Univerza v Ljubljani
Fakulteta za elektrotehniko

David Femc

Analiza zaznane varnosti sodelujočih robotov v sodelovalni aplikaciji

Diplomsko delo visokošolskega strokovnega študija

Mentor: doc. dr. Janez Podobnik
Somentor: as. dr. Sebastjan Šlajpah

Ljubljana, 2020
Zahvala

Zahvaljujem se mentorju doc. dr. Janezu Podobniku za vse strokovne nasvete in usmeritve, ki so me pripeljali do cilja. Velika zahvala gre tudi somentorju as. dr. Sebastjanu Šlajpahu za vso potrežljivost, podano znanje, ideje in uporabne napotke, brez katerih mi ne bi uspelo. Zahvala tudi vsem osebam, ki so sodelovale v eksperimentu.

Hvala tudi mojim cimrom za čudovita študentska leta, podporo in pomoč pri nastajanju diplomskega dela. Še posebej bi izpostavil Kiti in Staneta, ki sta mi pomagala pri obdelavi slik, in Kajo, ki je poskrbela, da je moje delo slovnično ustrezno.

Nenazadnje gre ogromna zahvala še moji družini za vso podporo in pozitivne spodbude v času študija.
Vsebina

1 Uvod 1
1.1 ISO/TS 15066 ................................................................. 2
1.2 Raziskave na temo zaznane varnosti .................................. 3
1.3 Cilji diplomskega dela ................................................... 4

2 Robotska celica 5
2.1 Robot UR5e ....................................................................... 5
2.2 Ročna učna enota .............................................................. 6
2.3 Prijemalo Robotiq Hand-e .................................................... 7
2.4 Varnostni senzor ............................................................... 7

3 Metodologija 9
3.1 TCP/IP povezava ................................................................ 9
3.2 Konfiguracija varnostnega senzorja ..................................... 10
  3.2.1 Priklop varnostnega senzorja ........................................ 14
3.3 Meritve hitrosti .................................................................. 15
  3.3.1 Izračun hitrosti ............................................................. 16
3.4 Grafični uporabniški vmesnik ............................................. 18
  3.4.1 Izdelava ankete ............................................................. 20
3.5 Zasnova aplikacije ............................................................ 24
  3.5.1 Varno in nevarno orodje ............................................... 25
  3.5.2 Spodnja in zgornja meja hitrosti ................................. 26
  3.5.3 Trajektorija linearnega in naključnega giba .................... 27
  3.5.4 Robotski program ....................................................... 28
4 Eksperiment ................................................................. 33
  4.1 O študiji ........................................................................ 34
    4.1.1 Anketna vprašanja ............................................. 35
  4.2 Rezultati ....................................................................... 36
    4.2.1 Varnost ................................................................... 38
    4.2.2 Prijetnost ............................................................... 39
    4.2.3 Umirjenost ............................................................. 41
    4.2.4 Sodelovanje ......................................................... 42
    4.2.5 Primerjava strokovnjakov in nestrokovnjakov ......... 44
  4.3 Ugotovitve .................................................................. 45

5 Zaključek ...................................................................... 46

Literatura ............................................................................... 48
Seznam uporabljenih okrajšav

TCP/IP – Protokol za nadzor prenosa/Internetni protokol (angl. Transmission Control Protocol / Internet protocol)


IP – Naslov internetnega protokola (angl. Internet Protocol address)

GUI – Grafični uporabniški vmesnik (angl. Graphical User Interface)

HDDM – Merjenje razdalje visoke ločljivosti (angl. High Definition Distance Measurement)

3D – Tridimenzionalen (angl. Three-Dimensional)

TCP – Delovna točka (angl. Tool Center Point)
Povzetek

Na področju robotike se vse bolj razvijajo in uveljavljajo sodelujoči roboti. Za razliko od klasičnih industrijskih robotov so opremljeni z varnostnimi senzorji, s katerimi dosegajo višje standarde varnosti in so zato primerni za sodelovanje s človekom. S tem se povečata še hitrost in učinkovitost dela. Kljub temu da so takšni roboti varni in namenjeni sodelovanju, pa se ljudje pogosto ob njih počutijo nevarno.

V diplomskem delu smo opisali robotsko celico in pripadajočo strojno opremo. Predstavili smo postopek konfiguracije varnostnega sistema, izdelave grafičnega vmesnika in izdelave sodelovalne aplikacije z robotom UR5e. V nadaljevanju smo razložili potek eksperimenta, s katerim smo analizirali zaznano varnost in občutke s strani uporabnika. Za metodo raziskave smo uporabili anketo. Na koncu smo predstavili rezultate in ugotovitve, ki kažejo na to, da na občutje osebe ob robotu največ vpliva vrsta giba, ki ga robot izvaja. V nekoliko manjši meri pa na počutje osebe vpliva tudi izbira orodja ter hitrost, s katero se giblje.

Ključne besede: sodelujoči roboti, varnost, eksperiment
Abstract

In the field of robotics there's a constant growth in innovation and implementation of collaborative robots. They differ from classical industrial robots in terms of safety, because they are equipped with safety sensors, and are able to ensure a higher standard of safety and efficiency for the people who are working with them. Even though that this type of robots are un-harmful towards workers and are meant for collaboration with human users, some people still have prejudice against them.

In this thesis we have described the collaborative robot cell and all hardware included in the application. We have presented the procedure of configuration the safety system, the development of the graphic interface and the app for experiments with the robot UR5e. In the continuation we have explained the protocol of the experiment, with which we analysed the perception of safety from the viewpoint of the user. Our method of research was a survey using questions. At the end we have presented the outcome of the survey, which shows that the type the robot moves has the most effect on the response of the person working with it. A lesser effect on the response has the type of the tool the robot is working with and the velocity at which the robot moves.

Key words: collaborative robots, safety, experiment
1 Uvod

Dandanes si skoraj ne moremo več predstavljati industrijske panoge, ki bi nemoteno delovala brez robotske podpore, ki namesto človeka opravljajo monotona, težka ali nevarna dela. Za preprečevanje poškodb in nesreč so v robotiki in splošno v industriji razviti varnostni standardi, ki delovni prostor robota in človeka strogo razmejjujo in ne dovoljujejo hkratnega dela na istem prostoru. Tako so za zagotavljanje varnosti industrijski roboti pogosto obdani z varnostno kletko ali pa s pomočjo senzorjev in drugih varnostnih sistemov preprečujejo, da človek vstopi v delovni prostor robota med njegovim delovanjem. V določenih primerih lahko sicer človek neko stvar naredi hitreje in natančneje od robota, zato pogosto naletimo na situacije, kjer bi bila interakcija med robotom in človekom zelo zaželena. Tukaj pa nastopijo sodelujoči roboti. [1]

Sodelujoči roboti so oblikovani tako, da nimajo ostrih robov in s tem zagotavljajo še večjo varnost za človeka. Pogosto so opremljeni z dodatnimi senzorji, ki zaznavajo ovire v delovnem prostoru robota ter spremembo hitrosti in navorov. Posledično se ob morebitnem trku s človekom nemudoma zaustavijo.

Glavni proizvajalci sodelujoči robotov so:

- Universal Robots,
- Techman Robot,
- ABB,
- FANUC,
- Kuka in
- Yaskawa.
1.1 ISO/TS 15066


Ključna ideja, na kateri temelji ISO/TS 15066, je:

Če je dovoljeno sodelovanje robota in človeka ter med njima pride do kontakta, to ne sme povzročiti bolečine ali poškodbe.

V standardu so podane smernice za oblikovanje in izdelavo robotov ter določene maksimalne dovoljene hitrosti, ki jo lahko nastavimo robotu.

Standard opisuje tudi metode za sodelovalne aplikacije [3]. Glavne štiri metode so:

- Varnostno nadzorovana ustavitev: pri tej metodi mora biti robotski sistem opremljen z dodatnimi senzorji, ki zaznavajo prisotnost operaterja v delovnem področju robota in aktivirajo varnostno nadzorovano ustavitev. Motorji pri tem ostanejo prižgani; ko operater zapusti delovni prostor, pa se robotski program nadaljuje.
- Vodenje z roko: robotski sistem je pripravljen na vodenje z roko takrat, ko vstopi v sodelovalno delovno območje in aktivira varnostno nadzorovano zaustavitev. Potem lahko operater vstopi v delovno območje robota in izvaja vodenje robota z roko. Ko operater zapusti delovni prostor robota, se robotski program nadaljuje.
- Nadzorovanje hitrosti in varnostne razdalje: pri tej metodi mora biti robotski sistem opremljen s senzorji, ki zaznavajo razdaljo med robotom in operaterjem in glede na oddaljenost operaterja se robot upočasnjuje ali pa se zaustavi, če se operater robotu preveč približa.
- Omejitev moči in sile: pri tej metodi je kontakt med operaterjem in robotom, ki se lahko zgodi namerno ali nenamerno, dovoljen. Takšna metoda pa zahteva omejevanje sile in navora na robotu, s čimer dosežemo, da robot operaterja v primeru trka ne poškoduje, saj se ob prekoračitvi izvede ustavitev.
1.2 Raziskave na temo zaznane varnosti


Slika 1.1: Baxter robot.
1.3 Cilji diplomskega dela

Glavni cilj diplomske naloge je bil ugotoviti, kako ljudje, ki nimajo izkušenj z roboti, dojemajo varnost in sodelovanje z robotom. Želeli smo ugotoviti, kako varno se počutijo v njegovi bližini in v kolikšni meri bi bili pripravljeni sodelovati z njim. V ta namen smo pripravili eksperiment s sodelovalno robotsko aplikacijo in izvedli anketo z osebami, ki se z robotiko še niso srečale. Namen je bil, da bi izvedba ankete potekala v celoti preko računalnika, zato smo ustvarili še vmesnik med računalnikom in robotom. Da bi zagotovili še večjo varnost anketirancev smo dodali še varnostni senzor.

Če povzamemo cilje diplomskega dela:

- vzpostavitev povezave med računalnikom in robotom,
- izdelava ankete,
- izdelava sodelovalne aplikacije,
- konfiguracija varnostnega senzorja in
- analiza ankete.
2 Robotska celica

Pred začetkom izdelave diplomske naloge smo se najprej seznanili z robotom in vso pripadajočo strojno opremo.

2.1 Robot UR5e

Robot UR5e (Slika 2.1) pripada družini sodelujočih robotov e-serije proizvajalca Universal Robots. Ostali člani družine so UR3e, UR10e in UR16e. Vsak robot ima drugačen doseg in dovoljeno obremenitev, vendar enako natančnost, ponovljivost in zanesljivost, s čimer predstavljajo dragocen dodatek v vsaki proizvodnji liniji. UR5e lahko prenaša 5 kg in ima radij dosega 850 mm, idealen je za avtomatizacijo nalog z majhno težo. Ostale tehnične specifikacije so zbrane v tabeli 2.1. [14]

Slika 2.1: Robotska celica UR5e.
Tabela 2.1: Tehnične specifikacije robota UR5e

<table>
<thead>
<tr>
<th>Specifikacije</th>
<th>Najvišja hitrost po oseh</th>
<th>Delovno območje po oseh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masa</td>
<td>20,6 kg</td>
<td>Os 1 ±180°/s</td>
</tr>
<tr>
<td>Osi</td>
<td>6</td>
<td>Os 2 ±180°/s</td>
</tr>
<tr>
<td>Nosilnost</td>
<td>5 kg</td>
<td>Os 3 ±180°/s</td>
</tr>
<tr>
<td>Doseg</td>
<td>850 mm</td>
<td>Os 4 ±180°/s</td>
</tr>
<tr>
<td>Ponovljivost</td>
<td>±0,03 mm</td>
<td>Os 5 ±180°/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Os 6 ±180°/s</td>
</tr>
</tbody>
</table>

2.2 Ročna učna enota

Za konfiguracijo robota, vodenje in izdelavo programov uporabljamo ročno učno enoto (angl. teach pendant), ki ima naložen grafični uporabniški vmesnik Polyscope.
Ročna učna enota (Slika 2.2) je opremljena z 12-palčnim zaslonom na dotik, tipkami za izklop v sili, vklop/izklop enote ter tipko za vodenje robota z roko. S krmilno omarico je povezana preko šestmeterskega kabla, vsebuje pa tudi režo za USB priklop.

![Slika 2.2: Ročna učna enota.](image)

Elementi, ki sestavljajo ročno učno enoto:
1. tipka za izklop v sili,
2. tipka za vodenje robota z roko,
3. tipka za vklop/izklop,
4. reža za priklop USB,
5. zaslon na dotik,
6. priključek za komunikacijski kabel.
2.3 Prijemalo Robotiq Hand-e

Prijemalo Robotiq Hand-e (Slika 2.3) je primerno za uporabo v različnih industrijskih aplikacijah. Na prijemalo lahko namestimo prste različnih oblik, kar omogoča enostavno pobiranje in odlaganje predmetov različnih velikosti in oblik. Prijemalu lahko določimo razmak med prsti, hitrost in maksimalno silo prijema. V tabeli 2.2 so podane tehnične specifikacije prijemala.

![Prijemalo Robotiq Hand-e](image)

Slika 2.3: Prijemalo Robotiq Hand-e.

<table>
<thead>
<tr>
<th>Razmak med prsti</th>
<th>50 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sila prijema</td>
<td>60 do 130 N</td>
</tr>
<tr>
<td>Hitrost zapiranja</td>
<td>20 do 150 mm/s</td>
</tr>
<tr>
<td>Teža</td>
<td>1 kg</td>
</tr>
</tbody>
</table>

2.4 Varnostni senzor

Da bi pri aplikaciji zagotovili še večjo varnost anketirancev, smo vpeljali še varnostni senzor, s katerim smo zaznali, če se je uporabnik robotu preveč približal, na podlagi česar smo upočasnili robota oziroma ga popolnoma zaustavili.
Uporabili smo optični skener SICK nanoscan3 (Slika 2.4), ki je najmanjši varnostni senzor proizvajalca SICK. Zaradi njegove majhne velikosti je zelo uporaben v mobilnih sistemih. Po zaslugi tehnologije safeHDDM je ekstremno odporen na svetlobo, prah in umazanijo [7]. V tabeli 2.3 so podane tehnične specifikacije uporabljenega senzorja.

Slika 2.4: Varnostni senzor SICK nanoscan3.

**Tabela 2.3: Tehnične specifikacije senzorja nanoscan3**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Doseg zaščitenega območja</td>
<td>3 m</td>
</tr>
<tr>
<td>Doseg opozorilnega območja</td>
<td>10 m</td>
</tr>
<tr>
<td>Kot zaznavanja</td>
<td>275°</td>
</tr>
<tr>
<td>Odzivni čas</td>
<td>70 ms</td>
</tr>
</tbody>
</table>

Slika celotne robotske celice je prikazana na sliki 2.5.

Slika 2.5: Robotska celica
3 Metodologija

3.1 TCP/IP povezava


```matlab
obj1 = tcpip('192.168.65.105',50000,'NetworkRole','server')
open(obj1)
```

Slika 3.1: Vzpostavitev povezave.
3.2 Konfiguracija varnostnega senzorja

Konfiguracija varnostnega senzorja SICK poteka v programu Safety Designer in je precej enostavna. Napravo povežemo z računalnikom preko ethernet povezave in zaženemo programsko okolje Safety Designer. V začetnem meniju (Slika 3.2) izberemo search for devices in počakamo, da program poišče vse povezane naprave.

Izberemo senzor nanoscan3 in ga z dvojnim klikom dodamo v izbor, ki je prikazan v oknu Device overview (Slika 3.3). Varnostni senzor je tako dodan v projekt, potrebna je še konfiguracija. Z dvojnim klikom na povezano napravo ustvarimo nov projekt z dodanim varnostnim senzorjem.
3.2 Konfiguracija varnostnega senzorja


![Slika 3.4: Konfiguracija senzorja](image)

Za vsako nastavljeno polje moramo dodeliti še izhode, ki se bodo aktivirali, ko senzor zazna oviro v določenem polju. V zavihku *Inputs and outputs* (Slika 3.6) izberemo digitalne izhode 17-pinskega konektorja, ki jih želimo uporabiti. Izbrali smo štiri izhode in za vsakega določili še napetostni nivo, na katerega se postavi ob aktivaciji.

![Slika 3.5: Konfiguracija varnostnega območja](image)

![Slika 3.6: Konfiguracija digitalnih izhodov senzorja](image)
Izbrane digitalne izhode potem še dodelimo ustreznemu polju. Obema poljema, ki smo jih zgoraj ustvarili, dodelimo po dva digitalna izhoda, saj varnostni vhodi v krmilnik zahtevajo po dva digitalna vhoda. To storimo v zavihku Monitoring cases (Slika 3.7).

S tem je konfiguracija varnostnega senzorja končana. Sedaj izberemo zavihek Transfer (Slika 3.8) in s klikom na Transfer to device naložimo izbrano konfiguracijo na napravo.

Slika 3.7: Konfiguracija izhodov opozorilnega in prepovedanega območja

Slika 3.8: Prenos konfiguracije na napravo
3.2.1 Priklop varnostnega senzorja

Digitalna izhoda, ki pripadata zaščitenemu območju, smo na krmilniku (Slika 3.9) priklopili na vhoda SI0 in SI1 ki sprožita ustavitev robota (angl. Safeguard stop), če se izhoda senzorja postavita na nizek napetostni nivo. Ko senzor ni več aktiviran, robot nadaljuje s programom od tam, kjer se je zaustavil.

Digitalna izhoda opozorilnega območja pa smo priklopili na sklop nastavljivih vhodov (angl. Configurable Inputs) (CI0 in CI1), v robotskem programu smo hkrati nastavili, da ob aktivaciji teh dveh vhodov robot deluje v reduciranem režimu (angl. Reduced mode). V tem načinu se hitrost robota zmanjša na hitrost 60 mm/s, ne glede na hitrost, s katero se premika v normalnem načinu delovanja.
3.3 Meritve hitrosti

Zanimala nas je razlika med željeno hitrostjo, ki jo nastavimo na robotu in dejansko hitrostjo, s katero se robot giblje. Predvsem so nas zanimala odstopanja pri višjih hitrostih. Za izvajanje meritev smo najprej sprogramirali nekaj osnovnih linearnih in ponavljajočih se gibov, pri tem pa smo uporabili privzeto vrednost pospeška, ki je 1200 mm/s². Hitrost smo spreminjali od 100 mm/s do 1000 mm/s in izvedli deset meritev. Lego vrha robota smo merili s sistemom Optotrak. Pri tem smo na prijemalo namestili triado markerjev, kot je prikazano na sliki 3.11.

Optotrak (Slika 3.10) je sistem kamer, s katerimi v realnem času merimo pozicijo markerjev v prostoru z napako manjšo od 0,1 mm. Podatke iz Optotraka smo zajemali s simulinkom in podatke shranili v .mat file. [10]
3.3.1 Izračun hitrosti

Lego markerja opišemo s krajevnim vektorjem \( \vec{r} = (x, y, z) \) v kartezijanem koordinatnem sistemu \((x, y, z)\). Krivuljo, ki določa \( \vec{r}(t) \), imenujemo trajektorija (Slika 3.12), [11].

![Slika 3.12: Trajektorija](image)

Zanimalo nas je, kako hitro se spreminja lega markerja. Zapišemo vektor hitrosti:

\[
\vec{v} = \lim_{\Delta t \to 0} \frac{\Delta \vec{r}}{\Delta t} = \frac{d\vec{r}}{dt} = \left[ \frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt} \right] = [v_x, v_y, v_z]
\]

Velikost hitrosti pa dobimo s kvadratnim korenom vektorja hitrosti.

\[
v = |\vec{v}| = \sqrt{(v_x^2 + v_y^2 + v_z^2)}
\]

Z grafa (Slika 3.13) je razvidno, da dejanska linearna hitrost, s katero se giblje robot pri nizkih hitrostih, dobro sledi referenčni hitrosti. Pri hitrostih višjih od 600 mm/s pa se odstopanje povečuje.

![Slika 3.13: Graf hitrosti](image)
Zanimal nas je še vpliv spremembe pospeška na rezultate, zato smo izvedli nekaj meritev, pri katerih smo za gibe s konstantno hitrostjo spreminjali pospešek. Spodnja grafa (Slika 3.14) prikazujeta časovni potek hitrosti, ki je nastavljena na vrednost 500 mm/s. V prvem grafu robot pospešuje s pospeškom 500 mm/s² v drugem pa s pospeškom 3000 mm/s².

![Slika 3.14: Časovni potek hitrosti v pri različnih vrednostih pospeška a](image1)

Slika 3.15 prikazuje potek hitrosti pri željeni vrednosti 1000 mm/s. Zgornji graf prikazuje potek hitrosti pri pospešku 500 mm/s², spodnji pa pri pospešku 3000 mm/s².

![Slika 3.15: Časovni potek hitrosti pri različnih vrednostih pospeška](image2)
Če primerjamo grafe med seboj, opazimo, da pri manjšem pospešku robot potrebuje več časa, da doseže željeno hitrost in se posledično manj časa giblje z željeno hitrostjo. V primeru, da je željena hitrost prevelika in pospešek zelo majhen, kot prikazuje zgornji graf na sliki 3.15, pa robot v celotnem gibu niti ne doseže željene hitrosti, saj je gib prekratek in začne z zaviranjem še predno doseže željeno vrednost. Pri velikem pospešku pa robot doseže željeno hitrost v precej krašem času, kar je zelo pomembno pri višjih hitrostih. S spodnjega grafa na sliki 3.15 je razvidno, da pri pospešku 3000 mm/s² robot v manj kot pol sekunde doseže hitrost 1000 mm/s. Na ta način smo določili ustrezen pospešek za izvedbo našega eksperimenta.

3.4 Grafični uporabniški vmesnik


Za urejanje vmesnika sta na voljo naslednja dva načina, in sicer:

1. Grafični način:
   V grafičnem načinu (angl. Design view) izdelamo grafično podobo vmesnika. Na voljo je knjižnica komponent (Slika 3.16), v kateri najdemo raznovrstne tipke, drsnike, polja za besedila in polja za grafe, ki jih povlečemo v polje urejevalnika in po želji oblikujemo.
2. Kodni način:


Z uporabo callback funkcij lahko tipkam in ostalim elementom iz grafične knjižnice dodamo kodo, ki se bo izvedla ob pritisku na to tipko. Funkcijo ustvarimo z desnim klikom na element, za katerega želimo ustvariti funkcijo in izberemo dodaj klicno funkcijo (Slika 3.17). V urejevalniku kode se je tako ustvarila nova funkcija (Slika 3.18), v katero vnesemo kodo, ki se bo izvedla ob pritisku na tipko.

![Slika 3.17: Dodajanje callback funkcije](image)

![Slika 3.18: Polje v kateri definiramo funkcijo](image)
3.4.1 Izdelava ankete

Najprej smo uredili grafično podobo vmesnika. Iz knjižnice komponent smo izbrali Tab Group in ustvarili več oken, med katerimi preklapljamo med anketo.

V prvo okno (Slika 3.19) smo dodali okno z besedilom Text Area, kjer so napisana navodila z namenom, da anketiranca seznanimo s potekom ankete. Iz knjižnice dodamo še tipko, s katero anketiranec potrdi, da je prebral navodila in da želi pričeti z anketo.

![Navodila in potek aplikacije](image)

Slika 3.19: Navodila za izvedbo ankete

Za tipko Potrdi smo ustvarili klicno funkcijo in vanjo zapisali ukaz za aktivacijo drugega zavihka:

```javascript
% Button pushed function: PotrdiButton

function PotrdiButtonPushed(app, event)
    app.TabGroup.SelectedTab = app.ena
end
```
V drugem oknu (Slika 3.20) anketiranev vnese svoje osebne podatke. Zanimal nas je nivo tehničnega znanja anketirancev in izkušnje na področju robotike. Za vnos imena, priimka in starosti smo v knjižnici komponent izbrali edit field(text), za ostala vprašanja s ponujenimi odgovori pa smo uporabili radio button group. Dodali smo tipko Začetek, s katero pošljemo robotu ukaz za začetek in aktiviramo naslednje okno.

![Slika 3.20: Okno za vnos osebnih podatkov](image)

Funkcija tipke Začetek, s katero v objekt (obj1) zapišemo vrednost parametra hitrosti. (0.8 = 800 mm/s) in aktivirano naslednje okno.

```matlab
% Button pushed function: ZacetekButton
function NadaljujButtonPushed(app, event)
global obj1
fwrite(obj1,'0.8')

    app.TabGroup.SelectedTab = app.dva

end
```
V tretjem (Slika 3.21) in četrtem oknu (Slika 3.22) se nahajajo anketa vprašanja, za katere smo uporabili devetstopenjsko Likertovo lestvico [13]. Ustvarili smo `radio button group` z devetimi možnimi odgovori. Za lažje razumevanje anketnih vprašanj smo k ponujenim odgovorom dodali še slike. Za dodajanje slik smo izbrali orodje `image` in ustvarili polje, v katerega smo dodali sliko. Sliko dodamo z desnim klikom na polje, izberemo `browse` in izberemo željene slike, ki se morajo obvezno nahajati v isti mapi kot GUI.
Ko anketiranec odgovori na zadnji dve vprašanj v četrtem oknu, vse odgovore zapišemo v matriko z rezultati, odgovore pobrišemo in aktiviramo tretje okno, kjer je prvi del anketnih vprašanj. To se ponavlja, dokler robot ne izvede celotne aplikacije do konca.

Del kode, s katero odgovore shranimo v matriko:

```matlab
global v1
    v1 = app.NeprijetnoPrijetnoButtonGroup.SelectedObject
    v1 = str2num(v1.Text);
    .
    .
    .

global odg1 odg
    odg1 = [v1 v2 v3 v4 v5]
    odg(i,:) = odg1;
```

Po vseh izvedenih korakih aplikacije se odpre zadnje okno (Slika 3.23), v katerem se anketirancu zahvalimo za sodelovanje. Dodamo še tipko Zapri, s katero shranimo vse zabeležene odgovore in zapremo aplikacijo.
3.5 Zasnova aplikacije


Slika 3.24: Kos pohištva

Slika 3.25: Obrat vijaka z imbus ključem

<table>
<thead>
<tr>
<th></th>
<th>ORODJE</th>
<th>HITROST</th>
<th>GIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Varno</td>
<td>Hitro</td>
<td>Linearni</td>
</tr>
<tr>
<td>2</td>
<td>Nevarno</td>
<td>Počasi</td>
<td>Linearni</td>
</tr>
<tr>
<td>3</td>
<td>Nevarno</td>
<td>Hitro</td>
<td>Naključni</td>
</tr>
<tr>
<td>4</td>
<td>Varno</td>
<td>Počasi</td>
<td>Linearni</td>
</tr>
<tr>
<td>5</td>
<td>Nevarno</td>
<td>Počasi</td>
<td>Naključni</td>
</tr>
<tr>
<td>6</td>
<td>Varno</td>
<td>Hitro</td>
<td>Naključni</td>
</tr>
<tr>
<td>7</td>
<td>Nevarno</td>
<td>Hitro</td>
<td>Linearni</td>
</tr>
<tr>
<td>8</td>
<td>Varno</td>
<td>Počasi</td>
<td>Naključni</td>
</tr>
</tbody>
</table>
3.5 Zašnova aplikacije

3.5.1 Varno in nevarno orodje

Za varno orodje smo izbrali brusilno gobico, ker je mehka, brez ostrih robov in ni nevarnosti za poškodbo človeka, tudi če pride do trka. Za nevarno orodje smo izbrali imbus ključ, saj je zaradi trdote in špicaste oblike pri višjih hitrosti za človeka precej bolj nevaren od brusilne gobe. Da je robot orodja s prijemalem lahko čvrsto prijel, smo s 3D tiskalnikom izdelali držalo za obe orodji (Slika 3.26).

![Slika 3.26: Varno in nevarno orodje](image)

Robot je med izvajanjem aplikacije menjaval orodji, zato je bilo pomembno, da sta obe orodji vedno postavljeni na isto mesto. S 3D tiskalnikom smo zato izdelali še stojalo za obe orodji. (Slika 3.27)

![Slika 3.27: Stojalo za orodji](image)
3.5.2 Spodnja in zgornja meja hitrosti

Vrednosti hitre in počasne hitrosti, ki smo jih uporabili pri aplikaciji, smo določili s kraškim eksperimentom. Sodelovalo je osem oseb, ki nimajo izkušenj s področja robotike. Vsak izmed njih se je postavil pred robota in preko grafičnega uporabniškega vmesnika (Slika 3.28) po metodi bisekcije spreminjal hitrost robota in nato določil hitrost, ki je po njegovem mnenju počasna, in hitrost, ki se mu zdi hitra. Prvi gib robota se je izvedel z 50 % maksimalne hitrosti, ki je bila 1000 mm/s. Nato je oseba s tipko počasneje zmanjšala za 25 % ali pa s tipko hitreje povečala za 25 %, v naslednjem koraku pa je trenutno hitrost lahko povečala ali zmanjšala za 12,5 % in po enakem postopku vse do najmanjše možne spremembe ±3,13 %. Celoten postopek spreminjanja hitrosti po metodi bisekcije je prikazan na sliki 3.29. Iz zbranih rezultatov smo izračunali povprečje in dobili vrednost spodnje hitrostne meje 330 mm/s, zgornje hitrostne meje pa 800 mm/s.

Slika 3.28: GUI za določitev hitrostne meje

Slika 3.29: Spreminjanje hitrosti po metodi bisekcije
3.5.3 Trajektorija linearnega in naključnega giba

Če se gibi robota ves čas ponavljajo, se uporabnik na njih hitro navadi, zato je bilo potrebno določiti dva različna gibra oziroma trajektorije, po kateri se robot giblje od točke, kjer prime orodje, do obdelovanca.

Prvi gib (Slika 3.30) je linearen in je definiran z dvema točkama. Prva točka se nahaja nad stojalom z orodji, druga pa se nahaja nad obdelovancem.

Naključno gibanje (Slika 3.31) robota je sestavljeno iz več točk. Sestavljeno je tako, da se robot z orodjem premika na različni višini in naredi tudi več gibov, ki so usmerjeni proti uporabniku. Tako gibanje robota je manj predvidljivo od linearnega giba.
3.5.4 Robotski program

Izdelali smo aplikacijo, v kateri robot večkrat izvede vijačenje in brušenje lesenega kosa pohištva. Da se izognemo večkratnemu pisanju istih ukazov, smo ustvarili več podprogramov, v katerih smo definirali ponavljajoče akcije, v glavnem programu pa smo v vsakem koraku izvedli klice podprogramov.

Podprogrami:

1. Prijem imbus ključa
2. Odlaganje imbus ključa
3. Prijem brusilne gobe
4. Odlaganje brusilne gibe
5. Vijačenje
6. Brušenje
7. Linearni gib od stojala do obdelovanca
8. Linearni gib od obdelovanca do stojala
9. Naključni gib od stojala do obdelovanca
10. Naključni gib od obdelovanca do stojala

Slika 3.32 prikazuje, vrstni red klicev podprogramov za izvedbo brušenja, na sliki 3.33 pa je prikazan klic podprogramov za izvedbo vijačenja.
3.5 Zasnova aplikacije


![Slika 3.34: Konfiguracija gibanja robota](image)

Pod izbranim ukazom `move` definiramo najmanj eno točko. Točke dodajamo z ukazom `waypoint` (Slika 3.35) in jih poimenujemo. Za definiranje točke v koordinatnem sistemu izberemo `set waypoint` in odpre se okno za vođenje robota (Slika 3.36).

![Slika 3.35: Konfiguracija točke](image)
Robota postavimo v željeno pozicijo in pritisnemo OK, da shranimo točko. Vodimo ga s tipkami za spreminjanje pozicije in orientacije na levi strani ali pa izberemo Freedrive funkcijo in robota z roko vodimo v željeno pozicijo.

V podprogramu vijačenja je bilo potrebno z imbus ključem, ki je bil nameščen na prijemalo robota, zadeti vijak. Uporabili smo funkcijo insertion, ki je namenjena za natančno pozicioniranje predmetov in zaznavanje stika s površjem. Funkcija izvaja spiralne, rotacijske in linearne premike, dokler ne doseže določene pozicije. Izbrali smo linear insertion (Slika 3.37), s katero imbus ključ potiskamo proti vijaku, dokler ne preseže maksimalne sile, ki smo jo definirali v programu.
3.5 Zasnova aplikacije

Pri programiranju gibov med stojalom z orodji in obdelovanjem je bil postopek drugačen kot pri ostalih gibih. Premiki, ki jih definiramo z ukazom move, se bodo premikali s hitrostjo, ki jo nastavimo v robotskem programu in med izvajanjem programa tega ni mogoče spreminjati, mi pa smo želeli spreminjati hitrost iz ukazi iz Matlaba. Potrebno je bilo uporabiti fukcijo script, ki sprejema ukaze iz URScript predloge. Uporabili smo ukaz za linerani premik movel, v kateri lahko definiramo pet parametrov.

```plaintext
movel( pose, a=?, v=?, t=?, r=?)
```

- **pose** – lega robota glede na bazni koordinatni sistem;
- **a** – pospešek [m/s²];
- **v** – hitrost [m/s²];
- **t** – čas [s] v katerem se gib izvede. V primeru, da definiramo parameter *t*, bosta parametra *a* in *v* zanemarjena. Robot bo sam izračunal hitrost in pospešek, ki sta potrebna, da se gib izvede v določenem času;
- **r** – radij [m] okoli definirane točke, v kateri robot že začne z naslednjim premikom, radij nastavimo zaradi bolj tekočega izvajanja gibov;

Na sliki 3.38 je prikazan podprogram naključnega premika do obdelovanca, ki je viden na sliki 3.31. Z ukazom assignment smo ustvarili sedem spremenljivk, v katerih smo zapisali koordinate robota. Nato smo za vsako lego zapisali še ukaz za premik. Pri vseh smo pospešek nastavili na vrednost 3 m/s², radij na 2 cm, hitrost gib pa je odvisna od spremenljivke hitrost2, ki smo jo med izvajanjem programa spreminjali preko grafičnega vmesnika.

---

Slika 3.38: Definiranje gibov z URScript ukazi
Vrednost parameta hitrost2 smo med izvajanjem programa spreminjali preko grafičnega vmesnika. V funkciji tipke potrdi smo v objekt obj1 zapisali vrednost željene hitrosti (str. 21). V robotskem programu pa smo ustvarili thread funkcijo, v kateri smo prejeti string z grafičnega vmesnika pretvorili v številsko vrednost in jo zapisali v spremenljivko hitrost2.

```java
Thread_1
hitrost = socket_read_string("internal")
if hitrost != ""
    hitrost2 = to_num(hitrost)
pozeni = 1
Wait: 1.0
```

Slika 3.39: Spremenljivka hitrost2 v robotskem programu
4 Eksperiment

Osebi smo naročili, da vzame kos pohištva, ga zloži skupaj in postavi na označeno mesto. Nato se oseba postavi na narisano črto pred robotom, s katere opazuje (Slika 4.1) gibanje robota ko ta privijači kos pohištva z imbus ključem in nato še zbrusi z brusilno gobo. Po vsaki izvedeni kombinaciji, ko je robot privijačil ali zbrusil kos pohištva, ga je uporabnik vzel v roke, pregledal, razstavil kos pohištva in ga postavil nazaj na označeno mesto. Potem je odgovoril na pet anketnih vprašanj, nato pa pognal naslednjo kombinacijo programa in se zopet postavil na svoje opazovalno mesto. Postopek je ponavljal za vseh osem kombinacij, v katerih je robot z različnimi gibi, hitrostmi in orodji obdeloval kos pohištva.

Slika 4.1: Opazovanje robota
4.1 O študiji

V eksperimentu je sodelovalo 26 oseb, ki niso imele znanja in izkušenj s področja robotike. Od vseh anketiranih oseb je bilo 17 žensk in 9 moških, torej 65 % žensk ter 35 % moških. Grafični prikaz porazdelitve po spolu je prikazan na sliki 4.2. 62 % vprašanih je bilo v starostnem razredu 20–30 let, 12 % 31–40 let, 12 % 41–50 let in 15 % 51–60 let. (Slika 4.3)

Slika 4.2: Porazdelitev anketirancev glede na spol

Slika 4.3: Porazdelitev anketirancev glede na starostno skupino
4.1.1 Anketna vprašanja

Za vseh osem kombinacij orodij, hitrosti in gibov, ki jih je robot izvedel, je anketiranec odgovoril na naslednja vprašanja:

1. Kako varno se vam je zdelo gibanje robota? (1– zelo varno, 9 – zelo nevarno)  
   S tem vprašanjem smo želeli preveriti, ali različne kombinacije vplivajo na to, kako varno se oseba počuti in v kakšni meri.

2. Kako prijetno ste se počutili ob gibanju robota? (1– neprijetno, 9 – prijetno)  
   To vprašanje ugotavlja, ali različne kombinacije vplivajo na prijetno počutje osebe.

3. Kako umirjeno ste se počutili ob gibanju robota? (1– umirjeno, 9 – vznemirjeno)  
   Tu nas je zanimalo, ali različne kombinacije vplivajo na to, kako umirjeno se oseba počuti.

4. V kolikšni meri bi bili pripravljeni sodelovati z robotom? (1 – sploh nič, 9 – povsem)  
   Spraševali smo se, ali bi bila oseba med delovanjem robota pripravljena z njim sodelovati in delati v istem delovnem prostoru in v kolikšni meri se ta pripravljenost do sodelovanja spremeni ob različnih kombinacijah.
4.2 Rezultati

Vsako vprašanje smo analizirali in prikazali, kako spreminjanje gibov, hitrosti in orodja vplivajo na občutke anketirancev. Preverili smo, kako prijetno, umirjeno in varno se oseba počuti ob gibanju robota, pa tudi v kakšni meri bi bili z robotom pripravljeni sodelovati.

Za prikaz podatkov smo uporabili škatlični diagram (angl. box plot). To je posebna vrsta grafov, ki nam na ilustrativen način prikaže več podatkov [15]. Uporabni so predvsem v primeru, ko grafično predstavimo porazdelitev iste spremenljivke v različnih skupinah, torej ko primerjamo več skupin na isti sliki. Dobimo celostno sliko o vplivu skupine na porazdelitev spremenljivke.

Škatlični diagram (Slika 4.4 ) prikazuje:

1. Minimum (najmanjša vrednost vseh rezultatov),
2. Spodnji ročaj (meja pod katero pade 25 % rezultatov),
3. Mediana (deli rezultate v okvirju na dva dela. Polovica rezultatov je večja ali enaka tej vrednosti, polovica pa manj),
4. Zgornji ročaj (meja pod katero pade 75 % rezultatov),
5. Maksimum (najvišja vrednost rezultatov).

Slika 4.4: Škatlični diagram
Za primerjavo rezultatov smo uporabili statistično metodo T test. Ta se najpogosteje uporablja za neodvisne vzorce, ko nas zanima, če obstaja statistično pomembna razlika med hipotetičnim povprečjem, ki ga določimo sami, in tistem dejanskim. T test opravljamo tudi takrat, ko ugotavljamo, ali med dvema neodvisnima vzorcema obstajajo statistično pomembne razlike v povprečni vrednosti.
Da test lahko sploh izvedemo, morajo biti izpolnjeni določeni pogoji: spremenljivke morajo biti številске, in sicer intervalne ali razmernostne, poleg tega pa morajo biti podatki, pridobljeni npr. z anketnim vprašalnikom, normalno porazdeljeni, ali če povemo z drugimi besedami, porazdeljeni po Gaussovi krivulji (ustreznost porazdelitve lahko preverimo s Kolmogorov-Smirnov testom). Poleg tega morata biti varianci med skupinama, v kolikor izvajamo test na dveh skupinah, na enaki stopnji vsaj približno enaki[16,17].
4.2.1 Varnost

Spodnji graf (Slika 4.5) prikazuje vpliv varnega in nevarnega orodja na zaznano varnost. Opazimo, da ima orodje, s katerim robot dela, velik vpliv na občutek varnosti. Pri vseh kombinacijah gibov z izjemo linearnega in počasnega giba (p > 0.05) in hitrosti so se ankentirane osebe počutile bolj varno, ko je robot uporabljal varno orodje.

Slika 4.5: Vpliv orodja na varnost (1 – zelo varno, 9 – zelo nevarno)

Z grafa na sliki 4.6 lahko opazimo, da je pri linearnem gibu ne glede na hitrost in orodje zaznana varnost precej višja kot pri naključnem gibu. Če primerjamo tretji in četrti stolpec, ki prikazuje hitri gib z varnim orodjem je vidno, da ko se izvede linearni gib, se uporabnikom zdi varno, če pa se izvede orodje z isto hitrostjo premakne po neki naključni poti, pa se uporabnikom ne zdi več tako varno.
4.2 Rezultati

Graf 4.7 pa prikazuje vpliv hitrosti. Pri varnem in linearnem gibu opazimo, da hitrost nima nobenega vpliva na občutek varnosti. Pri ostalih kombinacijah je zaznana varnost večja, ko se robot giblje z upočasnjenno hitrostjo, vendar ni toliko razlike v primerjavi s hitrim gibanjem.

\[ Slika \ 4.7: \ Vpliv \ hitrosti \ na \ varnost \ (1 – zelo \ varno, \ 9 – zelo \ nevarno) \]

4.2.2 Prijetnost

Na sliki 4.8 graf prikazuje, kako prijetno so se osebe počutile ob gibanju robota pri varnem in nevarnem orodju. Opazimo, da v primeru, ko robot izvaja linearne premike, se uporabnikom ni zdelo pomembno, kakšno orodje ima robot na prijemalu in se v obeh primerih počutijo prijetno. Če pa so giba naključni in na robotu nameščeno nevarno orodje, pa se počutijo manj prijetno.

\[ Slika \ 4.8: \ Vpliv \ orodja \ na \ prijetnost \ (1 – neprijetno, \ 9 – prijetno) \]
Vpliv giba na občutek prijetnosti je še lepše viden na grafu 4.9, kjer opazimo, da se v primeru naksločnega gibanja oseba počuti veliko manj prijetno.

Z grafa na sliki 4.10 je razvidno, da pri linearnem gibanju hitrost nima takšnega vpliva na dobro počutje osebe. Pri naksločnih gibih robota, ki so sami po sebi nevarnejši, pa opazimo, da se uporabnikom zdi pomembno, s kakšno hitrostjo se robot giblje. Čim manjša je hitrost, tem bolj se oseba počuti prijetno.
4.2.3 Umirjenost

Tudi pri vprašanju kako umirjeno se počuti oseba pri varnem in nevarnem orodju (Slika 4.11) smo dobili podobne rezultate kot zgoraj ter ugotovili, da pri linearnem gibu orodje nima vpliva na občutek umirjenosti. V primeru ko se robot giblje naključno, pa so se uporabniki počutili bolj vznemirjeno, ko je robot uporabljal nevarno orodje.

Slika 4.11: Vpliv orodja na umirjenost (1 – umirjeno, 9 – vznemirjeno)

Iz grafa na sliki 4.12 je razvidno, da način gibanja robota precej vpliva tudi na umirjenost. Pri vsaki kombinaciji z naključnim gibanjem robota je večina oseb odgovorila, da se počutijo precej bolj vznemirjeno kot takrat, ko se robot giblje linearno.

Slika 4.12: Vpliv giba na umirjenost (1 – umirjeno, 9 – vznemirjeno)
Graf vpliva hitrosti (Slika 4.13) prikazuje, da se pri vseh kombinacijah gibov in orodij osebe počutijo bolj umirjeno, če se gib izvaja počasneje. Najbolj pa je razlika opazna pri naključnem gibanju.

Slika 4.13: Vpliv hitrosti na umirjenost (1 – umirjeno, 9 – vznemirjeno)

4.2.4 Sodelovanje

Osebe so bolj pripravljene sodelovati z robotom takrat, ko se ta giblje linearno. Pri tem ni pomembno, s katerim orodjem upravlja. Pri naključnem gibanju pa ima pri pripravljenosti za sodelovanje izbira orodja nekoliko večji vpliv. (Slika 4.14)

Slika 4.14: Vpliv giba na sodelovanje (1 – brez sodelovanja, 9 – s sodelovanjem)
4.2 Rezultati

Anketiranci so odgovarjali, da so bolj pripravljeni sodelovati z robotom, če so njegovi gib linearni. Ko se premika po naključju, so glede sodelovanja bolj zadržani. (Slika 4.15)

![Slika 4.15: Vpliv giba na sodelovanje (1 – brez sodelovanja, 9 – s sodelovanjem)](image1)

Na grafu, ki je prikazan na sliki 4.16 lahko opazimo, da pri linearnem gibanju z varnim orodjem hitrost ne vpliva veliko, pri ostalih kombinacijah pa vidimo, da se osebam pri vprašanju o sodelovanju hitrost zdi bolj pomembna.

Iz rezultatov o sodelovanju, lahko opazimo, da je razpon različnih odgovorov zelo velik, kar si lahko razlagamo, da na pripravljenost sodelovanja z robotom vplivajo tudi drugi dejavniki, kot npr. individualne razlike med posamezniki.
4.2.5 Primerjava strokovnjakov in nestrokovnjakov

Zanimalo nas je kakšna je razlika, če primerjamo rezultate oseb, ki so robota pri eksperimentu videli prvič in strokovnjaki, ki se z robotiko vsakodnevno ukvarjajo. Izvedli smo dodaten eksperiment v katerem je sodelovalo 11 strokovnjakov s področja robotike.


Vizualno lahko opazimo, da pride do razlike med zaznanimi občutki strokovnjakov in nepoznalcev, vendar statistično gledano signifikantnega odstopanja ni. Iz tega lahko sklepamo, da tudi v primeru dolgoletnih izkušenj nevarni gibi robota še vedno vzbudijo manj prijetna občutja in se jih oseba nikoli povsem ne navadi.
4.3 Ugotovitve

Če torej povzamemo, na počutje oseb ob robotu vplivajo način gibanja, izbira orodja in hitrost. Ugotovili smo, da ima največji vpliv način gibanja. Če je torej gib predvidljiv in ponovljiv (kar velja za linearni gib), se oseba ob robotu počutí bolj umirjeno, prijetno in varno, če pa je gib nepredvidljiv oz. ko govorimo o naključnem gibanju, se osebe ob robotu počutijo manj prijetno in sproščeno. Od gibanja robota je odvisno tudi v kolikšni meri so osebe pripravljene sodelovati z robotom in istem delovnem prostoru. V primeru linearnega gibanja so osebe bolj pripravljene delati z njim kot pa takrat, ko se giblje naključno.

V nekoliko manjši meri je počutje odvisno od izbire orodja. Ko je robot opravljal z varnim orodjem, so se osebe počutile precej udobno in varno. Ko pa so prisostvovali delu robota z nevarnim orodjem, so se počutile manj varno.

Najmanj vpliva na razpoloženje oseb pa smo pripisali hitrosti. Rezultati niso pokazali bistvenih razlik glede samega počutja oseb predvsem pri linearnem gibanju, kjer so se osebe počutile enako varno, prijetno in umirjeno, ko se je robot premikal počasi ali hitro. Nekoliko več odstopanj je bilo pri naključnem gibanju. Hitreje kot se je robot premikal, manj prijetno so se osebe počutile ob njem.
5 Zaključek

Pri pripravi naloge smo podrobneje spoznali sodelujočega robota UR5e, programsko okolje Matlab ter naprave za dodatno zagotavljanje varnosti. Na začetku naloge smo prikazali ugotovitve drugih raziskav in študij s tega področja, nadaljevali pa smo z opisom robotske celice. Sledi poglavje o metodologiji, v katerem smo predstavili konfiguracijo varnostnega senzorja, izdelavo uporabniškega vmesnika in sodelovalne aplikacije. V diplomskem delu nas je zanimalo, kako se osebe počutijo ob robotih in v kolikšni meri so pripravljene sodelovati z njim. V ta namen smo pripravili eksperiment in anketo, s pomočjo katere smo analizirali rezultate. Ti so pokazali, da je počutje oseb ob robotu v veliki meri odvisno od načina gibanja robota, nekoliko manjši vpliv pa imata izbira orodja in hitrost, s katero se robot premika. Anketirance smo povprašali tudi, če so pripravljeni sodelovati z robotom, na kar so nam ponudili velik razpon ocen. Iz tega ni točno razvidno, tako kot npr. pri zaznavanju počutja, kdaj bi bili pripravljeni sodelovati z robotom in kdaj ne. Menimo, da so tu prisotni tudi vplivi drugih dejavnikov, kot npr. osebnostne značilnosti vsakega anketiranca. Lahko pa tudi sklepamo, da vprašanje o pripravljenosti na sodelovanje ni bilo ustrezno definirano, saj v vprašanju nismo jasno razložili, kako bi sodelovali z robotom. Posledično si je lahko sodelovanje vsak anketiranece po svoje predstavljal.

Med izvajanjem eksperimenta so nekatere osebe omenile, da se jim zdi razlika med hitrostmi premajhna, in da bi morala biti hitrost občutno večja. Tudi pri analizi rezultatov smo ugotovili, da spreminjanje hitrosti ni v tolikšni meri vplivalo na občutke oseb. V kolikor bi eksperiment ponovili, bi zagotovo upoštevali te pomanjkljivosti. V anketnem vprašalniku so bile ocenjevalne lestvice neenotne, s čimer naj bi se izognili rutinskem izpolnjevanju odgovorov in zagotovili, da si anketiranci podrobno preberejo vprašanja. Na nekaterih mestih je tako 1 pomenila najboljšo možno skrajnost, na drugih pa najslabšo izbiro. To se je izkazalo za neustrezno, saj je mnogo anketirancev ta porazdelitev zmotila, zato so v določenih primerih odgovarjali narobe,
kar so ugotovili na sredi eksperimenta in je bilo potrebno odgovore popraviti ali ponoviti cel eksperiment. Še vedno pa obstaja možnost, da so nekateri posamezniki odgovarjali v nasprotju s svojimi občutki, saj jih je vrsta vprašanj zmedla. To lahko vpliva na rezultate, zato bi bilo v primeru ponovitev eksperimenta potrebno vprašanja jasno definirati in lestvice poenotiti.

Literatura


[7] »nanoScan3, The world's smallest safety laser scanner – highly precise and extremely robust.« Dosegljivo:
4.3 Ugotovitve


