjene u odnosu na Newtonovsku mehaniku, no dok u Newtonovskoj formulaciji moraju biti eksplicitno prikazane konzervativne sile i koja je prikladna za Kartezijeve koordinate, Lagrangeova mehanika se može koristiti u bilo kojem koordinatnom sustavu. Opće koordinate se mogu odabrati proizvoljno što pomaže ako je promatrani sustav simetričan ili geometrijski ograničen u nekom pogledu [1]. Lagrangeova formulacija je našla primjene i u kvantnoj mehanici i teoriji relativnosti te se također može primijeniti za modeliranje električnih krugova. Oblici energije u Lagrangeovoj mehanici su: kinetička, potencijalna i disipacijska. Sustavi se oblikuju rješavanjem Euler-Lagrangeove jednadžbe koja glasi:

$$\frac{d}{dt}\left(\frac{\partial K_j}{\partial \dot{q}_i}\right)-\left(\frac{\partial K_j}{\partial q_i}\right)+\frac{\partial P_j}{\partial q_i}=T_{i,j}\ , \tag{2.1.}$$

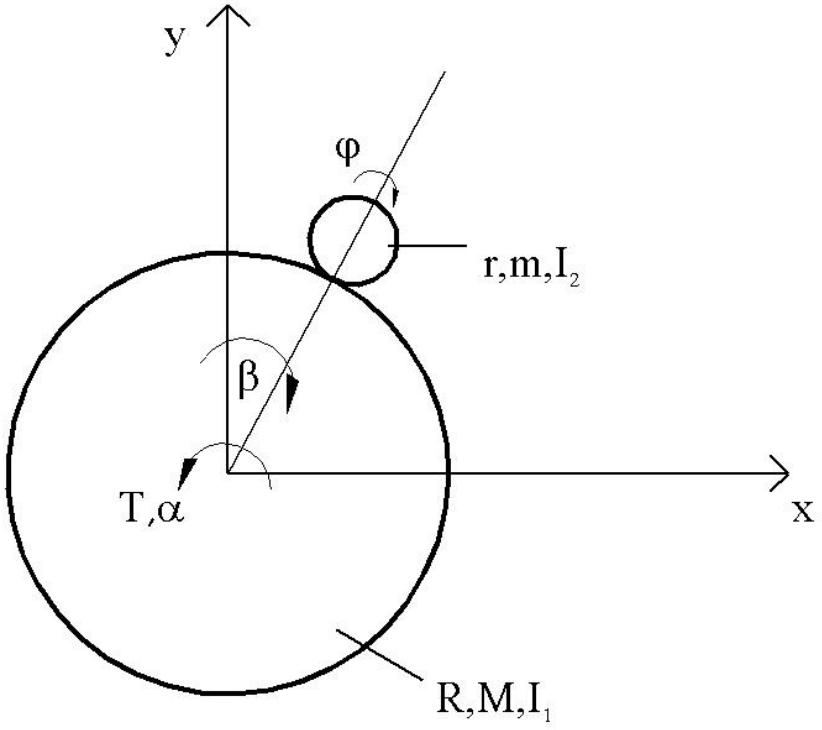
gdje je:

- K_j – kinetička energija j-te mase,
- P_j – potencijalna energija j-te mase,
- q_i – i-ta upravljana koordinata,
- $\dot{q}_i$ - brzina i-te upravljane koordinate,
- $T_{i,j}$ – sila ili moment i-te koordinate potreban za pokretanje j-te mase.

Lagrangeova mehanika je također prikladna za dinamiku čestica, polja se opisuju pomoću Lagrangeove gustoće. Lagrangeove jednadžbe se također koriste u optimizacijskim problemima dinamičkih sustava. Postoje Lagrangeove jednadžbe prve i druge vrste. Jednadžbe druge vrste se koriste dosta više nego one prve vrste [1].

2.2. PRIKAZ SUSTAVA

Na Slika 2.1. prikazana je skica sustava kuglice na kotaču za koji je potrebno izvesti dinamički model primjenom Euler-Lagrange-ove jednadžbe



Slika 2.1. Skica sustava kuglice na kotaču

Gdje je:

- R – radijus kotača,
- M – masa kotača,
- I_1 – moment inercije kotača,
- r – radijus kuglice,
- m – masa kuglice,
- I_2 – moment inercije kuglice,
- T – moment aktuatora,
- x,y – koordinatne osi,
- α – kut zakreta kotača,
- β – kut pomaka kuglice na kotaču,
- φ – kut zakreta kuglice.

2.3. PODSUSTAVI

Sustav kuglice na kotaču podijeliti će se u dva podsustava: podsustav kotača i podsustav kuglice. Za svaki podsustav će se postaviti odgovarajuće jednadžbe na temelju kojih će se doći do jednadžbi za kinetičku i potencijalnu energiju pojedinog podsustava.

2.3.1. PODSUSTAV KOTAČA

- Kinetička energija kotača:
  Kotač rotira kutnom brzinom $\dot{\alpha}$ i ima moment inercije I_1. Kinetička energija rotacije kotača tako glasi:

$$K_1 = \frac{1}{2} \cdot \dot{\alpha}^2 \cdot I_1 . \tag{2.2.}$$

- Potencijalna energija kotača:
  Centar mase kotača se nalazi u središtu referentnog koordinatnog sustava pa tako kotač u odnosu na njega nema potencijalnu energiju.

$$P_1 = 0 \ . \tag{2.3.}$$

2.3.2. PODSUSTAV KUGLICE

- Kinetička energija kuglice:
  Kuglica rotira kutnom brzinom $\dot{\varphi}$ i ima moment inercije I_2. Jednadžba za moment inercije kuglice glasi:

$$I_2 = \frac{2}{5} \cdot m \cdot r^2 \ . \tag{2.4.}$$

  Uz rotaciju, kuglica se također giba po rubu kotača pa tako brzina gibanja kuglice po rubu iznosi:

$$v_2 = (R+r) \cdot \dot{\beta} \ . \tag{2.5.}$$

  Tako izraz za kinetičku energiju kuglice glasi:

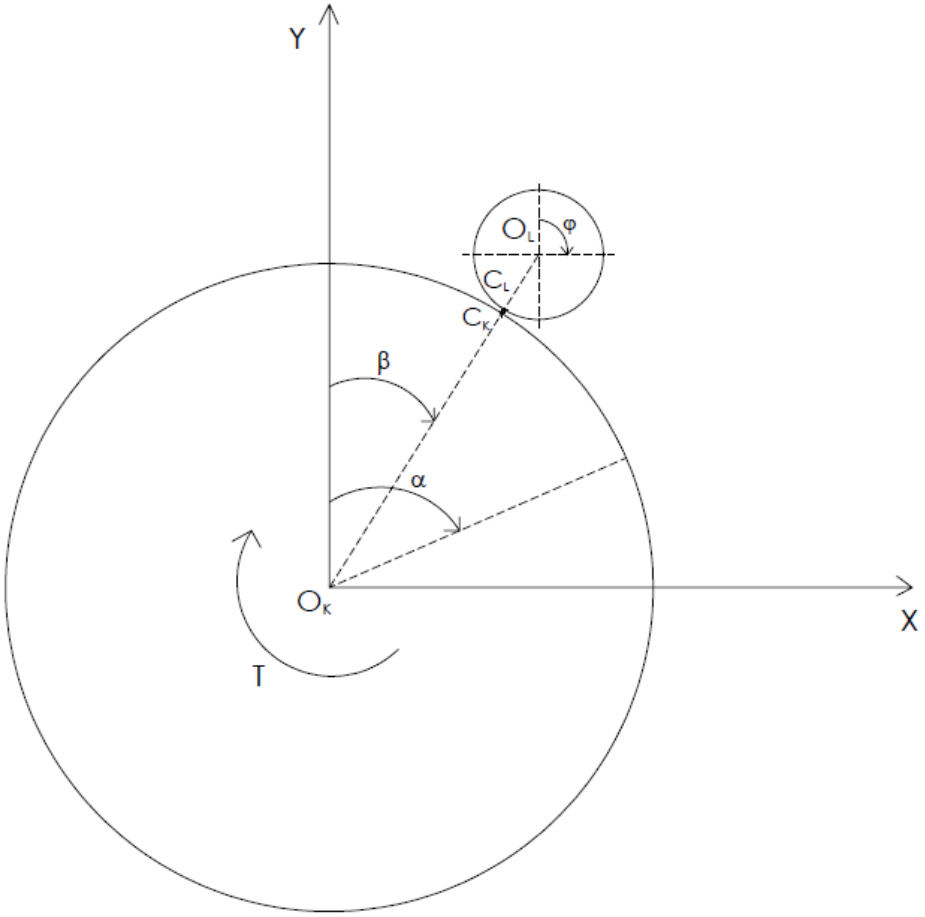
$$K_2 = \frac{1}{2} \cdot \dot{\varphi}^2 \cdot I_2 + \frac{1}{2} \cdot m \cdot v_2{}^2 \ . \tag{2.6.}$$

  Dalje nam treba izraz kojim ćemo opisati kut zakreta kuglice, $\dot{\varphi}$. Postupak dobivanja tog izraza komentirati će se po Slika 2.2. i izvesti prema [2]. Kako se kotač i kuglica rotiraju i kako se kuglica valja po rubu kotača, javlja se jedan ograničavajući uvjet. Neka je $v_{Ck/Ok}$ brzina točke kontakta C_k u odnosu na centar kotača O_k i koja je promatrana iz referentnog koordinatnog sustava po Slika 2.2. Tako slijedi da je:

$$v_{C_k/O_k} = \dot{\alpha} \cdot R \cdot \mathbf{e} , \tag{2.7.}$$

  gdje je **e** jedinični vektor koji je tangentan na kontaktnu točku i na točke u smjeru

povećanja $\dot{\alpha}$. Također neka je $v_{Cl/Ol}$ brzina točke kontakta C_l u odnosu na središte kuglice O_l i koja je promatrana iz referentnog koordinatnog sustava po Slika 2.2. Tada vrijedi:



Slika 2.2. Brzina rotacije kuglice

$$v_{C_l/O_l} = \dot{\varphi} \cdot r \cdot \mathbf{e} \ . \tag{2.8.}$$

Uz pretpostavku da nema klizanja u gibanju kuglice na kotaču, točka kontakta C_l uvijek ima relativnu brzinu jednaku nuli u odnosu na točku kontakta C_k. Tako je:

$$v_{C_l/C_k} = 0 \, . \tag{2.9.}$$

Brzina centra kuglice u odnosu na centar kotača dana je izrazom:

$$v_{O_l/O_k} = \dot{\beta} \cdot (R + r) \cdot \mathbf{e} \, . \tag{2.10.}$$

Iz gornjih izraza dolazi se do sljedeće jednadžbe:

$$\begin{aligned} v_{O_l/O_k} &= v_{O_l/C_l} + v_{C_l/C_k} + v_{C_k/O_k} \\ v_{O_l/O_k} &= -\dot{\varphi} \cdot r \cdot \mathbf{e} + \dot{\alpha} \cdot R \cdot \mathbf{e} \end{aligned} \ . \tag{2.11.}$$

Izjednačavanjem izraza (2.10.) i (2.11.) dolazi se do konačne jednadžbe za kutnu brzinu kuglice:

$$\dot{\varphi} = \frac{1}{r}\left(R \cdot \dot{\alpha} - \left(R + r\right) \cdot \dot{\beta}\right). \tag{2.12.}$$

Jednadžbe (2.5.) i (2.12.) uvrštavaju se u izraz (2.6.) te se tako dobiva kinetička energija kuglice:

$$K_2 = \frac{I_2}{2r^2}(R^2\dot{\alpha}^2 - 2Rr\dot{\alpha}\dot{\beta} - 2R^2\dot{\alpha}\dot{\beta} + \dot{\beta}^2(R+r)^2) + \frac{1}{2}m\dot{\beta}^2(R+r)^2. \tag{2.13.}$$

- Potencijalna energija kuglice:
  Prema Slika 2.1. potencijalna energija kuglice u odnosu na referentni koordinatni sustav glasi:

$$P_2 = mg(R+r)\cos\beta \ . \tag{2.14.}$$

2.4. JEDNADŽBE SUSTAVA

Ukupna jednadžba kinetičke energije sustava dobije se zbrajanjem izraza (2.2.) i (2.13.):

$$\begin{aligned} K &= K_1 + K_2 \\ K &= \frac{1}{2}\dot{\alpha}^2 I_1 + \frac{I_2}{2r^2}(R^2\dot{\alpha}^2 - 2Rr\dot{\alpha}\dot{\beta} - 2R^2\dot{\alpha}\dot{\beta} + \dot{\beta}^2(R+r)^2) + \frac{1}{2}m\dot{\beta}^2(R+r)^2 \ . \end{aligned} \tag{2.15.}$$

Ukupna jednadžba kinetičke energije jednaka je izrazu (2.14.) pošto je potencijalna energija kotača jednaka nuli:

$$P = P_2 = mg(R+r)\cos\beta \tag{2.16.}$$

2.5. PRIMJENA EULER-LAGRANGEOVE JEDNADŽBE

Da bi gore izračunate izraze za ukupnu kinetičku i potencijalnu energiju mogli primijeniti u Euler-Lagrangeovoj jednadžbi, trebamo odrediti upravljane varijable sustava. One glase:

$$\begin{aligned} q_1 &= \alpha \\ q_2 &= \beta \\ T_1 &= T \\ T_2 &= 0 \end{aligned} \quad . \tag{2.17.}$$

Gornja jednadžba znači da smo za upravljane koordinate sustava odabrali kut zakreta kotača i kut položaja kuglice na kotaču. Pošto kutom zakreta kotača možemo upravljati, tada je vrijednost varijable T_1 jednaka momentu kojeg primjenjujemo na kotač, dok položajem kuglice na kotaču ne upravljamo direktno pa je vrijednost pripadajuće varijable jednaka nuli.

2.5.1. KUT ZAKRETA KOTAČA - VARIJABLA α

Jednadžbe (2.15.) i (2.16.) uvrštavaju se u izraz (2.1.) prvo koristeći varijablu α.

$$\begin{aligned} \frac{\partial K}{\partial \dot{\alpha}} &= \dot{\alpha} I_1 + \frac{I_2}{r^2}\left(R^2\dot{\alpha} - Rr\dot{\beta} - R^2\dot{\beta}\right) \\ \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\alpha}}\right) &= \ddot{\alpha} I_1 + \frac{I_2}{r^2}\left(R^2\ddot{\alpha} - Rr\ddot{\beta} - R^2\ddot{\beta}\right) \\ \frac{\partial K}{\partial \alpha} &= 0 \\ \frac{\partial P}{\partial \alpha} &= 0 \end{aligned} \quad . \tag{2.18.}$$

Tako se dobiva prva glavna jednadžba sustava koja glasi:

$$\ddot{\alpha} I_1 + \frac{I_2}{r^2}\left(R^2\ddot{\alpha} - Rr\ddot{\beta} - R^2\ddot{\beta}\right) = T \, . \tag{2.19.}$$

2.5.2. KUT POMAKA KUGLICE NA KOTAČU - VARIJABLA β

Postupak za varijablu β analogan je gornjem postupku.

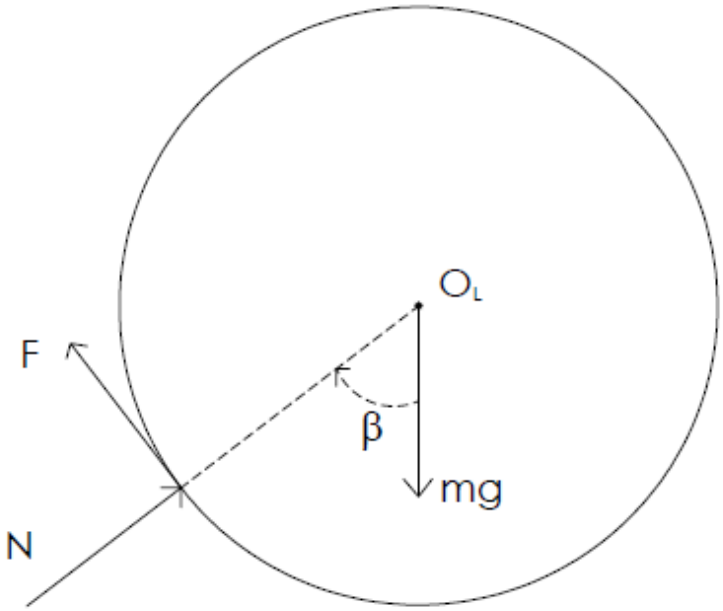
$$\begin{aligned} \frac{\partial K}{\partial \dot{\beta}} &= \frac{I_2}{2r^2}\left[-2Rr\dot{\alpha} - 2R^2\dot{\alpha} + 2\dot{\beta}(R+r)^2\right] + m\dot{\beta}(R+r)^2 \\ \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\beta}}\right) &= \frac{I_2}{r^2}\left[-Rr\ddot{\alpha} - R^2\ddot{\alpha} + \ddot{\beta}(R+r)^2\right] + m\ddot{\beta}(R+r)^2 \\ \frac{\partial K}{\partial \beta} &= 0 \\ \frac{\partial P}{\partial \beta} &= -mg(R+r)\sin\beta \end{aligned} \quad . \tag{2.20.}$$

Tako se dobiva druga glavna jednadžba sustava koja glasi:

$$\frac{I_2}{r^2}\left[-Rr\ddot{\alpha}-R^2\ddot{\alpha}+\ddot{\beta}(R+r)^2\right]+m\ddot{\beta}(R+r)^2-mg(R+r)\sin\beta=0\ . \tag{2.21.}$$

2.5.3. UVJET GIBANJA KUGLICE PO RUBU KOTAČA

Izrazi (2.19.) i (2.20.) vrijede jedino ako je centripetalna sila dovoljno velika da se održi kružno gibanje kuglice na kotaču, inače će kuglica odletjeti sa kotača [2]. Zadovoljenost uvjeta će se pobliže istražiti u nastavku koristeći Slika 2.3.



Slika 2.3. Uvjet gibanja kuglice

Jednadžba sila u radijalnom smjeru u odnosu na kuglicu dana je izrazom:

$$mg\cos\beta-N=m(\mathrm{R}+\mathrm{r})\dot{\beta}^2\ , \tag{2.22.}$$

gdje je N reakcijska sila između kuglice i kotača s obzirom na normalu. Sila F je sila trenja. Izraz na desnoj strani jednadžbe jednak je centripetalnoj sili koja djeluje na kuglicu. Kuglica i kotač gube kontakt kada je sila N jednaka nuli, pa se preformulacijom izraza (2.22.) dobiva sljedeći izraz koji mora biti zadovoljen tijekom rada sustava da bi loptica ostala na kotaču:

$$N=mg\cos\beta-m(R+r)\dot{\beta}^2>0\ . \tag{2.23.}$$

2.6. LINEARIZACIJA SUSTAVA

Ovako modeliran sustav potrebno je linearizirati. Linearizacija se provodi tako što će se pretpostaviti da će kut β biti toliko mali da vrijedi:

$$\sin\beta \cong \beta \ , \tag{2.24.}$$

čime se jednadžba (2.21.) pretvara u:

$$\frac{I_2}{r^2}\left[-Rr\ddot{\alpha} - R^2\ddot{\alpha} + \ddot{\beta}(R+r)^2\right] + m\ddot{\beta}(R+r)^2 - mg(R+r)\beta = 0\,. \tag{2.25.}$$

2.7. SREĐIVANJE JEDNADŽBI

Izraze (2.19.) i (2.25.) treba srediti tako da se grupiraju oko drugih derivacija upravljanih varijabli, tj. oko $\ddot{\alpha}$ i $\ddot{\beta}$. Da bi se pojednostavilo popisivanje, uvode se sljedeće zamjene za karakteristične dijelove jednadžbi:

$$\begin{aligned} A_1 &= I_2\left[\frac{R}{r} + \left(\frac{R}{r}\right)^2\right] \\ B_1 &= I_2\frac{(R+r)^2}{r^2} \\ C_1 &= R + r \\ D_1 &= I_1 + I_2\left(\frac{R}{r}\right)^2 \end{aligned} \quad . \tag{2.26.}$$

Tako (2.19.) prelazi u:

$$\ddot{\alpha} = \frac{T + \ddot{\beta}\cdot A_1}{D_1} \ . \tag{2.27.}$$

Iz dinamike sustava dobivamo dodatnu jednadžbu gibanja; kutna akceleracija kotača proporcionalna je dovedenom momentu i obrnuto proporcionalna momentu inercije:

$$\ddot{\alpha} = \frac{T}{I_1} \ . \tag{2.28.}$$

Sa svime prije navedenim, (2.25.) prelazi u:

$$\ddot{\beta} = \frac{\frac{T}{I_1}\cdot A_1 + mg\beta\cdot C_1}{B_1 + mC_1^{\,2}} \ . \tag{2.29.}$$

3. SIMULACIJA SUSTAVA

3.1. PROSTOR STANJA

Prostor stanja je matematički model fizikalnog sustava prikazanog kao skup ulaza, izlaza i varijabli stanja povezanih diferencijalnim jednadžbama prvog reda [3]. Ako je sustav linearan i vremenski neovisan, može se prikazati u matričnom obliku. Opći prikaz prostora stanja sa *p* ulaza, *q* izlaza i *n* varijabli stanja prikazuje se kao:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}x + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}x + \mathbf{D}u \end{aligned}, \tag{3.1.}$$

gdje je:

- **x** – vektor varijabli stanja,
- $\dot{\boldsymbol{x}}$ **–** vektor derivacija varijabli stanja,
- **u** – vektor ulaza,
- **y** – vektor izlaza,
- **A** – matrica koeficijenata sustava, dim **A** = *n* x *n*,
- **B** – matrica ulaza, dim **B** = *n* x *p*,
- **C** – matrica izlaza, dim **C** = *q* x *n*,
- **D** – matrica direktnog preslikavanja ulaza, dim **D** = *q* x *p*.

3.2. PREVOĐENJE SUSTAVA U PROSTOR STANJA

Kao varijable stanja odabrane su $\dot{\alpha}, \beta, \dot{\beta}$. Tako model preveden u prostor stanja ima oblik:

$$\begin{bmatrix} \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \mathbf{B} \cdot u$$

$$y = \mathbf{C} \cdot \begin{bmatrix} \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \mathbf{D} \cdot u \tag{3.2.}$$

gdje su matrice **A, B, C** i **D**, dobivene iz jednadžbi (2.27.) i (2.29.):

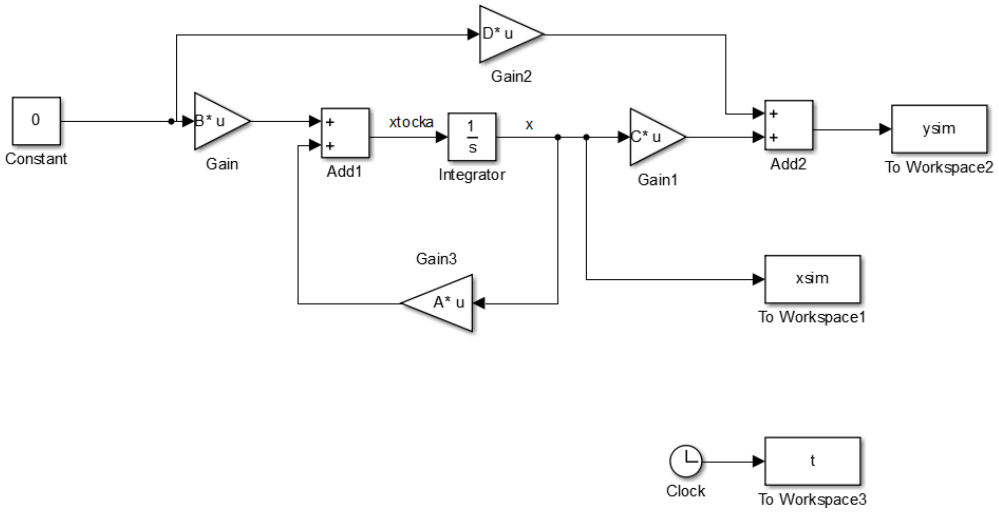
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \dfrac{mgC_1}{B_1 + mC_1^2} & 0 \end{bmatrix}, \tag{3.3.}$$

$$\mathbf{B} = \begin{bmatrix} \dfrac{1}{I_1} \\ 0 \\ \dfrac{A_1}{I_1\left(B_1 + mC_1^{\,2}\right)} \end{bmatrix}, \tag{3.4.}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.5.}$$

3.3. BLOK DIJAGRAM SUSTAVA

Na Slika 3.1. prikazan je blok dijagram sustava u programskom paketu Simulink. Sustav će se prvo pobuditi referencom 0, tj. pustiti će se prirodno ponašanje sustava s obzirom na neke početne uvjete te će se prikazati i komentirati odziv sustava.



Slika 3.1. Blok dijagram sustava

3.3.1. PARAMETRI SUSTAVA

Zadani su sljedeći parametri sustava. Njihove vrijednosti su stvarne vrijednosti, po kasnije izgrađenom realnom sustavu i referiraju se po Slika 2.1.

r = 0,03 m, R = 0,175 m, m = 0,15 kg, M = 0,6 kg,

$I_2 = 5{,}766*10^{-5}$ kg/m^2, I_1 = 0,0103 kg/m^2, g = 9.81 m/s^2.

3.3.2. POČETNI UVJETI SUSTAVA

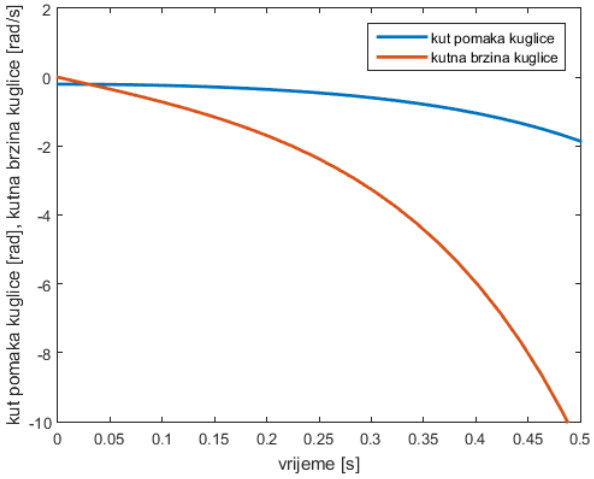
Sustav diferencijalnih jednadžbi mora imati neke početne uvjete, pa se u ovom slučaju ti početni uvjeti uvrštavaju u blok naziva *Integrator* na Slika 3.1. Početni uvjeti su u obliku vektora koji je veličine broja varijabla stanja. Početni uvjeti mogu se mijenjati proizvoljno da bi se promotrili razni odzivi sustava sa ili bez regulatora ovisno o njima. Glase:

$$\mathbf{X_0} = \begin{bmatrix} 0 & -0,2 & 0 \end{bmatrix}, \qquad (3.6.)$$

i znače redom:

- kotač se ne rotira,
- kuglica se nalazi na 0,2 rad u negativnom smjeru rotacije kotača,
- kuglica se ne giba.

Ovi početni uvjeti efektivno znače da se kuglica stavila rukom na početnu poziciju dok je kotač bio u mirovanju. Zbog djelovanja gravitacije, za očekivati je da će se kuglica nastaviti gibati u negativnom smjeru rotacije kotača i da će eventualno pasti sa kotača pošto se nikakvim regulatorom ne djeluje na sustav. Odziv sustava prikazan je na Slika 3.2.



Slika 3.2. Odziv sustava

Kao što se vidi na Slika 3.2. kuglica je nastavila padati u negativnom smjeru rotacije kotača i eventualno će pasti s njega jer se kotač nije zarotirao uslijed ne postojanja regulatora.

4. PROJEKTIRANJE REGULATORA

4.1. ODABIR REGULATORA

Pošto uobičajeni PID regulator ne bi mogao dobro odgovoriti na dinamiku ovog sustava zbog njegove inicijalne podupravljanosti i nestabilnosti, odabire se takozvani LQR regulator (*eng. Linear quadratic regulator*), tj. regulator po varijablama stanja koji je prikladan za ovaj problem.

4.2. REGULATOR PO VARIJABLAMA STANJA

Jedna od najčešće primjenjivanih tehnika za projektiranje linearnih regulacijskih sustava je metoda optimalnog upravljanja koja koristi povratne veze po varijablama stanja [4]. Ova metoda sastoji se u tome da je za sustav opisan u prostoru stanja potrebno uvesti povratnu vezu po varijablama stanja, tj. naći takav vektor upravljanja $\boldsymbol{u} = -\boldsymbol{K} * \boldsymbol{x}$, koji minimizira funkcional (indeks preformanse):

$$\mathbf{I} = \int_0^\infty \left(\mathbf{x}^\mathbf{T} \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{u}^\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{u} \right) dt \ . \tag{4.1.}$$

Matrica (vektor) pojačanja **K** određuje se iz izraza:

$$\mathbf{K} = \mathbf{R}^{-1} \cdot \mathbf{B}^\mathbf{T} \cdot \mathbf{H} \ , \tag{4.2.}$$

gdje je **H** pozitivno određeno rješenje algebarske matrične Ricattijeve jednadžbe:

$$\mathbf{H} \cdot \mathbf{A} + \mathbf{A}^\mathbf{T} \cdot \mathbf{H}^\mathbf{T} + \mathbf{Q} - \mathbf{H} \cdot \mathbf{B} \cdot \mathbf{R}^{-1} \cdot \mathbf{B}^\mathbf{T} \cdot \mathbf{H} = 0 \ . \tag{4.3.}$$

Optimalno upravljanje primijenjeno na sustav opisan u prostoru stanja dovodi do modela optimalnog sustava opisanog jednadžbom stanja u obliku:

$$\dot{\mathbf{x}} = \left[\mathbf{A} - \mathbf{B} \cdot \mathbf{K} \right] \cdot \mathbf{x} = \mathbf{A}_\mathbf{C} \cdot \mathbf{K} \ . \tag{4.4.}$$

Matrice **Q** i **R** su težinske matrice, pri čemu se matricom **Q** utječe na prijelazni proces vektora stanja, a preko matrice **R** na iznos energije upravljanja, pa se izborom elemenata matrica **Q** i **R** može utjecati na dinamičko ponašanje sustava u uvjetima zatvorene regulacijske petlje. Ukoliko se matrice **Q** i **R** izaberu kao dijagonalne matrice, dobiva se tzv. LQP (*eng. linear quadratic problem*), pa analogno tomu korišteni regulator naziva se LQR (*eng. linear quadratic regulator*). Za dijagonalno izabrane težinske matrice lakše je pratiti utjecaj pojedinih stanja i ulaza, jer svako stanje i ulaz ima svoj težinski element u matrici.

U procesu projektiranja optimalnog regulatora osobito važan korak je određivanje početnih vrijednosti težinskih matrica **Q** i **R**. Ukoliko se matrica **Q** izabere kao jedinična matrica onda je svakoj varijabli stanja dana ista težina odnosno isti utjecaj na dinamičko ponašanje sustava u uvjetima zatvorenog regulacijskog kruga. Veći iznos pojačanja u vektoru pojačanja **K** imat će one varijable stanja na čiju je promjenu sustav u procesu regulacije manje osjetljiv [4]. Dakle, iznosi pojačanja i osjetljivost sustava na varijable stanja odnose se obrnuto proporcionalno.

4.3. ODABIR TEŽINSKIH MATRICA Q I R

Elementi matrice **Q** su kvadrati recipročnih vrijednosti maksimalno dozvoljenih odstupanja varijabli ulaza i izlaza. Tako prvo treba definirati maksimalno dozvoljena odstupanja varijabli u slučaju da je kao ulazna referenca zadana vrijednost nule:

- $e_1 = 4 \cdot \pi$ - znači da je maksimalna dozvoljena brzina kotača 2 o/s,
- $e_2 = \pi / 12$ - znači da je maksimalni dozvoljen pomak kuglice po rubu kotača 15 stupnjeva,
- $e_3 = \pi / 6$ - znači da je maksimalna dozvoljena kutna brzina kuglice 30 stupnjeva/s

Pomoću gornjih vrijednosti dalje se računaju elementi matrice **Q**:

$$q_1 = \frac{1}{e_1^2}, \quad q_2 = \frac{1}{e_2^2}, \quad q_3 = \frac{1}{e_3^2} \ . \tag{4.5.}$$

Tako matrica **Q** glasi:

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix}, \tag{4.6.}$$

a matrica **R**:

$$\mathbf{R} = \begin{bmatrix} 1 \end{bmatrix} . \tag{4.7.}$$

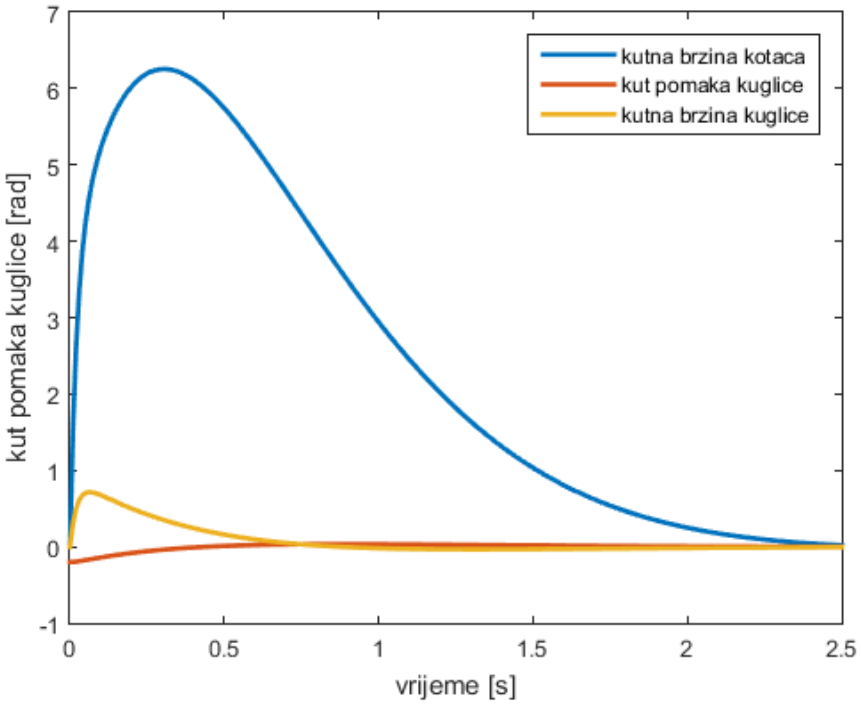
4.4. POJAČANJE REGULATORA

Vektor pojačanja **K** dobiva se pomoću MATLAB funkcije *lqr()*. Za gore definirane parametre sustava i za odabrane težinske matrice **Q** i **R** dobiva se sljedeći vektor pojačanja **K**:

$$\mathbf{K} = \begin{bmatrix} -0,0796 & 9,9324 & 2,4736 \end{bmatrix} . \tag{4.8.}$$

4.5. SIMULACIJA I ODZIV SUSTAVA

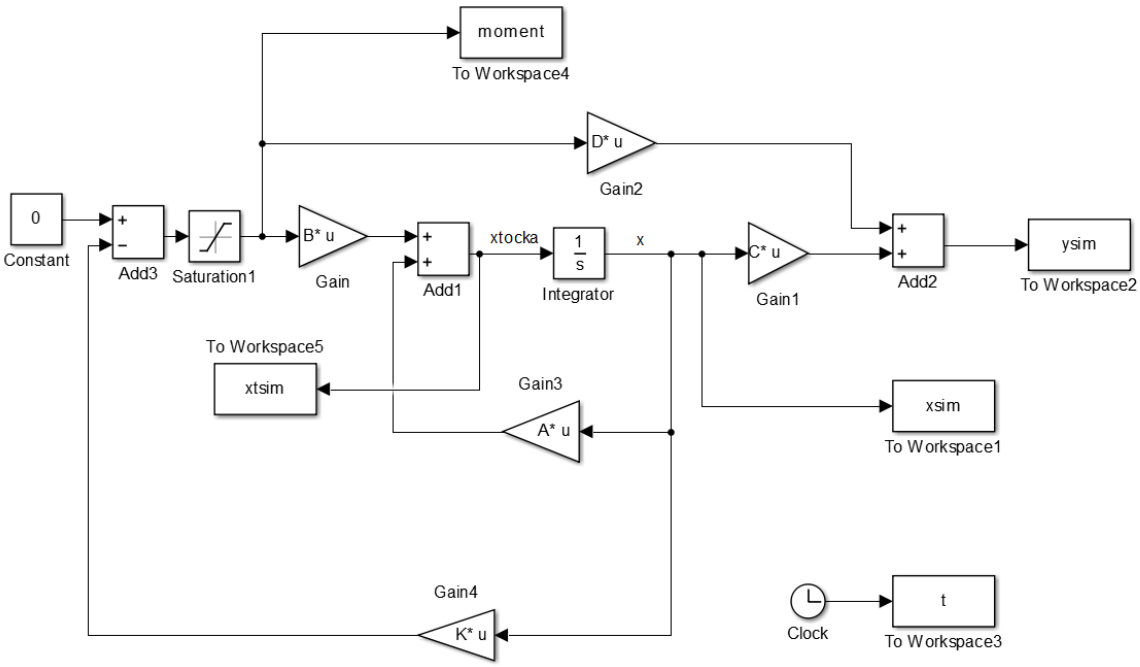
Uvrštavanjem izraza (4.8.) u izraz (4.4.) dobiva se nova matrica $\mathbf{A}_\mathbf{C}$. Tada se koristi već postojeća funkcija *lsim()* za simulaciju zatvorenog sustava. Odzivi kutne brzine kotača i kutnog pomaka loptice prikazani su na Slika 4.1. Uz drugačiji vektor pojačanja **K**, dobiti će se drugačiji odazivi, no svima će biti jedna stvar zajednička; sve varijable stanja će se smiriti u nuli, dok će se mijenjati trajanje smirivanja i amplituda oscilacija.



Slika 4.1. Odziv sustava uz LQR regulator

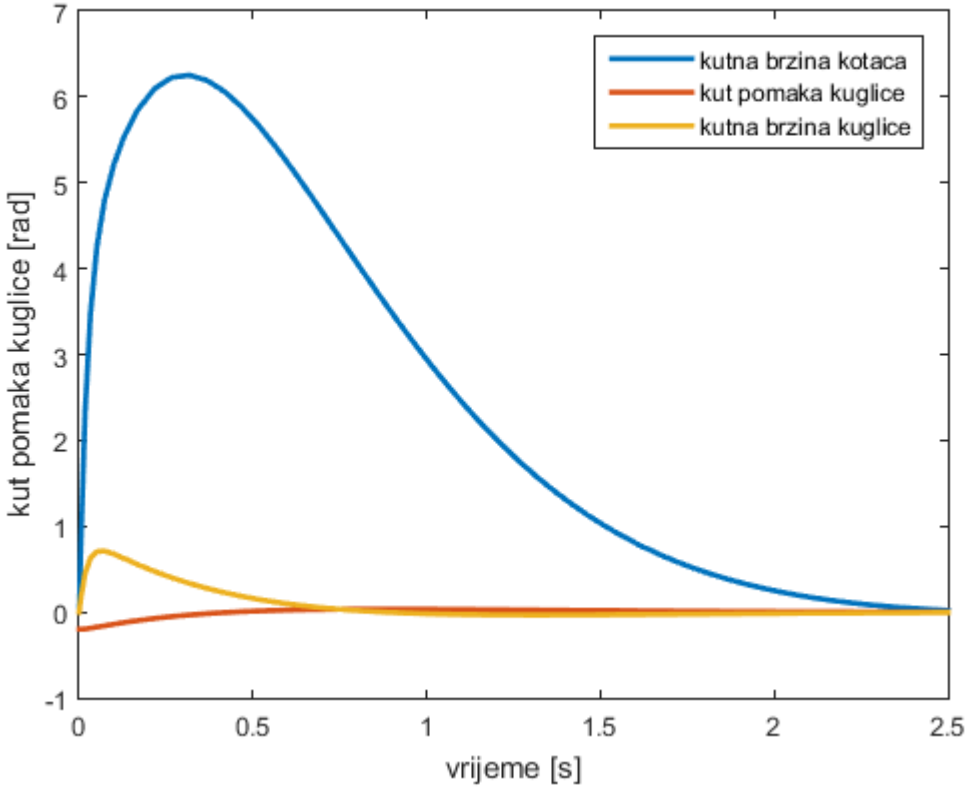
4.6. LQR REGULATOR U SIMULINK MODELU

Na Slika 4.2. prikazana je izvedba LQR regulatora u Simulink-u.



Slika 4.2. Blok dijagram sustava s regulatorom po varijablama stanja

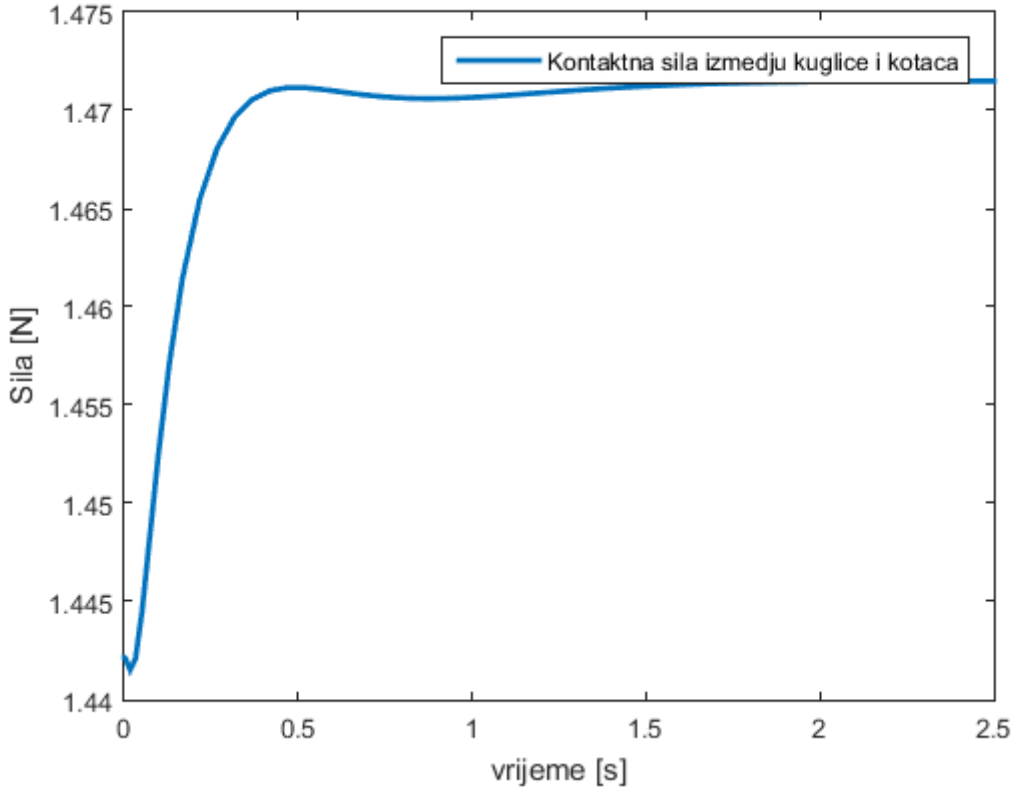
Kao što se vidi na Slika 4.2., dodana je povratna veza po varijablama stanja sa pojačanjem koje sadrži vektor pojačanja **K**. Tako se izvodi LQR regulator u blok dijagramu. Također se promatra izračunati potrebni moment za regulaciju sustava, a blok naziva *Saturation* koji postavlja gornje i donje limite ulazne vrijednosti biti će koristan kasnije kod izračuna za komponente realnog sustava. Odziv sustava po blok dijagramu prikazan je na Slika 4.3.



Slika 4.3. Odziv reguliranog sustava u Simulink-u

Uspoređujući Slika 4.1. i Slika 4.3. vidi se da su odzivi isti što znači da je sustav dobro izveden u Simulink-u. Također sustav u oba slučaja postiže referentnu vrijednost varijabli stanja koja iznosi nula što je bio i zadatak LQR regulatora.

Dodatno se provjerava uvjet gibanja kuglice po kotaču definiran u odjeljku *2.5.3.* Rezultati simulacije uvršteni su u izraz (2.23.) te je iscrtan sljedeći graf na Slika 4.4. iz kojega se vidi da je uvjet gibanja kuglice po kotaču zadovoljen, jer je sila *N* cijelo vrijeme veća od nule.



Slika 4.4. Kontaktna sila između kuglice i kotača

5. IZBOR KOMPONENTI REALNOG SUSTAVA

Da bi se sustav izveo u stvarnosti, trebaju se definirati potrebne komponente upravljačkog i mjernog sustava za realnu izvedbu. Za uspješan rad realnog sustava potrebna je točna identifikacija njegovih realnih parametara, sigurno upravljanje sustavom te točno mjerenje prije definiranih varijabli stanja. Komponenta upravljačkog sustava je naravno aktuator i njegov pripadajući upravljački sklop (*driver*) koji će upravljati rotacijom kotača. Komponente mjernog sustava su senzori koji moraju točno mjeriti varijable stanja. Obrada vrijednosti mjerenja i samo upravljanje aktuatorom se mora odvijati u zasebnoj upravljačkoj jedinici, tj. u mikrokontroleru. Odabir, proračun i obrazloženje odabranih komponenti slijedi u nastavku.

5.1. IZBOR AKTUATORA

Kako je ulazna varijabla simuliranog sustava moment, potrebno je odabrati aktuator koji će moći minimalno dati moment dobiven gornjom simulacijom. Najraširenij aktuatori su električni motori te se upravo oni razmatraju za izvedbu realnog sustava. Pošto postoji više vrsta električnih motora, za izvedbu ovog sustava u obzir dolaze oni napajani istosmjernim naponom jer je za izvedbu sustava dostupno istosmjerno napajanje promjenjivog napona i struje. Tako u izbor ulaze obični DC motori sa reduktorom na izlaznom vratilu za povećanje momenta i koračni motori sa većim nazivnim momentom. Pošto je cijena koračnih motora sa potrebnim momentom na temelju simulacija u vrijeme nabave bila manja od cijene DC motora sa reduktorom, nabavljen je koračni motor dovoljnog nazivnog momenta. Također je za DC motor otežavajući faktor bila obavezna kupnja enkodera s kojim bi se mjerio kut zakreta vratila, dok se za koračni motor može u kodu brojati broj koraka u oba smjera rotacije i time je određen točan kut zakreta i kutna brzina vratila.

5.2. KORAČNI MOTOR

Koračni motor je tip DC motora bez četkica koji se ne rotira kontinuirano već mu je rotacija podijeljena u određeni broj jednakih koraka. Tako se motor može ciljano rotirati za određeni kut, tj. za određeni broj koraka i držati tu poziciju. Podrazumijeva se naravno da je motor prilagođen operaciji koju izvodi u smislu zahtijevane brzine i momenta, u slučaju da nije doći će do proklizavanja motora, tj. gubljenja koraka i kut zakreta motora se više ne može točno izračunati [5].

Za pokretanje i rotaciju koračnog motora potreban je posebni upravljački sklop koji će obično pravokutne signale poslane sa kontrolera pretvarati u točno definiran redoslijed magnetiziranja određenih zavojnica koračnog motora. Tako se motor sa svakim signalom rotira za točno određeni kut koji je definiran samom geometrijom motora, tj. razmještajem i brojem polova na statoru i rotoru. Većina upravljačkih sklopova također ima opciju tzv . mikrokoraka (*eng. microstepping*) u kojem se motor više pri svakom signalu ne pomiče za jedan cijeli korak, nego za njegov dio ovisno o postavkama na upravljačkom sklopu.

Ovisno o vrsti podjele, postoje više vrsta koračnih motora. Po razlici u vrsti magneta, motori se dijele na motore sa permanentnim magnetima, motore sa varijabilnom reluktancijom i hibridne koračne motore koji su danas najviše u uporabi [5]. Po načinu namatanja zavojnica motori mogu biti unipolarni i bipolarni.

5.3. PRORAČUN AKTUATORA

U simulacijama iz prethodnog poglavlja, ulaz u sustav bio je moment koji je djelovao na kotač. Taj moment je u stvarnosti moment koji proizvodi motor koji je nekom vrstom spoja spojen na kotač. Ako se uzme pretpostavka da je spoj direktan, da nema nikakvih prijenosnika između vratila i motora u smislu reduktora ili remenskog prijenosa, tada postoji direktan prijenos između vratila motora i kotača i sav moment koji proizvede motor se prenosi na kotač.

Iz simulacija se može odrediti potreban moment koji će gore definirani sustav sa definiranim parametrima ustabiliti, tj. biti će u mogućnosti rotirati kotač sa dovoljnom kutnom brzinom i akceleracijom u oba smjera kako bi kuglica ostala na vrhu kotača. Taj moment se kroz simulaciju mijenja ovisno o položaju loptice na kotaču, no njegova maksimalna vrijednost je iznosa oko 2,77 Nm. Znači da je potrebno odabrati motor koji će imati minimalni nazivni moment tog iznosa, no poželjno je da postoji i neki faktor sigurnosti. Kako je odlučeno u prethodnom odjeljku, nabavlja se koračni motor sa dovoljnim nazivnim momentom.

Izabran je koračni motor tvrtke Nanotec kodnog broja ST6018L3008. On je dvofazni hibridni koračni motor, ima 8 izvoda zavojnica i konfiguracija zavojnica mu je takva da se može spajati bipolarno ili unipolarno te se navedenih 8 izvoda može spojiti u 4 izvoda serijski ili paralelno ovisno o namjeni. Neke njegove osnovne karakteristike dane su u nastavku, dok je *datasheet* dostupan u literaturi [6]. Odlučeno je da će se izvodi zavojnica spojiti paralelno čime se dobiva više struje u namotima, a time i veći moment. Paralelnim spojem se smanjuje otpor i induktivitet namota čime je njegova vremenska konstanta dosta manja i time se povećava moment motora pri većim brzinama jer se brže dođe do nazivne struje. Motor je prikazan na Slika 5.1. a u nastavku slijede osnovne karakteristike pri paralelnom spoju.

Slika 5.1. Koračni motor ST6018L3008 [7]

- Nazivni napon – 3,9 V,
- nazivna struja po fazi – 4,24 A,
- otpor po fazi – 0,65 Ω,
- induktivitet po fazi – 3,2 mH,
- statički moment – 3,54 Nm,
- korak – 1,8º,
- moment inercije rotora – 8,4 * 10^{-5} kg/m^2.

Pošto je moment inercije rotora za četiri reda veličine manji od momenta inercije kotača definiranog u odjeljku 3.3.1., njegovo sumiranje momentu inercije kotača u simulaciji neće dovesti do značajnog povećanja potrebnog okretnog momenta, što se i provjerilo. Viskozno trenje se zanemaruje.

5.4. ODABIR UPRAVLJAČKOG SKLOPA

S obzirom na odabrani aktuator treba se odabrati prikladni upravljački sklop. Odabrani koračni motor kao što je već napomenuto ima osam izvoda te korisnik može odlučiti kako će spojiti izvode, unipolarno ili bipolarno. Odabire se bipolarni spoj te se dalje treba odlučiti hoće li se izvodi spojiti serijski ili paralelno. Odabire se paralelni spoj zbog razloga navedenih u prijašnjem odjeljku. Sada treba naći upravljački sklop (u nastavku *driver*) koji će moći bipolarno upravljati odabranim koračnim motorom s obzirom na nazivnu vrijednost struje koračnog motora. No može se postaviti pitanje zašto se uopće koriste *driveri* kada bismo potrebne signale za pokretanje i rotaciju koračnog motora mogli jednostavno slati pomoću mikrokontrolera i određenog broja tranzistora. Razlog je jednostavan, *driver* je sigurniji za mikrokontroler, za njegovo programiranje potreban je samo jedan signal sa mikrokontrolera, logičko i motorno napajanje je u *driveru* većinom odvojeno optospojkama, *driver* omogućuje tako zvani *microstepping* pomoću kojeg se nazivni korak koračnog motora određenim konfiguracijama zavojničkih struja može podijeliti u broj manjih koraka čime se postiže veća finoća pokreta uz istu točnost, što je vrlo korisno za pozicijske sustave tipa CNC-a. *Driveri* se u radu mogu značajno zagrijati te je pri odabiru bitno ocijeniti njegovo hlađenje, tj. većinom veličinu pričvršćenog hladnjaka i prisustvo termopaste za što bolje odvođenje topline.

Odabrana je pločica naziva THB6064AH JP-6445A koja je napravljena oko glavnog *microchip-a* THB6064 proizvođača Toshibe, koji ima sljedeće karakteristike [8]:

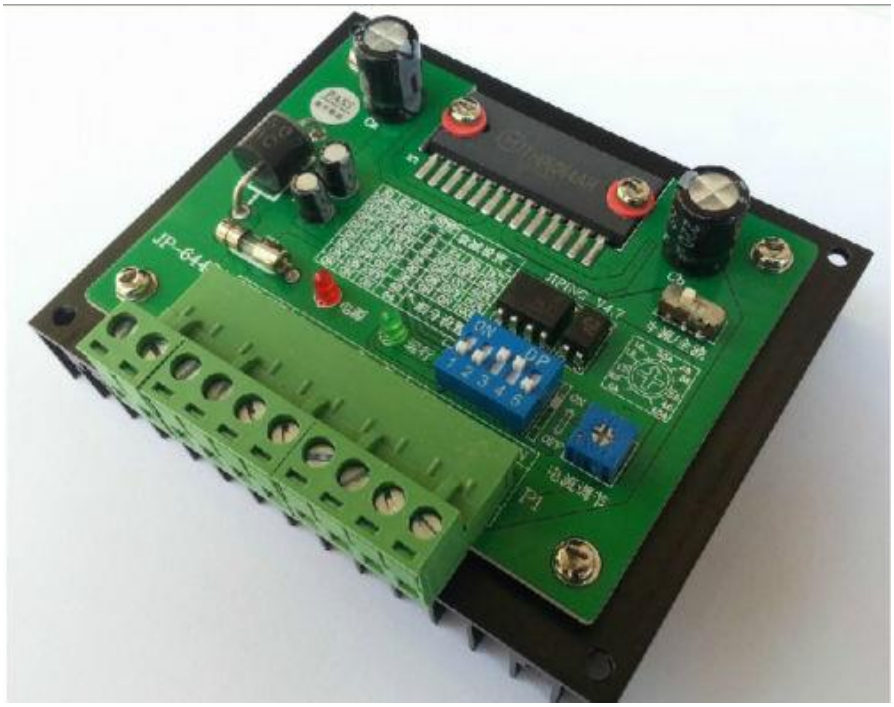
- nazivna struja do 4,5 A,
- mogući 2, 8, 10, 16, 20, 40, 64 *microstepping*;
- moguće limitiranje struje od 0 do 4,5 A,
- ulazni napon od 20-45 V.

Pločica ima deset konektora:

- dva su ulaza rezervirana za napajanje motora, plus i minus,
- četiri izlaza se spajaju na četiri izvoda koračnog motora i služe kao upravljački signali,
- logičko napajanje iz povezanog mikrokontrolera,
- ulaz koji definira smjer rotacije motora. Smjer ovisi o digitalnom stanju ulaza,

- ulaz koji prima signal za rotaciju,
- ulaz naziva *enable* koji ako se uzemlji trenutno zaustavlja rad *driver-a*.

Na Slika 5.2. prikazan je odabrani upravljački sklop.



Slika 5.2. Upravljački sklop THB6064AH JP-6445A [9]

5.5. ODABIR SENZORA U SUSTAVU

Da bi se sustav mogao uspješno realizirati u stvarnosti u smislu da kuglica ostane na vrhu kotača, moraju se regulirati prije navedene varijable stanja. Regulacija se odvija pomoću povratne veze koja sadrži trenutne vrijednosti varijabli stanja. U realnosti te vrijednosti se dobivaju pomoću raznih senzora koji mjere trenutne vrijednosti tih varijabli i koje se koriste u regulacijskom algoritmu.

Mjerenje dvije od tri varijable stanja će se odvijati u samom kodu. Kutna brzina kotača, $\dot{\alpha}$, mjeriti će se pomoću brojanja prijeđenih koraka koračnog motora tijekom rotacije. Pošto je za samu rotaciju koračnog motora potrebno imati dio koda koji se izvršava u točno definiranim vremenskim razdobljima, u taj dio je moguće ubaciti i liniju koja će inkrementirati broj koraka prilikom svakog izvršavanja tog dijela koda. Tako se tijekom izvršavanja dijela koda koji provodi mjerenje oduzimaju vrijednosti trenutnog i broja koraka u prošlom mjerenju. Ta vrijednost se dijeli sa nazivnim brojem koraka koračnog motora i množi sa 360 da bi se dobio prijeđeni kut koji je kotač napravio između dva mjerenja. Taj prijeđeni kut se tada pretvara u radijane te se dijeli sa proteklim vremenom između dva mjerenja čime se dobiva kutna brzina kotača u mjernoj jedinici rad/s. U prijašnjem tekstu prešutno se podrazumijeva da je kotač kruto vezan za izlazno vratilo motora i da ne postoji klizanje između vratila i kotača. Također se

podrazumijeva da motor ima dovoljni moment kroz cijelu izvedbu te da ne preskače korake uslijed prevelikog opterećenja.

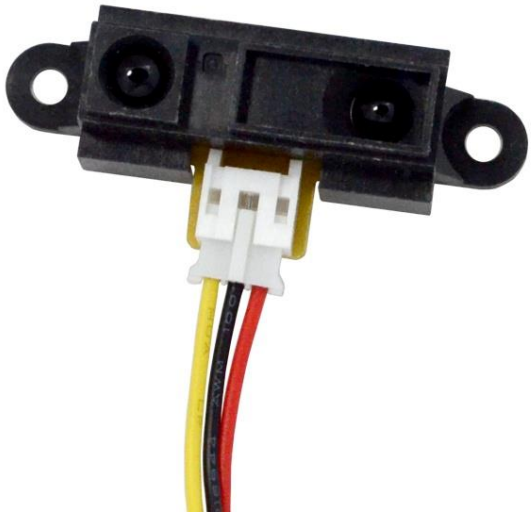
Mjerenje kutne brzine kuglice, $\dot{\beta}$, također će se odvijati u kodu. Odvijati će se na taj način da će se oduzeti trenutna i vrijednost kuta kuglice na kotaču u prošlom mjerenju. Ta vrijednost će se onda također podijeliti vremenom koje je proteklo između dva mjerenja te će se dobiti vrijednost kutne brzine kuglice po kotaču u rad/s.

5.5.1. ODABIR SENZORA UDALJENOSTI

Mjerenje kuta kuglice na kotaču, β, provesti će se korištenjem senzora udaljenosti. Senzori udaljenosti, ili senzori blizine, su uređaji koji su u mogućnosti detektirati prisustvo nekog predmeta u njihovoj blizini, a ponekad i izmjeriti ili procijeniti udaljenost od senzora do tog objekta bez ikakvog kontakta [10]. Postoje različiti načini rada senzora udaljenosti. Obično se odašilje elektromagnetsko polje ili zraka elektromagnetskog zračenja (npr. infracrvena zraka) i onda se promatra promjena u polju ili u povratnom signalu. Senzori udaljenosti uvijek imaju vrijednost do koje mogu detektirati predmete oko sebe te se ta vrijednost obično naziva „opseg detekcije“ senzora. Oni obično imaju dugi vijek trajanja i visoku pouzdanost zbog nepostojanja mehaničkih gibajućih dijelova i odsustva kontakta između senzora i predmeta detekcije. Najčešće vrste senzora udaljenosti su [10]:

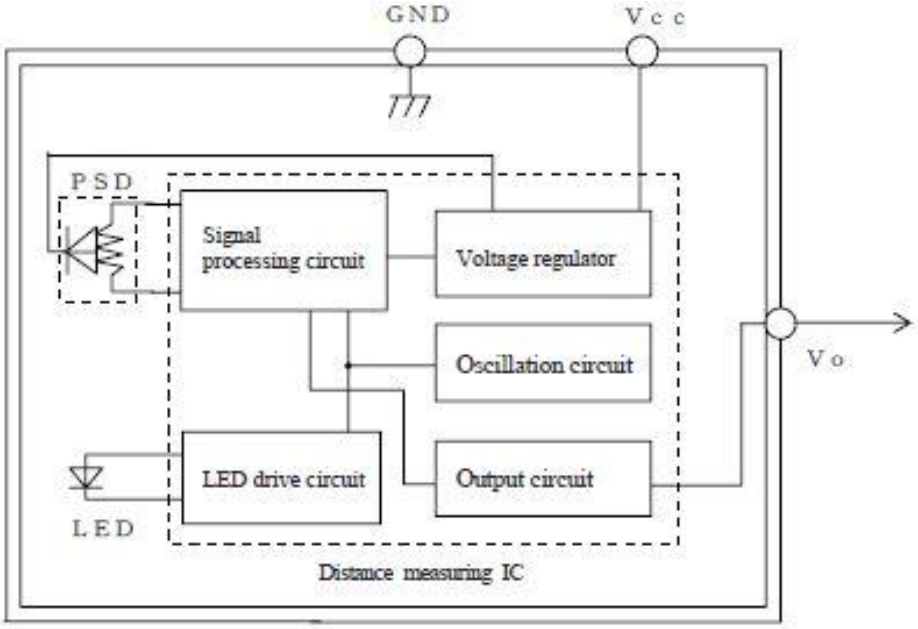
- kapacitativni,
- induktivni,
- magnetski,
- optički,
- radarski ili
- sonarski.

Za izvedbu realnog sustava odabran je infracrveni senzor proizvođača Sharp i naziva GP2Y0A21YK0F i prikazan je na Slika 5.3. Senzor radi na logičkom naponu.



Slika 5.3. Infracrveni senzor udaljenost GP2Y0A21YK0F [11]

Senzor ima mogućnost mjerenja udaljenosti od 10 do 80 centimetara, ima tri izvoda, napajanje, nulu i signal čija naponska vrijednost varira od udaljenosti promatranog predmeta od senzora [12]. Uređaj radi na principu da dioda koja odašilje infracrvenu svjetlost pošalje infracrvenu zraku u smjeru od senzora. Uređaj pored te infracrvene diode također sadrži i PSD – *position sensitive detector*. PSD je optički senzor pozicije koji može mjeriti poziciju točke svjetlosti u jednoj ili dvije dimenzije na površini senzora [13]. Tako se odašiljana infracrvena zraka reflektirana od predmeta vraća prema uređaju i pogađa PSD te se tada signal sa PSD-a obrađuje u sklopu za obradu signala. Tada se u sklopu generira napon, u ovisnosti o udaljenosti predmeta, koji se onda može mjeriti te na temelju njega zaključiti o udaljenosti predmeta od senzora. Shema sklopa senzora prikazana je na Slika 5.4.

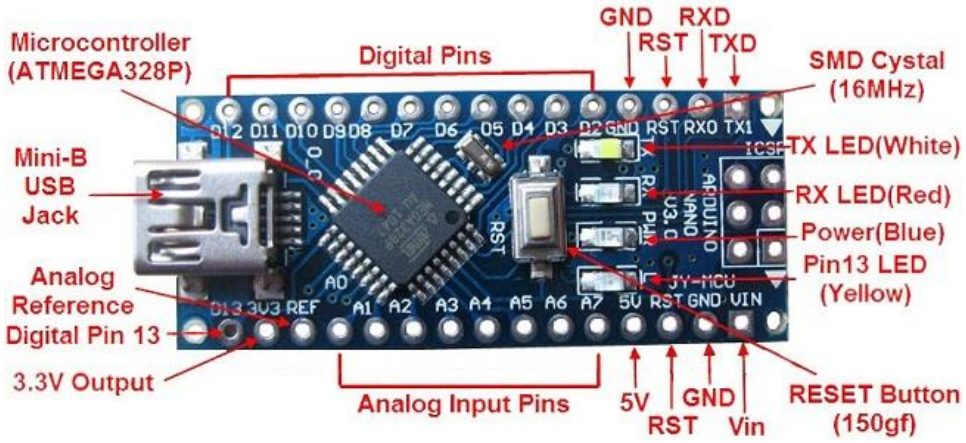


Slika 5.4. Shema električnog sklopa senzora [12]

Montažom gore odabranog senzora na prikladno mjesto na realnoj konstrukciji rezultirati će mjerenjem udaljenosti kuglice na kotaču od senzora. Tada se primjenom geometrije i aproksimacijom da se kuglica giba linearno a ne po kružnici (kotaču) dolazi do vrijednosti kuta β, odnosno kuta kuglice na kotaču. Aproksimacija linearnog gibanja je valjana jer bi vrijednosti kuta β ionako trebale biti male da bi sustav ostao u ravnoteži.

5.6. ODABIR MIKROPROCESORA

Odabirom aktuatorskih i senzorskih komponenti sustava riješen je veći dio zadatka. U nastavku treba odabrati jedinicu koja će obrađivati senzorske veličine, slati prikladne signale u aktuatorski dio sustava te sadržavati regulacijski algoritam koji će sustav održavati u ravnotežnom položaju. Za tu funkciju odabrana je pločica Arduino Nano i prikazana je na Slika 5.5.



Slika 5.5. Mikrokontroler Arduino Nano [14]

Pločica Arduino Nano sadrži Atmel-ov mikroprocesor oznake ATMEGA328P koji ima sljedeće karakteristike [15]:

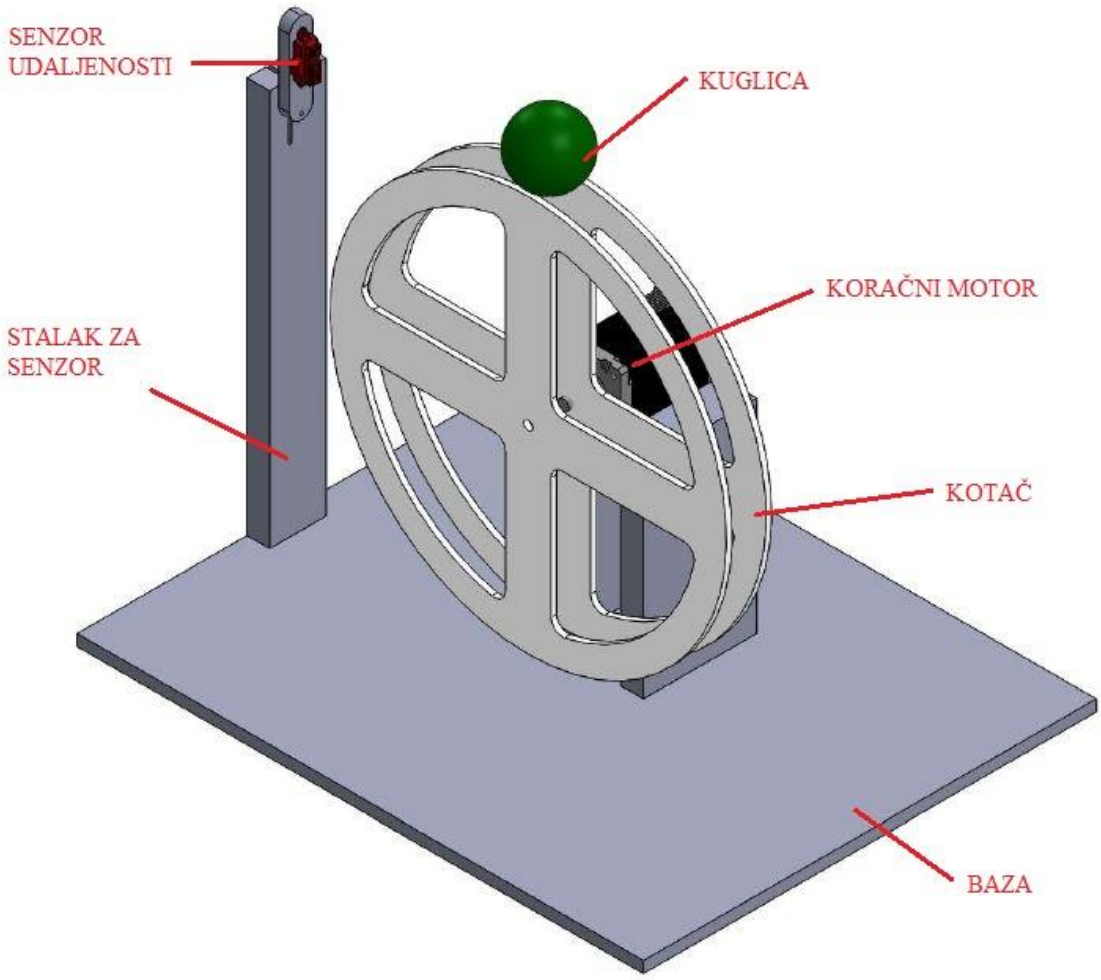
- logički napon – 5 V,
- *flash* memorija – 32 KB,
- radni takt – 16 MHz,
- analogni ulazno/izlazni pinovi – 8,
- digitalni ulazno/izlazni pinovi – 22,
- EEPROM – 1KB,
- maksimalna struja po pinu – 40mA,
- ulazni napon – 7-12 V,
- PWM izlazi – 6.

Pločica je odabrana iz više razloga. Kao što se vidi iz Slika 5.5. pločicu je moguće istovremeno i napajati i programirati pomoću mini-USB priključka koji se preko USB-a spaja na računalo, pločica je malih dimenzija i prikladna je za testiranje na testnoj ploči, sadrži oscilirajući kristal za davanje takta mikroprocesoru te sadrži LED-ice koje korisniku daju informaciju i stanju rada mikroprocesora, primanju i slanju podataka. Također sadrži regulator napajanja te omogućuje napajanje pločice i na načine drugačije od napajanja preko USB veze u odgovarajućem naponskom opsegu. Pločica je tako prikladna za testiranje sustava jer omogućuje lagano spajanje raznih komponenti na odgovarajuće ulaze i izlaze u pločicu te se preko USB veze može u par sekundi potpuno izmijeniti trenutni algoritam u mikroprocesoru.

6. KONSTRUKCIJA I UGRADNJA KOMPONENTI

6.1. MODELIRANJE I IZRADA KONSTRUKCIJE

Prije same izgradnje konstrukcije treba se pristupiti njenom modeliranju. Modeliranje konstrukcije provedeno je u programskom paketu SolidWorks. Prije samog modeliranja treba se definirati što se od konstrukcije očekuje. Konstrukcija treba imati neku vrstu baze na koju će se pričvrstiti ostali dijelovi konstrukcije i komponente sustava, mora se smisliti dio konstrukcije na kojem će se nalaziti koračni motor i mora se osmisliti način na koji će se koračni motor kvalitetno pričvrstiti na konstrukciju pošto će razvijati moment koji se mora prenositi samo na kotač. Također se morao smisliti kako će se kotač pričvrstiti na izlazno vratilo motora u smislu da se osigura direktan prijenos brzine vrtnje i momenta motora bez ikakvog proklizavanja. Također se mora oblikovati kotač da se kuglica može slobodno gibati po njemu te je potrebno prije odabrani senzor udaljenosti smjestiti u ravninu sa vrhom kotača da može dobro i točno izmjeriti poziciju kuglice na kotaču. Okvirno modelirana konstrukcija prikazana je na Slika 6.1. te će se njeni dijelovi opisati u nastavku.



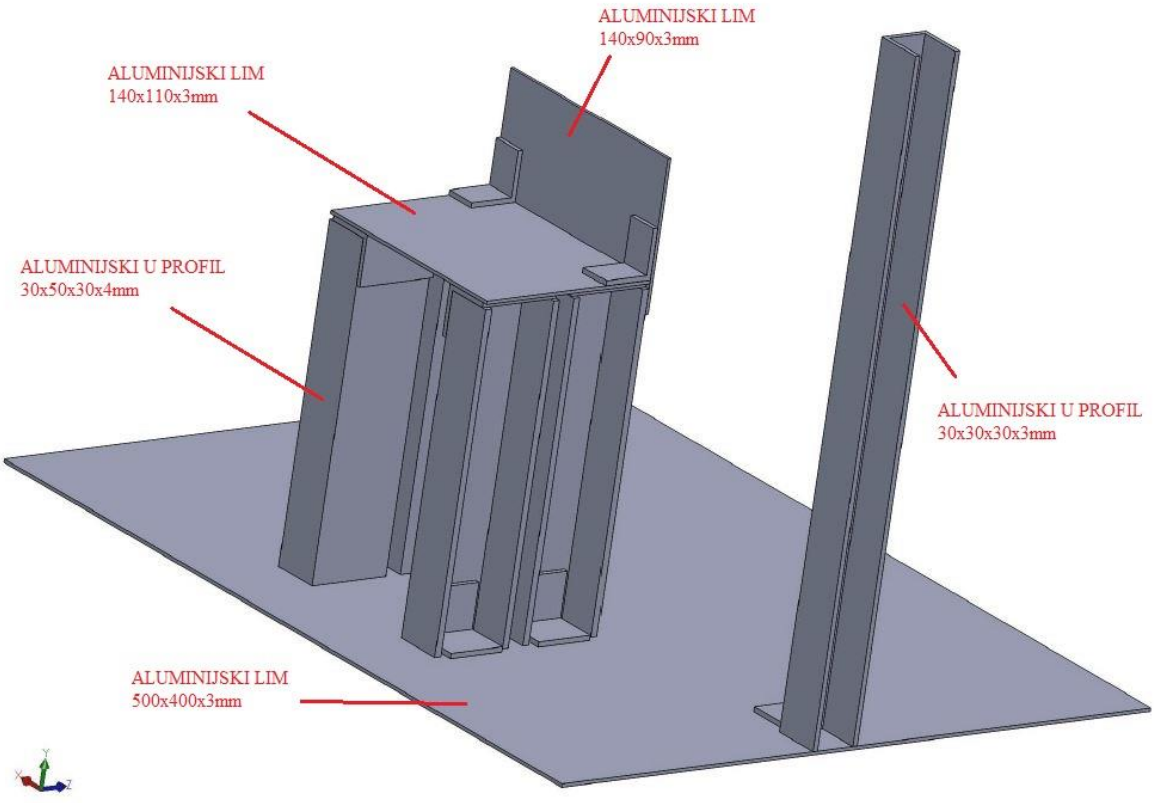
Slika 6.1. Okvirno modelirana konstrukcija

Na Slika 6.1. vide se osnovni dijelovi konstrukcije. Baza je pravokutni lim, senzor udaljenosti pričvršćen je za stalak izrađen od U-profila, koračni motor nalazi se na jednostavnoj konstrukciji, kotač se sastoji od dva diska koji su međusobno povezani i pričvršćeni za izlazno vratilo motora te se kuglica giba po rubovima dva diska.

No da bi se konstrukcija realno izradila, potrebno je točno definirati sve dimenzije, potrebne dijelove te izabrati materijal konstrukcije. Odlučeno je da će se cijela konstrukcija, osim kotača, izraditi od aluminija pošto bi čelik bio pretežak i mehanički zahtjevi konstrukcije nisu toliki da se zahtijevaju karakteristike koje ima čelična konstrukcija. Baza je izrađena od $AlMg_3$ lima debljine 3 mm i dimenzija 500x400 mm te je vodenim mlazom rezana. Od istog materijala iste debljine također su vodenim mlazom izrezana još dva lima, jedan na kojemu je smješten koračni motor, dimenzija 140x110 mm i drugi pomoću kojega je koračni motor pričvršćen na konstrukciju, dimenzija 140x90 mm.

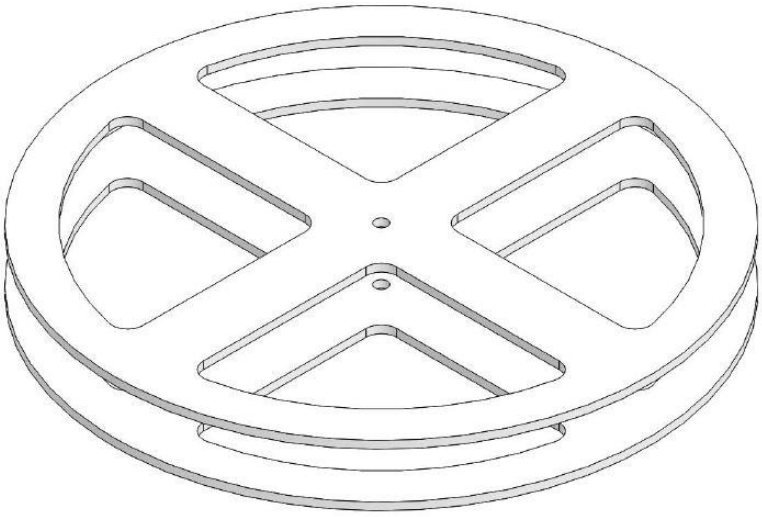
Dio konstrukcije na kojoj je koračni motor izrađen je od četiri komada aluminijskog U-profila dimenzija 30x50x30x4 mm i duljine 20 cm. Oni su spojeni na bazu također pomoću aluminijskih kutnih profila dimenzija 30x30x3 mm duljine 35 mm. Na te profile pričvršćen je gore navedeni lim također pomoću istih kutnih profila, duljine 50 mm. Za U-profile i za na njih pričvršćeni lim, pričvršćen je drugi lim. Stalak za senzor oblika je U-profila dimenzija 30x30x30x3 mm i duljine 45 cm te je za bazu također pričvršćen sa prije navedenim kutnim profilom. Svi spojevi na konstrukciju ostvareni su vijčano sa M4 vijcima različitih duljina i maticama.

Metalni dio konstrukcije prikazan je na Slika 6.2.



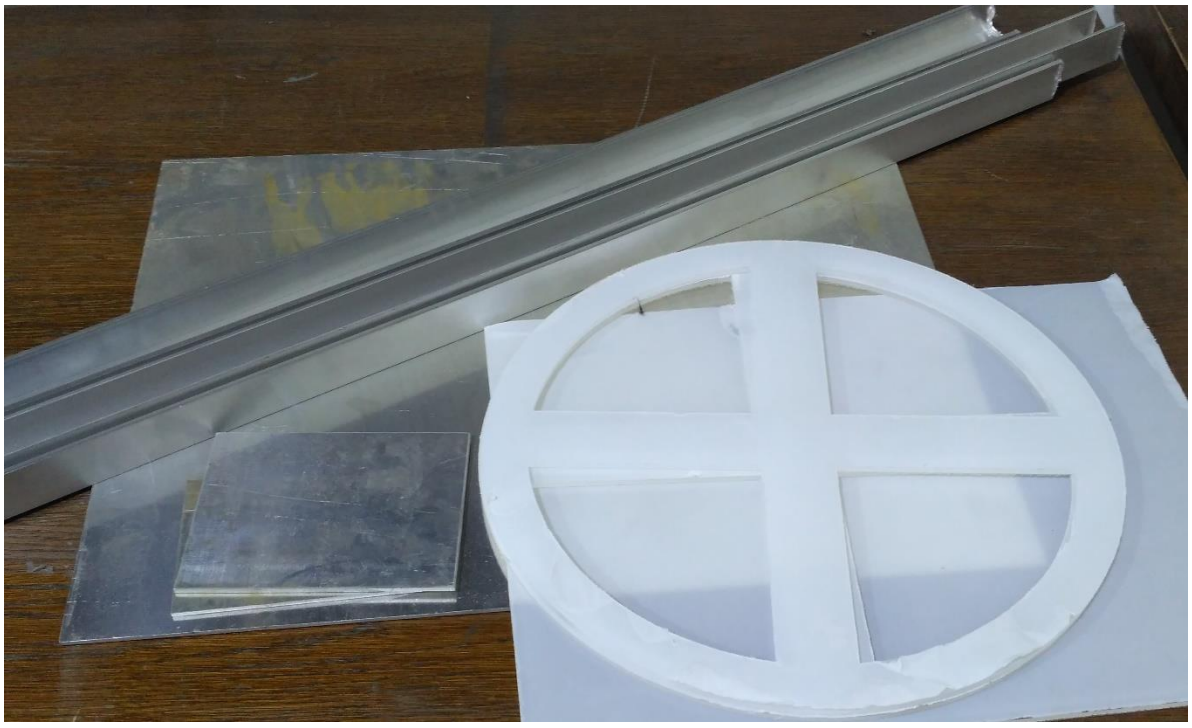
Slika 6.2. Metalni dio konstrukcije

Kotač je izrađen od pleksiglasa pošto bi bio pretežak ako bi se izradio od metala te bi time imao veću inerciju te bi bio potreban veći moment za njegovo pokretanje i rotaciju. Kao što je prije navedeno, kotač je izrađen od dva jednaka diska i to od pleksiglasa debljine 4 mm. Promjer diska je 350 mm te je izrađen sa šupljinama da bi mu se što više smanjila masa. Pošto se zahtijevala glatka kružnica po cijelom obodu oba diska, nije se pristupilo samostalnom rezanju pleksiglasa nego je on također vodeno rezan. Modeliran kotač prikazan je na Slika 6.3.



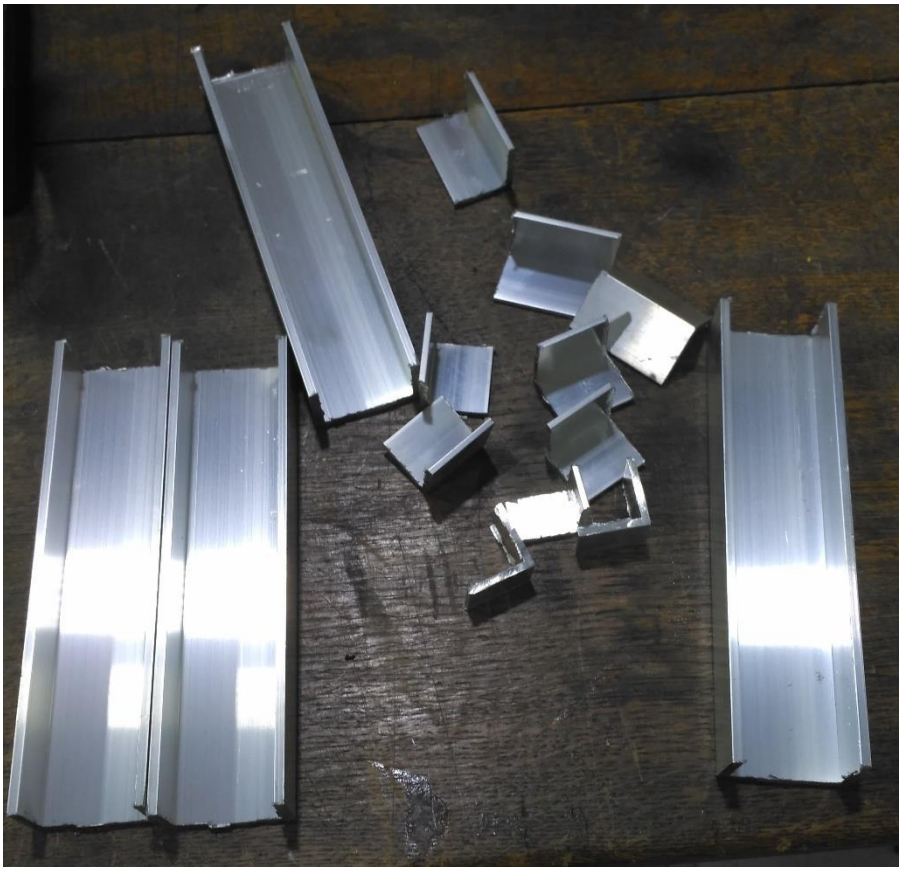
Slika 6.3. Modelirani kotač

Na Slika 6.4. prikazani su korišteni materijali.

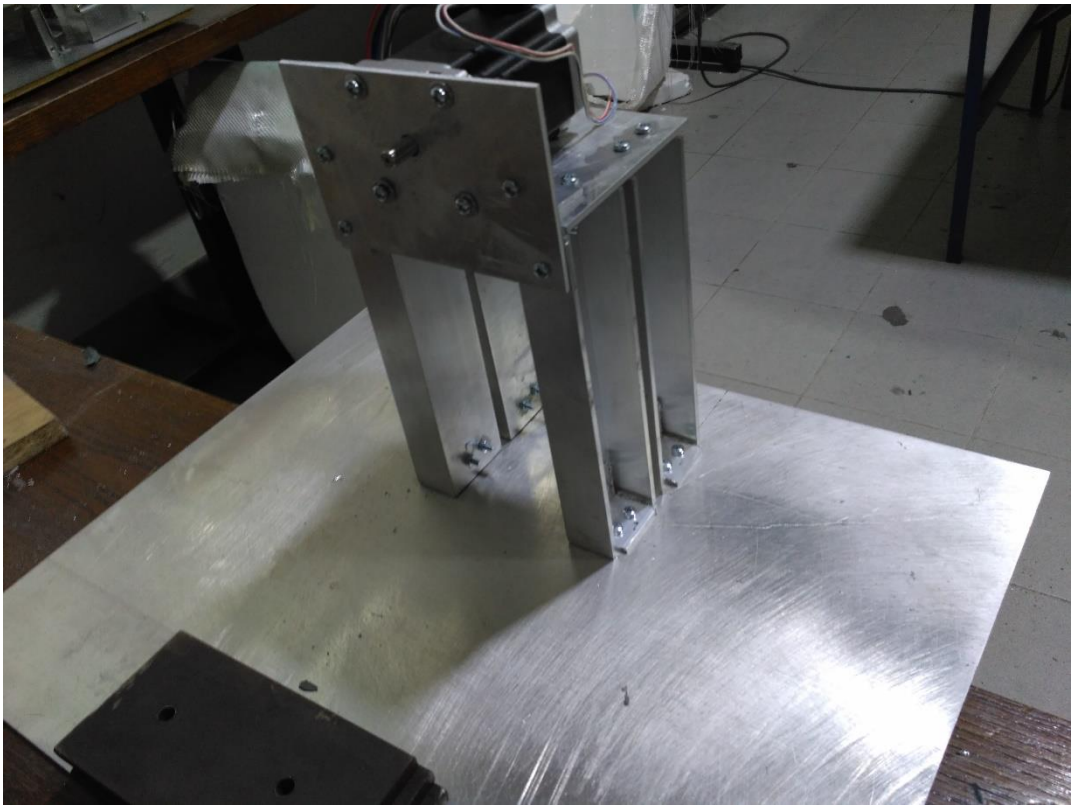


Slika 6.4. Svi dijelovi konstrukcije

Aluminijski profili rezani su brusilicom te kasnije istom i pobrušeni.



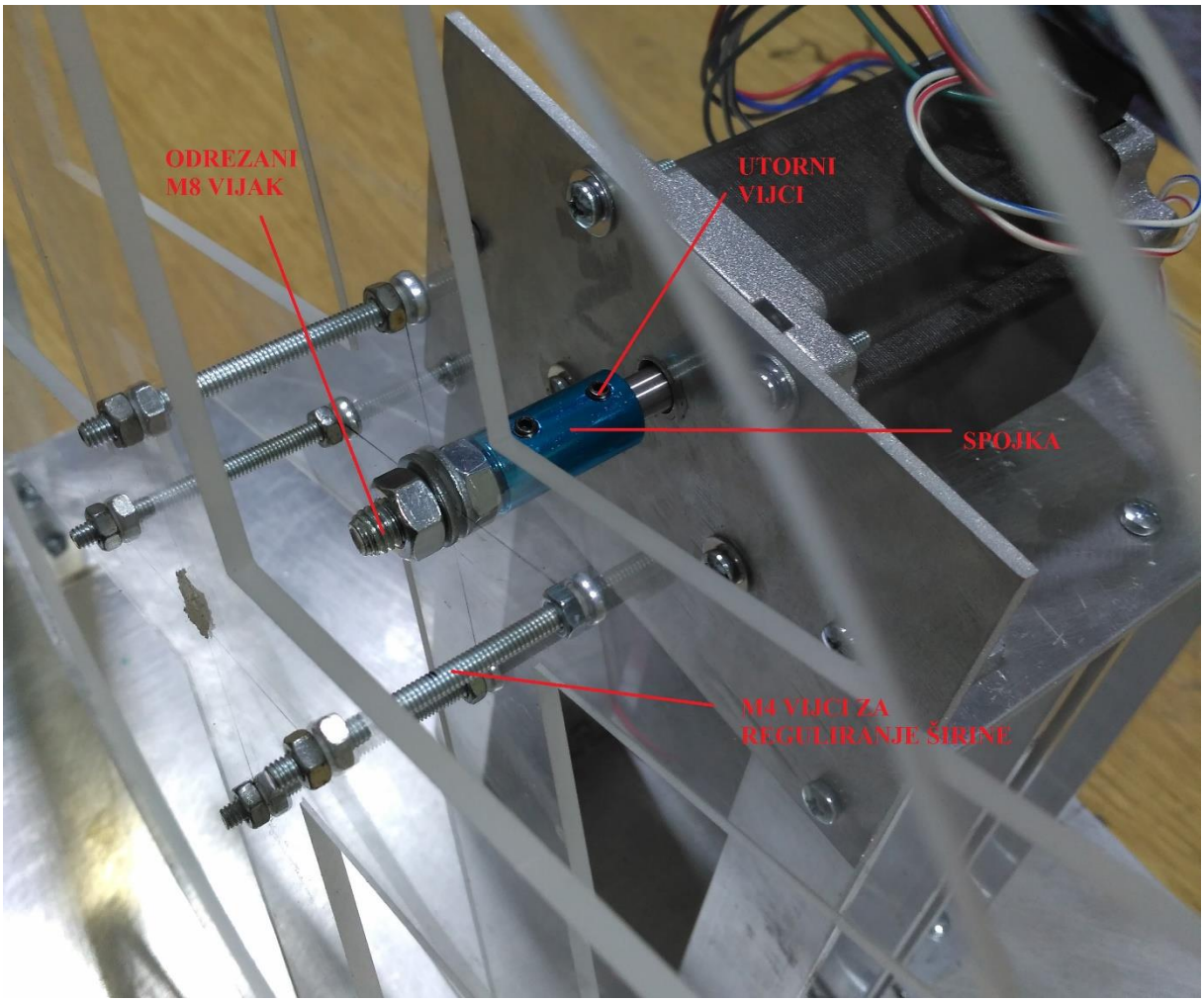
Slika 6.5. Izrezani aluminijski profili



Slika 6.6. Izrađeni dio konstrukcije za motor

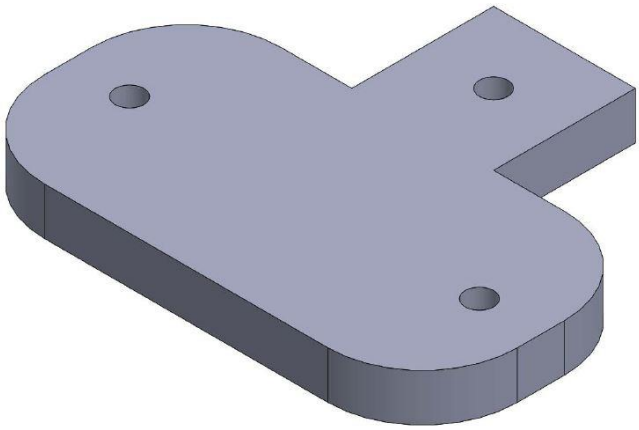
Kao što je vidljivo na Slika 6.6., koračni motor je na konstrukciju spojen pomoću lima i to sa četiri M4 vijka. Taj lim je za konstrukciju spojen također sa četiri vijka, sa dva je spojen za donje profile, a sa dva je preko kutnih profila spojen za lim na kojem se nalazi motor. Motor je time čvrsto spojen na konstrukciju te i u kasnijem radu ne dolazi do nikakvih vibracija ni pomaka konstrukcije.

Slijedi spoj kotača i motora. Spoj prvog diska i izlaznog vratila ostvaren je na sljedeći način. Nabavljena je spojka ulaznog i izlaznog promjera 8 mm koja se pričvršćuje utornim vijcima pošto je izlazno vratilo iz motora promjera 8 mm i punog je kružnog promjera, bez već izrađenih utora za lakše pričvršćivanje. U izlaznu stranu spojke umetnuta je glatka strana izrezanog M8 vijka sa glatkim vratom, dok je sa dvije podloške i sa dvije matice vijak čvrstim spojem povezan za prvi disk kotača. Drugi disk kotača je na prvi disk povezan pomoću četiri duga M4 vijka pa se tako razmak između dva diska a time i širina kotača može podešavati ovisno o veličini kuglice. Spoj je prikazan na Slika 6.7.



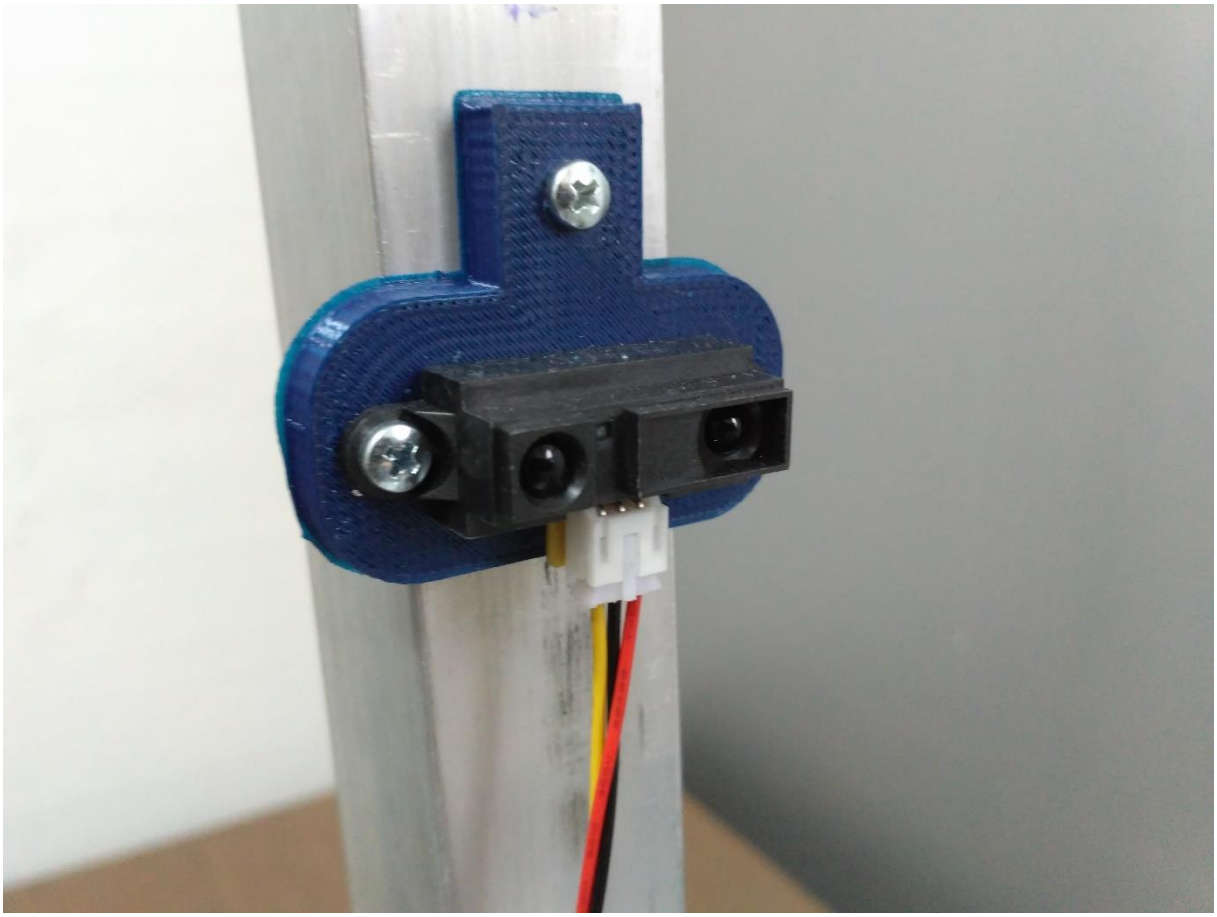
Slika 6.7. Spoj između motora i kotača

Slijedi montaža senzora udaljenosti. On na kućištu posjeduje dva provrta za vijčano pričvršćivanje. Modeliran je dio na koji će se pričvrstiti senzor i prikazan je na Slika 6.8.



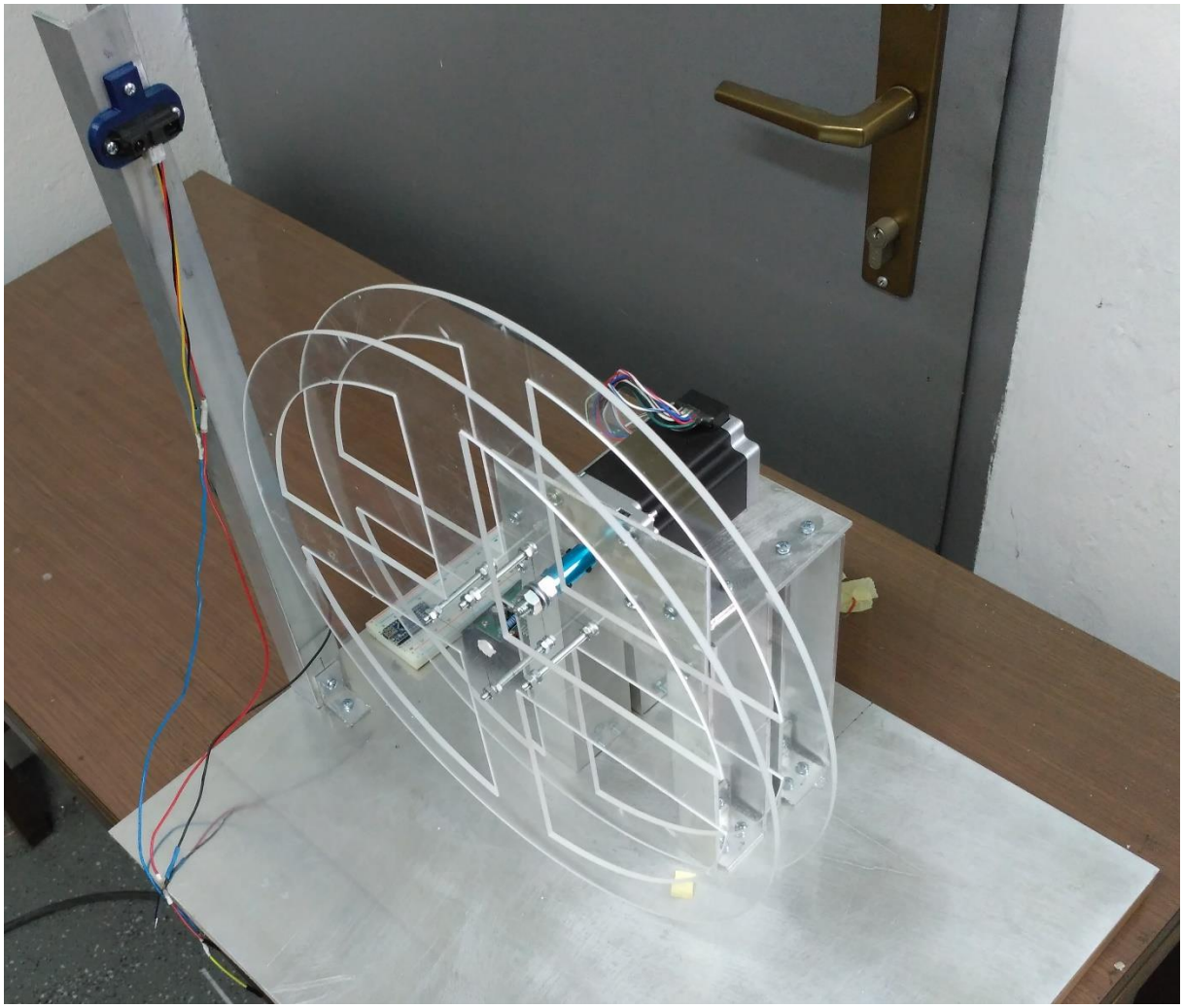
Slika 6.8. Modelirani dio za spoj senzora

Dio potreban za spoj senzora s aluminijskim profilom izrađen je 3D printanjem i on je također vijčano spojen na aluminijski U-profil sa Slika 6.2. u ravnini sa vrhom kotača. Spoj je prikazan na Slika 6.9.



Slika 6.9. Spoj senzora udaljenosti na konstrukciju

Slijedi slika gotove konstrukcije koja sadrži ugrađene sve potrebne komponente.

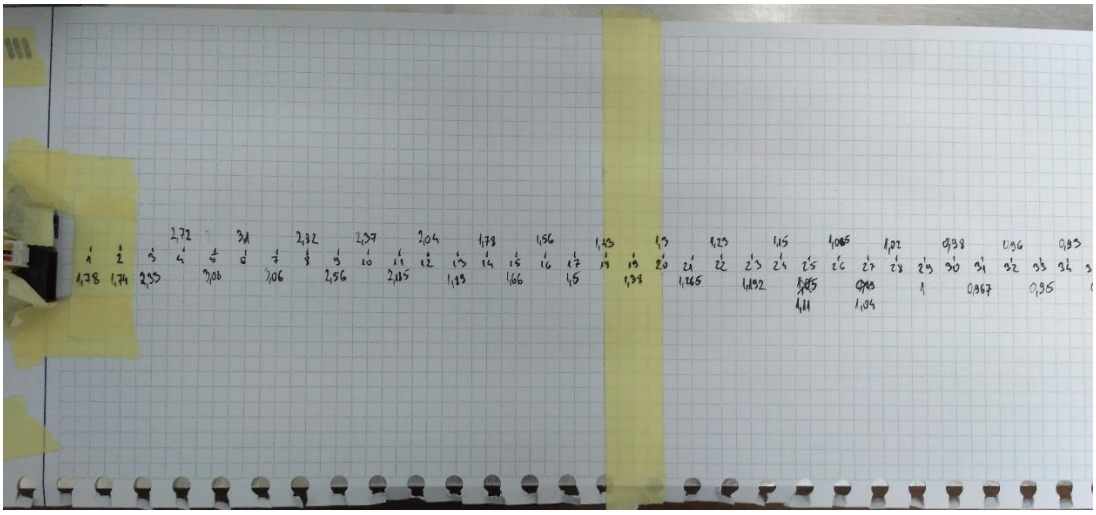


Slika 6.10. Gotova konstrukcija

6.2. KALIBRIRANJE SENZORA UDALJENOSTI

Senzoru udaljenosti u uputama proizvođača već je definirana izlazna karakteristika koja prikazuje ovisnost napona signala o udaljenosti promatranog predmeta od senzora. No pošto će svaki senzor imati blago drugačiju karakteristiku zbog nesavršenosti proizvodnje, provedena je samostalna kalibracija senzora da bi se postigla maksimalna dostupna preciznost.

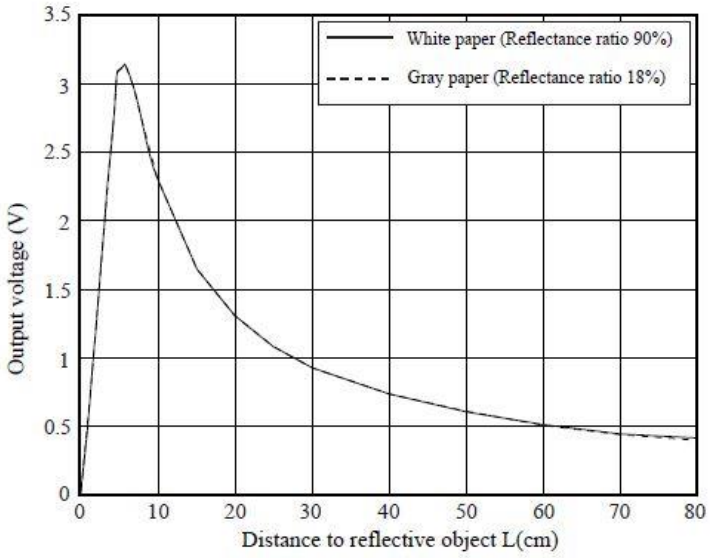
Kalibracija će se odvijati na način da će se senzor pričvrstiti za podlogu na koju je također zalijepljen papir sa označenim brojem centimetara udaljenosti od senzora [16]. Ispred senzora se tako redom svaki centimetar postavlja komad papira koji simulira prepreku i očitavaju se vrijednosti napona za svaku udaljenost. Postav je prikazan na Slika 6.11.



Slika 6.11. Kalibriranje senzora

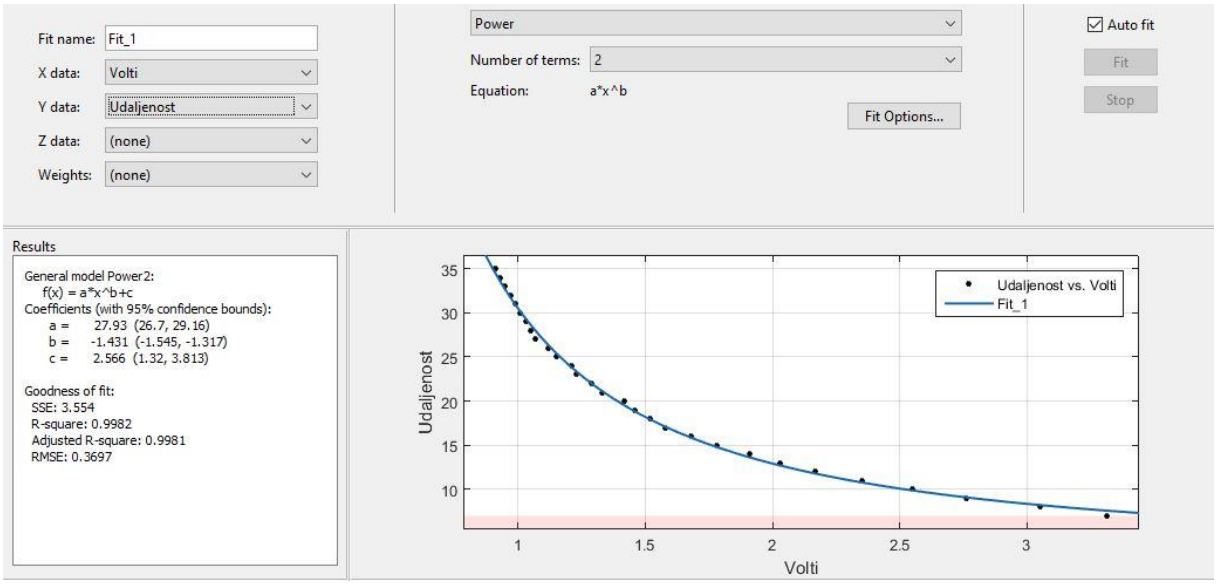
Kao što se vidi na Slika 6.11. izmjerene su vrijednosti napona za udaljenosti od 0 do 35 centimetara te su prepisane u skriptu u MATLAB-u. Potrebno je iz izmjerenih podataka doći do funkcije koja će aproksimirati karakteristiku senzora kako bi se u kodu moglo iz očitanog napona sa senzora odrediti udaljenost predmeta. Koristiti će se već ugrađena aplikacija u MATLAB-u, naziva Curve Fitting, koja iz učitanih podataka u MATLAB-u ovisno o željama korisnika generira krivulju za opisivanje tih podataka [16]. Krivulja može biti interpolacijska, polinomska, eksponencijalna, racionalna, spline, suma sinusa itd. Na korisniku je da definira koju će krivulju koristiti te također postoje dodatne opcije za svaki odabir u smislu npr. stupnja polinoma, dodatnog parametra kod eksponencijalne funkcije, linearne ili spline interpolacije itd.

Iz karakteristike senzora, dostupne iz uputa proizvođača, vidi se nepravilnost u radu senzora do udaljenosti od oko 8 centimetara jer senzor nije predviđen da radi za udaljenosti manje od te vrijednosti. Tako se mjerenja napona za manje udaljenosti od 8 cm isključuju iz podataka za generiranje krivulje. Na Slika 6.12. prikazana je karakteristika senzora iz uputa proizvođača.



Slika 6.12. Karakteristika senzora po uputama proizvođača [12]

Gornja karakteristika prikazuje ovisnost izlaznog napona o udaljenosti do promatranog predmeta. No za regulacijski algoritam potrebna je inverzna karakteristika koja će davati podatak o udaljenosti do promatranog predmeta u ovisnosti o signalnom naponu. Takva se krivulja zatim treba aproksimirati iz izmjerenih podataka. Izabran je eksponencijalni oblik krivulje jer je najmanji stupanj polinoma koji je približno dobro opisivao izmjerene podatke bio četvrtog ili petog stupnja te bi se trošilo previše procesorskog vremena za računanje udaljenosti pomoću takvih polinoma. Izabrana krivulja prikazana je na Slika 6.13.



Slika 6.13. Generirana karakteristika senzora

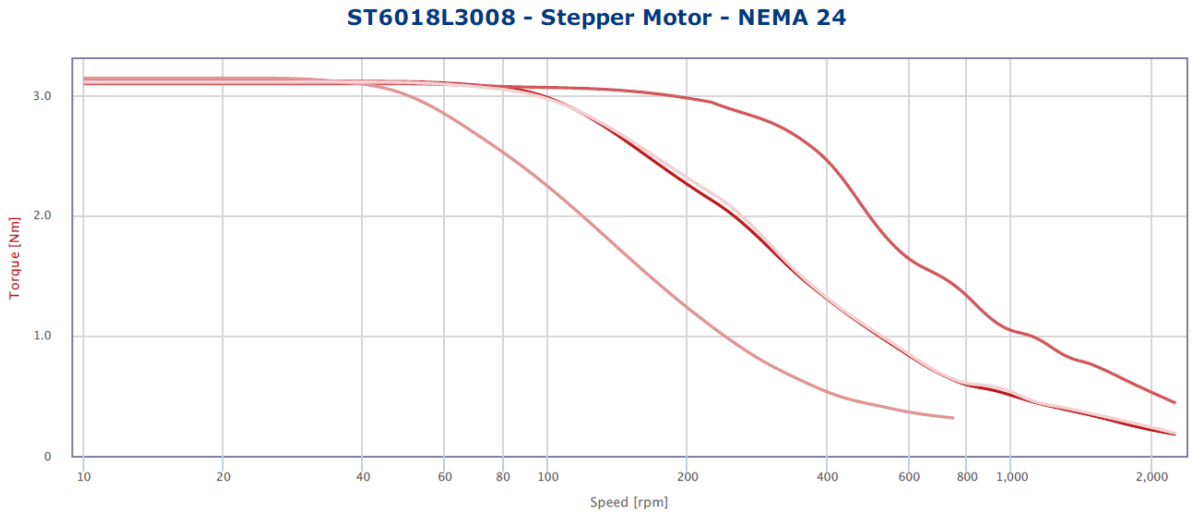
Prema Slika 6.13. generirana karakteristika senzora je sljedećeg oblika:

$$udaljenost = 27,93 \cdot volti^{-1,431} + 2,556 \ . \tag{6.1.}$$

Izraz (6.1.) uvršten je u kod za mjerenje udaljenosti te je testiran na postavu sa Slika 6.11. te se pokazalo da generirana krivulja dobro aproksimira mjerne podatke i da se može upotrijebiti u izvedbi realnog sustava.

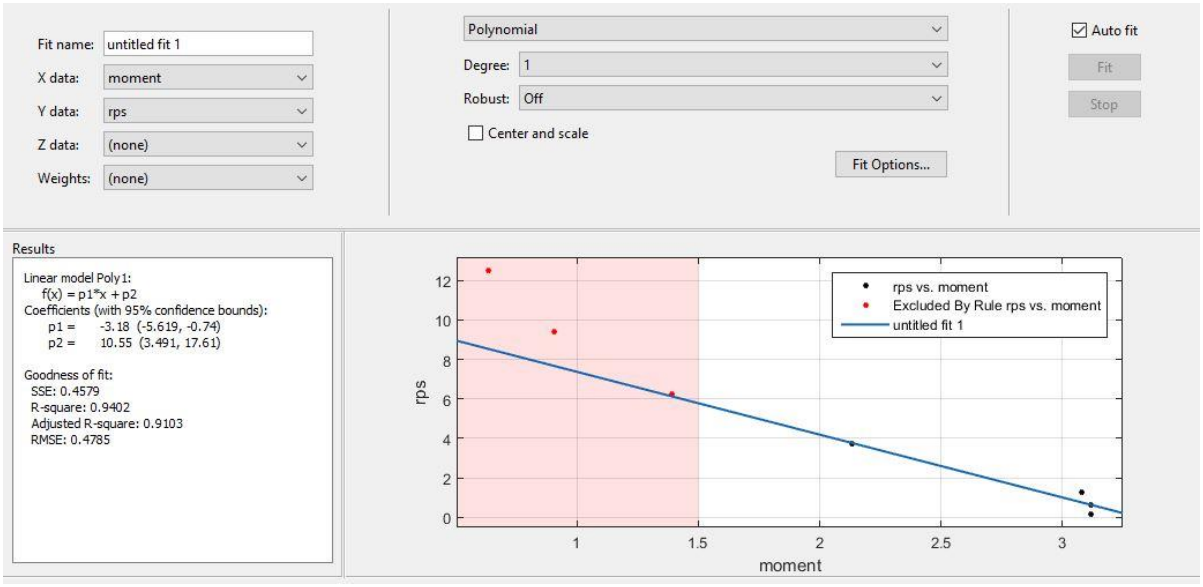
6.3. MOMENTNA KARAKTERISTIKA KORAČNOG MOTORA

U 2. poglavlju modeliran je sustav kojemu je ulazna varijabla moment. Taj moment će proizvesti koračni motor. Moment koračnog motora ovisi o broju okretaja, no pošto se on pokreće i rotira slanjem pulsnih signala, većina proizvođača momentne karakteristike koračnih motora prikazuje kao ovisnost momenta o broju pulseva po sekundi, tj. *pps – pulses per second*. No također se može prikazati kao ovisnost momenta o *rpm – revolutions per minute*, ili *rps – revolutions per second*. Tvrtka Nanotec, koja je proizvođač odabranog koračnog motora, momentnu karakteristiku je prikazala kao ovisnost momenta o *rpm* za napone napajanja od 24 i 48 volti te za serijski ili paralelni spoj izvoda. Momentna karakteristika prikazana je na Slika 6.14.



Slika 6.14. Momentna karakteristika odabranog koračnog motora [17]

Kao što je već ranije navedeno, izvodi motora spojeni su paralelno te je odabran napon napajanja od 24 volta. Na stranici proizvođača motora gornja karakteristika može se otvoriti u interaktivnom obliku te se mogu snimiti podaci za određeno napajanje i način spoja. Za odabranu karakteristiku preuzeti su željeni podaci i generirana krivulja u Curve Fitting aplikaciji. Kao i za senzor udaljenosti, potrebna je inverzna karakteristika jer se moment izračunava u regulacijskom algoritmu te je onda iz tog iznosa potrebno odrediti potrebnu brzinu rotacije da bi se taj moment u stvarnosti i dobio. Isključeni su podaci za veće brzine okretaja jer se ionako kotač ne bi smio vrtjeti toliko brzo da kuglica ne bi odletjela sa njega. Odabrana je polinomska krivulja prvog stupnja jer može dovoljno dobro opisati uključene podatke i prikazana je na Slika 6.15.



Slika 6.15. Generirana krivulja momentne karakteristike

Prema Slika 6.15. generirana krivulja momenta iznosi:

$$rps = -3{,}18 \cdot moment + 10{,}55 \ . \qquad (6.2.)$$

7. IZVEDBA EKSPERIMENTA I REZULTATI

7.1. REGULACIJSKI ALGORITAM

Za slanje i primanje signala koristiti će se već prije navedena pločica Arduino Nano koja sadrži Atmelov ATMega328P mikroprocesor. Programiranje mikroprocesora se odvija preko USB veze i to u programskom okruženju Arduino IDE. U njemu je napisan regulacijski algoritam čiji opis slijedi u nastavku. Sastojati će se od četiri dijela; deklaracija raznih potrebnih varijabli, funkcije za mijenjanje Timer-a, funkcija koja će u pravilnim vremenskim razmacima obavljati sva potrebna mjerenja i dio koji će se odvijati kroz cijelo vrijeme rada mikroprocesora.

7.1.1. DEKLARACIJA POTREBNIH VARIJABLI

Na Slika 7.1. prikazan je dio koda gdje su deklarirane razne potrebne varijable za izvedbu regulacijskog algoritma te je kraj svake varijable opisana njena funkcija.

```
//kotac
byte directionPin = 7; //izlaz preko kojeg se mijenja smjer vrtnje
byte stepPin = 8; //izlaz preko kojeg se zadaje puls driveru
long broj_koraka=0L; //brojanje koraka koracnog motora
float delta_alfa; //promjena kuta izmedju dva mjerenja
float alfa_om; //kutna brzina kotaca
boolean smjer=0; //smjer vrtnje; 0=CCW; 1=CW,
//time je i pozitivan smjer vrtnje i pozicije CW, a negativan smjer vrtnje i pozicije CCW
long broj_koraka_prosli=0L; //broj prijedjenih koraka u proslom mjerenju

//kuglica
int udpin = 4; //analogni ulaz za senzor
float tocnaudalj; //tocna udaljenost iz senzora
float zadnjaudalj; //tocna udaljenost iz senzora u proslom mjerenju
float beta_tocni; //kut kuglice beta
float beta_zadnji; // kut kuglice beta u proslom mjerenju
float betaom; //kutna brzina kuglice
float x; //objasnjeno kasnije

//regulator
float moment; //moment koji je izlaz iz regulatora
const float K1=-0.0796; //pojacanje za varijablu stanja kutne brzine kotaca
const float K2=9.9324; // pojacanje za varijablu stanja kuta kuglice
const float K3=2.4736; // pojacanje za varijablu stanja kutne brzine kuglice
float os; //izracunata brzina kotaca pri kojoj se dobiva gornje izracunati moment
float os_pravi;//signal koji saljemo

long zadnje_vrijeme=0L; //vrijeme zadnjeg mjerenja u milisekundama
long trenutno_vrijeme=0L; //trenutno vrijeme u milisekundama
int period_mjerenja=25; //period mjerenja u milisekundama
```

Slika 7.1. Deklarirane varijable

7.1.2. MIJENJANJE TIMER-A

Kako bi se osigurao pravilan rad koračnog motora, a time i točna i tražena brzina vrtnje, on mora dobivati pulsne signale za korake u točnim i pravilnim vremenskim razmacima. Kako bi se to ostvarilo, u kodu se mora osigurati da se pošalje pulsni signal svako definirano vremensko razdoblje. To se može ostvariti *if* petljom na početku svakog *loop-a,* gdje je *loop* u Arduino IDE-u dio koda koji se neprestano vrti u mikroprocesoru. No u slučaju da bi se dio koda u *loop-u* vrtio dulje nego što je definirani vremenski razmak, ili neka *while* petlja ne bi bila ispunjena, treba postaviti dodatne provjere u obliku dodatnih *if* petlji tijekom *loop* dijela koda čime se opet samo usporava izvršenje programa.

Zato je kao način osiguravanja da se pulsni signali za izvršavanje koraka zadaju u pravilnim vremenskim razmacima odabrana upotreba brojača Timer-a. Timer-i su unutrašnji prekidi (*eng. internal interrupts*) koji kada se aktiviraju izvršavaju neki prekidni dio koda (*eng. interrupt routine*) bez obzira u kojem dijelu koda se izvršavanje prije događalo jer prekidi imaju prioritet nad svim ostalim dijelovima koda [18]. To je posebno korisno kod npr.:

- mjerenje ulaznih signala u točno određenim intervalima,
- izračunavanje vremena između dva događaja,
- slanje signala točno određene frekvencije,
- periodična provjera dolaznih podataka preko serijske veze itd.

ATMega328P ima tri Timer-a, nazvani *timer0, timer1* i *timer2* [15]. Svaki od njih ima brojač koji se inkrementira pri svakom taktu Timer-a. Takt Timer-a će se objasniti u nastavku. Prekidni dio koda dodijeljen određenom Timer-u se aktivira kada brojač dosegne neku prije definiranu vrijednost nakon čega se brojač resetira na nulu i izvršava prekidni dio koda. Odabirom te definirane vrijednosti i postavljanja brzine takta Timer-a može se kontrolirati i definirati frekevencija izvršavanja prekidnih dijelova koda. Svaki od tri Timer-a kontrolira PWM signale za dva digitalna izlaza, dok je *timer0* odgovoran za Arduino IDE-ovu funkciju *millis()* koja mjeri trenutno vrijeme rada procesora u milisekundama što je većinom jako bitno tijekom programiranja realnih sustava. Dalje *timer0* i *timer2* su 8-bitovni što znači da njihova vrijednost brojača može dosegnuti maksimalnu vrijednost od 255. Tako se odabire *timer1* koji je 16-bitni i može spremiti vrijednost brojača do 65535.

Kao što je već napomenuto, ATMega328P radi na frekvenciji takta od 16MHz, znači da može obaviti 16 milijuna operacija u sekundi. To također znači da ako bi se brojač od *timer1* inkrementirao svaki takt mikroprocesora, vrlo bi brzo dosegao vrijednost od 65535 nakon čega bi se vrijednost brojača resetirala dok bi u stvarnosti prošlo vrlo malo vremena. Tako se uvode takozvani preskalari koji imaju vrijednosti 1, 8, 64, 256 i 1024 i definiraju svakih koliko će se inkrementirati vrijednost brojača po sljedećoj jednadžbi:

$$(brzina\ timer\text{-}a(Hz)) = (brzina\ takta\ ATMega328P(16MHz))/preskalar\ . \qquad (7.1.)$$

U slučaju da se želi aktivacija Timer-a određenom frekvencijom, mora se izračunati s kojom će se vrijednošću uspoređivati vrijednost brojača tog Timer-a, znači mora se izračunati vrijednost

usporedbe (*eng. compare match register*), u daljnjem tekstu *cmr*, i to se radi pomoć u sljedećeg izraza:

$$željena\ frekvencija\ prekida(Hz) = \frac{16.000.000}{preskalar(cmr+1)}$$

$$cmr = \frac{16.000.000}{preskalar \cdot željena\ frekvencija\ prekida} - 1 \quad , \qquad (7.2.)$$

i ona ne smije, kao što je prije rečeno, prijeći vrijednost od 65535. S obzirom na predviđeni broj okretaja po sekundi kotača, odabran je iznos preskalara od 1024. Slijedi Slika 7.2. na kojoj je prikazana sintaksa definiranja vrijednosti Timer-a u Arduino IDE i izračun potrebne željene frekvencije prekida.

```
//izracun za timer, ulaz je zeljeni broj okretaja po sekundi
float izracun_cmr(float okposek) {
  float herci=400*okposek; //iako koračni motor radi 200 koraka za jedan okretaj,
  //driver radi u 2 microsteppingu pa mu treba 400 koraka za jedan okretaj
  float cmr=(16000000/(1024*herci))-1;
  return cmr;
}

//cmr iz gornje funkcije ulazi u donju
//pomocu donje funkcije mijenja se iznos timera a time i brzina vrtnje koracnog motora
void mijenjanje_timera(float cmr_izracunati){
  cli();
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0;
  OCR1A = cmr_izracunati;
  TCCR1B |= (1 << WGM12);
  TCCR1B |= (1 << CS12) | (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);
  sei();
}

//interrupt rutina koja se vrti svaki put kad timer odbroji
//rotira motor za jedan korak i inkrementira broj koraka
ISR(TIMER1_COMPA_vect){
  digitalWrite(stepPin,HIGH);
  digitalWrite(stepPin,LOW);
  broj_koraka++;
}
```

Slika 7.2. Mijenjanje vrijednosti Timer-a

7.1.3. FUNKCIJA MJERENJA

U funkciji *mjerenje()* definiraju se nove vrijednosti varijabli stanja te se kao takve ubacuju u regulator. Na Slika 7.3. prikazana je funkcija i svaka linija sadrži komentare o tome što se točno događa kroz funkciju.

```
//funkcija za mjerenja
void mjerenja() {
  //prvo se mjeri kutna brzina kotaca
  delta_alfa=((broj_koraka-broj_koraka_prosli)/400.0)*360.0; //promjena kuta od zadnjeg mjerenja,
  //iako motor ima 200 koraka, microstepping je iznosa 2 pa je broj koraka za jedan krug 400,
  //dijeljenjem sa 360 se dobiva kut
  broj_koraka_prosli=broj_koraka; //resetiranje vrijednosti proslih koraka
  if (smjer==0){
    delta_alfa=-delta_alfa;  //ako se kotac vrti ccw promjena je negativna
  }
  alfa_om=1000.0*(delta_alfa*DEG_TO_RAD)/(trenutno_vrijeme-zadnje_vrijeme); //kutna brzina kotaca
  //dobiva se pretvaranjem promjene kuta u radijane i podjelom sa protecenim vremenom izmedju
  //dva mjerenja.onda se mjeri kut kuglice i kutna brzina kuglice
  int p=0; //broj mjerenja udaljenosti kuglice od senzora
  float sum=0; //suma mjerenja
  for (int i=0; i<25; i++){ //radi se 25 mjerenja
    float raw=analogRead(udpin); //ocitavaju se volti na ulazu udpin u 10bitnoj vrijednosti
    float volti = map(raw,0,1023,0,5000); //ti volti se mapiraju u vrijednost
    //od 0 do 5000 radi bolje preciznosti
    float udalj=28.46*pow(volti/1000,-1.263)+1.304;
    sum=sum+udalj;
    p++;
  }
  tocnaudalj=sum/p; //uzima se srednja vrijednost svih mjerenja
  tocnaudalj=tocnaudalj+3.0; // 3cm je radijus kuglice, posto senzor ocitava rub kuglice,
  //kuglica se stvarno nalazi na +3
  x=tocnaudalj-19.0; //19.5 je udaljenost od senzora do vrha kotaca, ovime varijabla x
  //postaje ili pozitivna ili negativna ovisno o poziciji kuglice na kotacu
  beta_tocni=atan(x/(17.50+3.0)); //17.5 je radijus kola, 3 je radijus kuglice.
  //tangens bete je udaljenost nasuprotne stranice x kroz prilezecu stranicu
  //koja je jednaka radijusu kotaca + radijusu kuglice
  betaom=1000.0*(beta_tocni-beta_zadnji)/(trenutno_vrijeme-zadnje_vrijeme);
  //kutna brzina kuglice je razlika vrijednosti
  //bete u dva mjerenja kroz proteceno vrijeme puta 1000 da bi se dobilo rad/s
  //resetiranje vrijednosti
  beta_zadnji=beta_tocni;
  zadnje_vrijeme=millis();
}
float mapiranje(float os,float a,float b,float c,float d)
{
  float f=(os-a)/(b-a)*(d-c)+c;
  return f;
}
```

Slika 7.3. Funkcija mjerenje()

7.1.4. SETUP I LOOP

U *setup()* dijelu programa obično se definiraju stanja određenih varijabli, jesu li određeni pinovi ulazni ili izlazni, njihova stanja (HIGH ili LOW), pokreće se serijska veza itd.

U *loop()* dio programa se ubacuje kod za koji se želi da se izvršava neprestano. U ovom slučaju *loop* sadrži digitalni regulator. Na početku svakog *loop*-a će se provjeravati da li je prošao neki period vremena od zadnjeg mjerenja, tj. od zadnji put kada je izvršena funkcija *mjerenje()*. Ako je taj period prošao, izvodi se *mjerenje()* te se sa dobivenim iznosima varijabli stanja i njihovim umnoškom sa pripadajućim pojačanjima dobivenih iz simulacija izračunava moment. Tada se dobiveni moment ograničava na određene vrijednosti i preko funkcije *mapiranje()* mapira u određene okretaje po sekundi. Dobiveni okretaji po sekundi se onda uvrštavaju u izraz (7.2.) da bi se izračunala potrebna vrijednost *cmr* sa kojom se onda mijenja vrijednost od *timer1* a time i brzina rotacije motora. Ovisno o poziciji loptice dobivene iz *mjerenje()* mijenja se stanje od *directionPin*, a time i smjer rotacije motora. Tada se ponovno iznova izvršava početak *loop-a* i provjeravanje je li prošao neki period vremena od zadnjeg mjerenja. *Loop* je prikazan na Slika 7.4.

```
120 void loop() {
121   trenutno_vrijeme=millis(); //svaki loop novo trenutno vrijeme,
122   //millis() milisekunde od zadnje reseta procesora
123   //ako je proslo period_vremena od zadnjeg mjerenja
124   if ((trenutno_vrijeme-zadnje_vrijeme)>=period_mjerenja){
125     mjerenja(); //odradi mjerenje
126     //odredi moment iz umnoska pojacanja i vrijednosti mjerenja
127     moment=(K1*alfa_om + K2*beta_tocni + K3*betaom);
128     moment=constrain(abs(moment),0.5,3.5);
129     os=mapiranje(abs(moment),0.5,3.5,0.2,0.6);
130     if (x>0.6){
131       mijenjanje_timera(izracun_cmr(os));
132       smjer=0;
133       digitalWrite(directionPin,HIGH);
134     }
135     else if (x<-0.6){
136       mijenjanje_timera(izracun_cmr(os));
137       smjer=1;
138       digitalWrite(directionPin,LOW);
139     }
140     else if (abs(x)<0.6){
141       mijenjanje_timera(izracun_cmr(0.005));
142     }
143   }
144 }
```

Slika 7.4. Digitalni regulator

7.2. REZULTATI EKSPERIMENTALNOG POSTAVA

S konstruiranim postavom i implementiranim regulatorom nisu dobiveni optimalni rezultati. Kuglica se u ravnotežnom položaju (na vrhu kotača) održava samo par sekundi i ulazi u nestabilnost uslijed bilo kakvog poremećaja. Kao uzroke navedenog ponašanja mogu se navesti: nedovoljno točno izrađena konstrukcija, šum u očitavanjima senzora te neoptimalni regulacijski algoritam.

Tako je otvorena mogućnost za daljnje nadograđivanje rada u smislu prerade nedostataka same konstrukcije, filtriranje očitavanja iz senzora, bolje napisani regulacijski algoritam te ostala poboljšanja koja će u konačnici dovesti rad do funkcionalnosti.

8. ZAKLJUČAK

Kroz rad je prikazana izvedba sustava kuglice na kotaču. Sustav kuglice na kotaču spada u inicijalno nestabilne i podupravljane sustave te je za njegovu izvedbu bilo potrebno primijeniti širok spektar znanja iz raznih područja znanosti: strojarstva, elektronike i računarstva.

Prvo je izveden dinamički model sustava te su provedene simulacije u programskom paketu MATLAB koje su dale zadovoljavajuće rezultate. U sljedećem dijelu izvedbe je izrađena konstrukcija te su ugrađene odabrane komponente i napisan regulacijski algoritam. Proveden je niz eksperimenata sa različitim pojačanjima regulatora i izmjenama konstrukcije te se do kraja pisanja ovog rada nije došlo do optimalnog ponašanja sustava što ne znači da do njega neće doći u budućnosti.

Iz prijašnjeg odlomka se može zaključiti da će se konačna izvedba i ponašanje realnog sustava uvijek razlikovati od rezultata simulacija zbog nesavršenosti izvedenog dinamičkog modela, raznih zanemarivanja, nedovoljno dobro napravljene i osmišljene konstrukcije i regulacijskog algoritma. Tako rezultate iz simulacija provedenih u bilo kojoj grani znanosti treba uvijek uzeti sa zadrškom te na svoju ruku procijeniti njihovu točnost i značenje jer će se oni gotovo uvijek razlikovati od realnog ponašanja.

To naravno ne treba biti obeshrabrenje za provođenje eksperimenata i izvedbu nekih realnih sustava jer se tijekom izvođenja nauči dosta novih stvari i osoba kroz pokušaje i pogreške neizmjerno proširi svoje znanje što i je cilj svakog obrazovanja.

LITERATURA

[1] Lagrangian mechanics, (n.d.), preuzeto s https://en.wikipedia.org/wiki/Lagrangian_mechanics, (10.1.2017.)

[2] Ming-Tzu Ho, Yi-Wei Tu and Hao-Shuan Lin, „*Controlling a Ball and Wheel system using full-state-feedback linearization*“, 2009., IEEE Control Systems Magazine

[3] State-space representation, (n.d.), preuzeto s https://en.wikipedia.org/wiki/State-space_representation, (11.1.2017.)

[4] Šitum Ž., Podloge s predavanja iz kolegija „Računalne simulacije“

[5] Stepper motor, (n.d.), preuzeto s https://en.wikipedia.org/wiki/Stepper_motor, (11.1.2017.)

[6] ST6018L3008-A datasheet, 2016., preuzeto s http://en.nanotec.com/fileadmin/files/Datenblaetter/Schrittmotoren/ST6018/L/ST6018L3008-A_neu.pdf, (20.1.2017.)

[7] ST6018L3008-A, (n.d.), preuzeto s http://www.motionusa.com/nanotec/steppermotors/SC60.png, (20.1.2017.)

[8] THB6064AH datasheet, 2011., preuzeto s http://www.massmind.org/images/massmind/THB6064AH_ENG.pdf, (2.2.2017.)

[9] THB6064AH, (n.d.), preuzeto s http://i.ebayimg.com/images/g/fUcAAOSwbqpTwkJE/s-l500.jpg, (2.2.2017.)

[10] Proximity sensor, (n.d.), preuzeto s https://en.wikipedia.org/wiki/Proximity_sensor, (10.2.2017.)

[11] GP2Y0A21YK0F, (n.d.), preuzeto s https://a.pololu-files.com/picture/938a294daf56169ef35, (10.2.2017.)

[12] GP2Y0A21YK0F datasheet, 2006., preuzeto s http://www.sharpsma.com/webfm_send/1489, (10.2.2017.)

[13] Position sensitive device, (n.d.), preuzeto s https://en.wikipedia.org/wiki/Position_sensitive_device, (10.2.2017.)

[14] Arduino Nano, (n.d.), preuzeto s http://www.arduino.org/media/k2/galleries/82/A000005-Arduino-Nano-2tri.jpg, (11.2.2017.)

[15] ATMega328P datasheet, 2015., preuzeto s http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf, (11.2.2017.)

[16] Vuković L., Šljivo A., „Snimanje statičke karakteristike IR senzora udaljenosti“, 2013.

[17] ST6018L3008-A torque curve , (n.d.), preuzeto s http://en.nanotec.com/products/541-st6018-stepper-motor-nema-24/, (12.2.2017.)

[18] Arduino Timer Interrupts, (n.d.), preuzeto s http://www.instructables.com/id/Arduino-Timer-Interrupts/?ALLSTEPS, (13.2.2017.)

PRILOZI

1. CD-R

2. MATLAB KOD

Slijedi dio MATLAB koda koji je korišten za izvođenje simulacija sustava kuglice na gredi.

```
clc
clear all
close all


%Parametri sustava
r=0.031; %radijus loptice [m]
R=0.175; %radijus kotaca [m]
m=0.15; %masa loptice [kg]
M=0.6; %masa kotaca [kg]
I2=2/5*m*r^2; %moment inercije loptice [kgm^2]
I1=0.01029318+8.4e-5;
g=9.81; %gravitacijsko ubrzanje [ms^-2]

%prostor stanja
A1=I2*(R/r + (R/r)^2);
B1=I2*(1+2*R/r + (R/r)^2);
C1=r+R;
D1=I1+I2*(R/r)^2;


%parametri
a32=(m*g*C1)/(B1+m*C1^2);
b11=1/I1;
b31=A1/((B1+m*C1^2)*I1);

A=[0 0 0; 0 0 1; 0 a32 0];
B=[b11; 0; b31];

C=eye(3);
D=zeros(3,1);

%Tezinske matrice Q i R
q1=(1/(4*pi))^2;  %max brzina kola je 2o/s
q2=(1/(pi/12))^2;  %max pozicija loptice je 15 stupnjeva
q3=(1/(pi/6))^2;  %max brzina loptice je 30 stupnjeva po sekundi

Q=diag([q1,q2,q3],0);
R=[1];

%LQR regulator
K=lqr(A,B,Q,R);
Ac=(A-B*K);
Bc=B;
Cc=C;
Dc=D;
```

```
t=0:0.01:2.5;
u=0*ones(size(t));

%Pocetni uvjeti
X0=[0 -0.2 0];

[y,x]=lsim(Ac,Bc,Cc,Dc,u,t,X0);
plot(t,x(:,1),t,x(:,2),t,x(:,3),'linewidth',2)
legend('kutna brzina kotaca','kut pomaka kuglice','kutna brzina kuglice');
xlabel('vrijeme [s]'),ylabel('kut pomaka kuglice [rad]')

%Plotanje
sim('simulacija_sustava');
figure
plot(t,xsim(:,2),t,xsim(:,3),'linewidth',2);
xlabel('vrijeme [s]'),ylabel('kut pomaka kuglice [rad], kutna brzina
kuglice [rad/s]')
legend('kut pomaka kuglice','kutna brzina kuglice');
axis([0 0.5 -10 2]);

sim('simulacija_uz_regulator');
figure
plot(t,xsim(:,1),t,xsim(:,2),t,xsim(:,3),'linewidth',2);
legend('kutna brzina kotaca','kut pomaka kuglice','kutna brzina kuglice');
xlabel('vrijeme [s]'),ylabel('kut pomaka kuglice [rad]')

%Provjera uvjeta kontakta
N=m*g*cos(xsim(:,2))-m*(r+r)*(xsim(:,3)).^2;
figure
plot(t,N,'linewidth',2)
legend('Kontaktna sila izmedju kuglice i kotaca');
xlabel('vrijeme [s]'),ylabel('Sila [N]')
```