

Šolski center Celje

Gimnazija Lava

# **PRIMERJAVA VOZEČEGA IN HODEČEGA ROBOTA**

RAZISKOVALNA NALOGA

Področje: mehatronika in robotika

**AVTOR**

Maks Jagodič, 4. a

**MENTORJA**

Karmen Kotnik, univ. dipl. inž.

Tomislav Viher, univ. dipl. org.

Mestna občina Celje, Mladi za Celje 2022

Celje, marec 2022

## **Zahvala**

Najprej se iskreno zahvaljujem mojima mentorjema in profesorjema informatike Karmen Kotnik in Tomislavu Viherju, za vso podporo in pomoč pri izdelavi raziskovalne naloge.

Seveda se zahvaljujem tudi mojima staršema, ki sta verjela vame, ko je bila to le ideja na papirju, sta me spodbujala in mi nudila finančno pomoč.

Zahvaljujem se profesorici slovenščine, Katarini Petač in profesorici angleščine, Andreji Vipotnik Ravnak, ki sta raziskovalno nalogo lektorirali.

## **Povzetek**

V nalogi sem preučil zgodovino robotov in različne vrste hodečih robotov, nato pa sem opisal postopek izdelave hodečega in vozečega robota. Predstavil sem način delovanja programov obeh robotov. Za izdelavo sem uporabil 3D tiskalnik in tudi CNC rezkalnik. Za nadzor obeh strojev sem uporabil Arduino, pri hodečem pa še Raspberry Pi. Napisal sem tudi Android aplikacijo, s katero ju lahko krmilim. Po tem, ko sem izdelal oba robota, sem ju med seboj primerjal in s pomočjo tega ovrgel ali pa sprejel hipoteze, ki sem si jih zastavil. Tako sem ugotovil prednosti enega in drugega robota. Ugotovil sem, da so vozeči stroji hitrejši, za razliko od hodečih pa se težje premikajo po neravnem terenu. Na koncu sem vse ugotovitve povzel in predstavil možne izboljšave obeh robotov.

**Ključne besede:** hodeči robot, vozeči robot, Python, 3D tisk, Arduino, Raspberry Pi

## **Abstract**

In this research paper I have studied the history of robots and different types of walking robots, and described the process of making a walking and driving robot. I have also described how the robots are programmed and how they work. I have used a 3D printer and a CNC milling machine to make them. For controlling the robots, I have used an Arduino to control both of them, and I also added a Raspberry Pi for controlling the walking robot. Besides this I have written an Android app that allows me to control the robots. After making both robots, I compared them to one another. Based on the results I have refuted or accepted the proposed hypotheses. By doing this I have found out the advantages and disadvantages of both robots. I have found that driving machines are faster, but unlike walking machines, they have a harder time moving on uneven terrain. Finally, I have summarized all the findings and presented possible improvements for both robots.

**Keywords:** walking robot, driving robot, Python, 3D printing, Arduino, Raspberry Pi

## **Kazalo vsebine**

<b>1 Uvod</b> .....	<b>1</b>
<b>2 Hipoteze</b> .....	<b>2</b>
<b>3 Zgodovina</b> .....	<b>2</b>
<b>4 Vrste hodečih robotov</b> .....	<b>4</b>
4.1 Enonožni roboti – monopodi .....	4
4.2 Dvonožni roboti .....	5
4.3 Štirinožni roboti .....	6
<b>5 Izdelava hodečega robota</b> .....	<b>6</b>
5.1 Teoretična pojasnitev delovanja .....	6
5.2 Prototip 1 .....	12
5.2.1 3D modeliranje .....	12
5.2.2 3D tiskanje .....	13
5.2.3 Krmiljenje – Android aplikacija .....	14
5.3 Prototip 2 .....	14
5.3.1 CNC rezkanje .....	15
5.3.2 Izdelava PCB .....	17
5.4 Programiranje robota .....	18
<b>6 Izdelava vozečega robota</b> .....	<b>20</b>
<b>7 Primerjava</b> .....	<b>22</b>
<b>8 Razprava</b> .....	<b>24</b>
<b>9 Zaključek</b> .....	<b>26</b>
<b>10 Viri in literatura</b> .....	<b>27</b>

## Kazalo slik

Slika 1: Robot ANYmal C podjetja ANYbotics.....	1
Slika 2: Da Vincijev »robot«.....	3
Slika 3: William Grey s svojim robotom.....	4
Slika 4: Primer enonožnega robota.....	5
Slika 5: Robot Atlas.....	5
Slika 6: Robot BigDog .....	6
Slika 7: Noga robota.....	7
Slika 8: Človeške kosti .....	7
Slika 9: Različne osi .....	7
Slika 10: Izpeljava formule IK .....	9
Slika 11: Izpeljava formule IK .....	10
Slika 12: Hoja žuželk.....	11
Slika 13: Prva skica .....	12
Slika 14: Prvi model noge .....	13
Slika 15: Prikaz programa SuperSlicer.....	13
Slika 16: Zaslonska slika Android aplikacije .....	14
Slika 17: Moja dopolnjena knjižnica za uporabo protokola I2C .....	15
Slika 18: Simulacija poti za CNC v programu Fusion 360 .....	16
Slika 19: CNC, ki reže material.....	16
Slika 20: Shema vezja iz programa KiCad.....	18
Slika 21: 3D render vezja, izdelan v programu KiCad.....	18
Slika 22: Hodeči robot.....	19
Slika 23: Vozeči robot .....	20
Slika 24: Hodeči in vozeči robot na 4-metrski poti.....	22
Slika 25: Hodeči in vozeči robot na travniku .....	23
Slika 26: Hodeči in vozeči robot na klančini.....	23
Slika 27: Robot Centauro .....	24
Slika 28: Robot Handel .....	25

## 1 Uvod

Z robotiko se ukvarjam od sedmega razreda osnovne šole, sodeloval sem na različnih tekmovanjih – od First Lego League do RoboCupJunior. Čeprav se ta tekmovanja med sabo zelo razlikujejo, imajo nekaj skupnega in to je, da uporabljajo vsi roboti za premikanje kolesa ali gosenice.

Večja podjetja, kot so Boston Dynamics, ki s posnetki svojih robotov navdušuje obiskovalce medmrežja, Agility Robotics, ANYbotics (slika 1) in druga vlagajo veliko denarja v razvoj hodečih robotov. Iz tega sem sklepal, da morajo imeti takšni roboti neke prednosti pred vozečimi roboti, zakaj bi jih drugače sploh razvijali. Ker pa nisem našel dobrega odgovora na to vprašanje in ker se ne morem upreti dobremu izzivu, sem se odločil, da bom izdelal robota obeh tipov in ju med sabo primerjal.



*Slika 1: Robot ANYmal C podjetja ANYbotics*

Za uspešno primerjavo si bom moral postaviti več hipotez. S hipotezami in dobljenimi rezultati si bom pomagal ugotoviti prednosti in slabosti obeh vrst robotov. Preučil bom različne vrste hodečih robotov in izdelal svojim zmožnostim najprimernejšo. Opisal bom razvoj obeh robotov, predstavili probleme, na katere bom naletel, in opisal njihove rešitve. Dotaknil se bom tudi zgodovine obeh vrst robotov.

Glavna metoda, ki jo bom uporabljali pri raziskovanju, bo eksperimentalna metoda.

## 2 Hipoteze

Že pred izdelavo obeh robotov sem predvideval, kaj bodo prednosti enega in drugega. Te sem želel dokazati s končnim poligonom, ki bo sestavljen tako, da bom lahko svoja predvidevanja ovrigel ali potrdil.

### **Hipoteza 1: Vozeči robot bo na ravnih tleh brez ovir hitrejši kot hodeči.**

Predvideval sem, da lahko motorji vozečega robota več časa nemoteno pospešujejo in zato tudi dosežejo večjo hitrost. Pri hodečem robotu pa morajo motorji neprestano menjati smer obračanja, kar pomeni, da težje doseže visoko hitrost.

### **Hipoteza 2: Hodeči robot se bo lahko premikal po grobem neravnem terenu, vozečemu robotu pa bo to predstavljalo težavo.**

Ideja te hipoteze je, da bo imel vozeči robot težave pri premagovanju ovir, kot so ležeči policaj, kamenje in podobno, saj se načeloma obnaša podobno kot avto, ki mu takšne ovire predstavljajo težavo. Hodeči robot pa bo lahko tovrstne ovire preprosto prestopil.

### **Hipoteza 3: Hodeči robot bo lahko premagal klančino z večjim naklonskim kotom.**

Sklepal sem, da se bo lahko vozeči robot povzpenjal po klančini, dokler ne bo statična komponenta sile teže večja od sile trenja med tlemi in gumami robota. Hodeči robot pa bo lahko premaknil svoje težišče, tako bo lahko dosegel večji naklon.

## 3 Zgodovina

Preden sem se lotil izdelave robotov, me je zanimala tudi zgodovina, saj lahko le z razumevanjem zgodovine razumemo sedanost in hkrati dobimo vpogled v prihodnost.

Ljudje so že v antičnih časih poskušali zgraditi stroje, ki bi opravljali delo sami. Eni prvih takšnih »robotov« so bili Arhitasovi umetni golobi, ki so delovali na vodno paro. V antiki je deloval tudi Heron, grški matematik in izumitelj, ki je ustvaril številne avtomatizirane naprave in opisal stroje, ki jih poganjajo zračni tlak, para in voda. Vidnejši napredek se je zgodil v renesansi z Leonardom da Vincijem, ki je okoli leta 1495 skiciral in izdelal prvega človeku podobnega »robot« (slika 2). Menijo, da je »robot« zgradil po raziskavi človeške anatomije. Da Vincijev »robot« je bil zmožen premakniti roke, se usedi in premikati glavo in usta.





*Slika 2: Da Vincijev »robot«*

Besedo robot je prvič uporabil češki pisatelj Karel Čapek v drami R. U. R (češ. Rossumovi Univerzální Roboti). Čapek je tudi zapisal, da mu je besedo robot predlagal njegov brat Jožef Čapek, sam pa je želel uporabiti besedo labori. V drami predstavi Čapek idejo o kreaciji umetnega človeka brez duše, ki je opravljal nevarna dela. To je podobno današnji predstavi o tem, kaj je robot. V Slovarju slovenskega knjižnega jezika je beseda robot definirana takole:

robót -a m (ô)

- 1. elektronsko vodena naprava, ki enakomerno opravlja vnaprej programirana, pogosto človekovemu zdravju škodljiva dela*

Prvi takšni napravi je izdelal britanski nevrofiziolog William Grey Walter med letoma 1948 in 1949 in ju poimenoval Elmer in Elsie (slika 3). Z njima je želel dokazati, da je delovanje možganov odvisno od povezave posameznih delov možganov. Robota sta imela svetlobni senzor, ki je bil povezan z motorjema. To jima je omogočala t. i. fototaksija ali gibanje proti svetlobi. Walterjeva robota sta delovala analogno, saj je Walter menil, da lahko le tako posnemamo delovanje možganov. Njegovi sodobniki Alan Turing, John von Neumann in Norbert Wiener pa so zagovarjali uporabo digitalne tehnologije. Prvega digitalnega robota z imenom Unimate je izumil George Devol leta 1954. Unimate je bil robotska roka, ki je v avtomobilskih tovarnah varil dele ali pa zlagal težje dele, ukazi za njegovo delovanje pa so bili shranjeni na magnetnem bobnu.



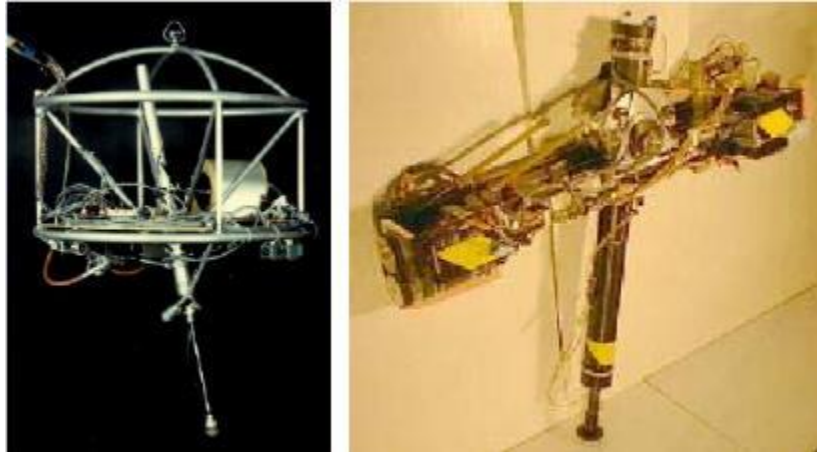
*Slika 3: William Grey s svojim robotom*

Prvi avtonomni hodeči robot sta razvila Bob McGhee in Andrew Frank leta 1966. Računalnik je nadziral štirinožnega robota, ker pa je imel le dva motorja na vsaki nogi, se je lahko premikal le v ravni črti in se ni mogel obrniti. Kljub temu je lahko hodil na dva različna načina (Robot, 2022).

## **4 Vrste hodečih robotov**

### **4.1 Enonožni roboti – monopodi**

Enonožni roboti se premikajo s pomočjo skokov, zato jih lahko primerjamo s kengurujem ali zajcem (slika 4). Zaradi svoje zgradbe so ti roboti nestabilni, kar rešujemo z dobrim sistemom za vzdrževanje ravnotežja. Na prvi pogled se zdi tovrstni roboti neuporabni in res jih ne uporabljajo veliko. Področje, na katerem bi bili uporabni, je raziskovanje različnih nebesnih teles, pri čemer bi imeli zaradi majhne gravitacije vozeči roboti težave s premikanjem. Primer takšne uporabe je ruski Skakač (rus. ПРОП-Φ), ki je leta 1998 pristal na luni Phobos.



*Slika 4: Primer enonožnega robota*

## 4.2 Dvonožni roboti

Dvonožni roboti so večinoma modelirani po človekovem izgledu. Čeprav se nam zdi, da smo ljudje stabilni, ko stojimo pri miru, v resnici podzavestno uravnavamo svoje telo. Zaradi tega imajo tudi dvonožni roboti (podobno kot enonožni) težave z ravnotežjem. Najpreprostejša rešitev tega problema so veliki podplati, vendar postane zaradi tega robot manj mobilni. Zato uporablja večina dvonožnih robotov žiroskop in senzorje za pospešek, s pomočjo katerih uravnava ravnovesje.

Za razliko od enonožnih robotov poznamo več primerov uporabe dvonožnih. Eden takšnih primerov je robot Atlas (slika 5), ki ga je razvilo podjetje Boston Dynamics, da bi pomagal reševalcem v nevarnih situacijah in pri upravljanju težkega orodja.



*Slika 5: Robot Atlas*

### 4.3 Štirinožni roboti

Štirinožni roboti so stabilnejši kot dvonožni, saj se lahko premikajo tako, da imajo naenkrat na tleh tri noge. Tudi ko so pri miru, so stabilnejši in jim ni potrebno vzdrževati ravnotežja. Nekateri štirinožni roboti imajo tudi nižje težišče, kar še dodatno pripomore k njihovi stabilnosti.

Robota BigDog (slika 6) je financirala ameriška Agencija za napredne obrambne raziskovalne projekte (DARPA). Z njim so želeli prevažati tovor po terenu, ki je neprehoden za običajna vozila (Silva, Tenreiro Machado, 2007).



*Slika 6: Robot BigDog*

## 5 Izdelava hodečega robota

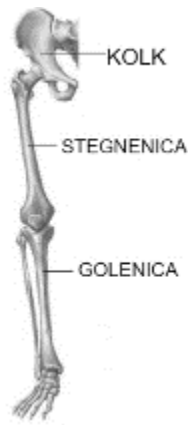
Po preučevanju različnih vrst hodečih robotov sem se odločil, da bom izdelal šestnožnega robota, saj bo zaradi števila nog zelo stabilen. Drugi dejavnik pa je bila tudi cena, saj potrebujem za vsako nogo najmanj tri motorje, vsaka dodatna noga pa naredi robota tudi kompleksnejšega. Tako je imel šestnogi robot dobro razmerje med ceno, kompleksnostjo in stabilnostjo.

### 5.1 Teoretična pojasnitev delovanja

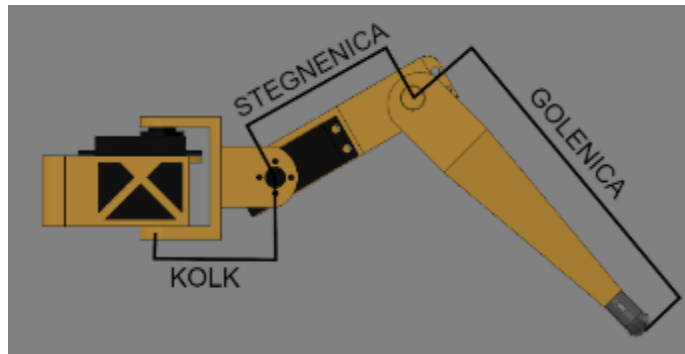
Premikanje hodečega robota lahko opišemo z direktno in inverzno kinematiko. Pri direktni kinematiki moramo podati vrednost, za koliko se naj premakne vsak robotov sklep, da bo robot dosegel želeno pozicijo.

Inverzna kinematika (v nadaljevanju IK) pa je matematični postopek, s katerim lahko iz želene končne pozicije robota izračunamo vrednosti premikov posameznih sklepov robota, da doseže robot to pozicijo. Formula IK je odvisna od konstrukcije in zahtev robota.

Moj robot ima šest nog, vsaka noga pa ima po tri sklepe, ki jih predstavljajo servo motorji. Razdalje med motorji se ne spreminjajo. Zaradi podobnosti hodečega robota in živali sem si pri reševanju IK pomagal tudi z anatomijo, zato sem te razdalje poimenoval po človeških kosteh. Na sliki 7 je prikazano njihovo poimenovanje, slika 8 pa prikazuje te kosti pri človeku.

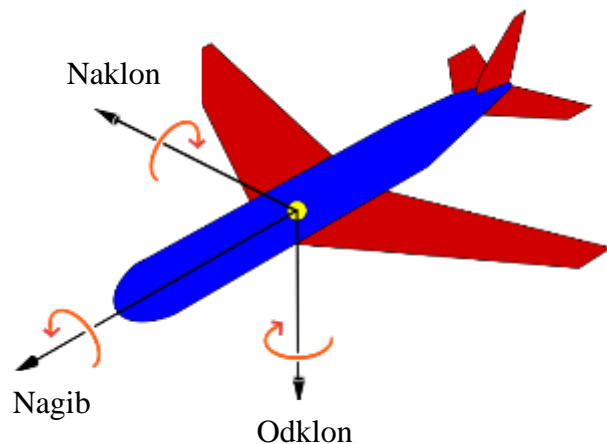


Slika 8: Človeške kosti



Slika 7: Noga robota

Ljudje in živali se lahko premikamo po x, y in z osi ali po katerikoli kombinaciji med njimi. Poleg tega pa poznamo še tri osi, po katerih se lahko premikamo, te so nagib (ang. roll), naklon (ang. pitch) in odklon (ang. yaw). Slika 9 prikazuje vseh šest osi. Da bi dosegel optimalno premikanje robota, sem moral v svoj matematični model IK vključiti vseh šest osi.



Slika 9: Različne osi

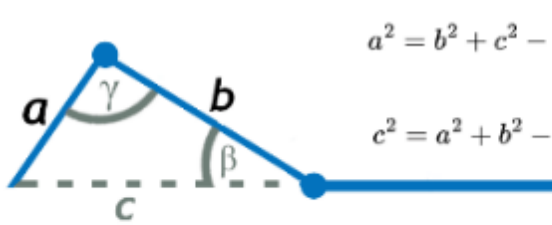
Čeprav šestnogi robot ni nekaj novega in že obstajajo programi z rešenim modelom IK, sem želel formule za premike posameznega motorja noge izraziti sam, saj bom le tako lahko razumel njegovo delovanje. Tako sem se lotil reševanja IK s pomočjo poenostavljenih skic noge robota in z osnovnim znanjem geometrije. Najprej sem si narisal nogo v pokrčenem položaju, ko robot ni dvignjen, in izrazil kota prvih dveh sklepov. Nato je sledila podobna skica, le da je bil tokrat robot dvignjen za poljubno razdaljo in sem na podlagi nove skice dopolnil prejšnji dve formuli. Tako sem postopoma dopolnjeval formule. Na slikah 10 in 11 je razviden cel postopek reševanja.

Ko je robot na tleh.

a = golenica

b = stegenica

c = zelena razdalja



$$a^2 = b^2 + c^2 - 2bc \cos \beta \Rightarrow \beta = \arccos \left( \frac{b^2 + c^2 - a^2}{2bc} \right)$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma \Rightarrow \gamma = \arccos \left( \frac{a^2 + b^2 - c^2}{2ba} \right)$$

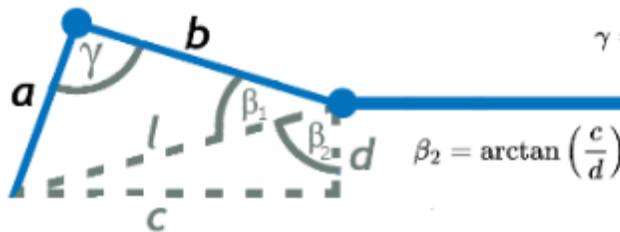
Ko se robot dvigne.

d = zelena višina

$$l = \sqrt{c^2 + d^2}$$

$$\gamma = \arccos \left( \frac{a^2 + b^2 - l^2}{2ba} \right) \Rightarrow$$

$$\gamma = \arccos \left( \frac{a^2 + b^2 - c^2 - d^2}{2ba} \right)$$



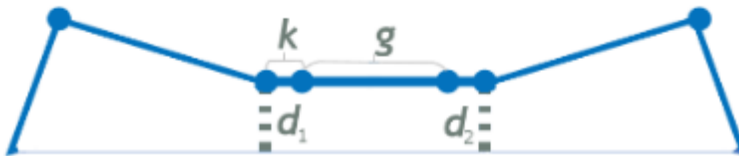
$$\beta_2 = \arctan \left( \frac{c}{d} \right) \quad \beta_1 = \arccos \left( \frac{b^2 + l^2 - a^2}{2bl} \right) \Rightarrow$$

$$\beta_1 = \arccos \left( \frac{b^2 + c^2 + d^2 - a^2}{2b\sqrt{c^2 + d^2}} \right)$$

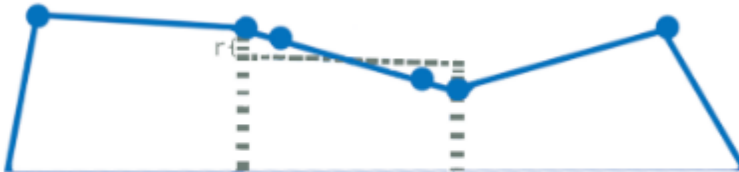
$$\beta = \arccos \left( \frac{b^2 + c^2 + d^2 - a^2}{2b\sqrt{c^2 + d^2}} \right) + \arctan \left( \frac{c}{d} \right)$$

Ko robot stoji.

k = kolk, g = dolžina trupa



r = zelen nagib

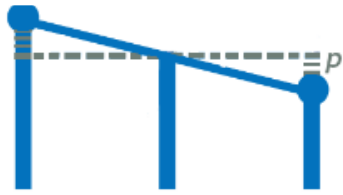


$$\beta = \arccos \left( \frac{b^2 + c^2 + (d \pm r)^2 - a^2}{2b\sqrt{c^2 + d^2}} \right) + \arctan \left( \frac{c}{(d \pm r)} \right)$$

$$\gamma = \arccos \left( \frac{a^2 + b^2 - c^2 - (d \pm r)^2}{2ba} \right)$$

Slika 10: Izpeljava formule IK

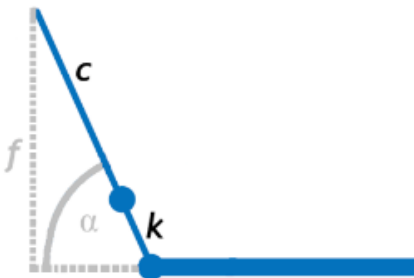
$p$  = želen naklon



$$\gamma = \arccos \left( \frac{a^2 + b^2 - c^2 - (d \pm r \pm p)^2}{2ba} \right)$$

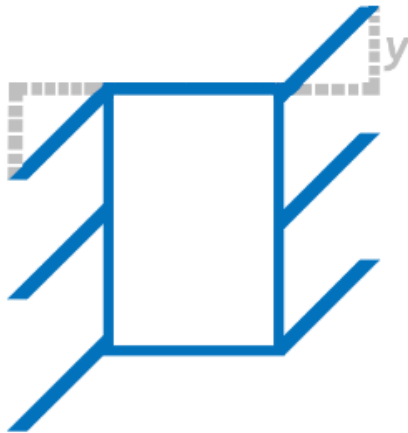
$$\beta = \arccos \left( \frac{b^2 + c^2 + (d \pm r \pm p)^2 - a^2}{2b\sqrt{c^2 + d^2}} \right) + \arctan \left( \frac{c}{(d \pm r \pm p)} \right)$$

$f$  = želen premik



$$\alpha = \arcsin \left( \frac{f}{c + k} \right)$$

$y$  = želen odklon



$$\alpha = \arcsin \left( \frac{f \pm y}{c + k} \right)$$

Slika 11: Izpeljava formule IK

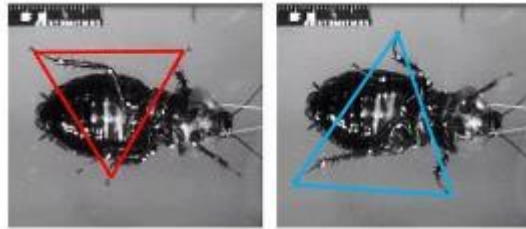


$$\beta = \arccos\left(\frac{\text{stegnenica}^2 + \text{želena razdalja}^2 + (\text{želena višina} \pm \text{želen nagib} \pm \text{želen naklon})^2}{2 * \text{stegnenica} * \sqrt{\text{želena razdalja}^2 + (\text{želena višina} \pm \text{želen nagib} \pm \text{želen naklon})^2}}\right) + \arctan\left(\frac{\text{želena razdalja}}{\text{želena višina} \pm \text{želen nagib} \pm \text{želen naklon}}\right)$$

$$\gamma = \arccos\left(\frac{\text{stegnenica}^2 + \text{golenica}^2 - \text{želena razdalja}^2 - (\text{želena višina} \pm \text{želen nagib} \pm \text{želen naklon})^2}{2 * \text{stegnenica} * \text{golenica}}\right)$$

$$\alpha = \arctan\left(\frac{\text{želen premik} + \text{želen odklon}}{\text{želena razdalja} + \text{kolk}}\right)$$

Le pravilno zaporedje premika nog lahko povzroči premik robota v želeno smer. Pri izdelavi robota in reševanju IK sem se zgledoval tudi po žuželkah, saj imajo, tako kot moj robot, šest nog. Po opazovanju žuželk v naravi sem ugotovil, da se večina žuželk premika v obliki trikotnika (ang. tripod gait), svojo teorijo pa sem preveril



Slika 12: Hoja žuželk

tudi v strokovni literaturi, v delu *Biologically Inspired Approaches for Locomotion, Anomaly Detection and Reconfiguration for Walking Robots*.

Večina žuželk najprej dvigne levo sprednjo in zadnjo nogo, hkrati pa dvigne desno sredinsko nogo, z ostalimi nogami se premakne naprej in noge spusti nazaj na tla, nato pa zamenja strani. Noge, ki so bile prej na tleh, so sedaj v zraku. Ta vzorec se nato ponavlja. Slika 12 prikazuje takšno premikanje žuželke.

Med raziskovanjem, kako bi se lahko premikal moj robot, sem naletel še na dva načina premikanja. Pri prvem bi lahko robot premikal eno nogo za drugo (ang. wave gait). Pri tem načinu hoje robot dvigne in premakne nogo eno za drugo, prvo na levi in nato na desni strani. Ta način hoje je zelo stabilen, saj je naenkrat dvignjena le ena noga, ampak je zaradi tega tudi počasen. Drugi način (ang. ripple gait) pa sledi zaporedju leva sprednja in desna srednja, potem desna sprednja in leva zadnja, za tem pa leva srednja in desna zadnja (Jakimovski, 2011).

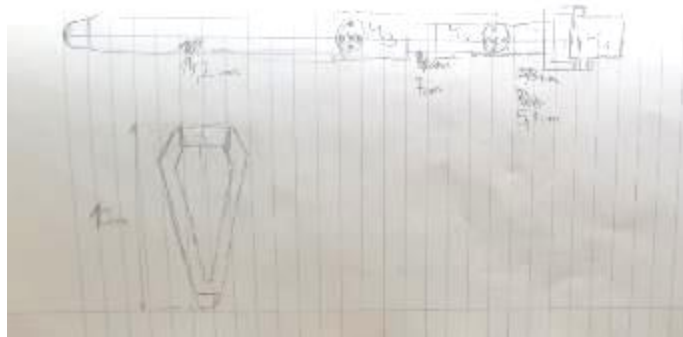
Premikanje v obliki trikotnika sem dosegel tako, da sem robotu s krmilnikom poslal želen premik v x in y smer, te vrednosti sem nato vstavil v formule IK kot želen premik in želen odklon ter izračunal nove vrednosti kotov, za katere se morajo premakniti noge, da se premaknejo za ti razdalji.

## 5.2 Prototip 1

Pri prvem prototipu sem uporabil Arduino Mega 2560. Zanj sem se odločil, saj sem imel že pred izdelavo raziskovalne naloge nekaj izkušenj z njegovim delovanjem. Pritegnila pa me je tudi velika podpora proizvajalca in skupnosti. Na Arduino sem priključil 18 servo motorjev MG995. Zanje sem se odločil, ker mi za razliko od navadnih DC motorjev omogočajo nadzor pozicije in so lažji ter manjši od koračnih motorjev. Njihova edina pomanjkljivost pa je, da se lahko zavrtijo le za 180 stopinj. Sicer obstajajo servo motorji, ki nimajo te omejitve, ampak so dosti dražji. Robota napaja ena 11 V Li-Po baterija, ki lahko odda veliko toka, ki je potreben za napajanje vzporedno vezanih motorjev. Ker pa motorji delujejo na največ 6 V, moram uporabiti pretvornik DC-DC, ki napetost iz 11 V pretvori na 5 V, da ne uniči motorjev.

### 5.2.1 3D modeliranje

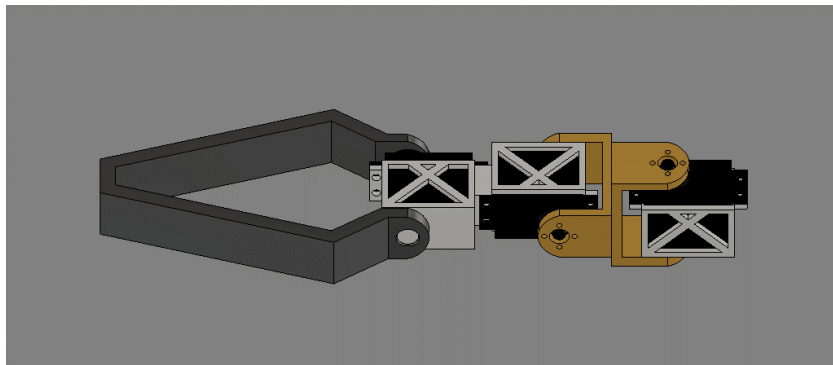
Preden sem se lotil izdelave 3D modela, sem navdih poiskal na spletu, kjer sem gledal že obstoječe šetnoge robote. Nato sem si skiciral, kako približno naj bi izgledala ena izmed nog. Skica je na sliki 13.



*Slika 13: Prva skica*

Ko sem bil zadovoljen s skico, sem se lotil izbire komponent in 3D modeliranja. Ker sem čakal, da prispejo motorji, sem njihove 3D modele poiskal na spletu in jih nato uvozil v program za modeliranje. Odločil sem se za uporabo programa Fusion 360, saj je ta program brezplačen za osebno uporabo, hkrati pa se ga je bilo preprosto naučiti.

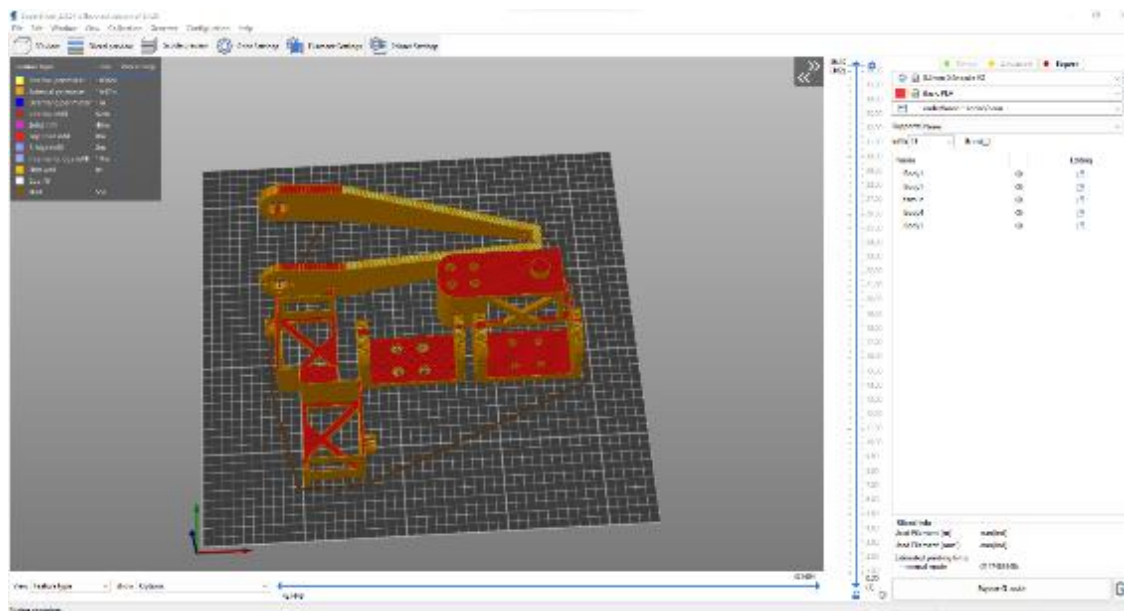
Eden izmed ciljev pri modeliranju je bil, da naj bo ogrodje robota čim preprostejše. Ker sem nameraval ogrodje 3D tiskati, sem pazil tudi na to, da bi potreboval čim manj podpornega materiala, hkrati pa sem tudi poskušal narediti ogrodje čim lažje, saj bi tako uporabil manj plastike in bi na motorje deloval manjši navor. Slika 14 prikazuje prvi model noge.



Slika 14: Prvi model noge

### 5.2.2 3D tiskanje

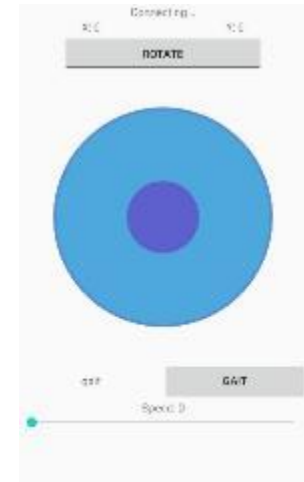
Dele, ki sem jih zmodeliral, sem nato izvozil v formatu 3mf in jih uvozil v SuperSlicer – program, ki ustvari navodila za delovanje mojega 3D tiskalnika. Na sliki 15 je prikazana zaslonska slika programa med delovanjem. Ker deli ne bodo izpostavljeni visoki temperaturi ali pa veliki obremenitvi, sem se odločil, da bom uporabil plastiko PLA, saj jo je najlažje tiskati. Uporabil sem 0.2 mm resolucijo, 0.4 mm šobo in 20 % polnila. Ker 3D tiskalnik deluje tako, da stopljeno plastiko polaga eno plast na drugo, vse strani končnega izdelka niso enako močne. Del je šibkejši na stičišču različnih plasti, in sicer v smeri osi z glede na položaj tiskanja. To sem upošteval pri delih, pri katerih sem moral v plastiko priviti vijak, saj bi lahko to povzročilo ločitev posameznih plasti.



Slika 15: Prikaz programa SuperSlicer

### 5.2.3 Krmiljenje – Android aplikacija

Za nadzor robota sem pri prvem prototipu uporabil Android aplikacijo, ki sem jo sam napisal v programu Android studio. Aplikacija je pošiljala podatke za premik preko bluetootha, zato sem moral na Arduino priključiti Bluetooth vmesnik. Aplikacija je zasnovana tako, da sta na sredini zaslona dva kroga, kot je vidno na sliki 16. Manjši začne slediti prstu, ko se dotaknemo zaslona. Večji krog pa predstavlja največji odklik od središča. Glede na razliko med začetno in končno pozicijo manjšega kroga lahko izračunam, za koliko se naj robot premakne. Pod tema krogoma je tudi drsnik za nadzor hitrosti robota. Čeprav je aplikacija delovala, z njo nisem bil zadovoljen, saj način, s katerim sem prenašal podatke med Arduinom in aplikacijo, ni dovolil prenosa več kot 255 različnih vrednosti. Zato sem jo v naslednjem prototipu opustil.



Slika 16: Zaslonska slika Android aplikacije

## 5.3 Prototip 2

V drugem prototipu sem Arduino zamenjal z mini računalnikom Raspberry Pi Zero, saj lahko njegov procesor doseže 1 GHz, medtem ko lahko Arduino doseže le 16 MHz. Poleg tega ima Raspberry Pi tudi več pomnilnika. Omogoča pa mi tudi lažjo povezavo Bluetooth krmilnika, ki prevzame vlogo Android aplikacije za nadziranje robota. Odločil sem se za uporabo Xboxovega krmilnika, saj sem ga že imel. Ker pa Linux, operacijski sistem, ki ga uporabljam, nima uradnega gonilnika za ta krmilnik, sem moral uporabiti neuradni gonilnik, zaradi česar povezava med krmilnikom in Raspberry Pi ni najboljša.

Raspberry Pi mi v prihodnosti omogoča tudi priključitev kamer za izvajanje prepoznave objektov. Ker potrebuje moj robot 18 servo motorjev, Raspberry Pi Zero pa ima le 4 pine, ki so primerni za priključitev servo motorjev, sem bil primoran povezati Raspberry Pi Zero in Arduino Mega 2560. Prvi izvaja vse kalkulacije IK in podatke posreduje Arduinu, ki ustrezno premakne posamezne motorje. Ker je Raspberry Pi tako kot vsak računalnik občutljiv na nenaden izpad elektrike, sem robotu dodal še prenosno baterijo (ang. power bank) z veliko kapaciteto za napajanje Arduina in Raspberry Pi. To mi nudi dovolj časa, da izklopim Raspberry Pi in preprečim poškodbe podatkov na SD kartici.

Za komuniciranje med Arduinom in Raspberry Pi sem uporabljali protokol I2C. Ta protokol deluje na principu nadrejeni/podrejeni (ang. master-slave), kot je razvidno iz imena, moramo določiti nadrejenega, ki lahko upravlja z večjim številom podrejenih. V našem primeru je nadrejeni Raspberry Pi, podrejeni pa Arduino. Podatki se prenašajo po dveh dvosmernih vodilih, SCL in SDA. Vodilo SCL (ang. serial clock line) prenaša takt, ki sinhronizira vse udeležence pogovora, določa pa tudi hitrost komunikacije, ponavadi od 100 kHz do 400 kHz.

SDA (ang. serial data line) pa, kot je razvidno iz angleškega imena, prenaša podatke med napravami. Vsaka naprava ima poseben naslov, tako je lahko povezanih med seboj 128 ali 1024 naprav, odvisno ali uporabljamo 7- ali 10-bitne naslove. Komunikacijo upravlja nadrejeni, ta generira takt in podrejenim pošilja različne ukaze in podatke. Hkrati pa lahko podrejeni tudi pošilja podatke nadrejenemu, ampak le, ko ta to zahteva. Podatki se pošiljajo bit po bit; komunikacije se začne s startnim bitom in konča s končnim bitom (I2C, 2022).

Tako Raspberry Pi kot Arduino imata že napisane knjižnice za uporabo protokola I2C. V primeru Raspberry Pi je to smbus, v primeru Arduina pa Wire.h. Kljub temu pa sem moral obe knjižnici za lažjo uporabo še dopolniti. Na sliki 17 je dopolnjena knjižnica, s katero lahko Arduino pošljem kakršen koli niz.

```
1 import smbus
2 import time
3
4 bus = smbus.SMBus(1)
5 address = 0x08
6
7 def writeData(value):
8     byteValue = StringToBytes(value)
9     bus.write_i2c_block_data(address,0x00,byteValue)
10    time.sleep(0.02)
11    return -1
12
13 def StringToBytes(val):
14     retVal = []
15     for c in val:
16         retVal.append(ord(c))
17     return
```

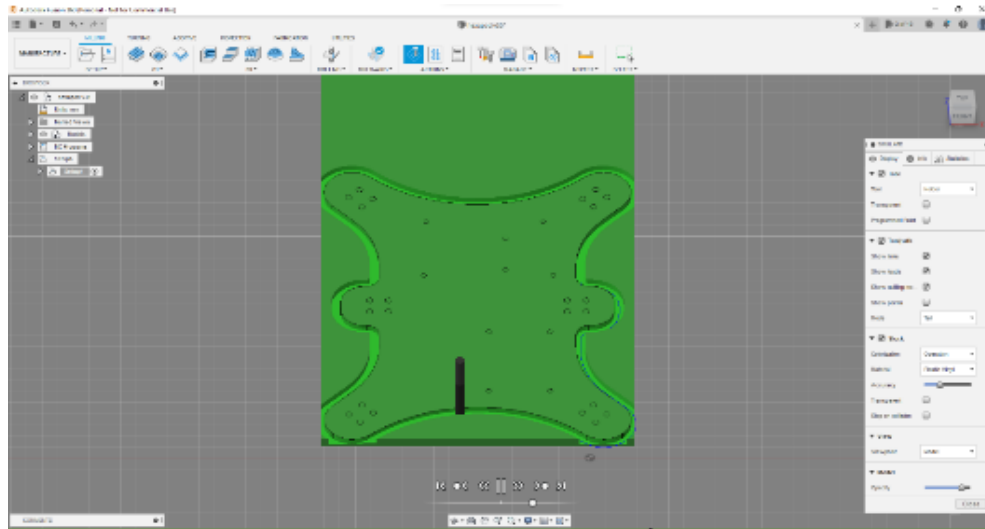
Slika 17: Moja dopolnjena knjižnica za uporabo protokola I2C

### 5.3.1 CNC rezkanje

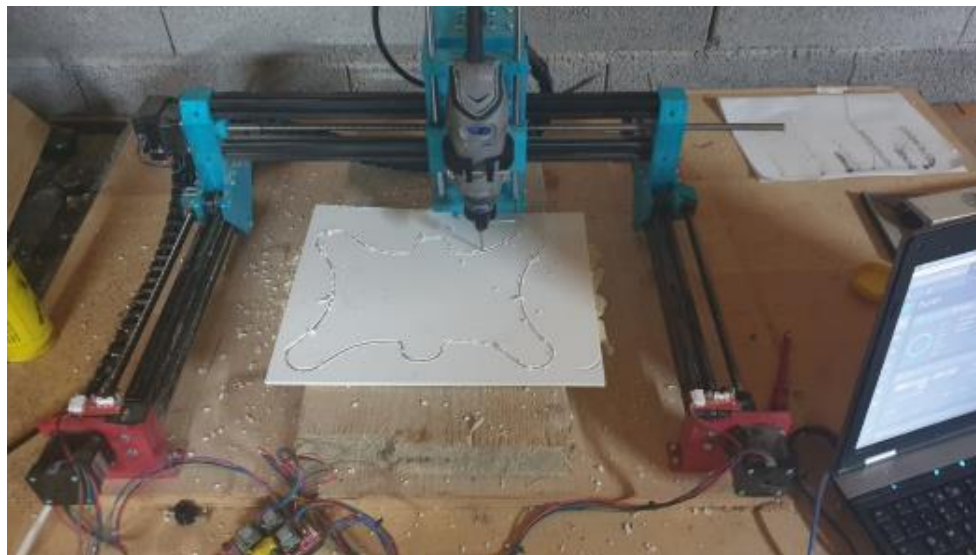
Pri prvem prototipu sem naletel na problem, da je bila plastika za trup preveč prožna, kar je naredilo robota premalo stabilnega. Ker je del za trup zavzel skoraj celotno površino mojega 3D tiskalnika in je porabil veliko plastike ter časa, sem se odločil, da bom naslednje prototipe izrezal na svojem rezkalniku CNC. Tudi tega sem izdelal sam. Pri gradnji sem se zgledoval po svojem 3D tiskalniku, saj delujeta oba na podoben način. Zaradi večjih obremenitev pri rezkanju CNC sem veliko pozornosti namenil trdnosti okvirja. Tako sem večinoma uporabil aluminijaste ali lesene dele, samo za povezovalne dele sem uporabil plastiko PETG, za rezanje materiala pa uporabljam ročni rezkalnik za les. Rezkalnik CNC je viden na sliki 19

S pomočjo Fusion 360 sem ustvaril navodila za CNC. Uporabil sem sveder s premerom 3 mm, hitrost rezkala 3000 obratov na minuto in hitrost rezanja 900 mm/min. Slika 18 prikazuje simulacijo poti CNC, ki reže ploščo ABS. Sprva sem nov trup izrezal iz 4 mm

debele plošče ABS, a je bila ta še kar preveč prožna, zato sem postopek ponovil. Tokrat sem uporabil 5 mm karbonsko ploščo. Ker se ne spoznam preveč na programiranje rezkalnika CNC in ker je karbon drag, sem trup izrezal ročno. Pri tem pa sem bil pozoren in uporabil ustrezno zaščito oči, nosil pa sem tudi masko FFP2 in rokavice, saj je lahko karbonski prah škodljiv za dihala in draži kožo.



*Slika 18: Simulacija poti za CNC v programu Fusion 360*

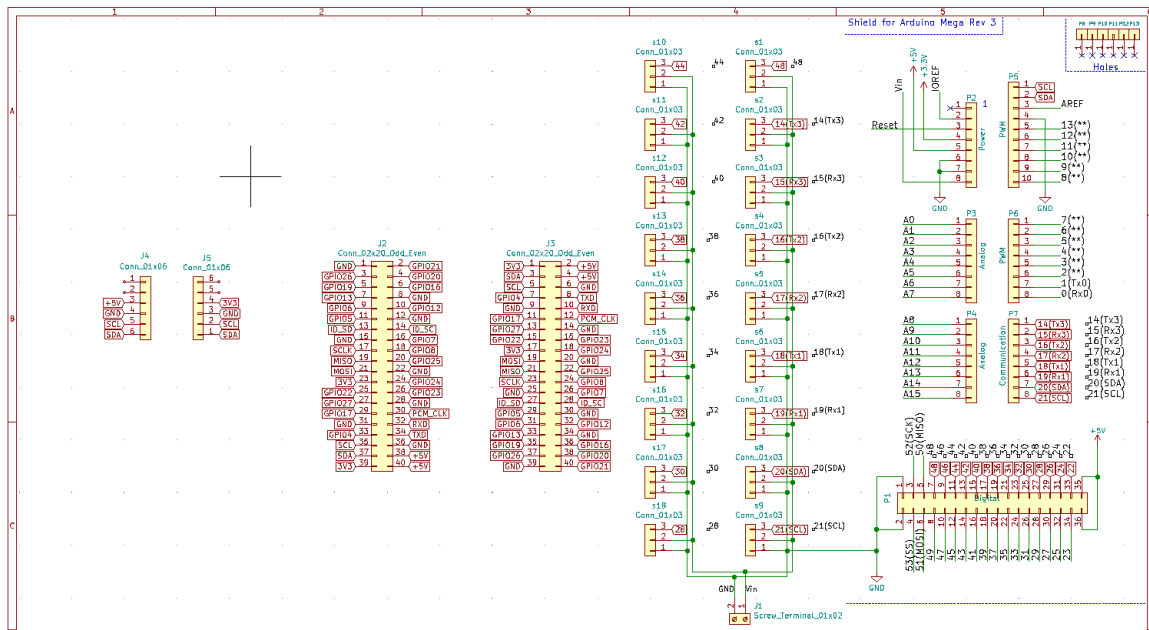


*Slika 19: CNC, ki reže material*

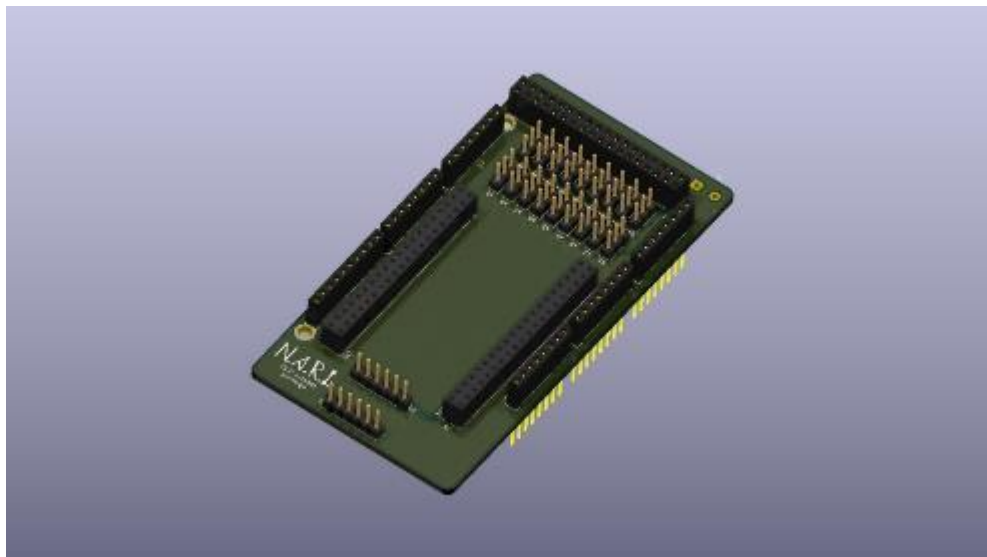
### 5.3.2 Izdelava PCB

Zaradi velikega števila žic in da bi bilo vse kompaktnije, sem se odločil, da bom naredil tudi vezje, ki bo skrbelo za povezavo med Arduinom in Raspberry Pi in mi omogočilo lažjo priključitev vseh 18 servo motorjev. Ker Arduino deluje na 5 V logični nivo, Raspberry Pi pa na 3.3 V, sem moral v vezje vključiti še dvosmerni logični pretvornik, ki lahko hkrati pretvori 5 V v 3.3 V in obratno. Vezje sem uporabil tudi za povezavo med servo motorji in baterijo.

Za izdelavo vezja sem uporabil odprtokodni program KiCad. Ker je bilo to moje prvo vezje, sem se poskušal izogniti uporabi komponent SMD. Najprej sem naredil shemo (slika 20), pri čemer mi je bila v pomoč shema Arduina. Nato pa sem moral razvrstiti komponente in jih povezati med seboj. Uporabil sem funkcijo avtomatske povezave, potem pa sem pogledal, če lahko katere povezave izboljšam. Dva sloja sta bila dovolj za povezavo vseh komponent, prvi sloj služi kot ozemljitveni sloj, drugi pa za povezavo ostalih komponent. Ko sem bil zadovoljen z vezjem, sem najprej izvozil 3D model vezja in ga uvozil v Fusion 360, kjer sem se prepričal, ali sem uporabil pravilne dimenzije. Končno vezje je vidno na sliki 21. Na koncu sem sledil navodilom na strani JLCPCB (<https://jlcpcb.com/>) in izvozil vse datoteke, ki so potrebne za izdelavo vezja. Te sem nato naložil na JLCPCB stran in čez dva tedna prejel vezje. Po prejemu sem spajkal vse komponente. Ugotovil sem, da sem naredil manjšo napako, in sicer sem za nadzor dveh motorjev uporabil dva pina, ki sta odgovorna za komunikacijo I2C, kar ne bi bil problem, če ne bi hkrati uporabljal tudi komunikacije I2C med Arduinom in Raspberry Pi. To je pomenilo, da je Arduino hitro preklapljal med vhodom, ko je potekalo komuniciranje I2C, in izhodom, ko je moral nadzirati motor, kar je pomenilo, da motorja nista ohranila svoje pozicije. Ta napaka je bila hitro popravljena, saj sem za nadzor teh motorjev samo uporabil druga dva pina.



Slika 20: Shema vezja iz programa KiCad



Slika 21: 3D render vezja, izdelan v programu KiCad

## 5.4 Programiranje robota

Pri prvem prototipu sem celotni program napisal v Arduinovem programskem okolju, v programskem jeziku C++. Vsak program za Arduino je sestavljen iz dveh delov. Prvi je `setup()`, ki se izvede le enkrat na začetku in v njem ponavadi definiramo, čemu služijo različni pini. Drugi del pa je neskončna zanka `loop()`. Tako se je program začel v delu `setup()` z definicijo vseh 18 motorjev, ki sem jim pripisal pine, ki bodo odgovorni za nadzor



posameznega motorja. Za preglednejšo kodo sem program razdelil v več funkcij in jih poklical v zanki `loop()`, ko je bilo to potrebno.

Ker sem v drugem prototipu (slika 22) dodal Raspberry Pi, sem moral večino programa napisati na novo. Poleg tega sem moral robota programirati v dveh delih, in sicer posebej Raspberry Pi in posebej Arduino. Raspberry Pi sem programiral v meni najbolj znanem programskem jeziku Python. Ker je Raspberry Pi čisto pravi računalnik s pravim operacijskim sistemom Linux, bi ga lahko priključil na zaslon in mu dodal še tipkovnico in programiral kar na njem samem. Vendar bi vsi dodatni kabli otežili testiranje delovanja robota, zato sem se odločil, da bom Raspberry Pi preko brezžične povezave povezal z računalnikom. Za to povezavo sem uporabil protokol SSH, ta omogoča varno povezavo med dvema oddaljenima računalnikoma. Poznati moramo IP naslov, uporabniško ime in geslo računalnika, s katerim se želimo povezati. Dostopamo lahko samo do tekstovnega uporabniškega vmesnika, kar pa mi ni predstavljalo težav, saj sem bil navajen uporabe terminala Linux. Napisal sem več posameznih programov, ki sem jih kot knjižnice nato uvozil v skupni program. Zaradi takšne delitve programa na manjše dele je končni izdelek lažje berljiv, v prihodnosti pa dovoljuje tudi lažjo uporabo te kode v kakšnem drugem projektu. Prvi podprogram skrbi za pošiljanje podatkov Arduino, drugi za reševanje IK modela in za pravilno zaporedje premikanja nog, tretji pa se odziva na ukaze, ki jih prejme od Bluetooth krmilnika. Ker je Raspberry Pi prevzel vlogo glavnega računalnika v robotu, je koda za Arduino postala preprostejša. Skrbi le za pravilno branje podatkov, ki jih pošilja Raspberry Pi, in se glede na prejete podatke pravilno odzove.



*Slika 22: Hodeči robot*

## 6 Izdelava vozečega robota

Kot sem omenil na začetku, sem tekmoval na različnih robotskih tekmovanjih, tako sem izdelal že veliko vozečih robotov, ki so bili vedno izdelani z uporabo Lego mindstorms ev3. Zato me je zanimalo, kako bi se takšen robot obnesel proti mojemu hodečemu robotu. Odločil pa sem se tudi za uporabo Arduina Uno s posebnim nastavkom EVShield, ki omogoča nadzorovanje motorjev in senzorjev Lego mindstorms ev3, ki sem jih že imel doma. Poleg tega pa mi Arduino omogoča uporabo boljših senzorjev in motorjev, ki jih, če bi uporabljal ev3 krmilnik, ne bi mogel uporabiti. Sama komunikacija med Arduinom in nastavkom poteka z uporabo protokola I2C.

Uporaba kock Lego mi je omogočala hitro izdelavo prototipov. Kljub temu se nisem mogel izogniti 3D tiskanju, saj sem moral natisniti ohišje za Arduino in nastavek. Ker nisem našel 3D modela za nastavek, sem uporabil optični čitalnik, da sem naredil posnetek vrha in dna nastavka. Ti sliki sem nato uvozil v Fusion 360 in obe tudi umeril. S pomočjo tega sem zmodeliral ohišje, ki sem ga uporabil, da sem lahko pritrdil Arduino in nastavek na ogrodje Lego, ki je sestavljeno iz dveh večjih Lego mindstorms ev3 motorjev in tretjega kolesa, ki se lahko premika v vse strani. Poleg tega uporabljam še Li-ion baterijo, ki je sestavljena iz treh celic, ki so vezane zaporedno. Skupna napetost te baterije je 10 V in je preko nastavka EVShield povezana z Arduinom. Robot je viden na sliki 23.



*Slika 23: Vozeči robot*

Ker morata biti oba robota vodljiva, da lahko med njima izvedem primerjavo, sem za nadzor vozečega robota uporabil Android aplikacijo, ki sem jo napisal za prvi prototip hodečega robota. To aplikacijo sem prilagodil potrebam vozečega robota. Odpravil pa sem tudi njeno pomanjkljivost, in sicer lahko sedaj pošiljam iz aplikacije kakršen koli niz, ki je manjši od 32 znakov, kar je za potrebe tega robota več kot dovolj.

Koda za Arduino je tudi pri tem robotu napisana v Arduinovem okolju. Nastavek EVShield ima že napisano knjižnico, ki mi omogoča preprost nadzor motorjev. Za komunikacijo z aplikacijo uporabljam Bluetooth vmesnik. Sam vmesnik pa podatke Arduino pošilja s serijsko komunikacijo, in sicer uporabljam protokol UART (ang. Universal Asynchronous Receiver-Transmitter). Pri tem protokolu se podatki prenašajo po dveh linijah, in sicer liniji za prenos podatkov (TX) in liniji za sprejem podatkov (RX). Na sprejemniku mora biti linija za sprejem podatkov povezana z linijo za prenos podatkov oddajnika. Linija za sprejem podatkov na oddajniku pa mora biti povezana z linijo za prenos podatkov na sprejemniku. Za razliko od I2C protokola UART ne uporablja taktne linije, ki bi sinhronizirala oddajnik in sprejemnik, zaradi česar oddajnik ne more biti prepričan, ali je sprejemnik dobil prave podatke. Zato razdeli podatek na točno določeno dolžino, poleg tega pošlje na začetku začetni bit in na koncu končni bit, sprejemnik pa potrди, ali je dobil vse podatke. Tako zagotovimo, da so prejeti podatki enaki kot oddani. UART ne dovoljuje komuniciranja med več kot dvema napravama. Poleg tega pa uporablja napol dvosmerno komuniciranje (ang. half duplex), kar pomeni, da lahko naprava le prejema ali oddaja podatke, ne more pa oddajati in sprejemati hkrati (Universal asynchronous receiver transmitter, 2022).

## 7 Primerjava

Po izdelavi obeh robotov je sledila primerjava med njima, s čimer sem želel dokazati, ali ovreči hipoteze. Robota sem tudi posnel.

### **Hipoteza 1: Vozeči robot bo na ravnih tleh brez ovir hitrejši kot hodeči.**

Za preverjanje te hipoteze sem meril čas, ki sta ga potrebovala robot, da sta opravila pot, dolgo 4 metre (slika 24). Tako sem lahko izračunal povprečno hitrost obeh robotov. Z vozečim robotom sem polje prevozil desetkrat in je v povprečju prevozil 4 metre v 11 sekundah, kar pomeni, da je njegova povprečna hitrost 0,36 m/s. Hodeči robot pa je potreboval kar 6 minut in 26 sekund ali 386 sekund, kar pomeni, da je njegova hitrost 0,01 m/s. Ker je bilo že v prvem poskusu razvidno, da je hodeči robot dosti počasnejši sem poskus ponovil le dvakrat. Torej lahko to hipotezo potrdim, saj je bil vozeči robot kar 40-krat hitrejši kot hodeči.



*Slika 24: Hodeči in vozeči robot na 4-metrski poti*

### **Hipoteza 2: Hodeči robot se bo lahko premikal po grobem neravnem terenu, vozečemu robotu pa bo to predstavljalo težavo.**

To sem testiral na travniku, saj ima veliko malih udrtin in izboklin, na katerih se lahko robot zatakne. Robota sta se naključno premikala po travniku (slika 25) eno minuto ali dokler se nista kam zataknila in poti nista mogla nadaljevati. Hodeči robot se je zlahka premikal po takšnem terenu, vozeči robot pa se je večkrat zataknil in na koncu tudi obtičal. Ta poskus sem nato opravil še desetkrat, a se rezultat ni spremenil. Tako lahko to hipotezo potrdim.



Slika 25: Hodeči in vozeči robot na travniku

**Hipoteza 3: Hodeči robot bo lahko premagal klančino z večjim naklonskim kotom.**

Izdelal sem premično klančino, da sem lahko spreminjal naklonski kot. Tega sem izračunal s formulo  $kot = \arcsin\left(\frac{višina}{52.5cm}\right)$ . Vse poskuse sem ponovil desetkrat, in sicer sem najprej višino klančine nastavljal na 10 cm, tako je bil naklonski kot 11 stopinj. Vozeči robot je to z lahkoto prevozil (slika 26), zato sem višino nastavljal na 20 cm ali 22 stopinj, tu pa je imel težave in ga ni mogel več prevoziti. Nato sem s hodečim robotom poskusil premagati klančino z 11 stopinjami, to mu je uspelo prehoditi le enkrat izmed desetih poskusov (slika 26). To je lahko vzrok njegove večje teže ali pa pomeni, da motorji niso dovolj močni, zato nisem poskušal na večjem naklonskem kotu. Ne glede na to hipoteze 3 ne morem potrditi.



Slika 26: Hodeči in vozeči robot na klančini

## 8 Razprava

Veliko Zemljinega površja je nedostopnega kakršnim koli vozečim se mehanizmom. Podobni problemi se pojavijo tudi na drugih planetih, zato so za takšne površine boljši hodeči roboti. Ti lahko prestopijo manjše ovire in se lahko tudi povzpnejo po stopnicah, česar večina vozečih robotov ne zmore. Največja pomanjkljivost hodečih robotov pa je njihova kompleksnost. Kot je tudi razvidno iz te raziskovalne naloge, je za izdelavo preprostega hodečega robota potrebno uporabiti veliko več motorjev, kar tudi vpliva na ceno, zato so ponavadi takšni roboti dražji. Potrebna je tudi izdelava kompleksnejšega algoritma za premikanje. S primerjavo obeh robotov pa sem ugotovil, da so hodeči roboti dosti počasnejši.

Zato se mi zdi smiselna uporaba vozečih robotov za dela, ki so točno določena in pri katerih lahko kontroliramo okolje, v katerem deluje robot. Tako lahko koristimo njihovo hitrost in se izognemo njihovim pomanjkljivostim. Hodeči roboti pa so bolj vsestranski, zato lahko uporabljamo en robot za različne namene. Na strani Boston Dynamics oglašujejo svoj štirinožni roboti Spot, kot primerne za pregledovanje rudarskih tunelov, območij z visoko napetostnim omrežjem in območij, kjer je veliko sevanja. Torej je Spot primeren za območja, ki so nevarna za človeka in kjer je nepredvidljiv teren.

Obstaja pa tudi možnost kombiniranja obeh mehanizmov premikanja, kar pomeni, da namesto podplata namestimo robotu kolo. To mu omogoči hitro premikanje na ravnih tleh, hkrati pa lahko robot preklopi na hojo, ko teren ne dopušča uporabe koles. Primer takšnega robota sta Handel (slika 28) podjetja Boston Dynamics in Centauro (slika 27), čigar razvoj financira Evropska unija.



*Slika 27: Robot Centauro*



*Slika 28: Robot Handel*

Moj hodeči robot je dosegel pričakovanja, saj je delal točno to, kar sem želel, a to še ne pomeni, da ga ne morem izpopolniti. Pri opazovanju robota med hojo sem ugotovil, da je robot skoraj pretežek za motorje in bi jih bilo smiselno zamenjati z motorji, ki imajo večji navor. Odpraviti bi bilo smiselno tudi napako na vezju, ki sem jo že omenil, in hkrati zamenjati način povezave med motorji in vezjem, saj je trenutna povezava s konektorji dupont nekoliko nezanesljiva. Čeprav Xboxov krmilnik deluje, ga moram vedno ob zagonu ročno povezati z Raspberry Pi in hkrati se po nekaj časa neaktivnosti sam izklopi, zato bi lahko naredil tudi svoj krmilnik. Robot lahko zaenkrat hodi le na en način, v prihodnosti pa bi program lahko dopolnil, da bi lahko izbral med vsemi tremi načini hoje, ki sem jih predstavil v poglavju 5.1. Zanimivo bi bilo tudi dodati žiro senzor, s pomočjo katerega bi bil trup robota vedno vzporeden s tlemi.

Tudi vozeči robot deluje točno tako, kot sem želel. Edina pomanjkljivost je bila slaba dokumentacija knjižnice za uporabo EVShield, ki je nekoliko otežila pisanje programa. Menim, da bi bilo smiselno poskusiti razviti svojo različico EVShielda, saj bi ga tako lažje prilagodil svojim potrebam, če bi robota želel uporabiti na tekmovanju, kot je RoboCupJunijor.

## 9 Zaključek

Z izdelavo teh robotov sem pridobil veliko praktičnega in tudi teoretičnega znanja programiranja, 3D modeliranja in tudi izdelave Android aplikacije. Najtežje se je bilo odločiti, kdaj sta robota dovolj dobra, saj je vedno nekaj, kar lahko popraviš ali izboljšaš, a časa ni nikoli dovolj. Z obema robotoma sem zelo zadovoljen in ju bom tudi v prihodnje poskušal še izpopolniti. Vse datoteke pa nameravam objaviti na strani GitHub, kjer bodo dostopne vsakomur, ki bi želel dopolniti moje delo.



## 10 Viri in literatura

- Aircraft principal axes. [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 9. feb. 2022 ;16:50] [Citirano 4. mar. 2022; 16.08] Dostopna na spletnem naslovu: [https://en.wikipedia.org/wiki/Aircraft\\_principal\\_axes](https://en.wikipedia.org/wiki/Aircraft_principal_axes)
- Arduino [Online]. Arduino. [Zadnja sprememba 2022] [Citirano 2. jan. 2022; 18.46] Dostopno na naslovu: <https://www.arduino.cc/>
- Elmer and Elsie (robots). [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 10. dec. 2021; 18:08] [Citirano 28. feb 2022; 17.03] Dostopna na spletnem naslovu: [https://en.wikipedia.org/wiki/Elmer\\_and\\_Elsie\\_\(robots\)](https://en.wikipedia.org/wiki/Elmer_and_Elsie_(robots))
- EVShield for Arduino Duemilanove or Uno. [Online]. Mindsensors. [Zadnja sprememba 1. maj 2018; 17:00] [Citirano 2. mar. 2022; 15.45] Dostopna na spletnem naslovu: <http://www.mindsensors.com/arduino/16-evshield-for-arduino-duemilanove-or-uno#:~:text=EVShield%20attaches%20directly%20to%20an,to%20attach%20existing%20Arduino%20shields.>
- Henkemans, D. in Lee, M. C Programming, Premier Press, 2002
- I2C. [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 8. feb. 2022; 18:49] [Citirano 25. feb. 2022; 19.25] Dostopna na spletnem naslovu: <https://en.wikipedia.org/wiki/I%C2%B2C>
- Jakimovski, B. Biologically Inspired Approaches for Locomotion, Anomaly Detection and Reconfiguration for Walking Robots, Springer Science & Business Media, 2011.
- Legged robot. [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 6. dec. 2021; 21:30] [Citirano 28. feb. 2022; 17.25] Dostopna na spletnem naslovu: [https://en.wikipedia.org/wiki/Legged\\_robot](https://en.wikipedia.org/wiki/Legged_robot)
- Neuroethology of Insect Walking. [Online]. Scholarpedia the peer-reviewed open-access encyclopedia. [Zadnja sprememba 9. nov. 2013; 19:56] [Citirano 6. jan. 2022; 15.45] Dostopna na spletnem naslovu: [http://www.scholarpedia.org/article/Neuroethology\\_of\\_Insect\\_Walking](http://www.scholarpedia.org/article/Neuroethology_of_Insect_Walking)
- Pysmbus. [Online]. GitHub. [Zadnja sprememba 30. aug. 2015] [Citirano 10. okt. 2021; 18.15] Dostopna na spletnem naslovu: <https://github.com/bjornt/pysmbus>
- Robot. [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 16. feb. 2022; 19:51] [Citirano 28. feb. 2022; 16.43] Dostopna na spletnem naslovu: <https://en.wikipedia.org/wiki/Robot>
- Robotics/Types of Robots/Walkers. [Online]. WikiBooks. [Zadnja sprememba 2. mar. 2018; 19:23] [Citirano 3. jan. 2022; 13.13] Dostopna na spletnem naslovu: [https://en.wikibooks.org/wiki/Robotics/Types\\_of\\_Robots/Walkers](https://en.wikibooks.org/wiki/Robotics/Types_of_Robots/Walkers)
- Silva, M. F., Tenreiro Machado, J. A. A Historical Perspective of Legged Robots. Journal of Vibration and Control. 2007, št 13, str. 1447–1486.

- Universal asynchronous receiver-transmitter. [Online]. Wikipedija, prosta enciklopedija. [Zadnja sprememba 25. feb. 2022; 01:27] [Citirano 2. mar. 2022; 17.45] Dostopna na spletnem naslovu: [https://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver-transmitter](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter)
- Wire. [Online]. Arduino. [Zadnja sprememba 2022] [Citirano 2. okt. 2021; 18.00] Dostopno na naslovu: <https://www.arduino.cc/en/reference/wire>

## IZJAVA\*

Mentorja Karmen Kotnik in Tomislav Viher v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljava, da je v raziskovalni nalogi z naslovom Primerjava vozečega in hodečega robota, katere avtor je Maks Jagodič:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, april 2022



Podpis mentorjev

Podpis odgovorne osebe

\*

### POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.