

# **Izdelava in vodenje avtonomnega vozila na gosenicah**

**Aleš Obal**

**Mentor: izr. prof. dr. Rajko Svečko**

**Univerza v Mariboru, FERI**

**Smetanova ul. 17, Maribor**

**alesobal1@gmail.com**

## ***Design and control of autonomous tracked vehicle***

The article describes a process of creating a control monitoring system for steering the robot in the two dimensional space. The key components of the system are sensors for measuring the orientation and position of the vehicle on the microcontroller system. In order to build such a system we created an adequate mathematical model which incorporates the dynamics and kinematics of the vehicle. On the basis of the mathematical model was made a steering algorithm. Vehicle was added communication with a PC for remote monitoring and simultaneous monitoring of its position on the basis of the feedback from the sensors which measure the position and orientation of the vehicle.

## ***Kratek pregled prispevka***

V članku je opisan postopek izdelave kontrolno nadzornega sistema za vodenje robota po prostoru. Ključni deli tega sistema so senzorji za merjenje orientacije in pozicije vozila na mikrokontrolniškem sistemu. Za izgradnjo takega sistema je bil izdelan ustrezen matematični model, ki zajema dinamiko in kinematiko vozila. Na podlagi tega modela smo izdelali algoritem za vodenje. Vozilu smo dodali komunikacijo z osebnim računalnikom za oddaljen nadzor in hkratno spremljanje njegovega položaja na podlagi povratnih informacij iz senzorjev za merjenje orientacije in pozicije.

## 1 Uvod

Avtonomni mobilni roboti so zelo zanimiv predmet raziskovanj in praktičnih aplikacij. Uporabljajo se vedno bolj za zunanje naloge, ki zahtevajo razširjeno stopnjo avtonomije, kot je iskanje, reševanje, planetarno raziskovanje, v kmetijstvu, vojski, gozdarstvu in rudarstvu.

Avtonomne mobilne robote lahko razdelimo na nožne in kolesne robote. Kolesno mobilni roboti so konstrukcijsko enostavnejši in učinkovitejši na utrjeni in ravni podlagi, nožni pa so boljši na neravnem terenu in pri izogibanju oviram.

Cilj zagotavljanja avtonomne mobilnosti robota je prenesti sistem iz trenutnega položaja na drugo lokacijo po vnaprej določeni ali sproti zaznani poti. Pri tem pa ne sme priti do poškodb mehanizma ali trkov.

Za izgradnjo takega sistema je treba upoštevati vse motnje, ki se lahko pojavijo med različnimi misijami. Nadzorni sistem avtonomnih mobilnih robotov mora hitro reagirati in se prilagoditi spreminjajočim se razmeram okolja. Da lahko uspešno zgradimo tak sistem, moramo najprej ustrezno pripraviti dinamični model. Ta model nam omogoča, da ob upoštevanju lastnosti, kot so masa, trenje, sile, navori, itd. določimo pospeške, hitrosti in poti vozilu. Če pa hočemo spremljati še trajektorijo vozila pa je nujen še kinematični model.

Cilj našega projekta je bil izdelati avtonomno vozilo, ki bo sposobno gibanja po prostoru z vnaprej določeno trajektorijo v ravninskem koordinatnem sistemu. Za določanje in spremljanje trajektorije je bil narejen uporabniški vmesnik v LabView programskem okolju.

Za orientiranje vozila uporabimo senzor kotne hitrosti (žiroskop), za merjenje poti pa optična senzorja. Informacije iz senzorjev so ključne pri takšnih sistemih, saj se na podlagi teh avtonomni mobilni roboti tudi premikajo. Te informacije uporabimo pri algoritmu vodenja, ki ga implementiramo na mikrokontrolniškem sistemu. Za delovanje pa je potrebna tudi

komunikacija vozila z računalnikom, ki je bila realizirana z APC220 radio komunikacijskim modulom.

Podatki, ki se prenašajo z komunikacijo sta dejanski in referenčni koordinati vozila. Dejanski koordinati vozila se v smeri računalnika pošiljata kontinuirano, referenčni koordinati za vodenje vozila pa v smeri vozila pošljemo le enkrat. Ko se vozilo dovolj približa referenčni točki, algoritem v LabView na podlagi povratne informacije dejanskih koordinat poskrbi, da se vozilu pošljeta novi koordinati. Vozilo lahko vodimo tudi ročno in hkrati spremljamo njegovo trajektorijo.

V prvem delu članka so prikazane uporabljene komponente sistema, njihov pomen v projektu ter fizični model. V drugem delu je opisano modeliranje sistema, ki vključuje dinamični in kinematični model ter simulacijo sistema zaprtega v regulacijsko zanko v MATLAB/SIMULINK. Tretji del pa obsega vodenje in implementacijo regulacijskega algoritma ter opis izdelave GUI grafičnega vmesnika (oddaljenega nadzornega sistema za vodenje in nadzor vozila).

## 2 Opis vozila

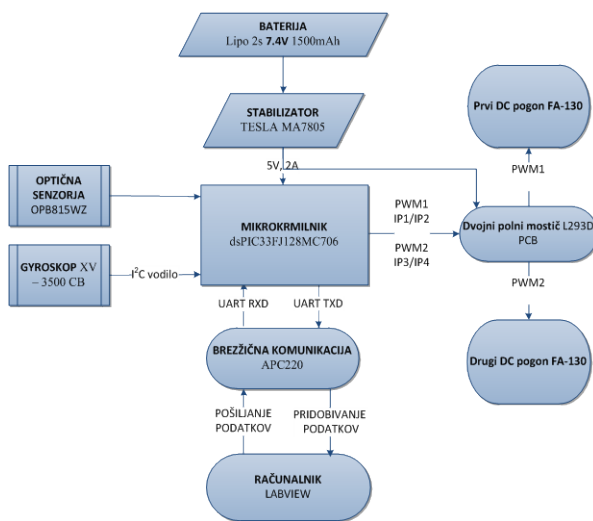
Vozilo sestavimo tako, da bo čim bolj kompaktno in funkcionalno. Ker želimo imeti natančen, stabilen in čim bolj zanesljiv sistem, moramo komponente namestiti na ogrodje tako, da bo masa celotnega vozila zbrana v središču vozila. To je pomembno zato, da preprečimo neenakomerno obremenjenost pogonov in čim bolj zmanjšamo zdrs gosenic na neravni površini.

Za razvoj avtonomnega vozila uporabimo mikrokontrolnik, ki vsebuje vse potrebne periferne komponente, ki jih potrebujemo za izgradnjo avtonomnega vozila. To so:

- generiranje dveh PWM signalov za vodenje pogonov,
- zajemanje podatkov preko I<sup>2</sup>C vodila iz senzorja kotne hitrosti (žiroskopa),

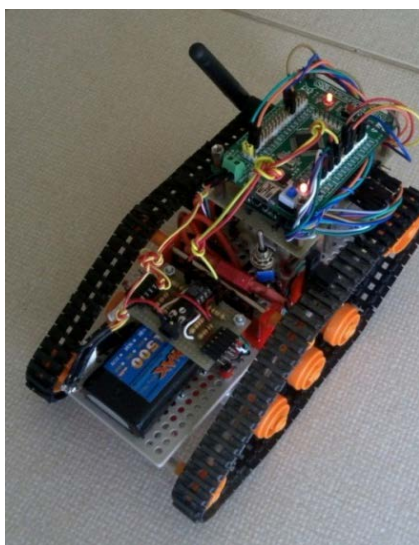
- dva digitalna vhoda za zajemanje pulzov iz gosenic,
- štiri digitalne izhode za določanje smeri pogonom in
- UART univerzalni asinhroni sprejemnik / oddajnik za brezžični APC220 modul.

Vse elemente povežemo in združimo v sistem kot je prikazano na *sliki 1*.



*Slika 1: Shema sistema*

Končno izdelano vozilo je prikazano na *sliki 2*.



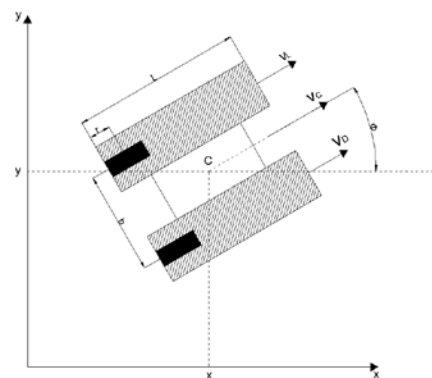
*Slika 2: Sestavljeno vozilo.*

### 3 Matematični model sistema

Za določanje trajektorije gibajočega vozila po prostoru zasnujemo dinamični in kinematični model vozila [1]. Najprej zapišemo diferencialne enačbe dinamičnega modela vozila, ki predstavljajo obodne hitrosti in pospeške gosenic glede na navore pogonov, ki pa so odvisni od napetosti.

Ker smo hoteli pokazati gibanje vozila po prostoru, smo zasnovali še kinematični model. Pri izpeljavi tega modela predpostavimo, da se vozilo giblje po ravni podlagi, ki se ne podaja. Da čim bolj poenostavimo model, pa zanemarimo zdrs gosenic glede na podlago.

Na *sliki 3* je prikazana pozicija vozila v ravnini.



*Slika 3: Pozicija vozila v ravnini*

#### 3.1 Dinamični model

Dinamični model sistema je zelo pomemben v primerih, ko hitrosti mobilnega robota tvorijo sile, katerih vpliv med gibanjem ne moremo zanemariti.

Model transformira vhodna navora levega in desnega pogona v kotne pospeške in hitrosti leve in desne gosenice. Do dinamičnega modela pridemo s pomočjo Euler Lagrangevih enačb [2]:

$$A\ddot{\theta}_D + B\ddot{\theta}_L = M_D - K\dot{\theta}_D, \quad (1)$$

$$B\ddot{\theta}_D + A\ddot{\theta}_L = M_L - K\dot{\theta}_L, \quad (2)$$

kjer je  $K$  viskozni faktor trenja na levi in desni strani gosenic,  $M_D$  in  $M_L$  navora desnega in levega pogona,  $\dot{\phi}_D$  in  $\dot{\phi}_L$  kotni hitrosti desne in leve gosenice,  $\ddot{\phi}_D$  in  $\ddot{\phi}_L$  kotna pospeška desne in leve gosenice ter  $A$  in  $B$  izračunana koeficienta momentov vozila.

### 3.2 Kinematični model

Izhodi iz dinamičnega modela so vhodni parametri v kinematični model [3], ki se tukaj transformirajo v hitrosti vozila v  $X$  in  $Y$  smeri v koordinatnem sistemu:

$$\dot{X}_C = \frac{r(\dot{\phi}_D + \dot{\phi}_L)}{2} \cos \phi, \quad (3)$$

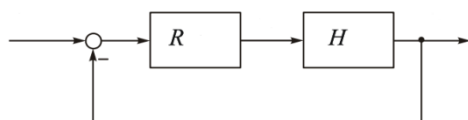
$$\dot{Y}_C = \frac{r(\dot{\phi}_D + \dot{\phi}_L)}{2} \sin \phi, \quad (4)$$

$$\dot{\phi} = \frac{r(\dot{\phi}_D - \dot{\phi}_L)}{b}, \quad (5)$$

kjer je  $\dot{X}_C$  hitrost centra vozila v smeri  $X$  osi v prostoru,  $\dot{Y}_C$  hitrost centra vozila v smeri  $Y$  osi v prostoru,  $\dot{\phi}$  kotna hitrost zasuka vozila,  $\dot{\phi}_D$  in  $\dot{\phi}_L$  kotni hitrosti desne in leve gosenice,  $r$  polmer kolesa in  $b$  osna širina med gosenicama.

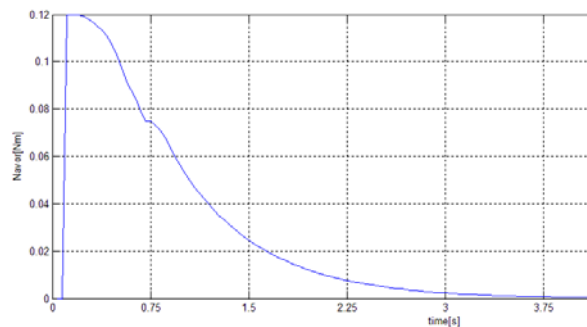
### 3.3 Regulator v MATLAB/SIMULINK

Po modeliranju smo na podlagi modela zasnovali regulator kota zasuka in poti. Uporabimo regulacijsko konfiguracijo [4] z regulatorjem  $R$  v direktni veji (slika 4).

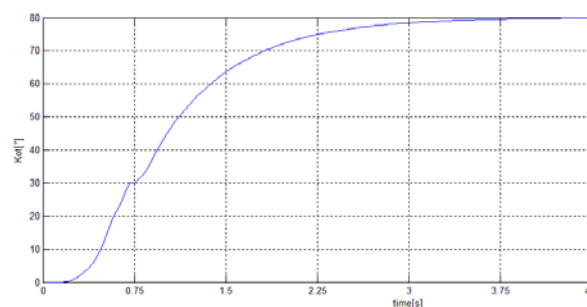


Slika 4: Regulacijski sistem

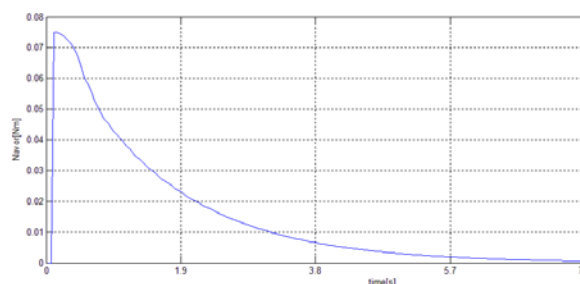
Za sistem v zaprti zanki določimo regulator  $P$ -tipa za kot zasuka in poti. Z simuliranjem določimo srednje vrednosti ojačenja tako, da je sistem stabilen in brez prenehaja. Pri tem pa dobimo naslednje rezultate (slika 5,6,7 in 8).



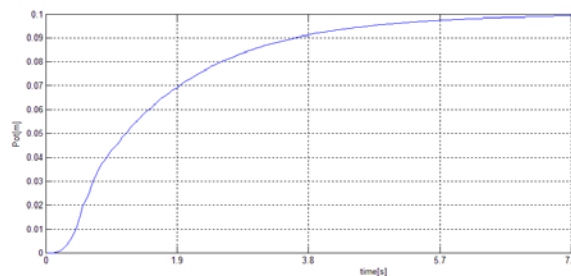
Slika 5: Izhod regulatorja pri referenčnem kotu  $80^\circ$



Slika 6: Izhod sistema pri referenčnem kotu  $80^\circ$



Slika 7: Izhod regulatorja pri referenčni poti 0,1 m



Slika 8: Izhod sistema pri referenčni poti 0,1 m.

#### 4 Algoritem za pozicioniranje vozila

Če enačbi (3) in (4) integriramo po času, dobimo pot v X in Y smeri, ki jo uporabimo pri algoritmu pozicioniranja:

$$X_C = \frac{1}{2} (X_D + X_L) \cos(\varphi), \quad (6)$$

$$Y_C = \frac{1}{2} (X_D + X_L) \sin(\varphi), \quad (7)$$

kjer je  $X_C$  pot centra vozila v smeri X-osi,  $Y_C$  pot centra vozila v smeri Y-osi,  $X_D$  in  $X_L$  pot desnega in levega kolesa ter  $\varphi$  kot iz žiroskopa.

Za natančno pozicioniranje vozila so ključnega pomena senzorji. Iz optičnih senzorjev dobivamo pulze, ki nam omogočijo določanje poti ( $X_D$  in  $X_L$ ), iz žiroskopa pa kotno hitrost. Kotno hitrost iz žiroskopa v mikrokrmilniku beremo v prekinitveni rutini oziroma časovniku. To kotno hitrost tako le še integriramo po času, da dobimo kot ( $\varphi$ ). Čas integracije je enak času proženja časovnika. Ko pridobimo podatke iz teh dveh senzorjev jih le še uporabimo pri enačbah pozicioniranja. Algoritem pozicioniranja uporabimo za ročni in prav tako za avtomatski način vodenja.

#### 5 Ročno vodenje

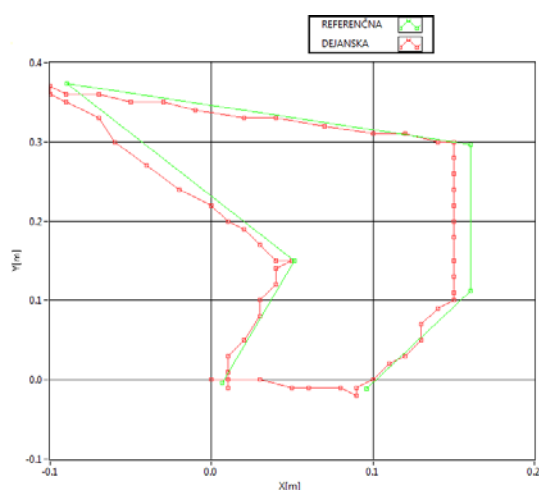
Ročno vodenje obsega komunikacijo preko UART vmesnika v smeri mikrokrmilnika in s tem tudi krmiljenje pogonov. Pogone krmilimo s tipkami. Če mikrokrmilniku pošljemo znak W se bo vozilo gibalo naprej. Če mu pošljemo znak A, se bo vozilo zasukalo v levo in če pošljemo znak D, v desno. V tem režimu vodenja lahko spreminjamo tudi hitrost gibanja. Sprejemanje ukazov v mikrokrmilnik realiziramo v časovniku, ki se vsako milisekundo proži kot prekinittev ter preverja, če smo po UART vodilu dobili katerega izmed teh ukazov.

#### 6 Avtomatsko vodenje

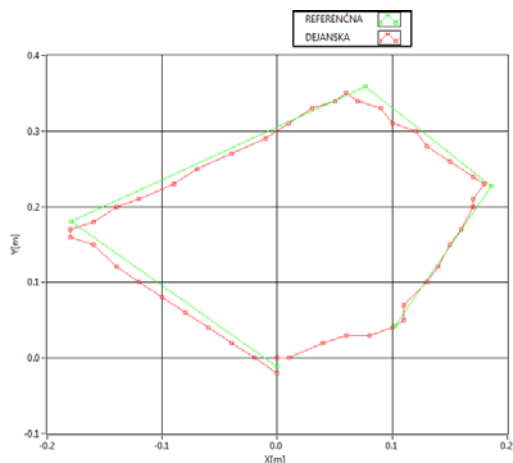
V avtomatskem načinu mikrokrmilniku pošljemo točki X in Y, ki ju lahko poimenujemo želeni vrednosti sistema oziroma, referenci sistema. Iz teh dveh želenih vrednostih, ki ju dobimo v obliki koordinat pa lahko izračunamo kot zasuka in pot, ki jo vozilo mora narediti, da doseže želeno pozicijo. Na podlagi teh zasujemo regulator, ki krmili napetosti na pogonih.

Ker imamo dve različni želeni vrednosti, smo zasnovali tudi dva regulatorja. Eden skrbi, da sistem sledi referenčnemu kotu, drugi pa, da sistem sledi referenčni poti. Za konfiguracijo regulatorjev uporabimo dobljena ojačenja iz simulacij v MATLAB/SIMULINK. Regulatorje v mikrokrmilniku implementiramo v časovnikih oziroma, prekinitvenih rutinah z vnaprej določenim prekinitvenim časom, ki ga lahko poimenujemo čas tipanja.

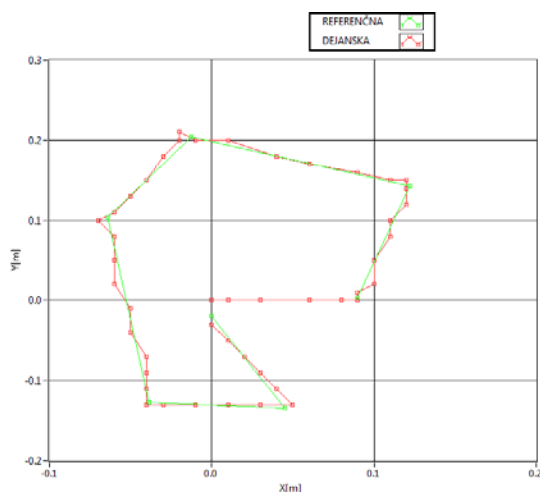
V namen primerjave vodenja vozila smo preizkusili nekaj različnih regulatorjev. Na slikah 9, 10 in 11 vidimo odstopanja med referenčnimi in dejanskimi potmi, ki jih vozilo opravlja z različno implementiranimi regulatorji.



Slika 9: Primerjava narisane in opravljene poti z regulatorjem poti brez popravljanja kota



Slika 10: Primerjava narisane in opravljene poti z dodanim regulatorjem kota na desnem kolesu



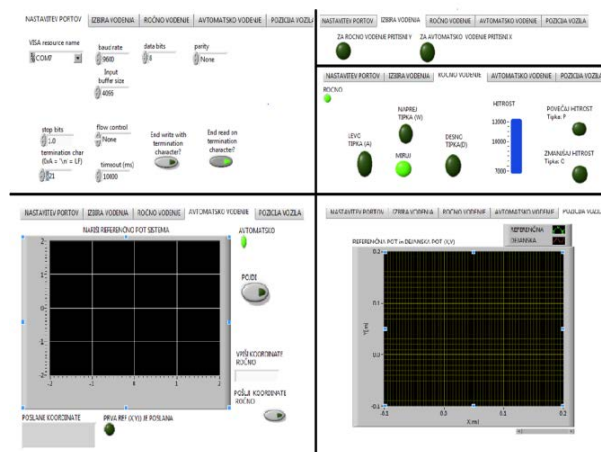
Slika 11: Primerjava narisane in opravljene poti z dodanim regulatorjem kota na obeh kolesih

Iz slike 9 vidimo da se pojavijo precejšnja odstopanja med referenčno in opravljeno potjo. Kar pa ne velja če regulatorju poti dodamo še regulator po kotu (slika 10 in 11).

Prednost regulatorja poti z dodanim regulatorjem kota je, da vozilo lahko vodimo tudi po neravni površini in bo še vedno sledilo začrtani poti. To lahko testiramo tako, da vozilo namerno odmaknemo iz začrtane poti, vozilo pa se bo znova vrnilo na začrtano pot.

## 7 Grafični vmesnik za vodenje in pozicioniranje vozila

Grafični vmesnik prikazan na sliki 12 smo naredili za namen vodenja in pozicioniranja vozila. Zaradi preglednosti smo ga razdelili na sekcije: inicializacijo, izbiro vodenja, ročno vodenje, avtomatsko vodenje in pozicioniranje.



Slika 12: Grafični vmesnik v LabView 9.0

V izbiri vodenja izbiramo med avtomatskim ali ročnim vodenjem. Ob izboru avtomatskega vodenja narišemo referenčno pot gibanja vozila, medtem pa lahko spremljamo dejansko opravljeno pot v sekciji pozicioniranje.

## 8 Sklep

V članku je opisana izdelava in princip vodenja avtonomnega vozila na gosenicah, ki definira fizično izvedbo vozila s pogonskimi aktuatorji ter vsemi potrebnimi sestavnimi deli za avtonomno delovanje.

Vozilo se je sposobno orientirati glede na v naprej podane referenčne vrednosti posameznih koordinat. Referenco gibanja vozila pa določamo v LabView uporabniškem vmesniku.

Vozilu je dodana tudi opcija ročnega oz. daljinskega upravljanja preko standardnih vhodnih naprav računalnika. Pri tem pa omogoča tudi dokaj natančno pozicioniranje v ravninskem koordinatnem sistemu in hkrati tudi popravljanje kota zaradi netočnosti vpetja gosenic.

Preizkušeni so bili različni načini vodenja. Pri vodenju prihaja do odstopanj zaradi nelinearne odvisnosti hitrosti od napetosti. To bi lahko izboljšali z identifikacijo napetosti na pogonih pri različnih hitrostih, na podlagi katere bi zasnovali regulator.

Regulacijo vozila bi lahko zasnovali še po številnih drugih variantah. Iz senzorjev poti bi lahko v časovnikih preračunavali hitrosti leve in desne gosenice in z hitrostmi regulirali kot gibanja vozila. Žiroskop pa bi uporabili samo za prečno popravljanje kota, ki ga iz senzorjev poti ni možno dobiti. Vozilu bi lahko dodali še regulator po dejanski krivulji in mu točke pošiljali zelo blizu skupaj. Pri tem pa bi uporabili regulator poti z dodanim regulatorjem kota. Tako bi dosegli, da bi vozilo sledilo krivulji.

Pri vodenju nastopi še en problem vreden omembe. Vrednosti, ki se pojavijo na izhodu iz žiroskopa v obliki kotne hitrosti, so zelo občutljive na premike in se pojavljajo tudi, ko vozilo miruje. Kot, ki ga dobimo ob integriranju teh vrednosti pa v sistemu ne smemo upoštevati. To napako zmanjšamo programske, če žiroskopu omejimo izhod do neke minimalne kotne hitrosti. Ko to kotno hitrosti presežemo, jo upoštevamo.

## 9 Literatura

- [1] Rajko Svečko, Teorija sistemov, Fakulteta za elektrotehniko, računalništvo in informatiko, 2000 Maribor
- [2] Edouard Ivanjko, Toni Petrini, Ivan Petrovi, Modelling of mobile robot dynamics, Faculty of Electrical Engineering and Computing 10000 Zagreb, Unska 3, Croatia.
- [3] Gyula Mester, Motion Control of Wheeled Mobile Robots, Department of Informatics, Polytechnical Engineering College, Marka Oreskovica 16, 24000 Subotica, Serbia
- [4] Rajko Svečko, Diskretni regulacijski sistemi, Fakulteta za elektrotehniko, računalništvo in informatiko, 2003 Maribor