



Nikolaj Semenič

STABILIZACIJA INVERZNEGA ROTACIJSKEGA NIHALA Z DSP

Diplomsko delo

Maribor, junij 2009

Diplomsko delo univerzitetnega strokovnega študijskega programa

STABILIZACIJA INVERZNEGA ROTACIJSKEGA NIHALA Z DSP

Študent: Nikolaj Semenič
Študijski program: UN ŠP - Elektrotehnika
Smer: Avtomatika
Mentor: izr.prof.dr. Rajko Svečko
Somentor: dr. Amor Chowdhury

Maribor, junij 2009

UNIVERZA V MARIBORU



Smetanova ulica 17
2000 Maribor

Številka: E-2558

Datum in kraj: 12. 06. 2009, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 90/2008)

SKLEP O DIPLOMSKEM DELU

1. **Nikolaju Semeniču**, študentu univerzitetnega študijskega programa Elektrotehnika, smer Avtomatika, se dovoljuje izdelati diplomsko delo pri predmetu Regulacije II.
2. **MENTOR:** izred. prof. dr. Rajko Svečko
3. **Naslov diplomskega dela:**
STABILIZACIJA INVERZNEGA ROTACIJSKEGA NIHALA Z DSP
4. **Naslov diplomskega dela v angleškem jeziku:**
STABILIZATION OF ROTARY INVERTED PENDULUM WITH DSP
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela" in ga oddati v treh izvodih ter en izvod elektronske verzije do 12. 06. 2010 v referatu za študentske zadeve.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.



Obvestiti:

- kandidata,
- mentorja,
- odložiti v arhiv.

“Prihodnost pripada tistim, ki verjamejo v lepoto svojih sanj.”

Eleanor Roosevelt

ZAHVALA

Zahvaljujem se mentorju za pomoč in vodenje pri opravljanju diplomskega dela. Prav tako se zahvaljujem komentorju. Hvala tudi Andreju Sarjašu in Andreju Knupležu, ki sta mi s svojimi nasveti vestno stala ob strani.

Posebna zahvala velja staršem, ki so me v času študija podpirali.

STABILIZACIJA INVERZNEGA ROTACIJSKEGA NIHALA Z DSP

Ključne besede: Regulacije, digitalni signalni procesor, regulator v prostoru stanj, linearni kvadratični regulator

UDK: 681.511.2(043.2)

Povzetek

V tem delu je predstavljena priprava programskih gradnikov na procesorju DSP TMS320F28335, ki so bili potrebni za razvoj okolja primernega izvajanju regulacijskih algoritmov. Sistem je bil zasnovan tako, da je prilagodljiv in primeren za nadaljnje eksperimente na področju vodenja in regulacij.

Izvedeno je bilo načrtovanje in izvedba algoritmov vodenja za stabilizacijo inverznega rotacijskega nihala.

STABILIZATION OF ROTARY INVERTED PENDULUM WITH DSP

Key words: Control, digital signal processor, state space regulator, linear quadratic regulator

UDK: 681.511.2(043.2)

Abstract

This work presents preparation of the software for different modules on DSP TMS320F28335, which are needed to create an environment that is suitable for realization of regulation algorithms. The system has been designed in way of adaptability for further use.

Design and realization of regulation algorithms for stabilization of rotary inverted pendulum have been performed.

VSEBINA

| | | |
|----------|---|-----------|
| 1 | UVOD | 1 |
| 2 | OPIS SISTEMA VODENJA IN RAZVOJNIH KOMPONENT | 3 |
| 2.1 | INVERZNO NIHALO APPARATUS KRI PP-300 | 3 |
| 2.2 | POTENCIOMETER | 3 |
| 2.3 | ENKODER..... | 4 |
| 2.4 | NAPAJALNIK..... | 5 |
| 2.5 | RAZVOJNA PLOŠČA ZA POGON MOTORJA | 5 |
| 2.6 | MIGRACIJA IZ INTEL MCS-96 N80C196KC-20 NA TMS320F28335..... | 6 |
| 2.7 | RAZVOJNA PLOŠČA Z DSP TMS320F28335..... | 6 |
| 2.8 | RAZVOJNO OKOLJE CODE COMPOSER STUDIO 3.3 | 8 |
| 3 | MATEMATIČNI MODEL IRN IN OCENA PARAMETROV | 9 |
| 3.1 | NELINEARNI MODEL | 10 |
| 3.2 | LINEARNI MODEL..... | 14 |
| 3.3 | REALIZACIJA MODELOV V SIMULINKU..... | 16 |
| 4 | PROGRAMSKI GRADNIKI KOMPONENT PROCESORJA | 19 |
| 4.1 | MODUL ZA PODPORO ENKODERJA (EQEP)..... | 20 |
| 4.2 | AD PRETVORBA..... | 29 |
| 4.2.1 | <i>Vgrajeni AD pretvornik</i> | <i>31</i> |
| 4.2.2 | <i>Zunanji AD pretvornik ADS1213</i> | <i>32</i> |
| 4.3 | SERIJSKO VODILO SPI..... | 36 |
| 4.4 | SERIJSKO VODILO SCI | 39 |
| 4.5 | PULZNO ŠIRINSKA MODULACIJA (PWM) | 44 |
| 4.6 | PREKINITVE IN ZGRADBA JEDRA PROGRAMA | 46 |
| 5 | NAČRTOVANJE REGULATORJEV V PROSTORU STANJ | 51 |
| 5.1 | REGULATOR ZA POMIKANJE POLOV | 53 |
| 5.2 | REGULATOR ZA SLEDENJE REFERENČNEMU VHODU..... | 54 |
| 5.3 | LINEARNI KVADRATIČNI REGULATOR (<i>LQR</i>)..... | 55 |

| | | |
|----------|--|-----------|
| 6 | REZULTATI REGULACIJSKIH ALORITMOV | 56 |
| 6.1 | IZVEDBA REGULATORJA ZA POMIKANJE POLOV | 60 |
| 6.2 | IZVEDBA REGULATORJA Z REFERENČNIM VHODOM..... | 65 |
| 6.3 | IZVEDBA LQR REGULATORJA..... | 71 |
| 7 | SKLEP | 77 |
| 8 | VIRI, LITERATURA..... | 78 |
| 9 | PRILOGE..... | 79 |
| 9.1 | MATEMATIČNI MODEL INVERZNEGA NIHALA [5] | 79 |
| 9.1.1 | <i>Model enosmernega motorja.....</i> | <i>79</i> |
| 9.1.2 | <i>Model inverznega nihala</i> | <i>80</i> |
| 9.2 | MATEMATIČNI MODEL NIHALA V OBIČAJNI LEGI [5] | 87 |
| 9.3 | LINEARIZACIJA [5]..... | 88 |
| 9.4 | MERITVE OJAČANJA NAPETOSTNEGA PRETVORNIKA..... | 92 |
| 9.5 | SEZNAM SLIK..... | 93 |
| 9.6 | SEZNAM PREGLEDNIC | 96 |
| 9.7 | NASLOV ŠTUDENTA | 96 |
| 9.8 | KRATEK ŽIVLJENJEPIS..... | 97 |

UPORABLJENI SIMBOLI

| | |
|------------|---|
| m_l | masa nihala |
| l_l | dolžina nihala |
| L_0 | dolžina prečke |
| α | kot prečke |
| θ | kot nihala (spodnja lega) |
| β | kot nihala (navpična lega) |
| x | os koordinatnega sistema |
| y | os koordinatnega sistema |
| z | os koordinatnega sistema |
| W_k | kinetična energija sistema |
| W_p | potencialna energija sistema |
| W_{kp} | kinetična energija prečke |
| W_{pp} | potencialna energija prečke |
| W_{kn} | kinetična energija nihala |
| W_{pn} | potencialna energija nihala |
| Q_i | doveden navor na i-ti osi |
| Θ_i | kot telesa glede na i-to os |
| L | Lagrange |
| M_p | skupen navor na prečko |
| M_n | skupen navor na nihalo |
| M_m | navor motorja |
| U | napajalna napetost motorja v obliki PWM |
| V_m | enosmerna napajalna napetost motorja |
| R_a | upornost kotve |

| | |
|-------------------|---|
| L_a | induktivnost kotve |
| I_a | tok skozi navitje |
| E_b | inducirana napetost |
| K_b | električna konstanta |
| K_t | navorna konstanta |
| T_s | čas tipanja |
| $\ddot{\alpha}$ | pospešek prečke |
| $\dot{\alpha}$ | hitrost prečke |
| α | položaj prečke |
| $\ddot{\beta}$ | pospešek nihala |
| $\dot{\beta}$ | hitrost nihala |
| β | položaj nihala |
| $v(k)$ | hitrost v časovnem koraku k |
| $x(k)$ | položaj v trenutnem časovnem koraku |
| $x(k-1)$ | položaj v prejšnjem časovnem koraku |
| T | konstanta, ki predstavlja čas tipanja |
| ΔX | sprememba poti v času tipanja |
| $t(k)$ | izmerjen čas v trenutnem dogodku |
| $t(k-1)$ | izmerjen čas v predhodnem dogodku |
| X | konstantni položaj |
| ΔT | čas, ki je pretekel pri opravljeni poti X |
| ω_{visoka} | hitrost motorja |
| n_E | število dogodkov v enem obratu |
| P_{CKL} | frekvenca procesorja, znaša 150 MHz |
| <i>Turbo Mode</i> | je faktor, ki vpliva na frekvenco delta sigma modulatorja |

Decimation Ratio je faktor, ki vpliva na število povprečenih rezultatov

| | |
|------------------|---|
| ω_{nizka} | hitrost motorja |
| D_e | del obrata, ki ga motor naredi med dogodkoma |
| n_C | število časovnih kvantov |
| $f_{podatki}$ | hitrost izhodnih podatkov AD pretvornika |
| f_{XIN} | frekvenca zunanega oscilatorja AD pretvornika |
| AC | vhodni analogni signal |
| $ADCIN0$ | vhodni pin AD pretvornika |
| R_8 | Vhodna upornost pina |
| R_{ON} | Upornost multiplekserja |
| C_p | Parazitna kapacitivnost |
| C_h | Kapacitivnost za shrambo vrednosti |
| R_{IN} | Vhodna upornost, ki omejuje vhodni tok (ne sem biti velika, tipično 50 Ω) |
| R_{SW} | Upornost multiplekserja |
| C_{IN} | Zunanji kondenzator, z uporom predstavljata filter |
| C_{SH} | Kapacitivnost za shrambo vrednosti |
| V_{PS} | Predstavlja prejšnjo zadržano vrednost |
| Q | utežitvena matrika spremenljivk stanj |
| R | utežitvena matrika vhodnih spremenljivk |
| K | matrika zaprtozančnih ojačanj |
| K | eksperimentalno določeno zaprtozančno ojačanje |
| Q_v | vodljivostna matrika |
| x_s | stacionarno odstopanje |
| y_s | stacionarno odstopanje na izhodu |
| r_p | prefilter |

UPORABLJENE KRATICE

| | |
|---------|--|
| DSP | Digital Signal Controller, digitalni signalni procesor |
| IRN | Inverzno Rotacijsko Nihalo |
| PWM | Pulse Width Modulation, pulzno širinska modulacija |
| AD | Analog to Digital, analogno digitalno |
| ADC | Analog to Digital Converter, analogno digitalni pretvornik |
| SPI | Serial Peripheral Interface ali serijsko sinhrono vodilo |
| SCI | Serial Communication Interface ali serijsko asinhrono vodilo |
| I2C | Serijsko vodilo |
| eCAN | Enhanced Controller Area Network |
| McBSP | Multichannel Serial Buffered Port, serijsko vodilo |
| RAM | Random Access Memory |
| OTP ROM | One Time Programmable Read Only Memory |
| DMA | Direct Memory Access |
| eCAP | Enhanced Capture Module, enota za zaznavanje dogodkov |
| eQEP | Enhanced Quadrature Encoder Pulse Module, enota za enkoder |
| GPIO | General Purpose Inputs Outputs, digitalni vhodi in izhodi |
| WD | Watch Dog |
| QUPRD | eQEP Unit Periode Register, register enote periode |
| QUTMR | eQEP Unit Timer, časovnik enote periode |
| QPOSLAT | eQEP Position Counter Latch Register, register položaja |
| QCTMR | eQEP Capture Timer Register, register časovnika, ki meri čas |
| QCPRD | eQEP Capture Periode Register; register, ki hrani vrednost časovnika |
| CAPCLK | Capture Timer Clock Prescaler, delilnik ure |

| | |
|------------|---|
| UPEVENT | Unit Position Event Prescaler, delilnik dogodkov |
| QPOSMAX | Maximum Position Count Register, maksimalni položaj |
| FIFO | First In First Out |
| SPISIMO | SPI Slave In Master Out, vhod za sužnja, izhod za gospodarja |
| SPISOMI | SPI Slave Out Mater In, Izhod za sužnja, vhod za gospodarja |
| LSPCLK | Low Speed Prescaler Clock, delilnik ure |
| SPIBRR | SPI Boud Rate Register, register, ki vpliva na hitrost prenosa podatkov |
| LSB | Least significant bit, najmanj pomemben bit |
| MSB | Most significant bit, najpomembnejši bit |
| TBPRD | Time Base Period Register, register za nastavitve periode PWM |
| TBCTL | Time Base Control Register, nastavitveni register |
| CMPA | Counter Compare A Register, primerjalni register A |
| CMPB | Counter Compare B Register, primerjalni register B |
| AQCTLA | Action-Qualifier Output A Control Register, nastavitveni register za obliko PWM signala |
| AQCTLB | Action-Qualifier Output B Control Register, nastavitveni register za obliko PWM signala |
| DBCTL | Dead-Band Generator Control Register, nastavitveni register za mrtvo področje |
| DBRED | Dead-Band Generator Rising Edge Delay Register, zakasnitev pozitivne fronte |
| DBFED | Dead-Band Generator Falling Edge Delay Register, zakasnitev negativne fronte |
| PIE | Periferial Interrupt Expansion, razširitev prekinitev |
| NMI | Nonmaskable Interrupt, nemaskirana prekinitev |
| INT1-INT14 | od 1 do 14 maskiranih prekinitev |

| | |
|---------|--|
| RTOSINT | Real Time Operating System Interrupt, prekinitev za podporo operacijskega sistema v realnem času |
| DLOGINT | Data Log Interrupt, prekinitev za podporo pri razhroščevanju |
| PIEIFR | PIE Interrupt Flag Register, register zastavic v PIE |
| PIEIER | PIE Interrupt Enable Register, register za omogočitev prekinitev |
| PIEACK | PIE Acknowledge Register, register za ponovno omogočitev prekinitev |
| IFR | Interrupt Flag Register, register zastavic za maskirane prekinitve |
| IER | Interrupt Enable Register, register za omogočitev prekinitev |
| MIMO | Multi Inputs Multi Outputs, sistem z večimi vhodi oz. izhodi |
| LQR | Liner Quadratic Regulator, linearni kvadratični regulator |

1 UVOD

Diskretne regulacije so danes eno izmed najpomembnejših področij znanosti in tehnologije. Skoraj ni več regulacijskega procesa oz. sistema vodenja, kjer ne bi našli diskretnega regulatorja. Prvotne analogne regulatorje so namreč skoraj v celoti zamenjali digitalni (diskretni) regulacijski sistemi. Velik razmah diskretnih regulacij v zadnjih 20-tih letih gre pripisati predvsem hitremu razvoju procesorjev. Zaradi njihove množične proizvodnje, predvsem pa zaradi relativno nizke cene, jih danes zelo pogosto uporabljamo tudi v regulacijske namene. Poznavanje diskretnih regulacij in ustreznih procesorjev, ki predstavljajo osnovna orodja za realizacijo, je tako bistvenega pomena v avtomatizaciji.

Diskretni regulacijski sistemi tvorijo tako zvezne (analogne) kot tudi diskretne komponente. Regulacijski objekti ali procesi so praviloma zvezne komponente, regulatorji pa diskretne. Senzorji in aktuatorji so lahko eno ali drugo. Glede na prisotnost zveznih in diskretnih komponent lahko tako konfiguracijo imenujemo hibridni regulacijski sistem. [1]

Glede na prisotnost zveznih in diskretnih komponent se tako pojavi potreba po ustrezni pretvorbi zveznih signalov v diskretne. Po diskretni obdelavi podatkov pa je potrebno rezultat prevesti nazaj v zvezno obliko in ga takega dovesti zveznemu sistemu.

Inverzno rotacijsko nihalo (INR) je v strokovni literaturi znano kot Furuta Pendulum. Ime je dobilo po avtorju, ki je prvi izpostavil problem inverznega nihala. Raziskave inverznega nihala so bile nujne zaradi težnje po zasnovi krmilnika za vodenje raket. Izstreljena raketa ima podobne lastnosti kot inverzno nihalo. Gre za nestabilen sistem, ki ga je potrebo ustrezno manevrirati v navpični legi. Danes pa se sistem IRN v splošnem uporablja za študij vodenja in regulacij nelinearnih sistemov.

Inverzno rotacijsko nihalo z digitalnim signalnim procesorjem (DSP) predstavlja hibridni sistem. Preden pa postane ta sistem regulacijski sistem je potrebno zagotoviti ustrezne funkcionalnosti na DSP-ju. DSP predstavlja bistvo izvajanja procesov, ki so povezani z vodenjem oz. s stabilizacijo inverznega nihala. Tako je potrebno zagotoviti ustrezno zajemanje fizikalnih veličin in obdelavo podatkov. Rezultat procesiranja pa mora biti v ustrezni obliki priveden zveznemu sistemu IRN.

V naslednjih poglavjih je podan natančnejši opis objekta vodenja oz. IRN, senzorjev, aktuatorjev, DSP-ja in razvojnega okolja. Nato je podan matematični opis IRN, ki je esencialen za izračun algoritmov vodenja. Zaradi same narave sistema, ki je izredno nelinearna, izračun algoritmov vodenja ni enostaven. Tako smo s pomočjo matematičnega modela, simulacij in v končni fazi z eksperimenti podali diskretne regulatorje.

Sledi natančen opis in opis izvedbe vseh potrebnih funkcionalnosti DSP-ja za zajemanje merilnih veličin in procesiranje regulacijskih algoritmov. Programski gradniki oz. posamezne komponente DSP-ja so bile izvedene tako, da so prilagodljive in primerne za najrazličnejše sisteme vodenja. Tako je bilo zgrajeno osnovno okolje, ki se bo lahko v prihodnje uporabljalo za različne sisteme vodenja za potrebe Laboratorija za sisteme in vodenje.

Ob koncu so podani izvedeni algoritmi vodenja in rezultati stabilizacije IRN.

2 OPIS SISTEMA VODENJA IN RAZVOJNIH KOMPONENT

Sistem vodenja predstavlja IRN z vrsto najrazličnejših komponent, ki jih bomo natančneje predstavili v tem poglavju.

2.1 Inveržno nihalo Apparatus Kri PP-300

Mehanski sistem Apparatus Kri PP-300 sestavlja mehansko ogrodje, nihalo, električni motor, napajalnik in procesor s prilagoditvenim vezjem.

Koncept stabilizacije inverznega nihala predstavlja na električni motor pritrjena prečka, na kateri je preko ležaja vpeto nihalo. Tako z ustreznim vodenjem električnega motorja vrtimo prečko v horizontalni ravnini, kar neposredno vpliva na položaj nihala. Merjene veličine so položaj nihala, položaj prečke in hitrost prečke. Položaj nihala merimo s potenciometrom, ki je nameščen za ležajem nihala. Položaj in hitrost prečke pa merimo z enkoderjem.

Ideja je stabilizacija nihala v navpični legi.

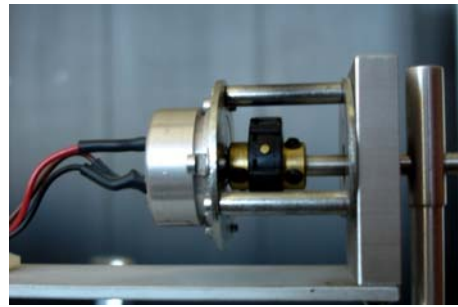


Slika 2.1: Inveržno rotacijsko nihalo

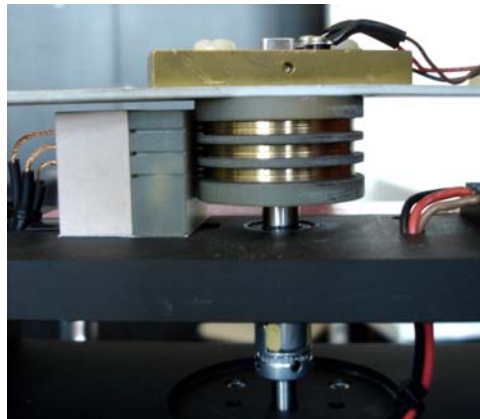
2.2 Potenciometer

S pomočjo 10 k Ω potenciometra, ki je nameščen za ležajem nihala lahko merimo položaj nihala. Potenciometer je napajan z napetostjo 5 V. Tako signal v območju od 0-5 V predstavlja položaj nihala 0-360 °. Ta napetostni signal merimo z AD pretvornikom. Zaradi nelinearnosti potenciometra je možno meriti na področju od 0-340 ° \pm 3 °.

Prenos signalov iz potenciometra je bil prvotno izveden preko žičnih drsnikov, ki pa niso predstavljali konstantnega trenja na os motorja, saj so žični drsniki popuščali in jih je bilo potrebno vedno znova zapenjati k osi motorja. To izvedbo prenosa signala smo nadomestili s ščetkami. Tako je postalo trenje konstantno, saj ščetke s konstantno silo pritiskajo na os motorja. Zmanjšali smo tudi silo lepenja.



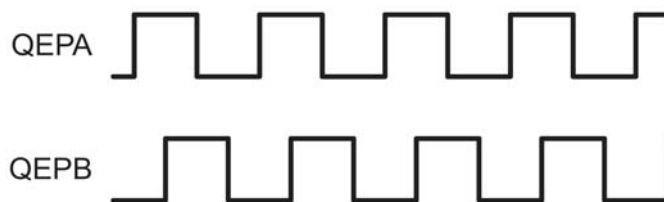
Slika 2.2: Potenciometer za merjenje položaja nihala



Slika 2.3: Prenos signalov je izveden preko ščetk

2.3 Enkoder

Enkoder omogoča merjenje položaja in hitrosti prečke s štetjem dogodkov. Dogodek predstavlja vsaka pozitivna ali negativna fronta signala A ali B, ki sta medsebojno premaknjena za 90° . Enkoder ima 1000 linij. Glede na to, da lahko z DSP-jem zaznavamo pozitivne in negativne fronte, lahko zaznamo 4000 dogodkov na obrat. Hitrost motorja je mogoče meriti tudi z vgrajenim tako merilnikom, ampak v tem delu smo se odločili za uporabo enkoderja.



Slika 2.4: Izhodna signala iz enkoderja [2]

2.4 Napajalnik

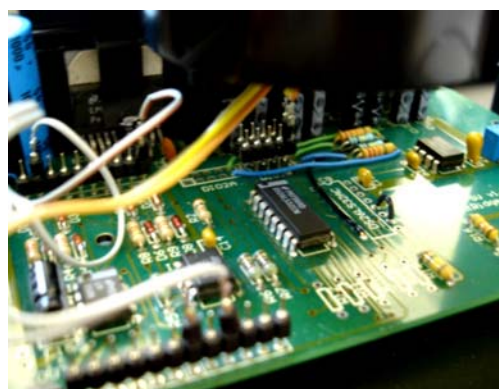
Omrežno napetost 230 V pretvori v 17 V enosmerne napetosti, ki napaja procesor in razvojno ploščo za pogon motorja.



Slika 2.5: Napajalnik

2.5 Razvojna plošča za pogon motorja

Na razvojni plošči za pogon motorja je prilagoditveno vezje, ki prilagaja napetostne nivoje med procesorjem in motorjem ter med procesorjem in senzorji. Na tej plošči je bil vgrajen procesor družine INTEL MCS-96 N80C196KC-20. Na razvojni plošči je H-mostič, ki zagotavlja motorju do 6 A toka v obliki pulzno širinsko moduliranega signala (PWM). H-mostič vodimo s procesorjem po principu PWM. Vgrajena je termična zaščita. Če temperatura preseže 170 °C se vsi izhodi izklopijo. Na razvojni plošči so prilagoditvena vezja za potenciometer, enkoder in tako. Zraven prilagoditvenih komponent, najdemo tukaj še napetostne regulatorje, ki prilagodijo napetost napajalnika, iz 17 V na ustrezna napetostna nivoja, 5 V za napajanje procesorja in 12 V za napajanje motorja.



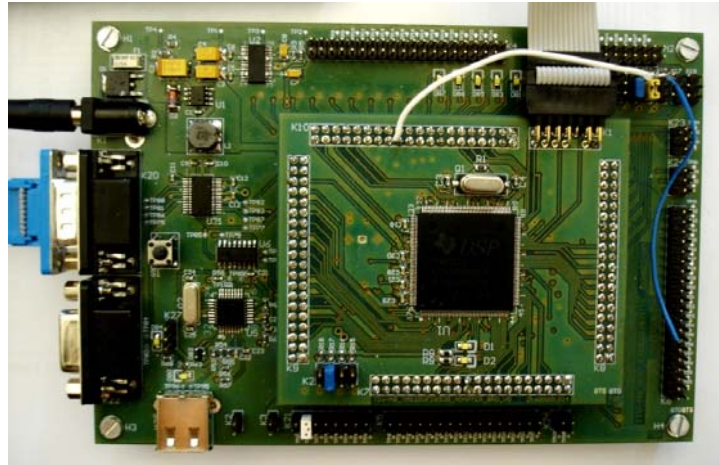
Slika 2.6: Razvojna plošča za pogon motorja

2.6 Migracija iz INTEL MCS-96 N80C196KC-20 na TMS320F28335

Sistem vodenja oz. IRN je bilo zasnovano z Intelovim procesorjem INTEL MCS-96 N80C196KC-20. Ideja je bila nadomestiti alternativni procesor z zmogljivejšim procesorjem TMS320F28335 proizvajalca Texas Instruments. Zaradi integracije drugega procesorja so se pojavile težave z obstoječo razvojno ploščo za pogon motorja. Tako jo je bilo potrebno prilagoditi novemu procesorju, ki pa ima za razliko od njegovega predhodnika namesto pet voltnih nivojev le tri voltne. Obstoječa prilagoditvena vezja so bila ustrezno zasnovana tako, da ni bilo potrebe po večjih spremembah. Potenciometer je bil napajen s petimi volti. Ustrezno prilagojen signal smo dobili z vgradnjo enostavnega napetostnega delilnika. Napetost namreč ni smela presežati treh voltov, saj bi lahko prišlo do poškodb analogno digitalnega (AD) pretvornika. Tudi enkoder je bil napajen s petimi volti. Za Texasov procesor smo morali zagotoviti tri voltne vhodni signal. Tudi tukaj smo uporabili enostavno rešitev prilagoditve napetosti z napetostnim delilnikom. Dodali smo še napetostni sledilnik. PWM signal za vodenje motorja iz procesorja je tudi tri volten, ampak to, za vodenje napajalnega vezja, povsem zadostuje.

2.7 Razvojna plošča z DSP TMS320F28335

DSP spada v družino *C2000*, ki zajema 32-bitne mikrokontrolerje namenjene aplikacijam vodenja v realnem času. *DSP* ima vgrajenih kar nekaj funkcionalnosti, ki omogočajo najrazličnejše aplikacije. Vgrajeno ima komunikacijsko periferijo kot so *SPI*, *SCI*, *I2C*, *CAN* in *McBSP*; *AD* pretvornik; omogoča izhode v obliki *PWM*; ima vgrajene časovnike; omogoča operiranje s prekinitvami; vgrajen ima sistem za enkoder oz. za štetje zunanjih dogodkov; vgrajen ima *DMA*, ki periferiji omogoča neposreden dostop do pomnilnika. Jedro omogoča operacije s plavajočo vejico, kar poviša hitrost delovanja procesorja. [3]



Slika 2.7: Razvojna plošča z DSP TMS320F28335

V tem delu smo razvili programsko podporo za naslednje komponente:

- *SPI* za komunikacijo z zunanjim *AD* pretvornikom,
- *SCI* za komunikacijo z osebnim računalnikom,
- notranji *AD* pretvornik,
- podpora za *PWM*,
- uporabili smo kar nekaj prekinitev tako smo lažje zagotovili časovno deterministične lastnosti izvajanja algoritmov,
- enoto za enkoder, ki omogoča merjenje hitrosti in položaja motorja.

Na razvojni plošči, kjer je vgrajen procesor se nahaja 22 bitni zunanji *AD* pretvornik *ADS1213*. Zaradi kvalitetnejše izvedbe zunanjega *AD* pretvornika v primerjavi z vgrajenim *AD* pretvornikom in določenih dodatnih zmožnostih, smo se odločili za prvega.

Na razvojni plošči najdemo še vrsto drugih stvari, ki pa jih v tem delu nismo uporabili, zato jih bomo samo omenili. Na plošči so še: zunanji *RAM*, *EEPROM*, dva zunanja *FLASH*-a, *LED* diode, integriran pretvornik *USB/UART*.

Tabela 1: Lastnosti procesorja [3]

| | | | |
|------------------|-------------------|------------------|--------------------|
| Vrsta procesorja | TMS320F28335 | Čas AD pretvorbe | 80 ns |
| Frekvenca (MHz) | 150 | McBSP | 2 |
| RAM | 68 KB | I2C | 1 |
| OTP ROM | 2 KB | UART | 3 SCI |
| Flash | 512 KB | SPI | 1 |
| DMA | 6-kanalov DMA | CAN | 2 |
| PWM | 18-kanalov | Časovniki | 3 32-Bit CPU, 1 WD |
| CAP/QEP | 6/2 | GPIO | 88 |
| ADC | 16-kanalov 12-Bit | | |

2.8 Razvojno okolje Code Composer Studio 3.3

Osnovno orodje za delo z DSP je bila programska oprema Code Composer Studio 3.3. Podpira jezik ANSI C. Predstavlja urejevalnik programske kode, prevajalnik, povezovalnik in razhroščevalnik. Za razhroščevanje in nalaganje programa na DSP potrebujemo še emulator oz. JTAG. Pri tem delu je bil uporabljen emulator XDS510.



Slika 2.8: JTAG XDS510

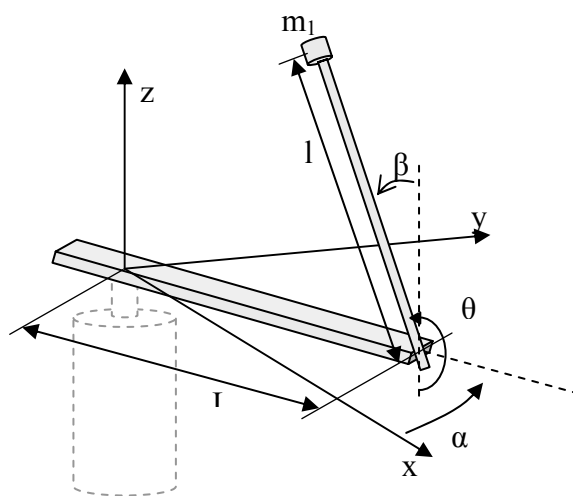
3 MATEMATIČNI MODEL IRN IN OCENA PARAMETROV

Matematični model je matematični opis sistema, ki je fizikalnemu objektu podoben v vseh najbolj bistvenih lastnostih [4]. Za izračun regulatorjev oz. algoritmov vodenja potrebujemo matematični model, ki kar se da natančno opisuje realni fizikalni objekt. Matematični model IRN je bil izpeljan s pomočjo Lagrange-ovih enačb gibanja. IRN velja za nelinearen sistem. Algoritme vodenja pa lahko izračunamo le za linearne sisteme, tako bo potrebno kompleksen sistem nelinearnih enačb linearizirati.

Ker načrtovanje regulatorjev ni enostavno, smo si pomagali tudi s simulacijami. Matematične modele nihala smo izvedli v Simulinku. Simulacije so nam bile v pomoč pri iskanju rešitev regulacijskih algoritmov. Regulatorje, ki smo jih izračunali z linearnimi modeli, smo preizkusili na nelinearnih modelih. Tako smo prišli do ustreznih rešitev za realni objekt.

Matematični model IRN je bil prevzet po [5], prav tako parametri modela. Model je izpeljan v dodatku .

Matematični model je osnovan na desno sučnem kartezijskem koordinatnem sistemu. Izhodišče kota β je v navpični legi nihala, kota θ pa v spodnji legi.



Slika 3.1: Model nihala

3.1 Nelinearni model

Nelinearni model IRN natančno opisuje odvisnosti med posameznimi spremenljivkami. Linearni model je izpeljan iz nelinearnega kot je razvidno iz izpeljav v dodatku. Nelinearni matematični model IRN:

$$\begin{bmatrix} J_0 + m_1 \cdot L_0^2 + m_1 \cdot l_1^2 \cdot \sin^2 \beta & -m_1 \cdot L_0 \cdot l_1 \cdot \cos \beta \\ -m_1 \cdot L_0 \cdot l_1 \cdot \cos \beta & J_1 + m_1 \cdot l_1^2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} C_0 + \frac{K_t \cdot K_b}{R_a} + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\beta} \cdot \sin 2\beta & m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta} \cdot \sin \beta + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\beta \\ -\frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\beta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} 0 \\ -m_1 \cdot g \cdot l_1 \cdot \sin \beta \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R_a} \\ 0 \end{bmatrix} V_m$$

Nelinearni matematični model, ki opisuje nihalo v običajni, spodnji legi:

$$\begin{bmatrix} J_0 + m_1 \cdot L_0^2 + m_1 \cdot l_1^2 \cdot \sin^2 \theta & m_1 \cdot L_0 \cdot l_1 \cdot \cos \theta \\ m_1 \cdot L_0 \cdot l_1 \cdot \cos \theta & J_1 + m_1 \cdot l_1^2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} C_0 + \frac{K_t \cdot K_b}{R_a} + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\theta} \cdot \sin 2\theta & -m_1 \cdot L_0 \cdot l_1 \cdot \dot{\theta} \cdot \sin \theta + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\theta \\ -\frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\theta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \end{bmatrix} \quad (3.2)$$

$$\begin{bmatrix} 0 \\ m_1 \cdot g \cdot l_1 \cdot \sin \theta \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R_a} \\ 0 \end{bmatrix} V_m$$

Če vpeljemo naslednje relacije, dobimo matrični zapis, ki ga bomo uporabljali za nadaljnje manipulacije.

$$A = J_0 + m_1 \cdot L_0^2 \quad (3.3)$$

$$B = m_1 \cdot l_1^2 \quad (3.4)$$

$$C = m_1 \cdot L_0 \cdot l_1 \quad (3.5)$$

$$D = C_0 + \frac{K_t \cdot K_b}{R_a} \quad (3.6)$$

$$E = \frac{K_t}{R_a} \quad (3.7)$$

$$F = J_1 + m_1 \cdot l_1^2 \quad (3.8)$$

$$H = m_1 \cdot g \cdot l_1 \quad (3.9)$$

S pomočjo podanih konstant enačbo (3.1) preoblikujemo v naslednjo obliko:

$$\begin{aligned} & \begin{bmatrix} A + B \cdot \sin^2 \beta & -C \cdot \cos \beta \\ -C \cdot \cos \beta & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} \\ & \begin{bmatrix} D + \frac{1}{2} B \cdot \dot{\beta} \cdot \sin 2\beta & C \cdot \dot{\beta} \cdot \sin \beta + \frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\beta \\ -\frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\beta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \\ & \begin{bmatrix} 0 \\ -H \cdot \sin \beta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m \end{aligned} \quad (3.10)$$

Podan sistem nelinearnih enačb je potrebno preoblikovati tako, da bodo primerne za izdelavo modela v Simulinku.

Matrični zapis (3.10) bomo preoblikovali v enačbi z uporabo naslednjih konstant:

$$a_1 = A + B \cdot \sin^2 \beta \quad (3.11)$$

$$a_2 = D + \frac{1}{2} B \cdot \dot{\beta} \cdot \sin 2\beta \quad (3.12)$$

$$b_1 = b = -C \cdot \cos \beta \quad (3.13)$$

$$b_2 = C \cdot \dot{\beta} \cdot \sin \beta + \frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\beta \quad (3.14)$$

$$c_1 = b = -C \cdot \cos \beta \quad (3.15)$$

$$c_2 = -\frac{1}{2}B \cdot \dot{\alpha} \cdot \sin 2\beta \quad (3.16)$$

$$d_1 = F \quad (3.17)$$

$$d_2 = C_1 \quad (3.18)$$

$$d_3 = -H \cdot \sin \beta \quad (3.19)$$

Nelinearni sistem opisujeta naslednji enačbi:

$$a_1\ddot{\alpha} + b_1\ddot{\beta} + a_2\dot{\alpha} + b_2\dot{\beta} = E \cdot V_m \quad (3.20)$$

$$c_1\ddot{\alpha} + d_1\ddot{\beta} + c_2\dot{\alpha} + d_2\dot{\beta} + d_3 = 0 \quad (3.21)$$

Enačbo (3.20) pomnožimo z d_1 , enačbo (3.21) pa z b . Ob upoštevanju enačb (3.13) in (3.15) dobimo:

$$a_1 \cdot d_1 \cdot \ddot{\alpha} + b \cdot d_1 \cdot \ddot{\beta} + a_2 \cdot d_1 \cdot \dot{\alpha} + b_2 \cdot d_1 \cdot \dot{\beta} = E \cdot d_1 \cdot V_m \quad (3.22)$$

$$b^2 \cdot \ddot{\alpha} + b \cdot d_1 \cdot \ddot{\beta} + b \cdot c_2 \dot{\alpha} + b \cdot d_2 \dot{\beta} + b \cdot d_3 = 0 \quad (3.23)$$

Enačbi odštejemo in dobimo:

$$(a_1 d_1 - b^2) \ddot{\alpha} + (a_2 d_1 - b c_2) \dot{\alpha} + (b_2 d_1 - b d_2) \dot{\beta} - b d_3 = E d_1 V_m \quad (3.24)$$

Najvišji odvod izpostavimo:

$$\ddot{\alpha} = -\frac{(a_2 d_1 - b c_2)}{(a_1 d_1 - b^2)} \dot{\alpha} - \frac{(b_2 d_1 - b d_2)}{(a_1 d_1 - b^2)} \dot{\beta} + \frac{b d_3}{(a_1 d_1 - b^2)} + \frac{E d_1 V_m}{(a_1 d_1 - b^2)} \quad (3.25)$$

V enačbo (3.25) ponovno vstavimo konstante (3.11)-(3.19), po ureditvi dobimo enačbo primerno za Simulink:

$$\begin{aligned}
\ddot{\alpha} = & -\frac{DF}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha} + \frac{1}{2} \frac{BC \cos \beta \sin 2\beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha}^2 \\
& - \frac{C_1 C \cos \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\beta} + \frac{CF \sin \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\beta}^2 \\
& - \frac{BF \sin 2\beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha} \dot{\beta} + \frac{CH \cos \beta \sin \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \\
& + \frac{EF}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} V_m
\end{aligned} \tag{3.26}$$

Enačbo (3.21) pomnožimo z a_1 , enačbo (3.20) pa z b . Ob upoštevanju enačb (3.13) in (3.15) dobimo:

$$a_1 \cdot b \cdot \ddot{\alpha} + a_1 \cdot d_1 \cdot \ddot{\beta} + a_1 \cdot c_2 \dot{\alpha} + a_1 \cdot d_2 \dot{\beta} + a_1 \cdot d_3 = 0 \tag{3.27}$$

$$a_1 \cdot b \cdot \ddot{\alpha} + b^2 \cdot \ddot{\beta} + a_2 \cdot b \cdot \dot{\alpha} + b_2 \cdot b \cdot \dot{\beta} = E \cdot b \cdot V_m \tag{3.28}$$

Enačbi odštejemo in dobimo:

$$(a_1 d_1 - b^2) \ddot{\beta} + (a_1 c_2 - a_2 b) \dot{\alpha} + (a_1 d_2 - b b_2) \dot{\beta} + a_1 d_3 = -E b V_m \tag{3.29}$$

Najvišji odvod izpostavimo:

$$\ddot{\beta} = -\frac{(a_1 c_2 - a_2 b)}{(a_1 d_1 - b^2)} \dot{\alpha} - \frac{(a_1 d_2 - b b_2)}{(a_1 d_1 - b^2)} \dot{\beta} - \frac{a_1 d_3}{(a_1 d_1 - b^2)} - \frac{E b V_m}{(a_1 d_1 - b^2)} \tag{3.30}$$

V enačbo (3.30) ponovno vstavimo konstante (3.11)-(3.19), po ureditvi dobimo enačbo primerno za Simulink:

$$\begin{aligned}
\ddot{\beta} = & -\frac{DC \cos \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha} + \frac{1}{2} \frac{AB \sin 2\beta + B^2 \sin 2\beta \sin^2 \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha}^2 \\
& - \frac{AC_1 + BC_1 \sin^2 \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\beta} - \frac{C^2 \sin \beta \cos \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\beta}^2 \\
& - \frac{BC \sin 2\beta \cos \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \dot{\alpha} \dot{\beta} + \frac{AH \sin \beta + BH \sin^3 \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} \\
& + \frac{CE \cos \beta}{(AF + BF \sin^2 \beta - C^2 \cos^2 \beta)} V_m
\end{aligned} \tag{3.31}$$

3.2 Linearni model

Linearni model je potreben za izračun regulacijskih algoritmov. S simulacijami smo preverili izračunane regulatorje. Linearni model za IRN z upoštevanjem enačb (3.3)-(3.9):

$$\begin{bmatrix} A & -C \\ -C & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} D & 0 \\ 0 & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -H \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m \tag{3.32}$$

Za običajno nihalo se spremenijo predznaki konstant C in H :

$$\begin{bmatrix} A & C \\ C & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} D & 0 \\ 0 & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m \tag{3.33}$$

Za realizacijo modela v Simulinku je potrebno enačbe preoblikovati.

$$A\ddot{\alpha} - C\ddot{\beta} + D\dot{\alpha} - EV_m = 0 \tag{3.34}$$

$$-C\ddot{\alpha} + F\ddot{\beta} + C_1\dot{\beta} - H\beta = 0 \tag{3.35}$$

Enačbo (3.34) pomnožimo s C , (3.35) pa z A . Nato enačbi odštejemo in dobimo:

$$(AF - C^2)\ddot{\beta} + CD\dot{\alpha} + AC_1\dot{\beta} - AH\beta - CEV_m = 0 \tag{3.36}$$

Izpostavimo najvišji odvod:

$$\ddot{\beta} = -\frac{CD}{(AF-C^2)}\dot{\alpha} - \frac{AC_1}{(AF-C^2)}\dot{\beta} + \frac{AH}{(AF-C^2)}\beta + \frac{CE}{(AF-C^2)}V_m \quad (3.37)$$

Enačbo (3.34) pomnožimo z F , (3.35) pa s C . Nato enačbi odštejemo in dobimo:

$$(AF-C^2)\ddot{\alpha} + DF\dot{\alpha} + CC_1\dot{\beta} - CH\beta - EFV_m = 0 \quad (3.38)$$

Izpostavimo najvišji odvod:

$$\ddot{\alpha} = -\frac{DF}{(AF-C^2)}\dot{\alpha} - \frac{CC_1}{(AF-C^2)}\dot{\beta} + \frac{CH}{(AF-C^2)}\beta + \frac{EF}{(AF-C^2)}V_m \quad (3.39)$$

Enačbi (3.37) in (3.39) lahko zapišemo v prostoru stanj tretjega reda:

$$\begin{bmatrix} \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = -\frac{1}{AF-C^2} \begin{bmatrix} -DF & CH & -CC_1 \\ 0 & 0 & AF-C^2 \\ -CD & AH & -AC_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \frac{1}{(AF-C^2)} \begin{bmatrix} EF \\ 0 \\ CE \end{bmatrix} V_m \quad (3.40)$$

Kot je razvidno iz enačb (3.37) in (3.39) dobimo sistem četrtega reda:

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = -\frac{1}{AF-C^2} \begin{bmatrix} 0 & AF-C^2 & 0 & 0 \\ 0 & -DF & CH & -CC_1 \\ 0 & 0 & 0 & AF-C^2 \\ 0 & -CD & AH & -AC_1 \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \frac{1}{(AF-C^2)} \begin{bmatrix} 0 \\ EF \\ 0 \\ CE \end{bmatrix} V_m \quad (3.41)$$

Kot nakazujeta enačbi (3.40) in (3.41) je sistem IRN lahko tretjega ali četrtega reda. Iz enačb (3.37) in (3.39) je razvidno, da nobeno stanje ni neposredno odvisno od položaja prečke (α). Tako sistem četrtega reda dobimo tako, da to stanje enostavno dodamo sistemu tretjega reda.

3.3 Realizacija modelov v Simulinku

Tako kot modeli, so bili prevzeti tudi parametri [5]. Pri simulacijah smo uporabili naslednje parametre:

Tabela 2: Parametri sistema

| POMEN | SIMBOL | VREDNOST | ENOTA |
|-------------------------------|--------|----------|----------------------------------|
| Navorna konstanta | K_t | 0,0706 | $\left[\frac{Nm}{A} \right]$ |
| Električna konstanta | K_b | 0,0707 | $\left[\frac{Vs}{rad} \right]$ |
| Upornost kotve | R_a | 0,9 | [Ohm] |
| Dolžina prečke | L_0 | 0,1375 | [m] |
| Masa nihala | m_1 | 0,03179 | [kg] |
| Dolžina nihala | l_1 | 0,1579 | [m] |
| Gravitacijska konstanta | g | 9,81 | $\left[\frac{m}{s^2} \right]$ |
| Vztrajnostni moment prečke | J_0 | 0,008591 | $[kgm^2]$ |
| Viskozno trenje ležaja prečke | C_0 | 0,006408 | $\left[\frac{Nms}{rad} \right]$ |
| Vztrajnostni moment nihala | J_1 | 0,000217 | $[kgm^2]$ |
| Viskozno trenje ležaja nihala | C_1 | 0,000158 | $\left[\frac{Nms}{rad} \right]$ |

Parametri so bili po večini izmerjeni ali prevzeti iz dokumentacije motorja. J_0 , J_1 , C_0 , C_1 pa so bili ocenjeni z identifikacijo. Uporabili so metodo najmanjših kvadratov. Zato, ker to ni namen tega dela, smo parametre prevzeli.

Identifikacijo so izvedli pri času tipanja $T_s=0.01$ s, zato je takšen čas tipanja smiseln tudi za izbiro časa tipanja sistema vodenja. Če pogledamo odprtozančne pole sistema:

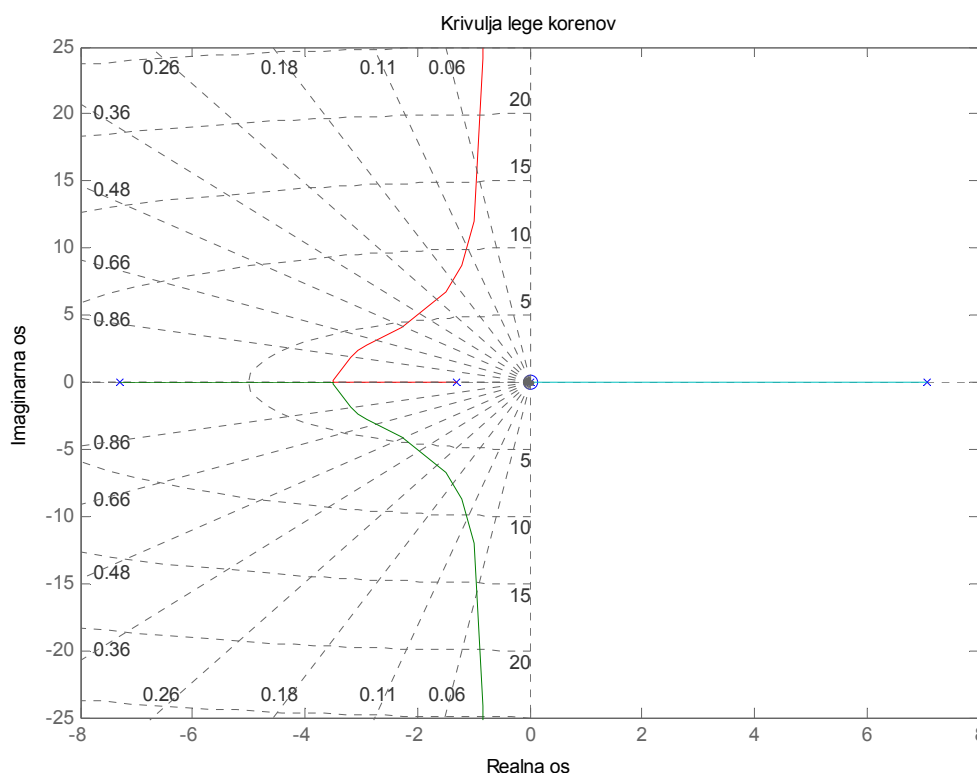
$$s_1 = -7.2967$$

$$s_1 = -1.2981$$

$$s_1 = 7.0590,$$

vidimo, da je časovna konstanta najhitrejšega pola okrog 1 s, tako je izbira časa tipanja $T_s=0.01$ s povsem upravičena.

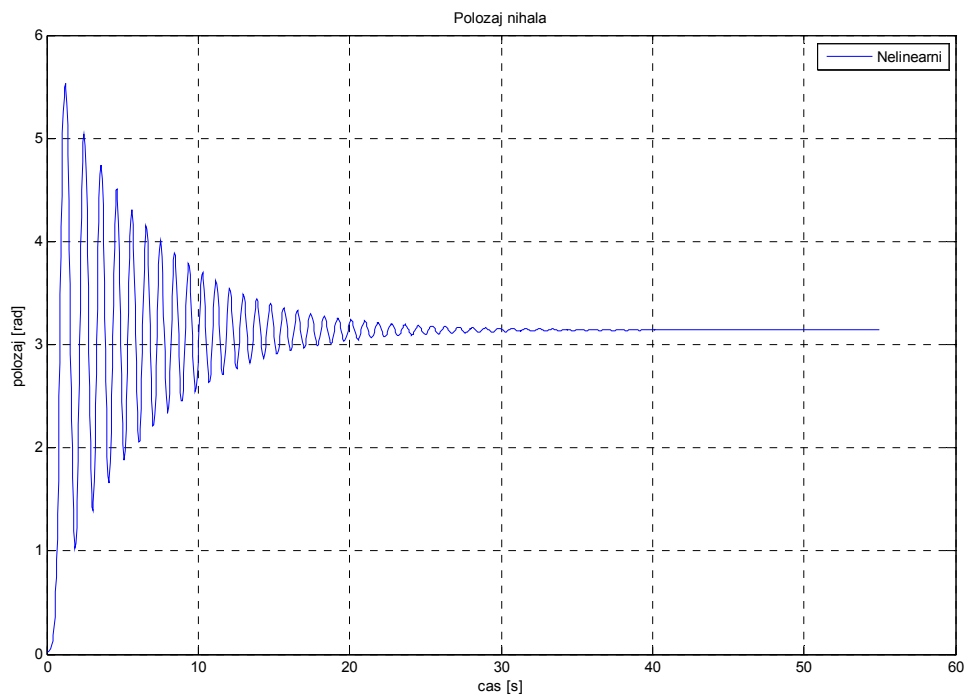
Lego odprto zančnih polov IRN prikazuje slika 3.2. Gre za nestabilen sistem, kar nakazuje lega pola, ki je pozitivna. Tako bo metoda načrtovanja regulatorja za pomikanje polov še kako smiselna.



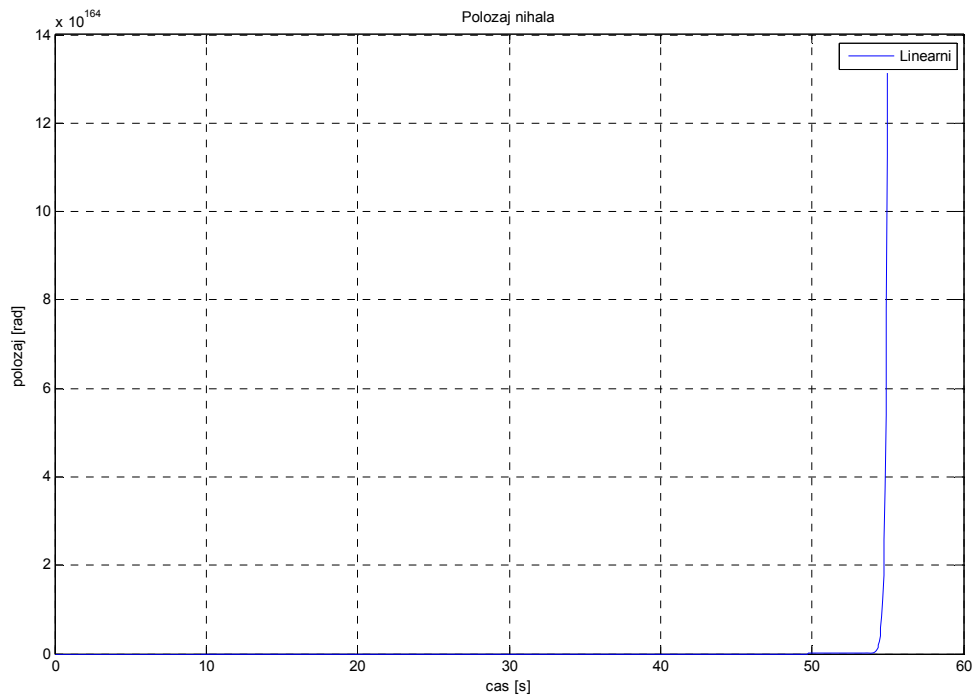
Slika 3.2: Krivulja lege korenov odprtozančnega sistema

Če v modelu nihalo le za stopinjo izmaknemo iz ravnovesne lege, pade iz stabilnosti. Na sliki 3.3 vidimo, da nihalo zaniha iz začetne lege okrog nasprotne lege, ki je za π večja od navpične lege. Pri linearnem modelu zaradi odsotnosti sinusnih in kosinusnih delov, ki smo se jih znebili pri linearizaciji, model ne predvidi nasprotne lege nihala tako natančno kot nelinearen model. Tako je lineariziran model le približek nelinearnega, saj smo pri linearizaciji z razvojem v Taylorjevo vrsto člene višjega reda zanemarili.

Kot smo že omenili lahko regulatorje izračunamo le z linearnim modelom. Tako bo potrebno izračunane regulatorje preizkusiti na nelinearnem modelu.



Slika 3.3: Odprtozančni odziv nelinearnega modela



Slika 3.4: Odprtozančni odziv linearnega modela

4 PROGRAMSKI GRADNIKI KOMPONENT PROCESORJA

Kot smo že ugotovili v prejšnjem poglavju, lahko sistem IRN opišemo s sistemom tretjega ali četrtega reda. IRN lahko stabiliziramo oz. vodimo po vseh štirih stanjih ali pa samo po treh. Izvedli bomo regulatorje v prostoru stanj kot bomo videli v nadaljevanju. Pogoj za vodenje sistema v prostoru stanj je, da imamo vsak periodičen čas tipanja informacije o notranjih stanjih sistema. Tako moramo z DSP-jem realizirati zajemanje ustreznih meritev. Kot je razvidno iz (3.40) in (3.41) potrebujemo stalne informacije o položaju prečke (α), hitrosti prečke ($\dot{\alpha}$), položaju nihala (β) in hitrosti nihala ($\dot{\beta}$), če vodimo IRN kot sistem četrtega reda in informacije o hitrosti prečke ($\dot{\alpha}$), položaju nihala (β) in hitrosti nihala ($\dot{\beta}$), če vodimo IRN kot sistem tretjega reda.

Hitrost in položaj prečke bomo zajemali z enkoderjem. Enkoder je naprava, ki nam daje merilni rezultat v obliki dveh medsebojno zamaknjenih kvadratastih napetosti. DSP ima vgrajen poseben strojni sistem, ki omogoča zaznavanje dogodkov v obliki pozitivnih in negativnih front kvadratastega signala.

Položaj nihala bomo merili s pomočjo potenciometra, ki je nameščen za ležajem nihala. Ob spremembi položaja nihala, se spremeni upornost na potenciometru, to pa zaznamo v obliki spremembe padca napetosti na potenciometru, ki ga merimo z AD pretvornikom. Hitrost nihala pa določimo s spremembo poti nihala v dveh sosednjih časih tipanja.

Z izmerjenimi vrednostmi in izračunano hitrostjo nihala regulator izračuna ustrezen vhod v sistem IRN v obliki širine PWM pulza, ki določa iznos enosmerne napetosti na motorju.

Preden se lotimo načrtovanja regulatorjev je na DSP-ju potrebno realizirati vse ustrezne komponente:

- enoto za enkoder, ki omogoča merjenje hitrosti in položaja motorja,
- notranji AD pretvornik,
- podpora za zunanji AD pretvornik in SPI za komunikacijo z zunanjim AD pretvornikom,

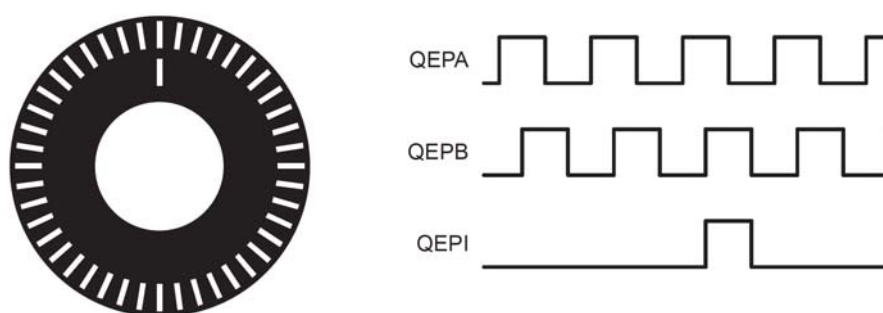
- SCI za komunikacijo z osebnim računalnikom,
- podpora za PWM, za vodenje motorja
- realizacija prekinitev, tako smo lažje zagotovili časovno deterministične lastnosti izvajanja algoritmov.

V nadaljevanju so natančneje predstavljeni posamezni moduli DSP-ja in kako jih je bilo potrebno nastaviti.

4.1 Modul za podporo enkoderja (eQEP)

Modul za podporo enkoderja (eQEP) je namenjen linearnim ali rotacijskim inkrementalnim enkoderjem za merjenje položaja, smeri vrtenja in hitrosti. Mi smo ga uporabili za merjenje hitrosti in položaja ročice.

Enkoder sestavlja inkrementalni disk. Kot prikazuje slika 4.1 je sestavljen iz mnogih svetlih in temnih linij. Te linije pa bereta dva optična foto elementa, ki sta med seboj zamaknjena za četrtnino razdalje med dvema svetlima ali temnima linijama. Ko se inkrementalni disk vrti, optična foto elementa generirata za 90° premaknjena signala, kot prikazuje slika 4.1.



Slika 4.1: Optični enkoder [2]

Ta dva signala imenujemo QEPA in QEPB. Pozitivna smer urinega kazalca je definirana takrat, ko QEPA signal prehiteva signal QEPB. Z hitrostjo motorja se spreminja frekvenca signalov QEPA in QEPB. Tako lahko s štetjem pulzov QEPA in QEPB merimo hitrost in položaj. Hitrost lahko merimo na dva različna načina:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad [2] \quad (4.1)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad [2] \quad (4.2)$$

Kjer so:

$v(k)$ – hitrost v časovnem koraku k

$x(k)$ – položaj v trenutnem časovnem koraku

$x(k-1)$ – položaj v prejšnjem časovnem koraku

T – je konstanta in predstavlja čas tipanja

ΔX – sprememba poti v času tipanja

$t(k)$ – izmerjen čas v trenutnem dogodku

$t(k-1)$ – izmerjen čas v predhodnem dogodku

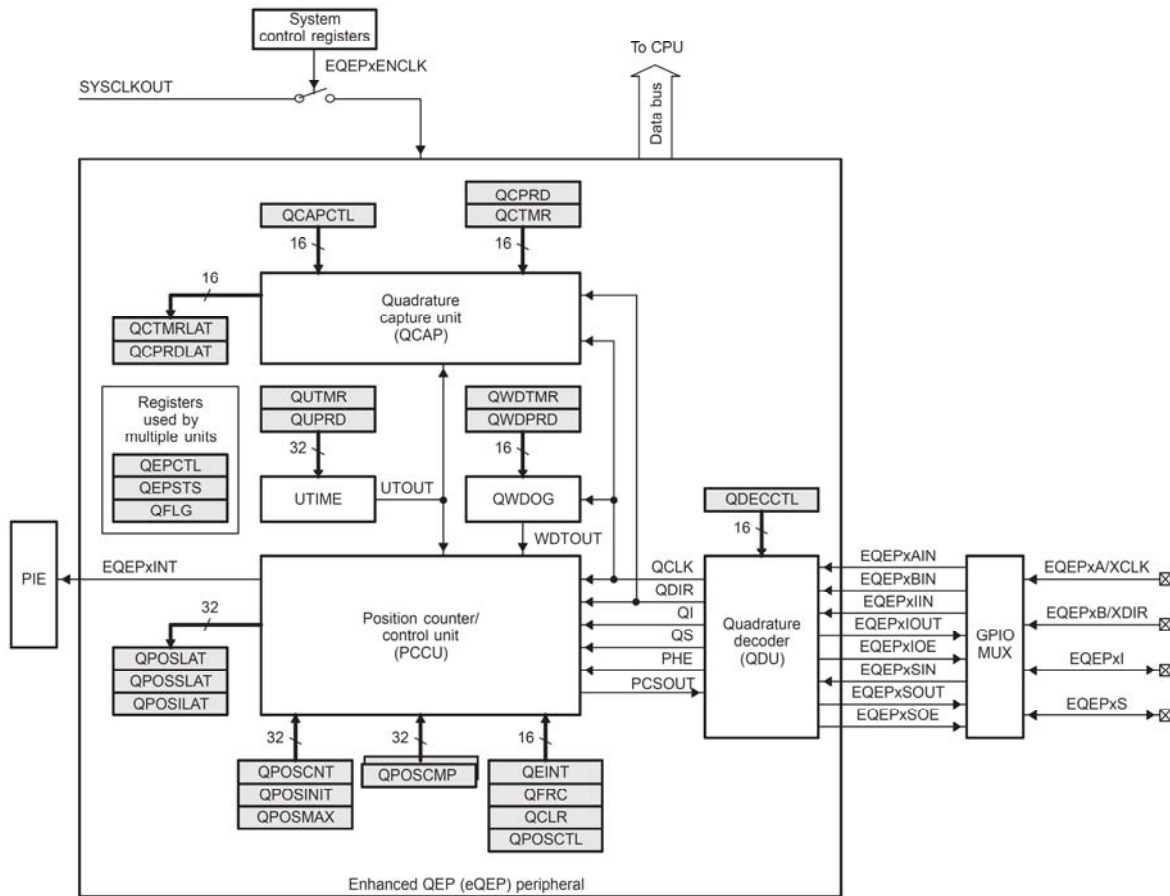
X – predstavlja konstanten položaj

ΔT – čas, ki je pretekel pri opravljeni poti X

Enačbi (4.1) in (4.2) predstavljata dva različna načina merjenja in izračunavanja hitrosti. Enačba (4.1) predstavlja število pulzov, ki jih DSP zazna v neki točno določeni periodi. Enkoder ima 1000 linij. DSP smo nastavili tako, da šteje pozitivne in negativne fronte signalov QEPA in QEPB. Tako dobimo 4000 dogodkov na obrat. Enoto periode (Unit periode) v kateri DSP šteje dogodke, smo nastavili na 0,01 s, kot znaša čas tipanja. Enoto periode nastavimo v registru QUPRD, ker vpišemo število do katerega naj števnik šteje. Čas enega časovnega kvanta je obratno sorazmeren s frekvenco jedra procesorja, ki znaša 150 MHz. Tako dobimo za register QUPRD:

$$QUPRD = 0,01s \cdot 150000000 = 150000 \quad (4.3)$$

Ob izteku periode se prepíše register položaja (QPOSLAT) s številom dogodkov, ki jih je DSP zaznal v enoti periode. V statusni register se postavi zastavica, ki javlja, da je v registru nova vrednost. Ta zastavica ima še eno funkcijo in sicer sproži prekinitvev, v kateri se obdela algoritem. Za shemo registrov oz. posameznih enot modula glej sliko 4.2.



Slika 4.2: Posamezne enote modula [2]

Ta način merjenja in izračunavanja hitrosti je primeren za višje hitrosti, saj pri velikem številu prešteti dogodkov, napaka štetja ne pride do izraza. Situacija je drugačna, ko je hitrost nizka in je število dogodkov le nekaj.

Hitrost se izračuna:

$$\omega_{visoka} = \frac{QPOSLAT \cdot 2\pi}{T_s \cdot n_E} \left[\frac{rad}{s} \right] \quad (4.4)$$

Kjer so:

ω_{visoka} - hitrost motorja

QPOSLAT – register, ki vsebuje število dogodkov v času T_s

T_s – čas tipanja

n_E – število dogodkov v enem obratu in znaša 4000

Za nizke hitrosti pa je primernejši način, ki ga podaja enačba (4.2). V tem primeru pa se meri čas med dvema dogodkoma. Opravljena pot med dvema dogodkoma pa je ena štiri tisočina obrata. V tem primeru pa je napaka števnikova časa manjša, večji je čas med dvema dogodkoma. Torej je napaka višja pri višjih hitrostih. Vrednost časovnika, ki meri čas QCTMR, se ob pripetitvi ustreznega dogodka shrani v register QCPRD¹. Vsakič, ko se register prepíše se pojavi zastavica, ki nam javi, da je v registru nova vrednost. Tako tega algoritma ne izračunavamo vsak čas tipanja, ampak le ob pripetitvi zastavice. S tem je obremenitev DSP-ja nižja.

Časovnik, ki meri čas (QCTMR) smo nastavili tako, da meri čas med vsakim drugim dogodkom kot prikazuje slika 4.3. Osnovni časovni kvant časovnika QCTMR smo povečali na:

$$QCTMR(k) - QCTMR(k-1) = \frac{CAPCLK}{P_{CLK}} = \frac{128}{150000000} \quad (4.5)$$

Kjer je:

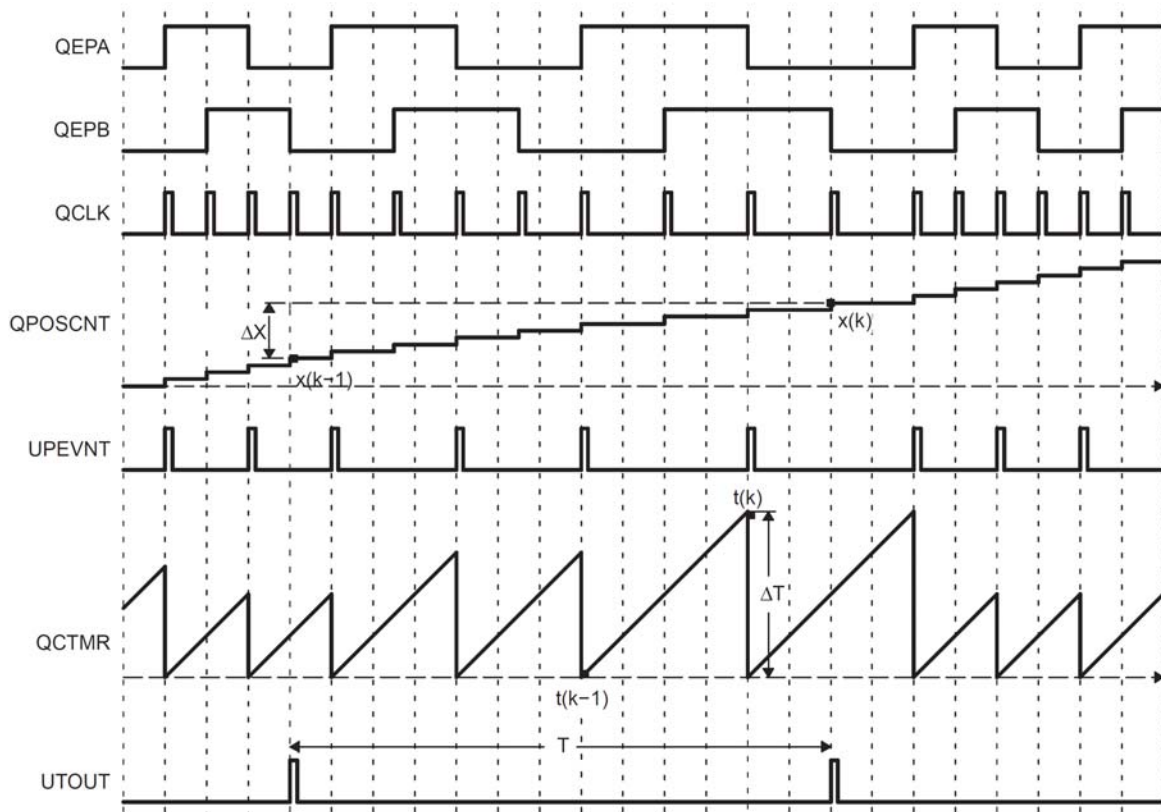
$CAPCLK$ – delilnik ure v našem primer ima vrednost 128

P_{CLK} – frekvenca procesorja znaša 150 MHz

Register QCPRD, v katerega se prepíše časovnik QCTMR je namreč 16-biten. To pomeni, da je najnižja možna hitrost, ki jo je še mogoče zaznati omejena za najvišjo možno vrednostjo registra QCPRD, ki znaša:

$$QCPRD_max = 2^{16} - 1 = 65535 \quad (4.6)$$

¹ Glej sliko 4.2.



Slika 4.3: Shema merjenja položaja in časa [2]

Tako je najnižja hitrost, ki jo lahko izmerimo:

$$\omega_{nizka} = \frac{UPEVENT \cdot P_{CLK} \cdot D_e \cdot 2\pi}{CAPCLK \cdot n_c} \left[\frac{rad}{s} \right] \quad (4.7)$$

$$\begin{aligned} \omega_{nizka} &= \frac{P_{CLK} \cdot D_e \cdot 2\pi}{CAPCLK \cdot n_c} \left[\frac{rad}{s} \right] = \frac{150 \cdot e^6 \cdot 2\pi}{128 \cdot 2000 \cdot 65535} \\ &= 0,056 \left[\frac{rad}{s} \right] = 0,53 \left[\frac{obratov}{min} \right] \end{aligned} \quad (4.8)$$

Kjer so:

ω_{nizka} - hitrost motorja

P_{CKL} – frekvenca procesorja znaša 150 MHz

D_e – del obrata, ki ga motor naredi med dogodkoma, je konstanta in znaša $\frac{1}{2000}$, glej sliko 4.3

UPEVENT – je delilnik dogodkov, med katerimi šteje časovnik

CAPCLK – delilnik ure v našem primer ima vrednost 128

n_C – število časovnih kvantov oz. vrednost registra QCPRD

Kot dokazuje enačba (4.8) je s tako konfiguracijo mogoče meriti zelo nizke hitrosti. Veliko nižje kot je potrebno.

V programu smo izvedli oba algoritma za merjenje hitrosti. Za višje hitrosti uporabljamo algoritem (4.1) oz. (4.4), za merjenje nižjih hitrosti pa algoritem (4.2) oz. (4.7). Mejo, ki določa katera hitrost ima nižji pogrešek, pa smo določili z izračunom relativnih pogreškov. Tako imamo v širokem območju merjenja hitrosti relativno nizek pogrešek. Zaradi večje nazornosti smo pogreške izračunali za hitrosti v obratih na minuto.

S pomočjo (4.4) dobimo:

$$\begin{aligned} \omega_{napaka} \left[\frac{\text{obratov}}{\text{min}} \right] &= \frac{(QPOSLAT + 1) \cdot 60}{4000 \cdot T_s} - \frac{QPOSLAT \cdot 60}{4000 \cdot T_s} \\ &= \frac{3}{200 \cdot T_s} \end{aligned} \quad (4.9)$$

$$\omega_{napaka} [\%] = \frac{\omega_{napaka}}{\omega} = \frac{3}{2 \cdot \omega \cdot T_s} \quad (4.10)$$

Za nizke hitrosti dobimo s pomočjo(4.7):

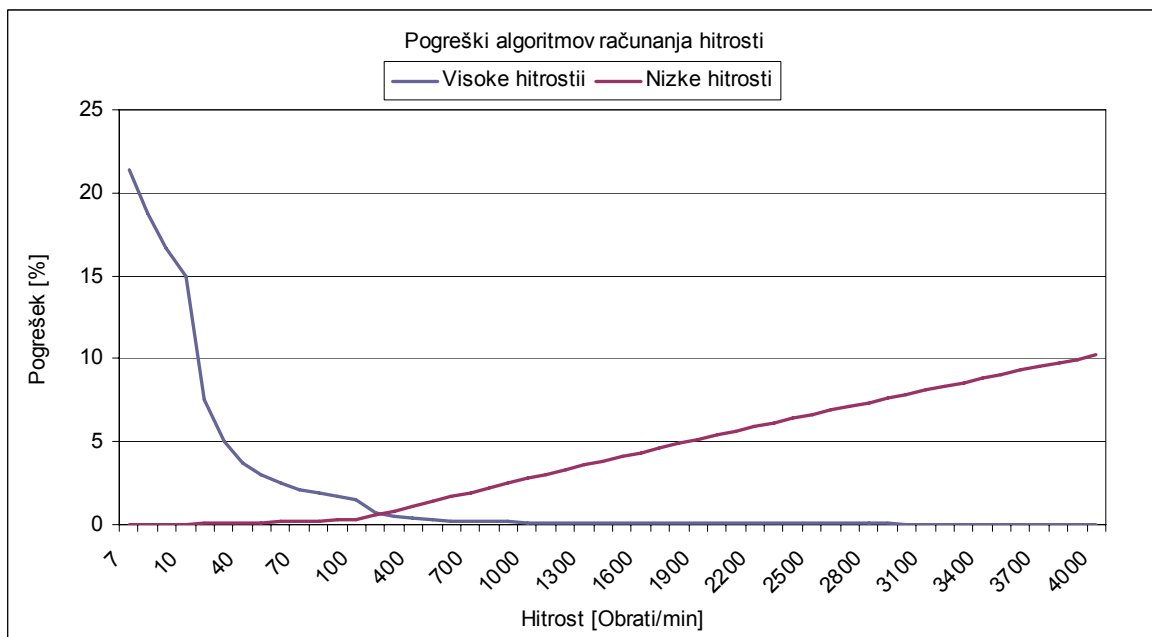
$$\begin{aligned} \omega_{napaka} \left[\frac{\text{obratov}}{\text{min}} \right] &= \frac{UPEVENT \cdot P_{CLK} \cdot D_e \cdot 60}{CAPCLK \cdot n_C} \\ &\quad - \frac{UPEVENT \cdot P_{CLK} \cdot D_e \cdot 60}{CAPCLK \cdot (n_C + 1)} \end{aligned} \quad (4.11)$$

Najprej moramo določiti število časovnih kvantov n_C v odvisnosti od hitrosti:

$$n_C = \frac{UPEVENT \cdot P_{CLK} \cdot D_e \cdot 60}{CAPCLK \cdot \omega} \quad (4.12)$$

in dobimo relativni pogrešek:

$$\omega_{napaka} [\%] = \frac{\omega - \frac{UPEVENT \cdot P_{CLK} \cdot D_e \cdot 60}{CAPCLK \cdot (n_C + 1)}}{\omega} \cdot 100 \quad (4.13)$$



Slika 4.4: Pogreški algoritmov merjenja hitrosti motorja

Tabela 3: Prikaz pogreškov algoritmov

| Hitrost [O/min] | Enačba (4.10) [%] | Enačba (4.13) [%] |
|-----------------|-------------------|--------------------|
| 210 | 0,714285714 | 0,593786449 |
| 220 | 0,681818182 | 0,621886152 |
| 230 | 0,652173913 | 0,649969974 |
| 240 | 0,625 | 0,678037928 |
| 250 | 0,6 | 0,706090026 |
| 260 | 0,576923077 | 0,734126284 |
| 270 | 0,555555556 | 0,762146713 |
| 280 | 0,535714286 | 0,790151328 |
| 290 | 0,517241379 | 0,818140142 |

Kot vidimo ima enačba (4.7) nižji pogrešek za hitrosti do 240 obratov v minuti, za višje hitrosti pa enačba (4.4). Tako smo program izvedli tako, da smo realizirali oba algoritma. V odvisnosti od izmerjene hitrosti smo uporabili ustrezen algoritem.

Modul za podporo enkoderja, natančneje dekoder¹ omogoča tudi merjenje smeri vrtenja motorja. Dekoder (*Quadrature decoder*) lahko posreduje informacijo o smeri motorja enoti za štetje dogodkov (*Positione counter*) na štiri različne načine. Dekoder generira tudi uro za štetje dogodkov.

- Kvadraturni način,
- neposredno podajanje smeri,
- štetje navzgor,
- štetje navzdol.

Kvadraturni način smo delno že opisali. Ura za štetje dogodkov se generira ob vsaki pozitivni in negativni fronti signalov QEPA in QEPB. Tako dobimo 4000 dogodkov na obrat. Smer motorja se določi glede na prehitevanje določenega signala. Če signal QEPA prehiteva signal QEPB je definirana pozitivna smer urinega kazala. V tem delu smo uporabili ta način, saj nam enkoder z ustreznostjo signalov to omogoča.

Drugi način predstavljata signal QEPA, ki zagotavlja uro in QEPB, ki posreduje informacijo o smeri vrtenja.

Zadnja dva načina sta podobna in primerna za štetje samo v eno smer. Signal QEPA pa zagotavlja uro.

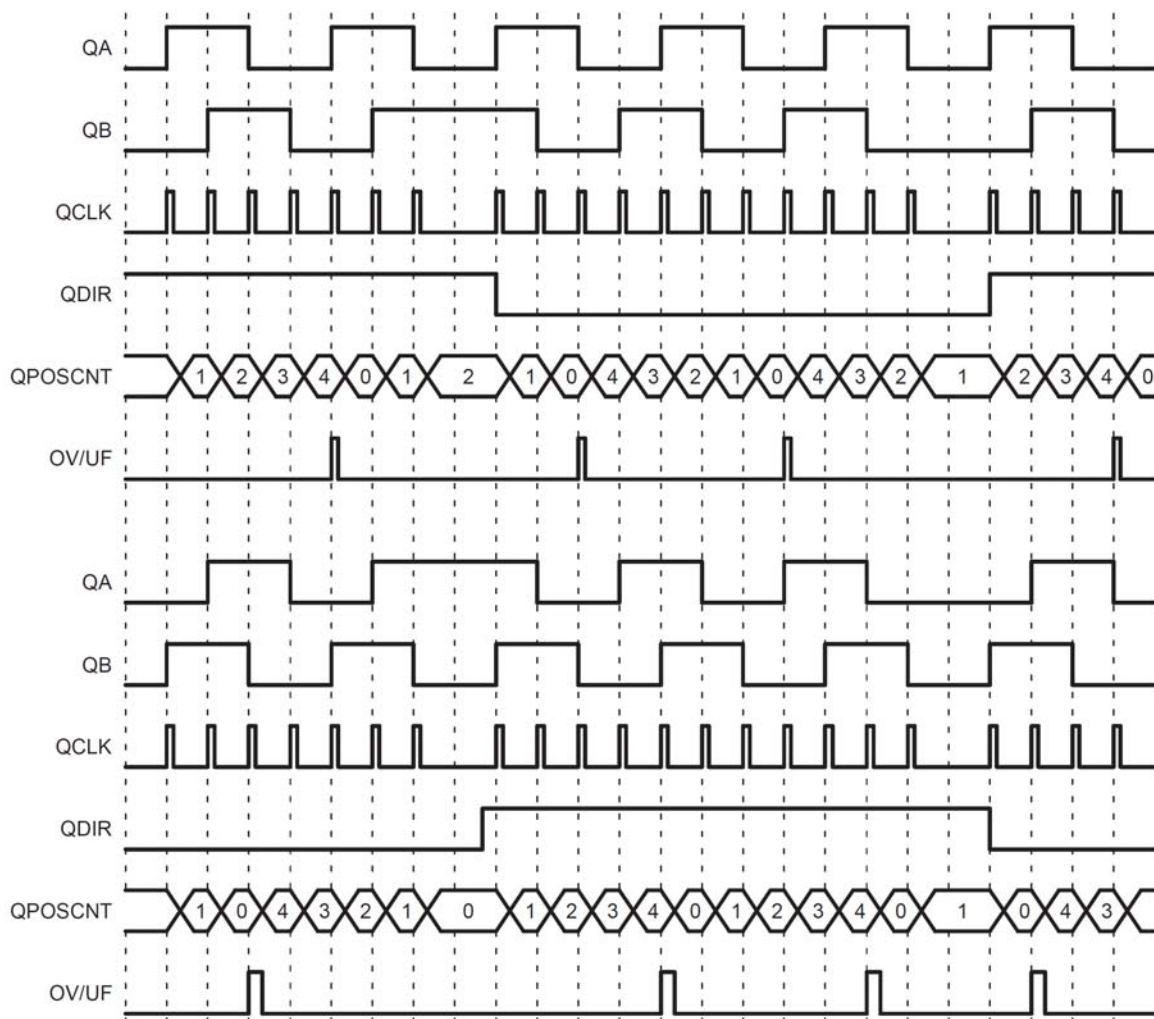
Modul omogoča štiri različne načine ponastavitve enote za štetje dogodkov (*Positione counter*):

- ponastavitev ob dogodku Index,
- ponastavitev ob maksimalnem položaju,

¹ Glej sliko 4.2.

- ponastavitev ob prvem dogodku Index,
- ponastavitev ob izteku enote periode (*Unit periode*).

Uporabili smo način ponastavitve enote za štetje dogodkov ob maksimalnem položaju. To pomeni, da ko naredi motor en obrat v pozitivni smeri, enota za štetje dogodkov prešteje do 4000 in se ponastavi. Če se motor vrti v obratni smeri, se ob prečkanju točke 0 nastavi na 4000 in odšteva naprej. Shematsko to prikazuje slika 4.5. Maksimalno vrednost do katere šteje enota za štetje dogodkov se nastavi v registru QPOSMAX [2].



Slika 4.5: QPOSMAX = 4 [2]

Modul omogoča tudi enoto za primerjanje pozicije. To pomeni, da lahko v točno določenem položaju sprožimo prekinitev. Vendar tega ne bomo natančneje opisovali, saj te funkcije pri našem sistemu vodenja nismo potrebovali.

Modul omogoča tudi proženje programskih prekinitev. Prekinitev lahko sprožajo najrazličnejše zastavice, ki se postavljajo ob različnih dogodkih. Program smo zasnovali tako, da prekinitev proži zastavica ob izteku enote periode (*Unit periode*), ki smo jo nastavili na čas tipanja. V tej prekinitvi¹ se izvede algoritem (4.4) in ob postavitvi ustrezne zastavice tudi algoritem (4.7).

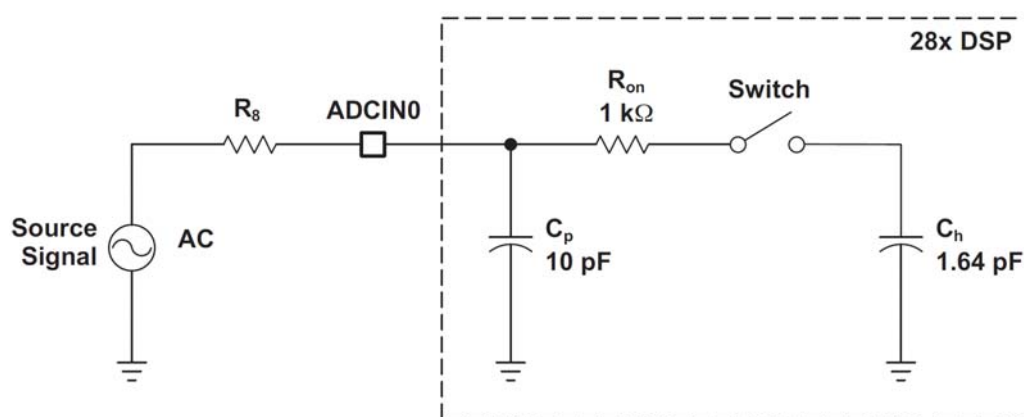
4.2 AD pretvorba

Pretvorba iz zveznega oz. analognega sveta v diskretni oz. digitalni je danes ena izmed zelo pogostih postopkov, ki so del vsakdana pri sistemih vodenja.

Sam DSP ima vgrajen 16 kanalni 12 bitni AD pretvornik, ki omogoča zajemanje podatkov s frekvenco tudi do 12,5 MHz. Vendar v primeru, če želimo, da so meritve natančne, je zajemanje meritev pri taki hitrosti skoraj nemogoče.

Na razvojni plošči se nahaja še zunanji AD pretvornik ADS1213, ki smo ga implementirali zaradi kvalitetnejših meritev in še vrsto drugih prednosti.

Osnovno zgradbo AD pretvornika prikazuje slika 4.6.



Slika 4.6: Shema AD pretvornika [6]

¹ Glej podpoglavje Prekinitve in zgradba jedra programa.

Kjer so:

AC – vhodni analogni signal

$ADCIN0$ – vhodni pin AD pretvornika

R_8 – Vhodna upornost pina

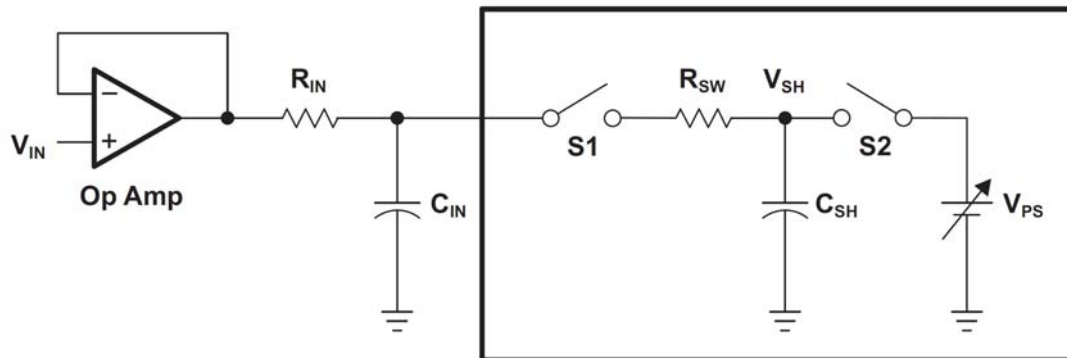
R_{ON} – Upornost multiplekserja

C_p – Parazitna kapacitivnost

C_h – Kapacitivnost za shrambo vrednosti

Zajemanje analogne vrednosti predstavlja stikalo, ki se sklene le v času tipanja. Ko je stikalo sklenjeno se kondenzator C_h napolni z analogno vrednostjo in jo zadrži. Med tem časom mora biti impedanca vira analognega signala čim nižja. Analogni signal pa mora ostati čim bolj stabilen.

AD pretvorbo lahko izboljšamo z dodatnim zunanjim nizkopasovnim filtrom, ki pomaga pri stabilizaciji merilne veličine. Z dodanim operacijskim ojačevalnikom pa zagotovimo vir z visoko vhodno upornostjo in zaščitimo vhod pred prekomerno napetostjo. [6]



Slika 4.7: Prilagoditveno vezje za AD pretvorbo [6]

Kjer so:

R_{IN} – Vhodna upornost, ki omejuje vhodni tok (ne sem biti velika, tipično 50Ω)

R_{SW} – Upornost multiplekserja

C_{IN} – Zunanji kondenzator, z uporom predstavljata filter

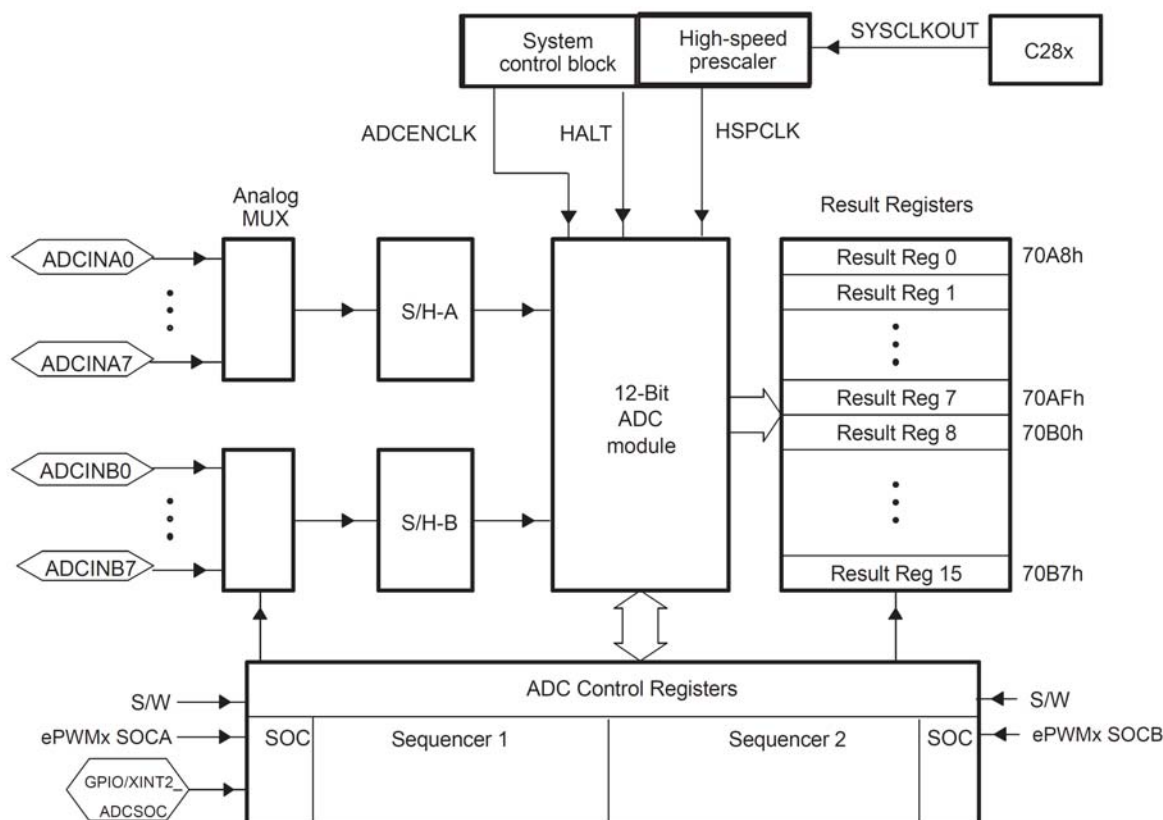
C_{SH} – Kapacitivnost za shrambo vrednosti

V_{PS} – Predstavlja prejšnjo zadržano vrednost

Moramo se zavedati, da z dodanim zunanjim filtrom povečamo čas stabilizacije napetosti. Zadržana vrednost pa je natančnejša. Z dodanim zunanjim filtrom moramo tako povečati čas tipanja. Dosežemo podoben učinek, kot če zajemamo merilno veličino pri višji frekvenci in jo v DSP-ju diskretno povprečimo. Ker je naš DSP dovolj zmogljiv ni bilo potrebe po dodajanju zunanjega filtra. Ustrezne ojačevalne stopnje pa so bile že pripravljene na razvojni plošči za pogon motorja. Prilagoditi smo morali le napetostne nivoje.

4.2.1 Vgrajeni AD pretvornik

DSP ima vgrajen 16 kanalni 12 bitni AD pretvornik kot vidimo na sliki 4.8. V DSP integriran AD pretvornik smo programsko realizirali predvsem zaradi morebitne kasnejše uporabe v laboratoriju. Pri eksperimentu stabilizacije IRN pa smo zaradi boljših lastnosti uporabili zunanji AD pretvornik ADS1213.

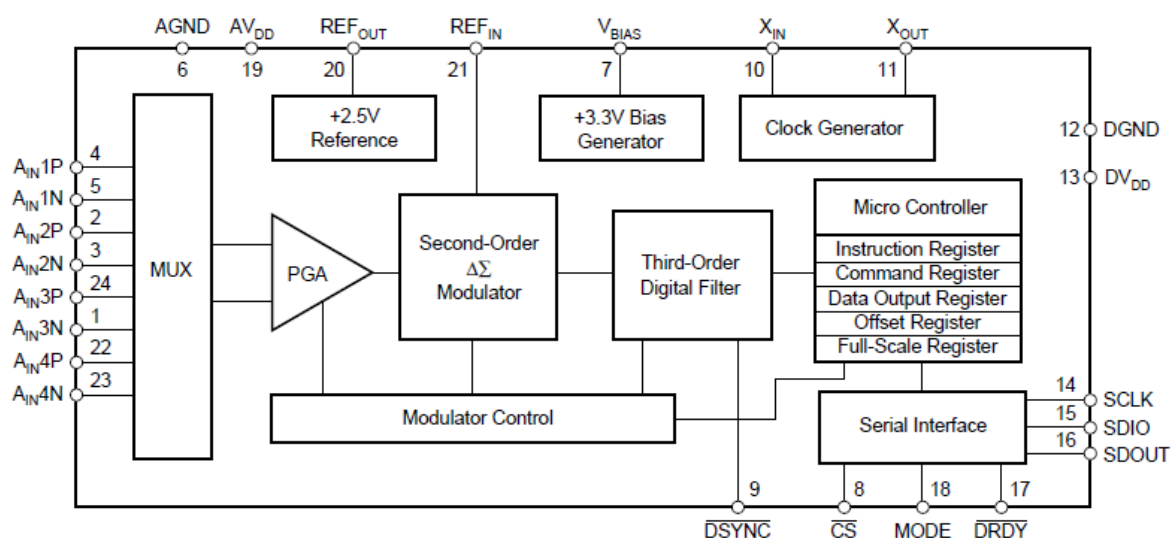


Slika 4.8: Shema vgrajenega AD pretvornika [7]

Integrirani AD pretvornik ima 16 vhodnih pinov, ki so razporejeni v dve skupini ADCINA in ADCINB. Vsaka skupina ima svoj zadrževalni člen, komplet kontrolnih registrov in komplet registrov v katere se shranjujejo rezultati. Vsaka skupina omogoča proženje prekinitve neposredno iz programa ali s signalom, ki ga generira poljubna PWM enota. Implementirali smo vse možnosti, ki so na voljo. Tako, da lahko AD enostavno poljubno nastavimo. AD smo tudi preizkusili. Nastavili smo zajemanje merilnih veličin z vseh kanalov. V programski prekinitvi, ki jo je prožil PWM signal s frekvenco 1 kHz, smo preverili merilne rezultate. AD pretvornik smo testirali z enosmernim napajalnim virom. Ugotovili smo, da ima za linearen faktor premaknjeno skalo. To se da seveda v DSP-ju diskretno popraviti oz. poljubno prilagoditi. Med drugim AD razpolaga s posebnim kalibracijskim registrom (*offset register*). Tega nismo realizirali, ker smo razvoj nadaljevali na zunanjem, veliko kvalitetnejšem AD pretvorniku ADS1213.

4.2.2 Zunanji AD pretvornik ADS1213

ADS1213 je 22 bitni delta sigma AD pretvornik z 20 bitno resolucijo pri 10 Hz in 16 bitno resolucijo pri 1 kHz. To je za naš sistem vodenja pri času tipanja 0,01 s več kot dovolj. Na vhodu ima štiri kanale, ki so preko multiplekserja povezani na diferenčni vhod. Ima integriran ojačevalnik za majhne signale. Podprto ima SPI ali I2C komunikacijo. Mi smo ga z DSP-jem povezali preko SPI komunikacijskega vodila. Omogoča zajemanje podatkov s frekvenco do 6,25 kHz. Čip ima integrirane posebne kalibracijske mehanizme, ki omogočajo precizno nastavitve območja merjenja in začetnega odmika (*offset*).



Slika 4.9: Shema zgradbe AD pretvornika [8]

AD pretvornik ima vgrajene različne komponente, ki omogočajo uspešno zajemanje merilne veličine. Vgrajen ima tudi lasten mikrokrmilnik, ki upravlja z instrucijskim, komandnim in podatkovnim registrom ter dvema kalibracijskima registroma. V te registre zapisujemo preko SPI vodila.

Ena izmed velikih prednosti tega pretvornika je vgrajen digitalni filter. Njegova funkcija je povprečenje trenutnih rezultatov, ki jih podaja delta sigma modulator. Število rezultatov, ki jih filter povpreči je odvisno od faktorja povprečenja (*Decimation Ratio*), ki ga nastavimo v komandnem registru. Faktor modulatorja (*Turbo Mode*) vpliva na frekvenco modulatorja. To pomeni, da ta dva faktorja neposredno vplivata na hitrost računanja in dostavljanja podatkov v podatkovni register. [8] Hitrost izhodnih podatkov podaja naslednja relacija:

$$f_{podatki} = \frac{f_{XIN} \cdot Turbo\ Mode}{128 \cdot (Decimation\ Ratio + 1)} \quad [8] \quad (4.14)$$

Kjer je:

$f_{podatki}$ - frekvenca izhodnih podatkov

f_{XIN} - frekvenca vhodnega oscilatorja, znaša 2 MHz

Tabela 4: Resolucija AD [8]

Tabela 5: Šum AD [8]

| Data Rate (Hz) | Effective Resolution (Bits rms) | | | | | DATA RATE (Hz) | NOISE LEVEL (μ Vrms) | | | | |
|----------------|---------------------------------|-------------------|-------------------|-------------------|--------------------|----------------|---------------------------|-------------------|-------------------|-------------------|--------------------|
| | Turbo Mode Rate 1 | Turbo Mode Rate 2 | Turbo Mode Rate 4 | Turbo Mode Rate 8 | Turbo Mode Rate 16 | | TURBO MODE RATE 1 | TURBO MODE RATE 2 | TURBO MODE RATE 4 | TURBO MODE RATE 8 | TURBO MODE RATE 16 |
| 10 | 20 | 21 | 21 | | | 10 | 7.6 | 3.8 | 3.8 | | |
| 20 | 19 | 20 | 21 | 21 | | 20 | 15 | 7.6 | 3.8 | 3.8 | |
| 40 | 18 | 20 | 21 | 21 | 21 | 40 | 30 | 7.6 | 3.8 | 3.8 | 3.8 |
| 50 | 17 | 19 | 20 | 21 | 21 | 50 | 60 | 15 | 7.6 | 3.8 | 3.8 |
| 60 | 17 | 19 | 20 | 21 | 21 | 60 | 60 | 15 | 7.6 | 3.8 | 3.8 |
| 100 | 15 | 17 | 19 | 21 | 21 | 100 | 240 | 60 | 15 | 3.8 | 3.8 |
| 250 | 12 | 14 | 16 | 19 | 20 | 250 | 1900 | 480 | 120 | 15 | 7.6 |
| 1000 | | | 12 | 14 | 16 | 1000 | | | 1900 | 480 | 120 |

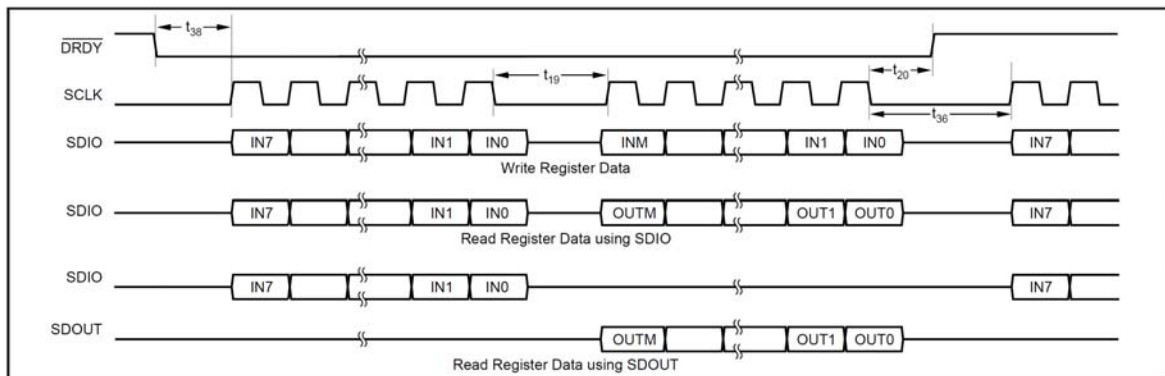
Čas tipanja sistema IRN bo 0,01 s. Tako je frekvenca tipanja 100 Hz. Ker si želimo čim natančnejše meritve, želimo čim višji faktor povprečenja (*Decimation Ratio*). To bomo dosegli z najvišjo nastavitvijo faktorja modulatorja (*Turbo Mode*). Kot vidimo iz tabel 4 in 5 je izbira faktorja modulatorja 16 zadovoljiva. Resolucija je pri frekvenci 100 Hz 21 bitna, kar je celo preveč. Izbrali smo 12 bitno resolucijo, ki povsem zadostuje. Meriti namreč ne

moremo natančneje od potenciometra. Tako zaradi pogreška potenciometra višja ločljivost AD pretvorbe ni ustrezna. Kot vidimo je pri frekvenci 100 Hz zanemarljiv tudi vpliv šuma.

Z ustrezno izbiro parametrov dobimo:

$$f_{podatki} = \frac{f_{XIN} \cdot Turbo Mode}{128 \cdot (Decimation Ratio + 1)} = \frac{2 \cdot e^6 \cdot 16}{128 \cdot (2499 + 1)} = 100 \text{ Hz} \quad (4.15)$$

Parametra *Turbo Mode*, *Decimation Ratio* in še mnoge druge je potrebno vpisati na ustrezna mesta v komandnem registru. Območje merjenja vhodnega signala smo z izbiro notranjega ojačanja 1, dobili od -5 do +5 V, kar ustreza signalu potenciometra. Komunikacijo med AD pretvornikom in DSP-jem smo realizirali preko SPI, sinhronega serijskega vodila. Način komunikacije je v obliki gospodar suženj (Master Slave), pri čemer je DSP gospodar, AD pretvornik pa suženj. To pomeni, da se odziva le na zahtevo DSP-ja. Komunikacija poteka tako, da je potrebno AD pretvorniku najprej poslati vsebino instruksijskega registra, s katerim povemo naslov ustreznega registra in ali bomo vanj pisali ali pa bomo iz njega brali. Potek komunikacije prikazuje slika 4.10.



Slika 4.10: Potek komunikacije [8]

Kjer so:

DRDY – Bit, ki je 0, ko ima AD pripravljene nove podatke v podatkovnem registru, ko ta bit pade na 0, se sproži zunanja prekinitve v DSP-ju¹

¹ Glej podpoglavje Prekinitve in zgradba jedra programa.

SCLK – notranja ura AD pretvornika

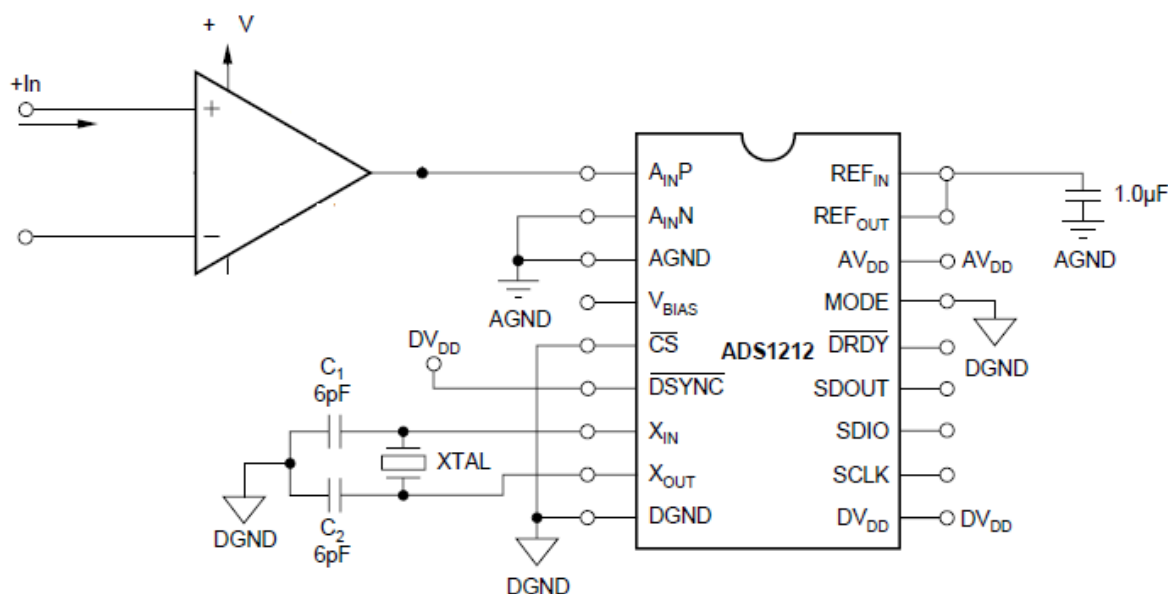
SDIO – vhodno izhodni pin

SDOUT – izhodni pin

Komunikacija lahko poteka po vodilu SPI ali I2C. Če uporabimo SPI, kot smo ga v našem primeru, je potrebno vhodno izhodni pin (*SDIO*) nastaviti kot vhodni. V primeru I2C ima pin *SDIO* vlogo vhoda in izhoda.

Tako kot integrirani AD ima tudi ta kalibracijske registre. V teh registrih je mogoče ročno vpisati vrednosti začetnega odklona (offset) in skaliranja, vendar ima ta AD vgrajene mehanizme samodejne kalibracije, ki so se pokazali kot zelo uspešni.

Primer vezja na sliki 4.11 prikazuje način povezave merilne veličine na vhod AD pretvornika.



Slika 4.11: Shema AD pretvornika [8]

Pri sistemu stabilizacije IRN smo zaradi kvalitetnejših meritev uporabljali ta AD pretvornik. Kot smo videli ima vgrajen digitalni filter, ki povpreči 2499 meritev. Še ena prednost je, da je AD pretvornik povsem avtonomen in deluje neodvisno od DSP-ja. Tako DSP-ju prihranimo kar nekaj procesiranja.

4.3 Serijsko vodilo SPI

Serijsko periferno vodilo (*SPI*) je sinhrono vhodno/izhodno vodilo, ki omogoča tok podatkov poljubne dolžine (od enega do šestnajst bitov) v in iz naprave pri določljivi hitrosti. Običajno se uporablja za komunikacijo DSP-ja s kakšno zunanjo periferno napravo ali za komunikacijo z drugim DSP-jem. SPI omogoča komunikacijo v obliki gospodar/suženj (Master/Slave). Vgrajena ima dva FIFO sklada, ki pripomoreta k zmanjšanju obremenitve DSP-ja.

Kot smo že omenili smo SPI vodilo nastavili za komunikacijo z zunanjim AD pretvornikom. Vodilo smo nastavili kot gospodar/suženj, pri čemer je gospodar DSP, suženj pa AD pretvornik.

Gospodar ima popoln nadzor nad uro vodila. Ko gospodar pošlje urin signal, se komunikacija prične s tem, da se začnejo podatki pošiljati in hkrati sprejemati iz pomičnega registra (*Shift Register*). Pošiljajo se ob pozitivni ali negativni fronti, prejemajo pa ob ravno nasprotni fronti. Mi smo nastavili prejemanje ob pozitivni fronti, pošiljanje pa ob negativni. Obstajajo tri možni scenariji komunikacije:

- gospodar pošlje koristne podatke, suženj pošlje neuporabne podatke,
- gospodar pošlje koristne podatke, suženj pošlje koristne podatke,
- gospodar pošlje neuporabne podatke, suženj pošlje koristne podatke.

Ker gospodar, v našem primeru DSP, nadzoruje urin signal, se pojavi potreba po detekciji, kdaj so podatki pripravljeni za pošiljanje v AD pretvorniku. AD to javlja DSP-ju preko zunanjega pina DRDY¹.

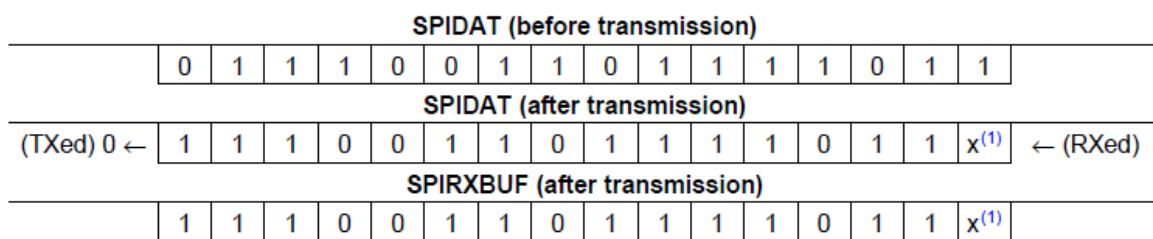
Princip komunikacije gospodar ali suženj prikazuje slika 4.12.

¹ Jedro programa preko tega pina zazna kdaj so podatki pripravljeni za pošiljanje v AD pretvorniku. Ko ta pin pade na 0 se sproži zunanja prekinitve. Glej podpoglavje Prekinitve in zgradba jedra programa.

SPISLK – Urin signal komunikacije.

Ko se paket pošlje iz *SPIDAT* registra, se statusnem registru postavi zastavica, ki omogoča programsko prekinitve. Mi prekinitve nismo uporabili.

Podatki, ki jih vpišemo v *SPITXBUF* morajo biti levo poravnani. Podatke, ki pa sprejmemo v *SPIRXBUF*, pa so desno poravnani. Primer pošiljanja podatkov prikazuje slika 4.13. [9]



Slika 4.13: Prikaz pošiljanja podatkov [9]

Pri SPI vodilu je bilo potrebno poskrbeti zraven ustreznih fizičnih povezav med DSP-jem in AD pretvornikom, tudi za nastavitve ustrezne dolžine podatkov, hitrosti vodila, načina konfiguracije gospodar ali suženj, nastavitve prekinitve in FIFO skladov.

Kot smo že omenili smo DSP nastavili kot gospodar. Prekinitve nismo uporabili, zato smo jih izklopili. Prav tako smo izklopili tudi FIFO sklade, ker jih nismo potrebovali. Namen FIFO skladov je manjše obremenjevanje jedra procesorja. Z uporabo skladov lahko namreč na enkrat v registre zložimo do 16 različnih 16 bitnih podatkov. Nato lahko jedro opravlja druge naloge. Med tem, pa bo SPI modul samodejno skrbel za ustrezno polnjenje registra *SPIDAT*. Ko se FIFO izprazni, lahko SPI to jedru sporoči preko zastavice ali pa ponovno prevzame jedro procesorja s pomočjo programske prekinitve.

Dolžino podatkov smo nastavili na 8 bitov, saj je bilo tako najlažje realizirati zahtevano obliko komunikacije z AD pretvornikom. Instrukcijski register v AD-ju je namreč 8 biten.

Hitrost vodila smo nastavili na 250 kHz. To je namreč maksimalna hitrost, pri kateri je AD pretvornik še uspešno sposoben komunicirati z DSP-jem. DSP podpira veliko višje hitrosti komunikacije, vse do 37,5 MHz.

$$SPI \text{ Baud Rate} = \frac{LSPCLK}{(SPIBRR + 1)} = \frac{25 \cdot e^6}{(99 + 1)} = 250 \text{ kHz} \quad (4.16)$$

Kjer je:

SPI Baud Rate – Hitrost prenosa podatkov

LSPCLK – Low Speed Prescaler Clock, delilnik ure

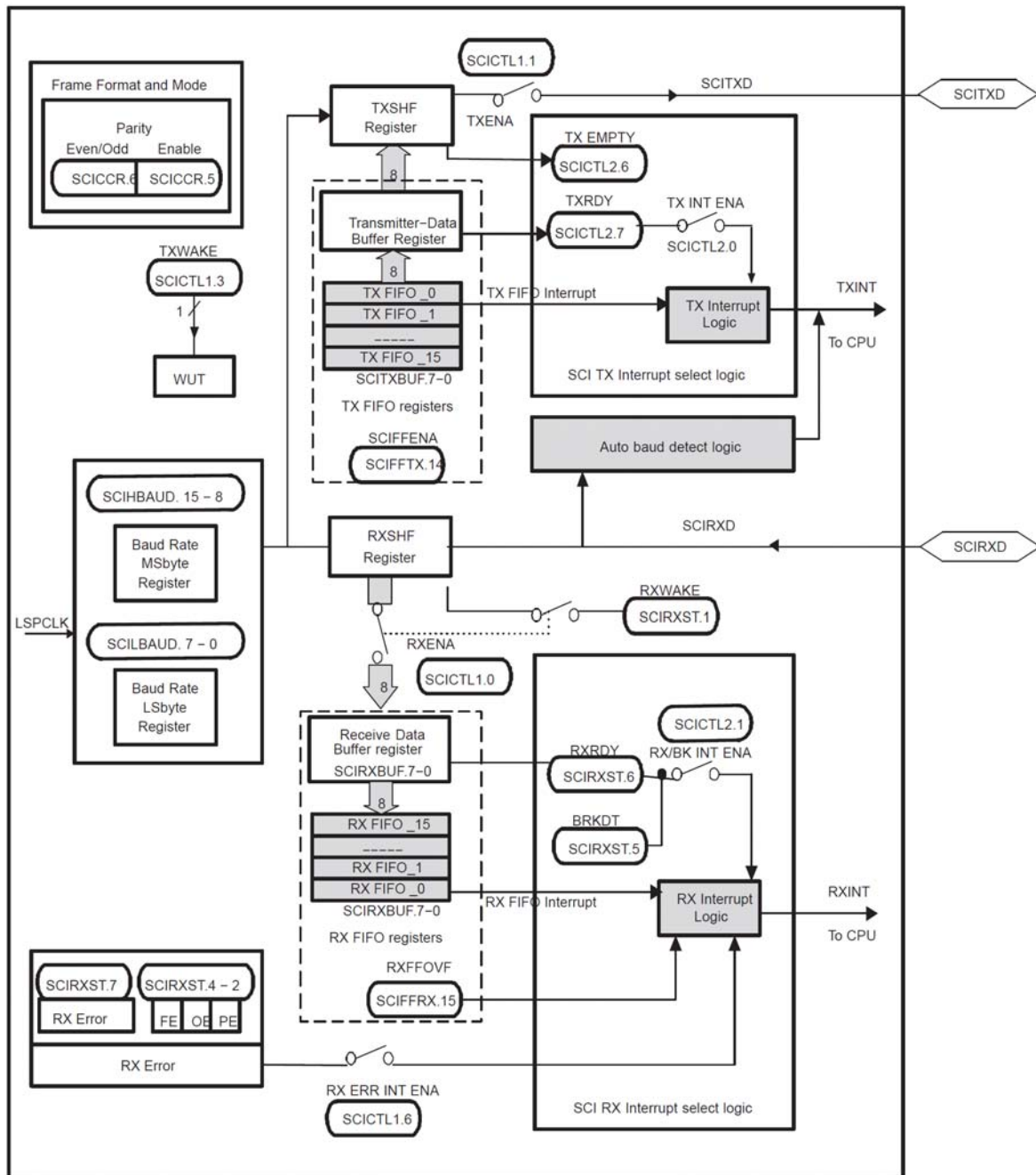
SPIBRR – SPI Boud Rate Register, Register, ki vpliva na hitrost prenosa podatkov

4.4 Serijsko vodilo SCI

Serijsko komunikacijsko vodilo (*SCI*), poznano tudi kot UART, je dvožično asinhrono vodilo, ki se uporablja za komunikacijo z drugimi DSP-ji ali napravami. Sestavljata ga ločeni enoti za pošiljanje in sprejemanje, ki omogočata poldupleksni ali dupleksni prenos podatkov. Vsaka komunikacijska enota razpolaga z lastnim FIFO skladom, kar pripomore k razbremenitvi procesorja.

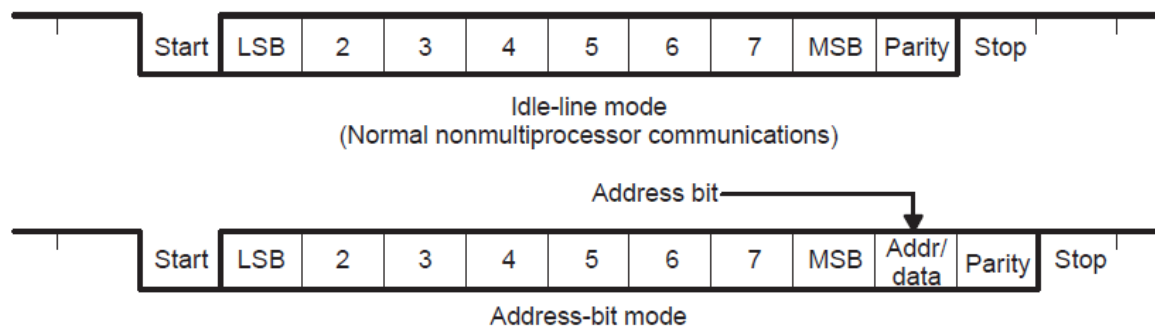
UART smo uporabili za komunikacijo z osebnim računalnikom. Tako smo lahko iz DSP-ja sprejemali meritve. Realizirali smo tudi povratno komunikacijo in sicer iz osebnega računalnika proti DSP-ju.

Kot vidimo na sliki 4.14, modul omogoča programske prekinitve za vsako enoto posebej. Torej za enoto, ki skrbi za pošiljanje posameznih bitov iz SCITXBUF registra na zunanji pin SCITXD in za enoto, ki sprejema posamezne bite preko pina SCIRXD in jih shranjuje v SCIRXBUF register. Posamezne operacije pošiljanja bitov, modul izvaja s pomočjo posameznih pomičnih registrov (*TXSHF* in *RXSHF*). Prekinitve so povezane s FIFO skladi, tako jih je mogoče prožiti pri poljubni zasedenosti skladov. Tudi ta modul ima namreč podobno kot SPI za vsako enoto posebej 16 16-bitnih registrov. Tako lahko jedro procesorja napolni FIFO z največ 16. različnimi podatki, nato pa se posveti drugim opravilom. Med tem pa bo modul samodejno skrbel za pošiljanje podatkov iz SCITXBUF. Podatke lahko tudi sprejema v FIFO, brez nenehnega posredovanja jedra procesorja. Ob določeni zasedenosti FIFO sklada, ki jo nastavimo, pa modul sporoči jedru preko zastavice, da je v skladu določeno število podatkov. Modul lahko jedro prevzame tudi s pomočjo prekinitve, ki se odzove s pojavitvijo zastavice v statusnem registru. [10]



Slika 4.14: Shema SCI Modula [10]

SCI podpira dva večprocesorska protokola, *idle-line multiprocessor* in *address-bit multiprocessor*. Razlikujeta se v osnovnem okvirju in sta prikazana na sliki 4.15.



Slika 4.15: Osnovna okvirja protokolov [10]

Kjer so:

LSB – Least significant bit, najmanj pomemben bit

MSB – Most significant bit, najpomembnejši bit

Address bit – naslovni bit

Parity – pariteta¹

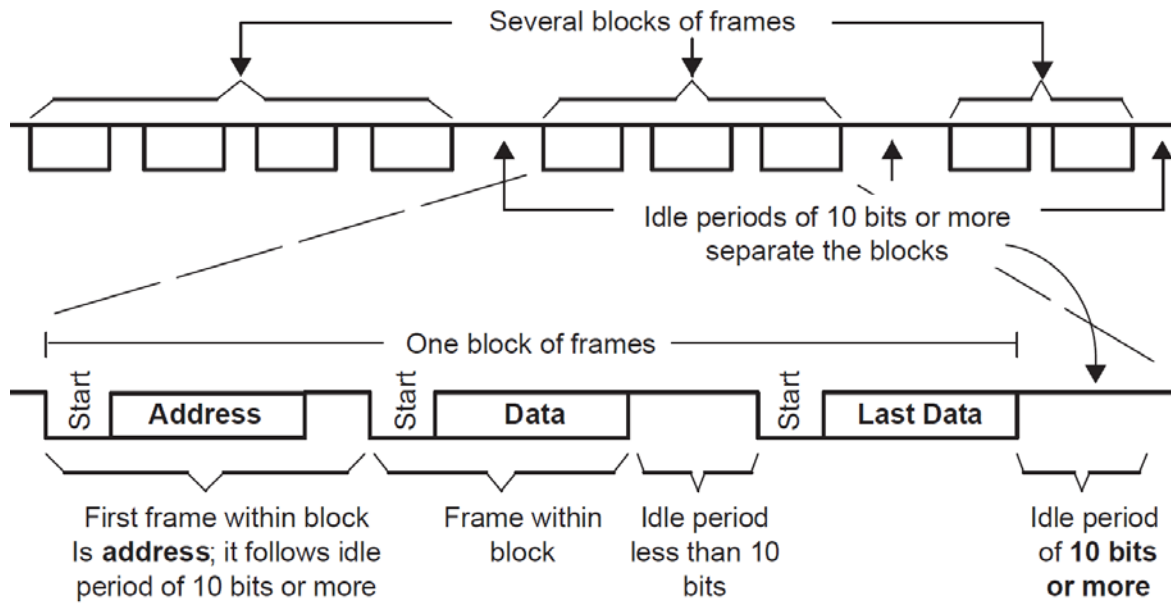
Osnovna filozofija večprocesorske komunikacije je takšna, da lahko v nekem času govori samo eden. Ostali morajo poslušati, saj lahko drugače pride do kolizij in tako posledično do izgube informacije. Princip asinhronne komunikacije temelji na naslovnem bajtu, ki se najprej pošlje po vodilu. Ta bajt sprejmejo vsi poslušalci. Nadaljnje podatke pa sprejme le poslušalec z ustreznim naslovom.

Zgoraj omenjena protokola se razlikujeta po načinu posredovanja naslovnega bajta. *Idle-line* protokol zahteva nek časovni prostor pred posredovanjem naslovnega bajta (slika 4.16). V osnovnem okvirju nima naslovnega bita, kar pomeni, da je učinkovitejši pri operacijah z več 10 bajtov velikimi bloki podatkov. Pred posredovanjem naslovnega bajta je potrebno pustiti najmanj 10 bitov prostora.

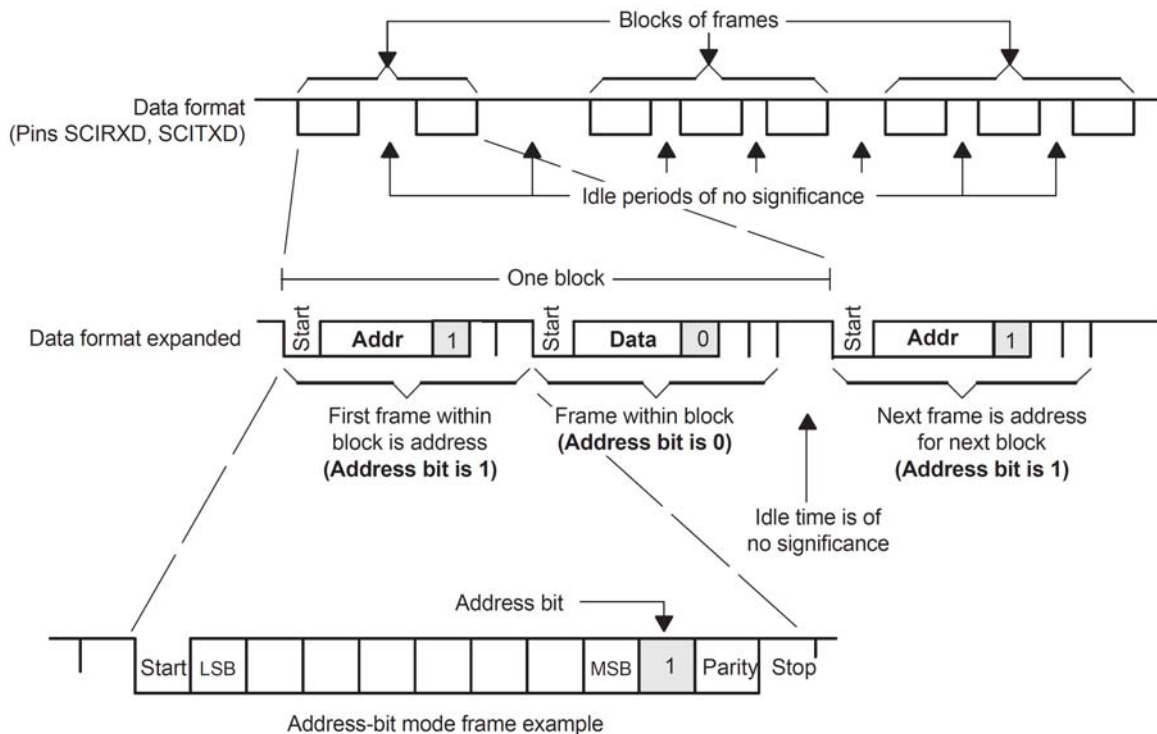
Address-bit protokol vsebuje v osnovnem okvirju dodaten bit, ki razlikuje naslovni bajt od podatkovnega (slika 4.17). Ta protokol je učinkovitejši pri operacijah z več manjšimi bloki

¹ Poznamo liho in sodo pariteto. Gre za preprost način razpoznavanja napake z dodanim bitom v osnovnem okvirju. Njegova vrednost je odvisna od števila enic v podatkovnem polju. Npr. če imamo liho pariteto in liho število enic v podatkovnem polju, potem je vrednost paritetnega bita 1. Drugače je 0.

podatkov, saj čakanje pred posredovanjem naslovnega bajta ni potrebno. Pošiljanje novega naslovnega bajta se začne tako, da se v osnovnem okvirju postavi naslovni bit na 1.

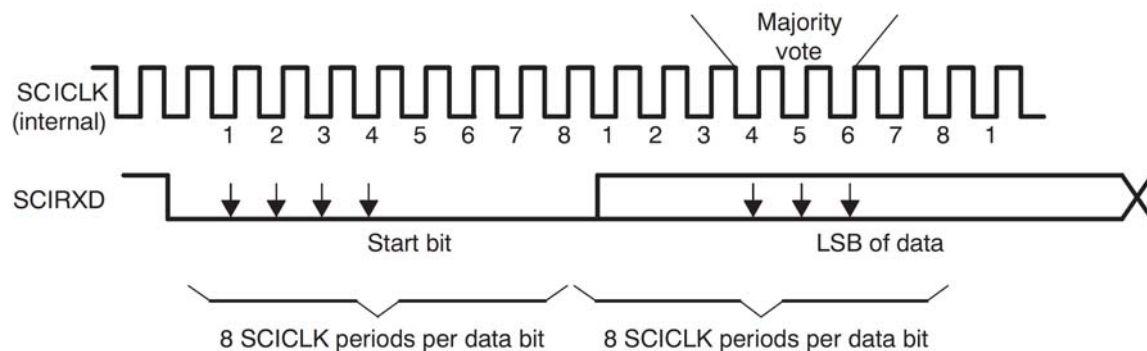


Slika 4.16: *Idle-line* protokol [10]



Slika 4.17: *Address-bit* protokol [10]

Osnovni okvir torej sestavlja 1 začetni bit (*start bit*), od 1 do 8 podatkovnih bitov, po izbiri naslovni bit in pariteta ter eden ali dva stop bita. Za vsak bit je potrebnih 8 pulzov urinega signala. Start bit tako predstavlja 4 zaporedne pulze urinega signala, pri čemer je vrednost na pinu 0. Vrednost podatkovnih bitov se določi tako, da se odvzame vrednost pri treh pulzih urinega signala točno na sredini bita (slika 4.18). Sprejemnik se tako sinhronizira z okvirjem in tako ni potrebe po skupni uri na vodilu. [10]



Slika 4.18: format komunikacije [10]

Hitrost vodila smo določili po naslednjem obrazcu:

$$SCI \text{ Baud Rate} = \frac{LSPCLK}{(BRR + 1) \cdot 8} = \frac{25 \cdot e^6}{(26 + 1) \cdot 8} \approx 115,2 \text{ kHz} \quad (4.17)$$

Kjer je:

SCI Baud Rate – Hitrost prenosa podatkov

LSPCLK – Low Speed Prescaler Clock, delilnik ure

BRR – SCI Boud Rate Register, Register, ki vpliva na hitrost prenosa podatkov

V DSP-ju smo realizirali pošiljanje in sprejemanje podatkov po protokolu *idle-line*, saj nismo imeli več udeležencev v komunikaciji. Pošiljanje podatkov oz. meritev osebnemu računalniku smo izvedli enostavno z ustreznim polnjenjem registra SCITXBUF. Realizirali smo posebne funkcije, ki omogočajo pošiljanje številke ali črk. Tako lahko pošiljamo na računalnik besede ali poljubne znake, ki jih najdemo v ANSI tabeli. Merilne rezultate

pošiljamo v jedru glavnega programa¹. Tako ne motimo časovno determinističnih programskih prekinitev.

Podobno smo izvedli sprejemanje. DSP lahko sprejema podatke v obliki besed ali v obliki števil. DSP-ju smo lahko tako preko osebnega računalnika pošiljali določene podatke, kot so smer vrtenja motorja, širina PWM pulza za pogon motorja. Lahko smo vključili ali izključili prikazovanje podatkov na zaslonu oz. pošiljanje podatkov proti računalniku. Sprejemanje podatkov smo izvedli s pomočjo FIFO skladov in aktiviranja prekinitve, ko se sklad napolni s sedmimi znaki. To pomeni, da DSP-ju pošljemo 7 znakov v obliki števil ali črk. Posamezni znaki se shranjujejo v FIFO sklad. Ko DSP sprejme sedmi znak, sproži prekinitve v kateri preberemo vsebino registrov.

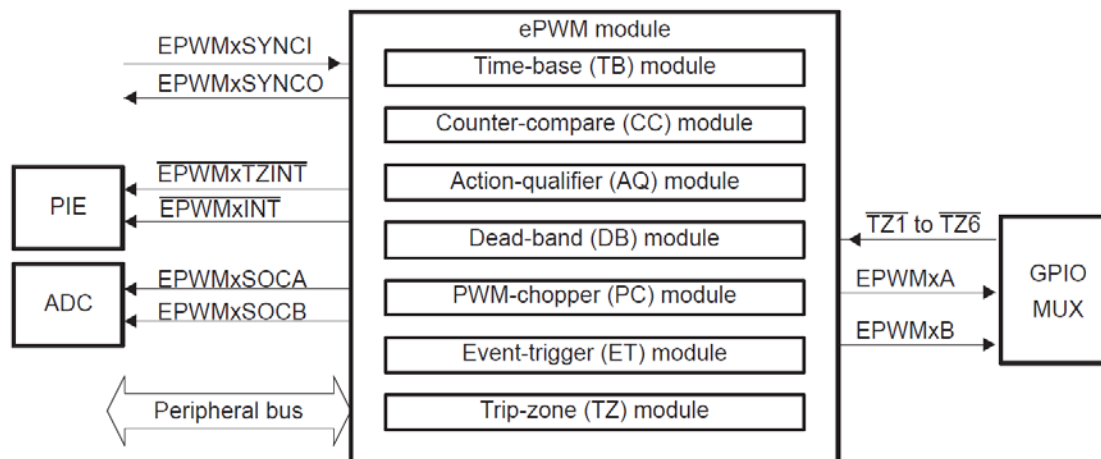
4.5 Pulzno širinska modulacija (PWM)

DSP podpira 6 PWM modulov, ki omogočajo na izhodnih pinih vsak po 2 različna pulzno širinsko modulirana signala. Vsak modul omogoča poljubno izbiro najrazličnejših lastnosti na izhodu generiranega signala:

- vsak modul ima svoj neodvisen časovnik z možnostjo nastavitve periode oz. frekvence PWM signala,
- možno poljubno oblikovanje izhodnega PWM signala,
- možnost sinhronizacije z drugim PWM modulom,
- možnost poljubnega načrtovanja mrtvega področja, s poljubnimi zakasnitvami pozitivne ali negativne fronte,
- vsak modul omogoča proženje prekinitve in internega AD pretvornika.

Lastnosti posameznega PWM modula prikazuje slika 4.19.

¹ Glej podpoglavje Prekinitve in zgradba jedra programa.



Slika 4.19: Lastnosti posameznega PWM modula [11]

Osnovna konfiguracija je dokaj enostavna. V register TBPRD vpišemo periodo PWM signala. Register TBCTL vsebuje delilnik ure, lastnosti števnik (gor, dol, gor in dol) in še druge možnosti. Vsak modul razpolaga z dvema primerjalnima registroma CMPA in CMPB. Vsak od teh registrov direktno vpliva na en izhodni pin. Ko števec doseže vrednost zapisano v posameznem registru se lahko spremeni logični nivo na izhodnem pinu. Tako lahko sestavimo poljubne konfiguracije PWM signala. To nastavimo v registrih AQCTLA in AQCTLB za vsak izhodni pin posebej. Generiranje mrtvega področja se nastavi v registru DBCTL. Zakasnitev pa se vpiše v registra DBRED in DBFED. [11]

V tem delu smo razvili programsko podporo za vseh 6 PWM modulov, ki so enostavno prilagodljivi in jih je mogoče poljubno nastaviti. PWM potrebujemo za vodenje enosmerne električne motorja, ki je aktuator sistema IRN. Predstavlja izhod oz. rezultat, ki ga poda regulator.

PWM modula smo uporabili tudi za testiranje modula za podporo enkoderja. V tem primeru smo naredili dva PWM signala, ki sta bila med seboj zamaknjena za četrtino periode. Nato smo periodo signalov interaktivno spreminjali preko osebnega računalnika povezanega z DSP-jem in spremljali izmerjene rezultate.

4.6 Prekinitev in zgradba jedra programa

Prekinitev so strojno ali programsko vodeni signali, ki povzročijo, da jedro procesorja opusti izvajanje trenutne rutine in začne z izvajanjem podrutine. Ponavadi jih prožijo periferni moduli procesorja ali kakšna druga strojna naprava, ki želi določene podatke izmenjati z jedrom procesorja. Predstavljajo lahko tudi določene dogodke.

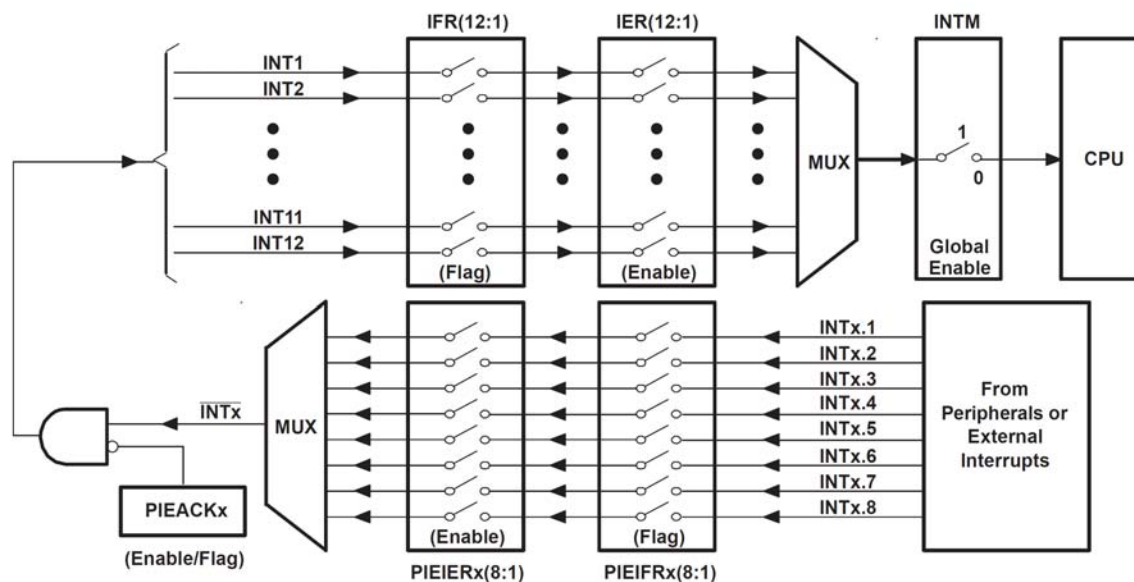
Če se v istem času sproži več prekinitev, jih poseben mehanizem jedra razvrsti po njihovi prioriteti. DSP razpolaga s posebnim razširitvenim modulom (*PIE*), ki omogoča razporeditev številnih možnih prekinitev tako, da izmed njih dostopi do jedra le prekinitev z najvišjo prioriteto. Poznamo dve vrsti prekinitev, maskirane in nemaskirane. Prve je mogoče onemogočiti, slednjih pa ne.

Jedro izvede prekinitev na naslednji način:

- Jedro prejme zahtevo za prekinitev. Dobi jo lahko s strani programa ali strojne opreme.
- Odobritev prekinitve. V primeru maskirane prekinitve morajo biti izpolnjeni določeni pogoji. Če gre za nemaskirano prekinitev, je odobritev takojšnja.
- Priprava na začetek izvajanja prekinitvene rutine. Zaključi se izvajanje trenutne instrukcije, registri se shranijo v sklad.
- Prekinitvena rutina se izvede.[12]

Jedro DSP-ja podpira eno nemaskirano prekinitev (*NMI*) in 16 maskiranih prekinitev (*INT1-INT14*, *RTOSINT* in *DLOGINT*). Najrazličnejše periferne naprave DSP-ja so sposobne prožiti svoje prekinitve ob najrazličnejših dogodkih. Nekateri moduli omogočajo proženje celo več kot ene prekinitve. Zaradi številnih možnih prekinitev se pojavi potreba po sistemu (*PIE*), ki možne prekinitve sistematizira preden jih dovede jedru procesorja. Gre za strojni sistem, ki ga sestavlja 12 skupin po 8 perifernih prekinitev. Znotraj posamezne skupine so prekinitve razporejene po prioriteti. To pomeni, če se znotraj skupine v istem času pojavi več zahtev po prekinitvi, bo posredovana jedru procesorja prekinitev z najvišjo prioriteto. Teh 12 skupin je povezanih z maskiranimi prekinitvami

INT1-INT12. Najvišjo prioriteto ima INT1, najnižjo pa INT12. Posledično so tako po prioriteti razporejene tudi skupine prekinitev. Sistem PIE prikazuje slika 4.20.



Slika 4.20: PIE [13]

Kjer so:

INTx – Predstavlja vseh 12 skupin perifernih prekinitev, ki so povezane z maskiranimi prekinitvami. Vsaka skupina vsebuje 8 različnih perifernih prekinitev.

PIEIFRx – Predstavlja zastavico določene prekinitve, ki je postavljena, ko je prekinitev aktivna. Ko se prekinitvena rutina izvede do konca, se zastavica samodejno zbriše.

PIEIERx – Postavitev določenega bita v tem registru predstavlja omogočitev določene periferne prekinitve na nivoju PIE. S tem omogočimo posredovanje prekinitve jedru s postavitvijo zastavice IFR. Če želimo, da se ustrezna prekinitev izvede, mora biti omogočena tudi na procesorskem nivoju.

PIEACKx – Ustrezen bit v tem registru se postavi v primeru, ko je bila prekinitev določene skupine posredovana jedru. S tem se onemogoči prejemanje nadaljnjih prekinitev iz iste skupine. Skupino moramo ponovno

aktivirati v programu, običajno na koncu prekinitvene rutine. To storimo s ponovno postavitvijo ustreznega bita v registru.

IFR – Predstavlja zastavico na procesorskem nivoju. Ta zastavica pomeni, da prekinitve čaka na začetek izvajanja. Ko se začne izvajati, se zastavica zbriše.

IER – Postavitev določenega bita v tem registru predstavlja omogočitev ustrezne procesorske prekinitve (INT1-INT12). Tako omogočimo sprejemanje in izvajanje prekinitve določene skupine perifernih prekinitve. [13]

Spodnja tabela prikazuje vse možne prekinitve, ki jih lahko omogočimo s pomočjo zgoraj omenjenih registrov. Nižje številke prekinitve predstavljajo višje prioritete. Tako po skupinah, kot tudi znotraj skupine.

Tabela 6: Periferne prekinitve [13]

| | INTx.8 | INTx.7 | INTx.6 | INTx.5 | INTx.4 | INTx.3 | INTx.2 | INTx.1 |
|----------------|-------------------------------|-------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| INT1.y | WAKEINT (LPMWDT) 0xD4E | TINT0 (TIMER 0) 0xD4C | ADCINT (ADC) 0xD4A | XINT2 Ext. int. 2 0xD48 | XINT1 Ext. int. 1 0xD46 | Reserved - 0xD44 | SEQ2INT (ADC) 0xD42 | SEQ1INT (ADC) 0xD40 |
| INT2.y | Reserved - 0xD5E | Reserved - 0xD5C | EPWM6_TZINT (ePWM6) 0xD5A | EPWM5_TZINT (ePWM5) 0xD58 | EPWM4_TZINT (ePWM4) 0xD56 | EPWM3_TZINT (ePWM3) 0xD54 | EPWM2_TZINT (ePWM2) 0xD52 | EPWM1_TZINT (ePWM1) 0xD50 |
| INT3.y | Reserved - 0xD6E | Reserved - 0xD6C | EPWM6_INT (ePWM6) 0xD6A | EPWM5_INT (ePWM5) 0xD68 | EPWM4_INT (ePWM4) 0xD66 | EPWM3_INT (ePWM3) 0xD64 | EPWM2_INT (ePWM2) 0xD62 | EPWM1_INT (ePWM1) 0xD60 |
| INT4.y | Reserved - 0xD7E | Reserved - 0xD7C | ECAP6_INT (eCAP6) 0xD7A | ECAP5_INT (eCAP5) 0xD78 | ECAP4_INT (eCAP4) 0xD76 | ECAP3_INT (eCAP3) 0xD74 | ECAP2_INT (eCAP2) 0xD72 | ECAP1_INT (eCAP1) 0xD70 |
| INT5.y | Reserved - 0xD8E | Reserved - 0xD8C | Reserved - 0xD8A | Reserved - 0xD88 | Reserved - 0xD86 | Reserved - 0xD84 | EQEP2_INT (eQEP2) 0xD82 | EQEP1_INT (eQEP1) 0xD80 |
| INT6.y | Reserved - 0xD9E | Reserved - 0xD9C | MXINTA (McBSP-A) 0xD9A | MRINTA (McBSP-A) 0xD98 | MXINTB (McBSP-B) 0xD96 | MRINTB (McBSP-B) 0xD94 | SPITXINTA (SPI-A) 0xD92 | SPIRXINTA (SPI-A) 0xD90 |
| INT7.y | Reserved - 0xDAE | Reserved - 0xDAC | DINTCH6 (DMA6) 0xDAA | DINTCH5 (DMA5) 0xDA8 | DINTCH4 (DMA4) 0xDA6 | DINTCH3 (DMA3) 0xDA4 | DINTCH2 (DMA2) 0xDA2 | DINTCH1 (DMA1) 0xDA0 |
| INT8.y | Reserved - 0xDBE | Reserved - 0xDBC | SCITXINTC (SCI-C) 0xDBA | SCIRXINTC (SCI-C) 0xDB8 | Reserved - 0xDB6 | Reserved - 0xDB4 | I2CINT2A (I2C-A) 0xDB2 | I2CINT1A (I2C-A) 0xDB0 |
| INT9.y | ECAN1INTB (CAN-B) 0xDCE | ECAN0INTB (CAN-B) 0xDCC | ECAN1INTA (CAN-A) 0xDCA | ECAN0INTA (CAN-A) 0xDC8 | SCITXINTB (SCI-B) 0xDC6 | SCIRXINTB (SCI-B) 0xDC4 | SCITXINTA (SCI-A) 0xDC2 | SCIRXINTA (SCI-A) 0xDC0 |
| INT10.y | Reserved - 0xDDE | Reserved - 0xDDC | Reserved - 0xDDA | Reserved - 0xDD8 | Reserved - 0xDD6 | Reserved - 0xDD4 | Reserved - 0xDD2 | Reserved - 0xDD0 |
| INT11.y | Reserved - - | Reserved - - | Reserved - - | Reserved - - | Reserved - - | Reserved - - | Reserved - - | Reserved - - |
| INT12.y | LUF (FPU) | LVF (FPU) | Reserved - | XINT7 Ext. Int. 7 | XINT6 Ext. Int. 6 | XINT5 Ext. Int. 5 | XINT4 Ext. Int. 4 | XINT3 Ext. Int. 3 |

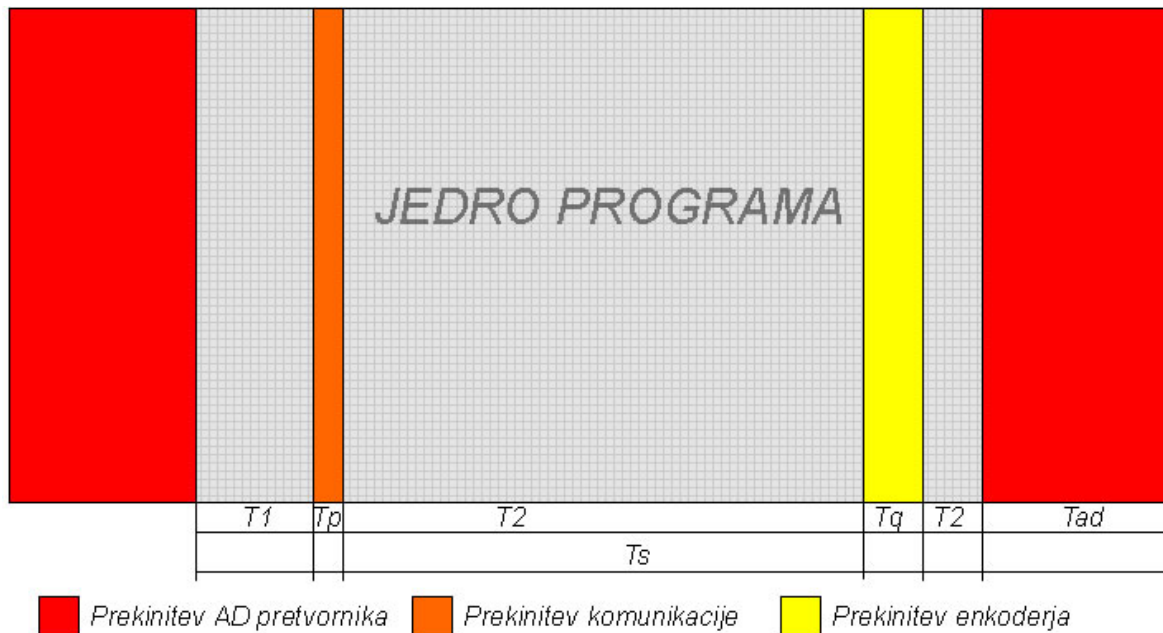
V DSP-ju smo realizirali vrsto najrazličnejših prekinitev. Vsaka prekinitev ima svojo specifično nalogo. Shema izvedbe jedra programa podaja slika 4.21. V osrednjem delu jedra programa, ki ga predstavlja neskončna zanka smo izvedli funkcije, ki skrbijo za pošiljanje merilnih rezultatov k osebnemu računalniku. Sinhronizacijo zunanjega AD pretvornika z DSP-jem smo izvedli preko zunanje prekinitev. Ko ima AD merilne podatke pripravljene v podatkovnem registru pade zunanji pin DRDY na 0. To se zgodi natanko vsake $0,01$ s. Ta dogodek sproži programsko prekinitev v kateri preko SPI vodila preberemo vrednost podatkovnega registra AD pretvornika.

V tej prekinitvi tudi prilagodimo vrednost registra QUTMR modula za enkoder. To je števniki, ki ob vrednosti, ki je vpisana v registru QUPRD, sproži prekinitev enkoderja. Tudi ta prekinitev se sproži vsake $0,01$ s oz. čas tipanja. Z vsakokratno prilagoditvijo časovnika enkoderja, dosežemo sinhronizacijo prekinitev enkoderja s prekinitvijo AD pretvornika. Tako poskrbimo za to, da se prekinitev enkoderja zagotovo pripeti nekoliko pred prekinitvijo AD pretvornika. Zunanji AD pretvornik ima namreč lasten vir urinega signala. Ta ura nikakor ne more teči popolnoma identično z uro DSP-ja. Tako pride do zelo majhnih razlik. Če ne bi poskrbeli za sinhronizacijo med prekinitvami, bi se lahko zgodilo, da se prekinitev enkoderja, ne bi zaključila pravočasno ali celo, da bi se zgodila za prekinitvijo AD pretvornika. Tega si nikakor ne moremo privoščiti, saj se algoritem regulatorja in tako ustrezen PWM signal za vhod v motor, izračunava ravno v prekinitvi AD pretvornika. Tako mora prekinitev enkoderja pravočasno poskrbeti za ustrezno izmerjeno hitrost in položaj prečke. Sistem smo načrtovali tako, da ima prekinitev AD pretvornika višjo prioriteto kot prekinitev enkoderja.

Načrtovali smo še eno prekinitev, ki pa ima nižjo prioriteto kot zgoraj omenjeni prekinitvi, saj v časovno determinističnem smislu ni tako pomembna. Gre za pošiljanje podatkov iz osebnega računalnika proti DSP-ju preko SCI vodila. Ta prekinitev je načrtovana tako, da se sproži, ko v FIFO sklad prispe 7. znak. V prekinitveni rutini preberemo ustrezne znake.

Če povzamemo, s pomočjo prekinitev zadostimo pogoju časovne determinističnosti časa tipanja. Tako imamo ustrezne meritve vsak cikel časa tipanja pripravljene na nadaljnjo obdelavo. S pomočjo AD pretvornika dobimo natančne podatke o položaju nihala. Hitrost in položaj prečke nam zagotovi enkoder. Hitrost nihala pa ocenimo na osnovi trenutne in prejšnje vrednosti položaja nihala:

$$\dot{\beta} = \frac{\beta(k) - \beta(k-1)}{T_s} \quad (5.1)$$



Slika 4.21: Zgradba programa

Kjer so:

T_1 - T_3 – Njihov seštevek predstavlja izvajanje jedra programa. V jedru programa se izvajajo funkcije za pošiljanje podatkov proti osebemu računalniku.

T_{ad} – Čas izvajanja prekinitve AD pretvornika. Ta prekinitvev ima najvišjo prioriteto.

T_q – Čas izvajanja prekinitvev enkoderja. Ta prekinitvev ima nižjo prioriteto od prekinitvev AD pretvornika.

T_p – Čas izvajanja prekinitvev v kateri DSP sprejema podatke poslane iz osebnega računalnika. Ta prekinitvev ima najnižjo prioriteto in se lahko pojavi kjer koli med izvajanjem jedra programa.

T_s – Čas tipanja.

5 NAČRTOVANJE REGULATORJEV V PROSTORU STANJ

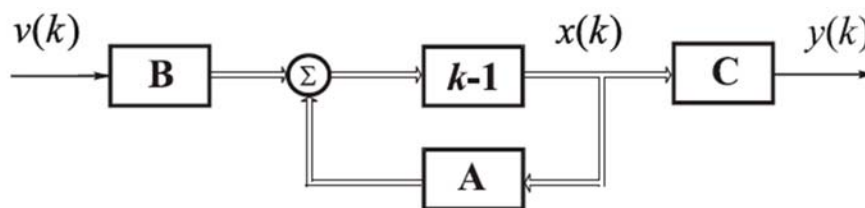
V svetu regulacij in vodenja je predstavitev sistema v prostoru stanj dejansko matematični model, ki opisuje nek fizikalni objekt, v obliki diferencialnih enačb prvega reda. Te enačbe predstavljajo poleg vhodnih in izhodnih spremenljivk, še notranje spremenljivke oz. spremenljivke stanj. Za boljši pregled nad številnimi spremenljivkami, jih lahko izrazimo v obliki vektorjev. Tako lahko pod pogojem, če gre za linearen in časovno invarianten¹ sistem, diferencialne in ostale algebrske enačbe zapišemo v matrični obliki. Takšen zapis je priročen za modeliranje in obravnavanje sistemov z mnogimi vhodi oz. izhodi (*MIMO* – Multi Inputs Multi Outputs).

Zapis v prostoru stanj izvedemo s sistemom linearnih enačb. V zvezni obliki:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (5.1)$$

V diskretni obliki:

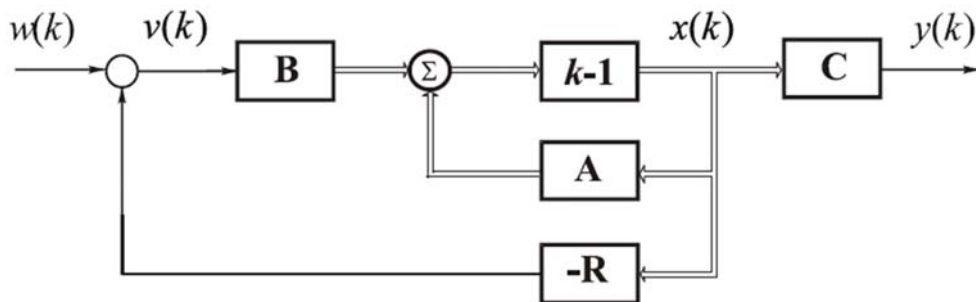
$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (5.2)$$



Slika 5.1: Primer modela v prostoru stanj [4]

¹ Sistem je časovno neodvisen ali invarianten, če oblika in potek izhodnega signala nista odvisna od izbranega začetka opazovanja [4]

V prostoru stanj imamo več informacij o objektu kot v vhodno izhodnem opisu. Zaradi tega je tudi regulacija toliko bolj uspešna. Kot vidimo na sliki 5.2 je osnovni princip regulacije v prostoru stanj takšen, da regulator v povratni veji tvori regulirano količino na osnovi spremenljivk stanj in tako posredno nastavlja želene izhodne vrednosti. Zaradi tega ga imenujemo regulator stanj. [4],[14]



Slika 5.2: Primer sistema s povratno povezavo [4]

Za uspešno načrtovanje regulatorja v prostoru stanj mora veljati, da je sistem vodljiv. To pomeni, da lahko vplivamo na vse spremenljivke stanj tako, da jih ojačujemo sli slabimo. S tem lahko vektor stanj v nekem končnem intervalu pripeljemo iz poljubnega začetnega stanja v poljubno končno stanje. [4] Za sistem mora veljati:

$$\text{rank} \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n, [14] \quad (5.3)$$

Spoznavnost je dualna lastnost vodljivosti in pravi, da je sistem spoznaven, če je mogoče iz končnega števila opazovanj izhodnih in vhodnih signalov enolično določiti začetno stanje. [4]

$$\text{rank} \begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix} = n, [14] \quad (5.4)$$

5.1 Regulator za pomikanje polov

Regulator stanj lahko določimo tako, da ima zaprtzančni sistem zahtevan karakteristični polinom oz. želeno lego korenov. Regulator, ki ga dobimo na ta način imenujemo regulator za pomikanje polov (*FSF* – Full State Feedback). Načrtovanje sistema na takšen način je zelo zaželeno, saj je lega korenov neposredno povezana z lastnimi vrednostmi sistema, ki pa direktno vplivajo na karakteristike izhodnega odziva sistema.

Pole diskretnega sistema predstavljajo koreni karakterističnega polinoma, ki ga dobimo kot:

$$|zI - A| = 0 \quad (5.5)$$

Če vstavimo enačbo:

$$u = -Rx(k) \quad (5.6)$$

v 5.2 dobimo:

$$x(k+1) = (A - BR)x(k) \quad (5.7)$$

Če vstavimo enačbo (5.7) v enačbo za karakteristični polinom (5.5) dobimo:

$$|zI - (A - BR)| = 0 \quad (5.8)$$

S primerjavo te enačbe s karakteristično enačbo zelenih korenov, lahko izračunamo vrednosti matrike R , ki predstavlja ustrezno povratnozančno ojačanje. Tako dobimo sistem z želeno lego korenov.

Pri sistemih višjega reda pa postane metoda primerjave koeficientov karakterističnih polinomov neuporabna, predvsem zaradi zahtevnega analitičnega izračuna determinante zaprtzančnega sistema. [4], [14], [15] Obstaja splošna rešitev sistema v obliki Ackermannove formule:

$$R = [0 \ 0 \dots 0 \ 1] Q_v^{-1} P(A) \quad (5.9)$$

Kjer so:

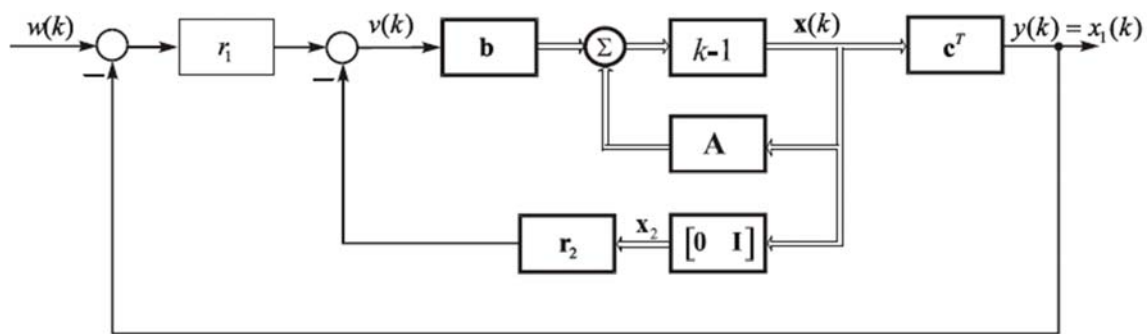
R – Matrika ustreznih ojačanj v povratnih vejah.

Q_v – Vodljivostna matrika. Inverzna vodljivostna matrika obstaja le v primeru, če je sistem vodljiv.

$P(A)$ – Matrični polinom systemske matrike A .

5.2 Regulator za sledenje referenčnemu vhodu

Regulator za pomikanje polov je mogoče modificirati tako, da bo izhod sistema sledil referenčnim vhodnim vrednostim. Za sistem v prostoru stanj lahko predpostavimo, da predstavlja spremenljivka stanj x_1 izhod sistema. Kot prikazuje slika 5.3, lahko regulator stanj razstavimo tako, da je r_1 njegov prvi element, vse ostale pa združimo v vektor r_2 . To lahko naredimo seveda za poljubno spremenljivko stanja, ki jo izključimo iz vektorja x_2 . Ojačanje te spremenljivke v povratni veji pa seveda izključimo iz vektorja r_2 .



Slika 5.3: Sledenje referenčnemu vhodu [4]

Kjer je enačba zaprtozančnega sistema:

$$x(k+1) = (A - bR)x(k) + r_1 b w(k) \quad (5.10)$$

Tak sistem se na stopnični odziv odzove tako, da po določenem času vektor stanj zavzame končno stacionarno vrednost x_s . Odziv je stabilen, zato namreč poskrbi regulator stanj.

$$x_s = r_1 (I - A + bR)^{-1} \quad (5.11)$$

Stacionarno odstopanje na izhodu:

$$y_s = c^T x_s = r^1 c^T (I - A + bR)^{-1} b \quad (5.12)$$

Kjer je c^T vektor, ki določeno ustrezno spremenljivko stanja kot izhod.

Če imamo sistem z integrirnim delovanjem, to pomeni, da ima sistem vsaj en integrator, potem tak sistem nima stacionarnega pozicijskega pogreška.

Če sistem nima integrirnega delovanja, pa lahko referenčni vhod skaliramo z obratno vrednostjo stacionarnega odstopanja na izhodu (y_s):

$$r_p = \frac{1}{r^1 c^T (I - A + bR)^{-1} b} \quad (5.13)$$

Dobljeno ojačanje imenujemo tudi predfilter. [4], [14], [15]

5.3 Linearni kvadratični regulator (LQR)

Teorija optimalnega vodenja se ukvarja z vodenjem dinamičnih sistemov pri najnižjih vrednostih kriterijske funkcije. S pomočjo kriterijske funkcije se meri uspešnost regulacijskega sistema. Sinteza optimalnega regulacijskega sistema je običajno iterativni proces, kjer spreminjamo kriterijsko funkcijo tako dolgo, dokler ne dosežemo želenega obnašanja regulacijskega sistema. Pri optimalnem vodenju poznamo LQ problem, kjer sistem opisujejo linearne diferencialne enačbe, kriterijsko funkcijo pa kvadratični funkcional. Bistvo linearnega kvadratičnega regulatorja je poiskati ustrezna povratno znančna ojačanja (K) sistema. Postopek iskanja rešitev določa kriterijska funkcija, ki s pomočjo posameznih utežitvenih faktorjev omogoča izračun ustreznih ojačanj. [4], [16], [17]

Kvadratična kriterijska funkcija:

$$J = \frac{1}{2} x^T(T) F(T) x(T) + \int_0^T (x^T Q x + u^T R u) dt \quad (5.14)$$

Kjer sta Q in R utežitveni matriki spremenljivk stanj in vhodnih spremenljivk.

Zaprtoznančno ojačanje K se določi po:

$$K = R^{-1} B^T P(t) \quad (5.15)$$

Kjer se ojačanje izračuna s pomočjo rešitev Riccatijeve enačbe:

$$A^T P(t) + P(t) A - P(t) B R^{-1} B^T P(t) + Q = -\dot{P}(t) \quad (5.16)$$

6 REZULTATI REGULACIJSKIH ALORITMOV

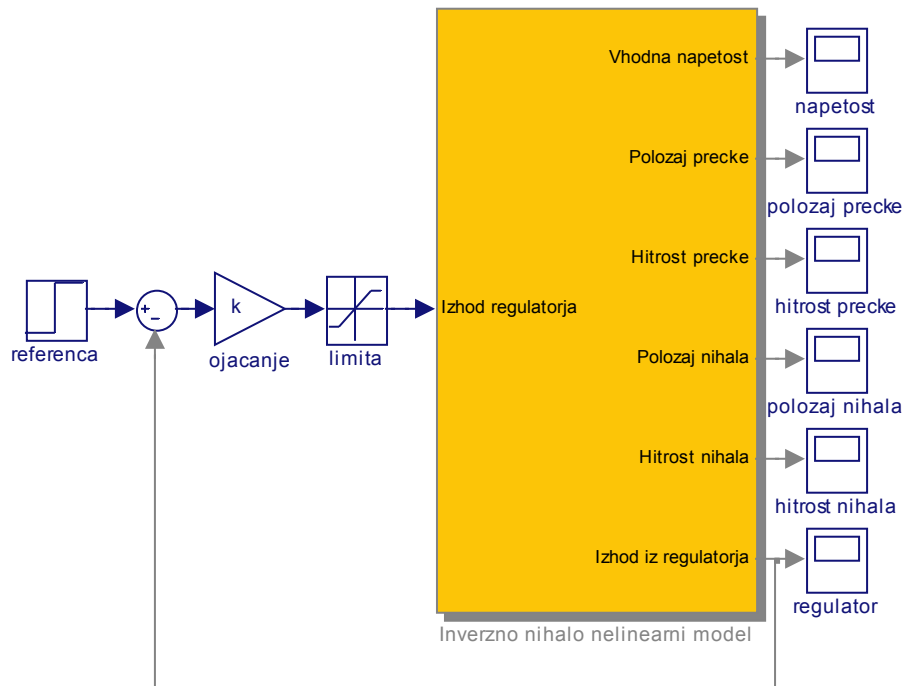
Regulator za pomikanje polov smo načrtali s pomočjo Ackermannove formule v Matlabu. Kot smo ugotovili, potrebujemo za določitev zaprtozančnega ojačanja želene zaprtozančne pole. S pomočjo izraza »acker« lahko v Matlabu izračunamo ustrezna ojačanja.

LQR regulator smo izračunali s pomočjo izraza »lqr« in ustreznih utežitvenih matrik Q in R .

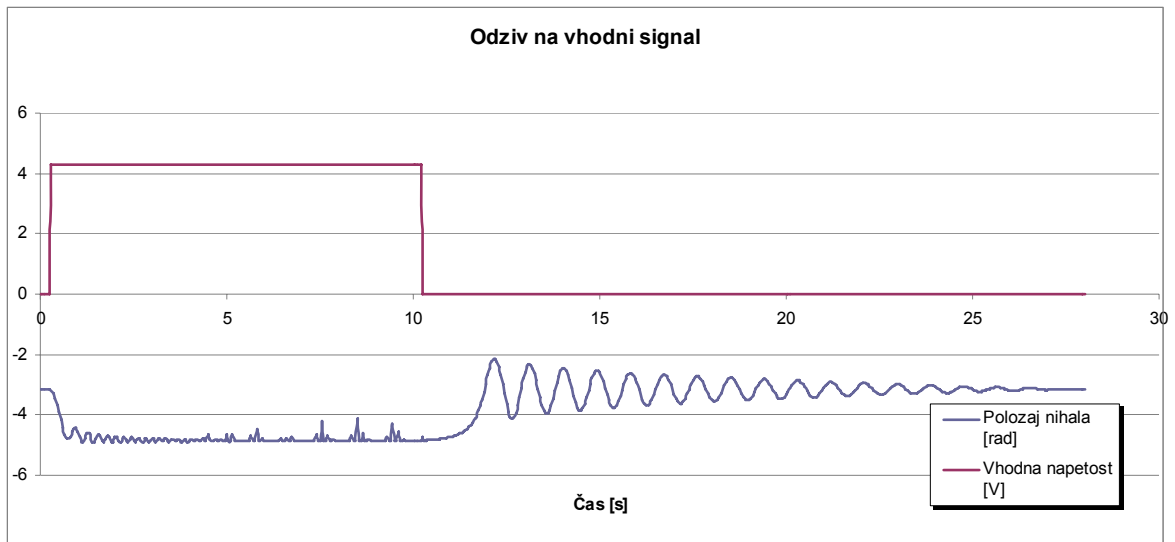
Tako izračunane regulatorje smo nato preizkusili na modelih v Simulinku. Glede na to, da smo regulatorje načrtovali za lineariziran sistem, smo jih morali preizkusiti na nelinearnem modelu, kjer smo preverili, ali rešitev ustreza kriterijem stabilnosti. S simulacijami smo lahko tudi preizkusili do kakšnega odklona nihala je sistem še stabilen.

Kot smo že ugotovili je razlika med linearnim in nelinearnim modelom kar precejšnja. Velika razlika pa je tudi med nelinearnim modelom IRN v Simulinku in realnim fizikalnim objektom oz. IRN. Matematični model je omejen s posploševanjem in nikakor ne zajema vseh fizikalnih pojavov, kot so na primer šumi in motnje. Trenje je le blago obravnavano v obliki konstantne navorne obremenitve. Model ne obravnava zračnega upora in lepenja, ki nikakor ni zanemarljivo.

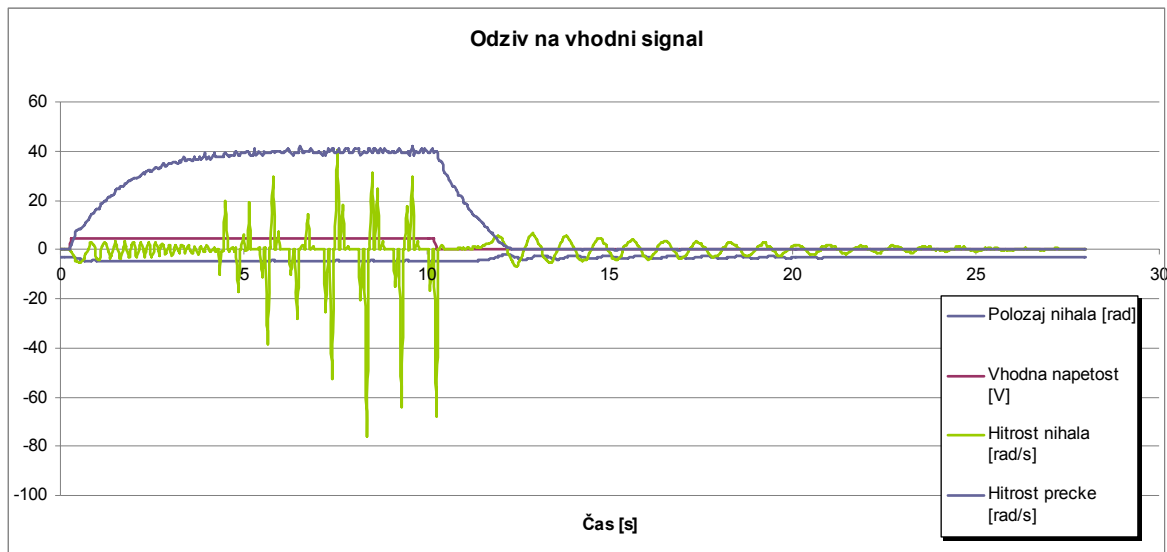
Po implementaciji regulatorjev v DSP, smo jih morali nekoliko prilagoditi v obliki povratnozančnega ojačanja k in limit. Kot je to prikazano na sliki 6.1. Uporaba limite vhodne napetosti je nujna, saj je prisotna tudi na realnem objektu. Napajalnik lahko elektromotorju namreč zagotovi največ 17 V enosmerne napetosti.



Slika 6.1: Model IRN v Simulinku



Slika 6.2: Odziv na pravokotni vhodni signal



Slika 6.3: Odziv na pravokotni vhodni signal

Zaradi kvalitetne izvedbe AD pretvornika in ugodnega principa merjenja enkoderja so merilni rezultati dokaj kvalitetni. Kot vidimo na sliki 6.2 pa AD pretvornik ni čisto neobčutljiv na šume. Ker je hitrost nihala, ki jo izračunamo neposredno iz meritev položaja nihala v bistvu integral položaja po času, integriramo zraven tudi vse merilne pogreške. To lepo vidimo na sliki 6.3. Krajši kot je čas zajemanja meritev oz. čas tipanja, večji bo vpliv pogreška pri hitrosti nihala.

Uporabljali smo naslednje modele¹:

$$\begin{bmatrix} \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} -1.3708 & 3.8605 & -0.0124 \\ 0 & 0 & 1.0 \\ -0.9372 & 51.4135 & -0.165 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 8.9957 \\ 0 \\ 6.1498 \end{bmatrix} V_m \quad (6.1)$$

$$\begin{bmatrix} \dot{\alpha}(k+1) \\ \beta(k+1) \\ \dot{\beta}(k+1) \end{bmatrix} = \begin{bmatrix} 0.9864 & 0.0383 & 0.0001 \\ 0 & 1.0026 & 0.0100 \\ -0.0093 & 0.5140 & 1.0009 \end{bmatrix} \begin{bmatrix} \dot{\alpha}(k) \\ \beta(k) \\ \dot{\beta}(k) \end{bmatrix} + \begin{bmatrix} 0.0893 \\ 0.0003 \\ 0.0611 \end{bmatrix} V_m(k) \quad (6.2)$$

¹ Parametre modelov smo prevzeli po [5]

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1.3708 & 3.8605 & -0.0124 \\ 0 & 0 & 0 & 1 \\ 0 & -0.9372 & 51.4135 & -0.165 \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 \\ 8.9957 \\ 0 \\ 6.1498 \end{bmatrix} V_m \quad (6.3)$$

$$\begin{bmatrix} \alpha(k+1) \\ \dot{\alpha}(k+1) \\ \beta(k+1) \\ \dot{\beta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.0099 & 0.0002 & 0 \\ 0 & 0.9864 & 0.0383 & 0.0001 \\ 0 & 0 & 1.0026 & 0.01 \\ 0 & -0.0093 & 0.5140 & 1.0009 \end{bmatrix} \begin{bmatrix} \alpha(k) \\ \dot{\alpha}(k) \\ \beta(k) \\ \dot{\beta}(k) \end{bmatrix} + \begin{bmatrix} 0.0004 \\ 0.0893 \\ 0.0003 \\ 0.0611 \end{bmatrix} V_m(k) \quad (6.4)$$

Pri izvedbi regulatorja za pomikanje polov, smo uporabili model tretjega reda, saj regulacija po položaju prečke ni bistvena za samo stabilizacijo nihala. Pri izvedbi regulatorja z referenčnim vhomom pa smo uporabili model četrtega reda, saj smo kot referenčno stanje uporabili položaj prečke.

Zaprtozančne pole pri regulatorju za pomikanje polov smo izbrali v obliki člena drugega reda, kjer smo nastavljali dušenje in lastno frekvenco, in enega ali dveh realnih polov. Odvisno od stopnje obravnavanega sistema.

Regulatorji izračunajo iznos napetosti za pogon enosmernega motorja in v katero smer se mora zavrteti. Kot smo že omenili uporabljamo za pogon motorja PWM signal, ki ima frekvenco 1 kHz. Sto procentno prevajalno razmerje dosežemo, če časovnik za PWM prešteje do maksimalne vrednosti, ki znaša 37500. Maksimalno prevajalno razmerje, pa kot smo izmerili, na motorju ustvari najvišjo napetost 16 V. Ustrezno prevajalno razmerje pa dobimo tako, da izračunano napetost povečamo za 2338 krat. Tabela meritev je v dodatku. Kot se je izkazalo pa smo morali to ojačanje zmanjšati.

6.1 Izvedba regulatorja za pomikanje polov

Za izračun regulatorja smo uporabili model (6.2). Naš cilj je stabilizacija nihala.

Izbrani parametri in s tem zaprtozančni poli sistema:

$$d = 0.0650$$

$$\omega_0 = 4.32$$

$$s_1 = -0.2808 + 4.3109i$$

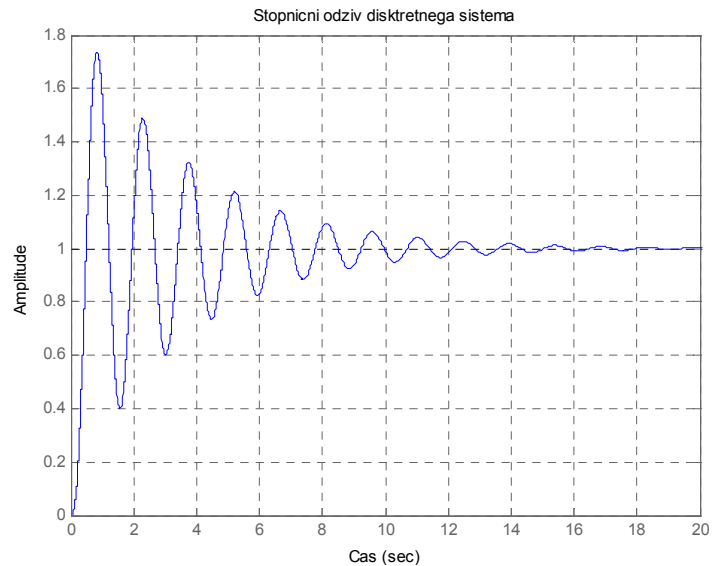
$$s_2 = -0.2808 - 4.3109i$$

$$s_3 = -9.2821$$

$$z_1 = 0.9963 + 0.0430i$$

$$z_2 = 0.9963 - 0.0430i$$

$$z_3 = 0.9114$$



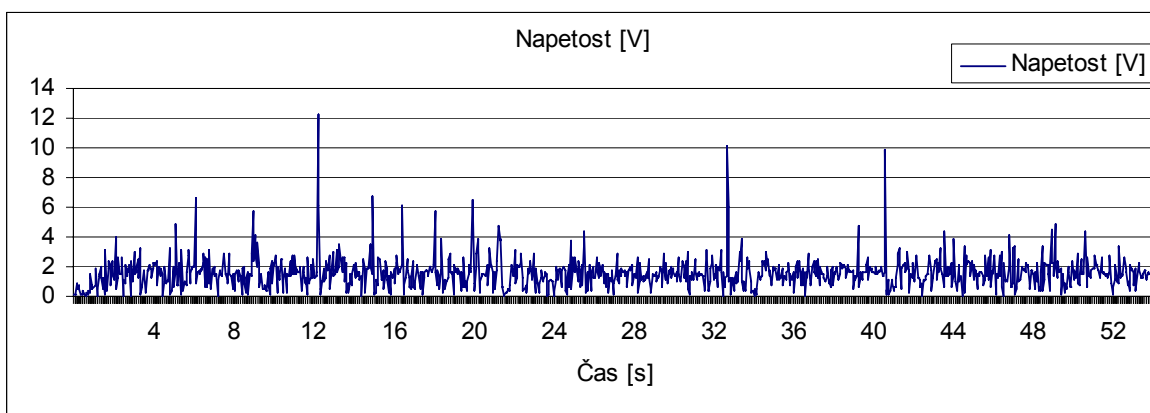
Slika 6.4: Stopnični odziv izbranega sistema

Izračunana ojačanja:

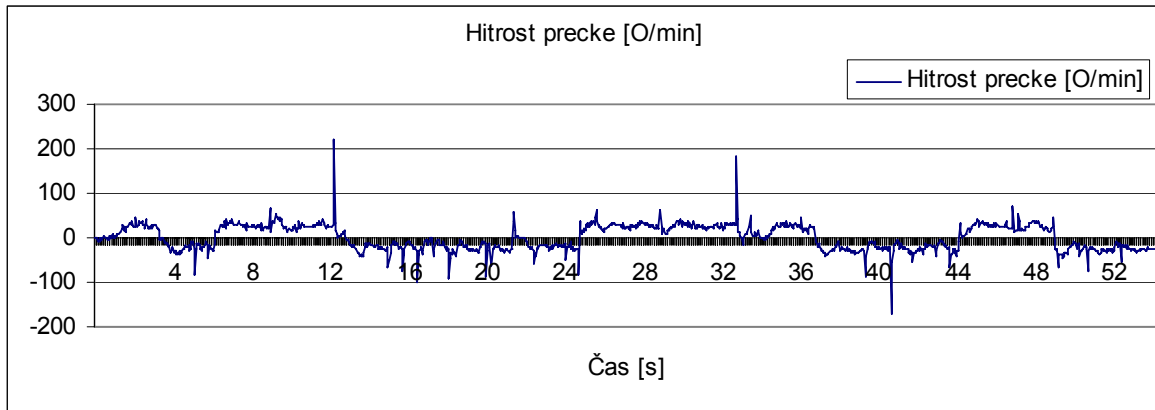
$$K_1 = -0.5310 \quad K_2 = 12.3609 \quad K_3 = 2.1225$$

Zaprtozančno ojačanje:

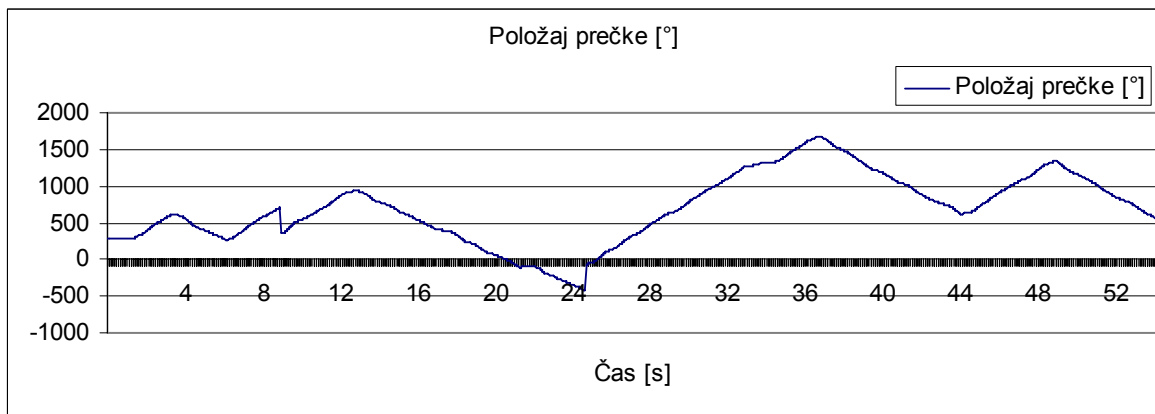
$$k = 2200$$



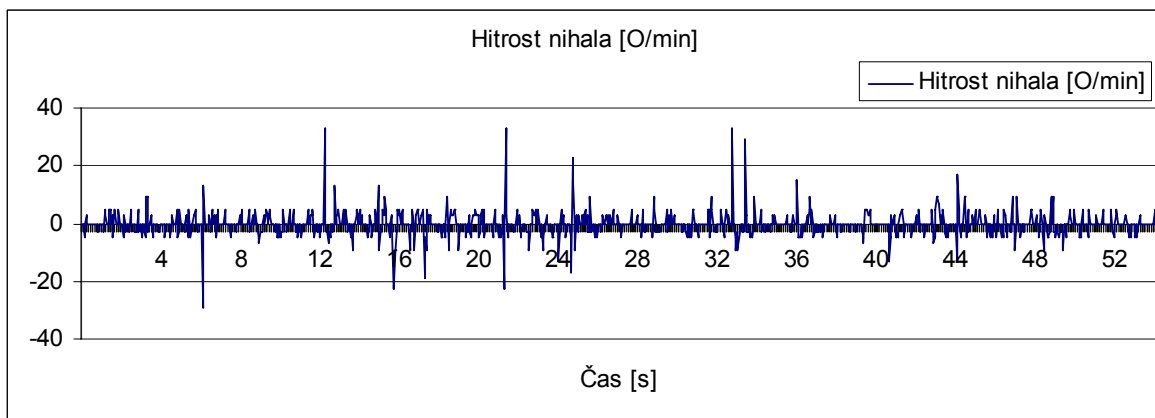
Slika 6.5: Meritev sistema tretjega reda



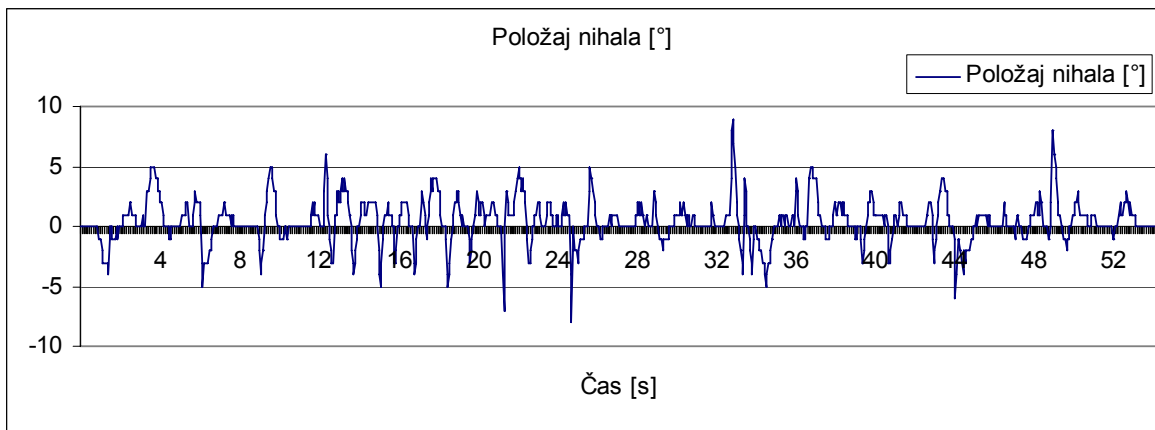
Slika 6.6: Meritev sistema tretjega reda



Slika 6.7: Meritev sistema tretjega reda



Slika 6.8: Meritev sistema tretjega reda



Slika 6.9: Meritev sistema tretjega reda

Izbrani parametri in s tem zaprtizančni poli sistema:

$$d = 0.5$$

$$\omega_0 = 8$$

$$s_1 = -4.0000 + 6.9282i$$

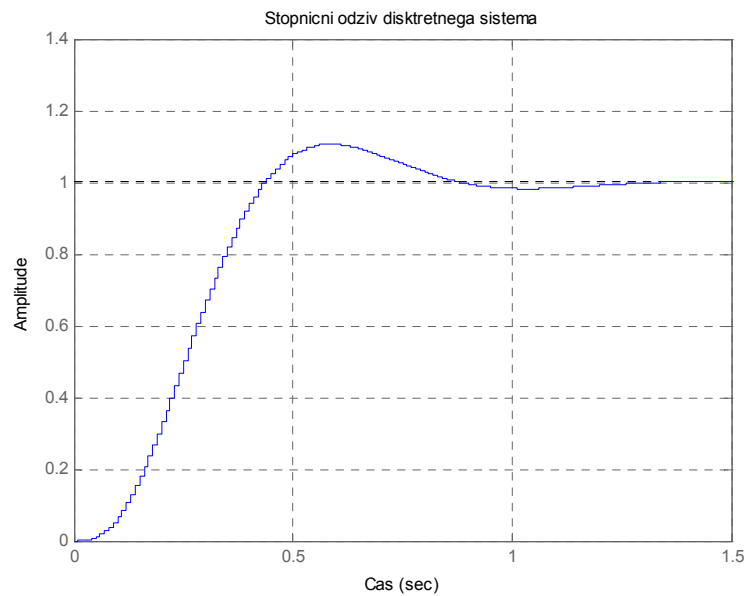
$$s_2 = -4.0000 - 6.9282i$$

$$s_3 = -10.0000$$

$$z_1 = 0.9585 + 0.0665i$$

$$z_2 = 0.9585 - 0.0665i$$

$$z_3 = 0.9048$$



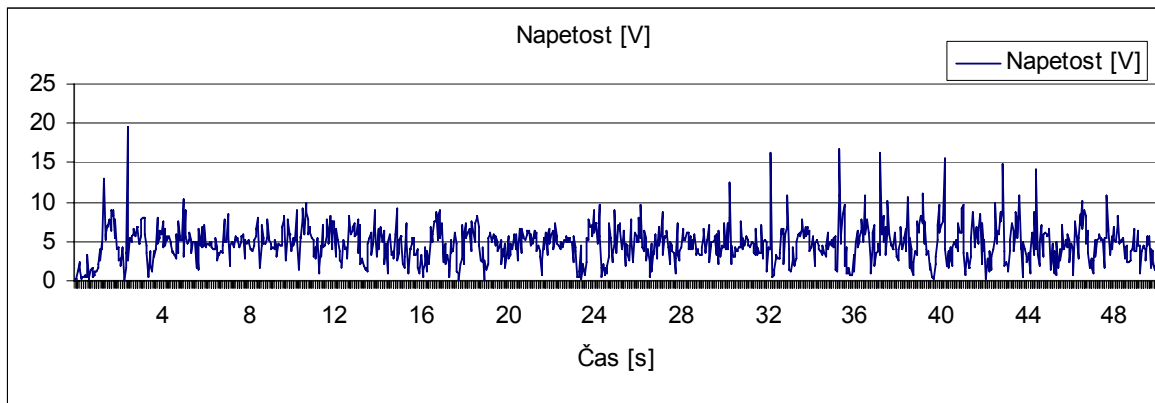
Slika 6.10: Stopnični odziv izbranega sistema

Izračunana ojačanja:

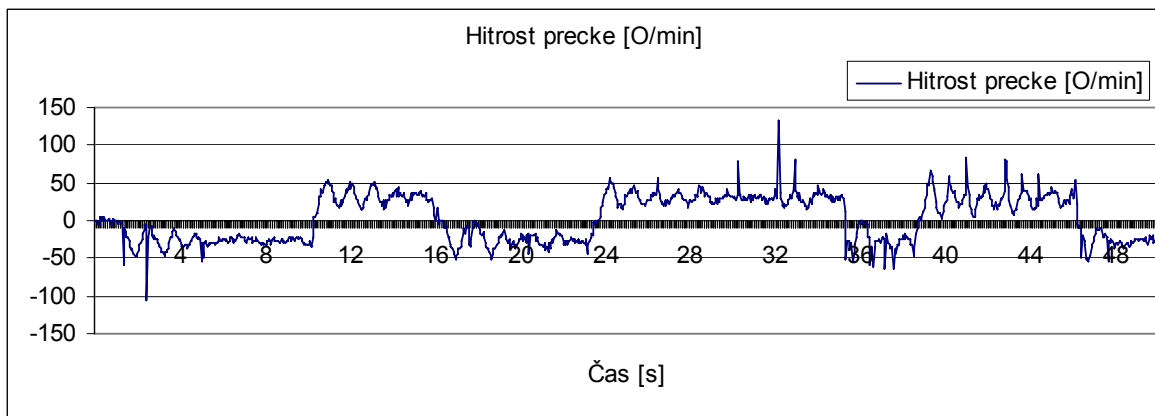
$$K_1 = -1.4954 \quad K_2 = 30.7716 \quad K_3 = 4.7847$$

Zaprtozančno ojačanje:

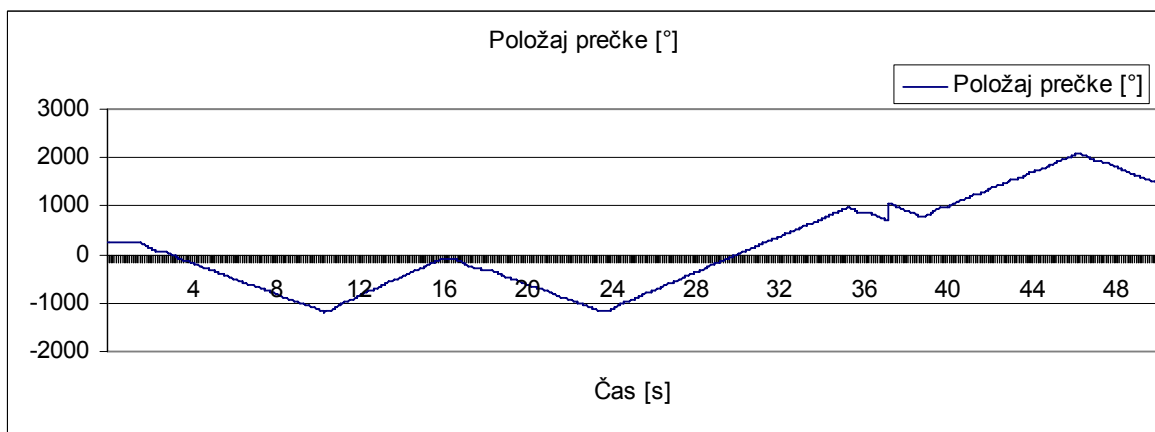
$$k = 720$$



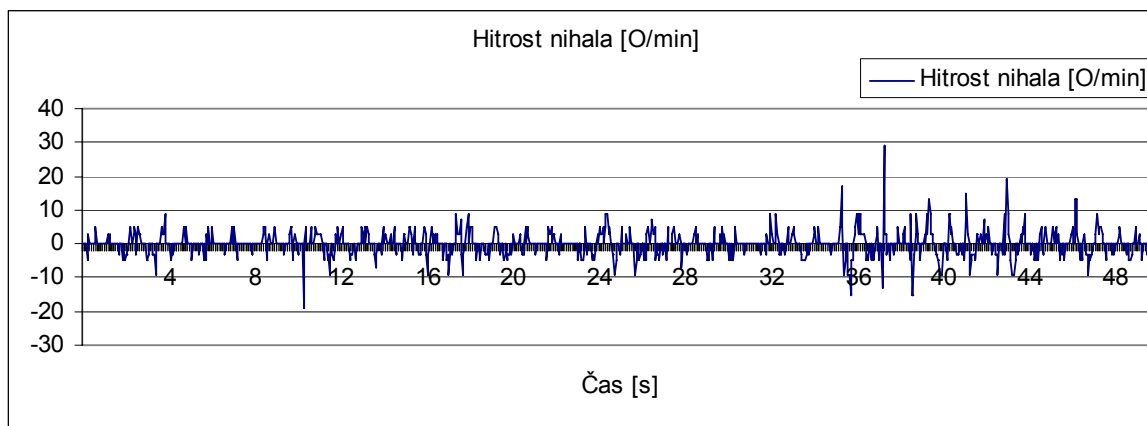
Slika 6.11: Meritev sistema tretjega reda



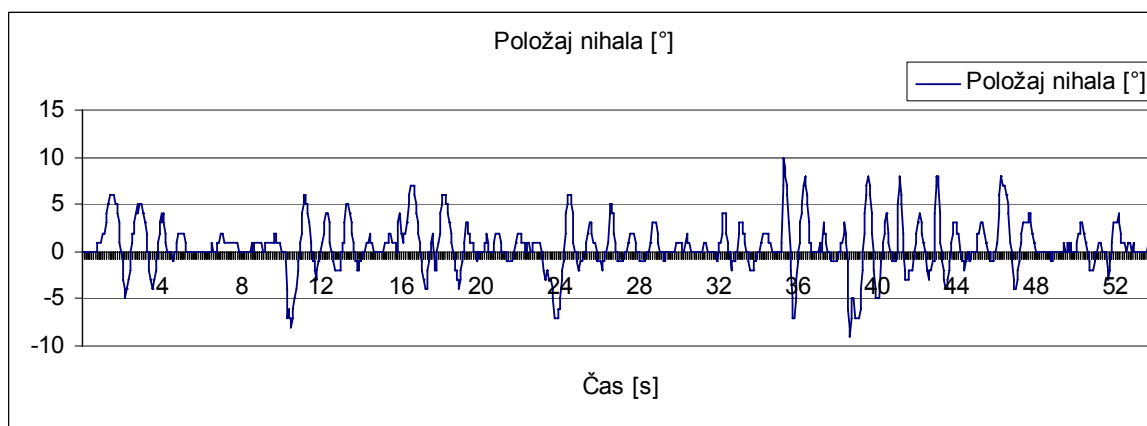
Slika 6.12: Meritev sistema tretjega reda



Slika 6.13: Meritev sistema tretjega reda



Slika 6.14: Meritev sistema tretjega reda



Slika 6.15: Meritev sistema tretjega reda

Regulator za pomikanje polov smo izračunali in izvedli s pomočjo modela (6.2). Tako iznos izhodne napetosti iz regulatorja ni bil odvisen od položaja prečke. Kot vidimo na zgornjih meritvah, je položaj prečke naraščal ali padal. Do spremembe položaja je prišlo enostavno takrat, ko smo nihalo ročno pahnili v nasprotno smer. Kot vidimo ja prišlo ravno ob spremembah položaja prečke, do spremembe smeri hitrosti in do večjih sprememb položaja nihala. Kot vidimo smo lahko pri prvem primeru regulatorja dopustili visoko zaprtozančno ojačanje k , v drugem primeru pa smo ga morali več kot za polovico zmanjšati. Dinamika drugega izračunanega regulatorja je bila enostavno previsoka. Tako smo v bistvu dobili dokaj podobne rezultate. Tudi če pogledamo razmerja med ojačanji teh dveh regulatorjev, ugotovimo, da ni večjih razlik. Z meritvami smo ugotovili, da je smiselna takšna izbira zaprtozančnih polov sistema, pri katerih dobimo regulator, ki ima najvišjo občutljivost pri položaju nihala.

6.2 Izvedba regulatorja z referenčnim vhodom

Za izračun smo uporabili model (6.3). Kot referenčno stanje smo vzeli položaj prečke. Kljub temu, da smo izvedli več različnih regulatorjev, smo se odločili, da v tem delu predstavimo le dva.

Izbrani parametri in s tem zaprtizančni poli sistema:

$$d = 0.0650$$

$$\omega_0 = 4.32$$

$$s_1 = -0.2808 + 4.3109i$$

$$s_2 = -0.2808 - 4.3109i$$

$$s_3 = -1.6142$$

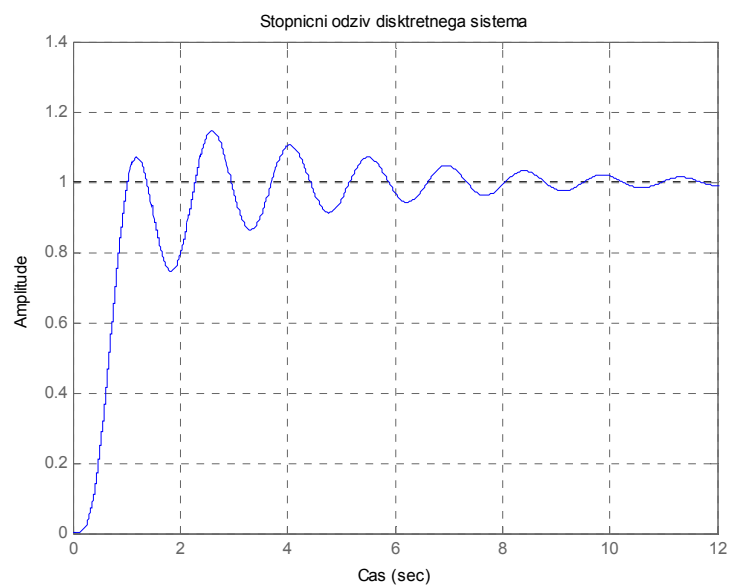
$$s_4 = -9.2821$$

$$z_1 = 0.9963 + 0.0430i$$

$$z_2 = 0.9963 - 0.0430i$$

$$z_3 = 0.9840$$

$$z_4 = 0.9114$$



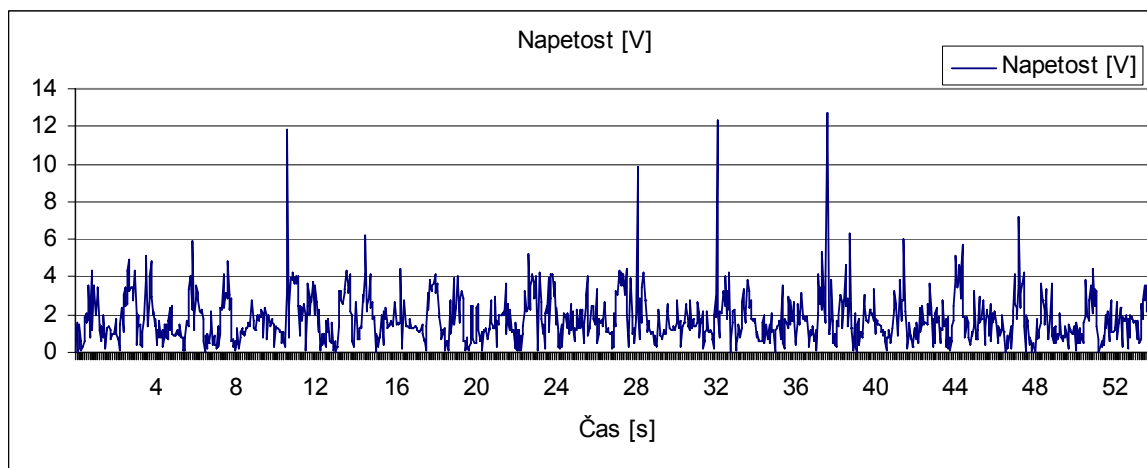
Slika 6.16: Stopnični odziv izbranega sistema

Izračunana ojačanja:

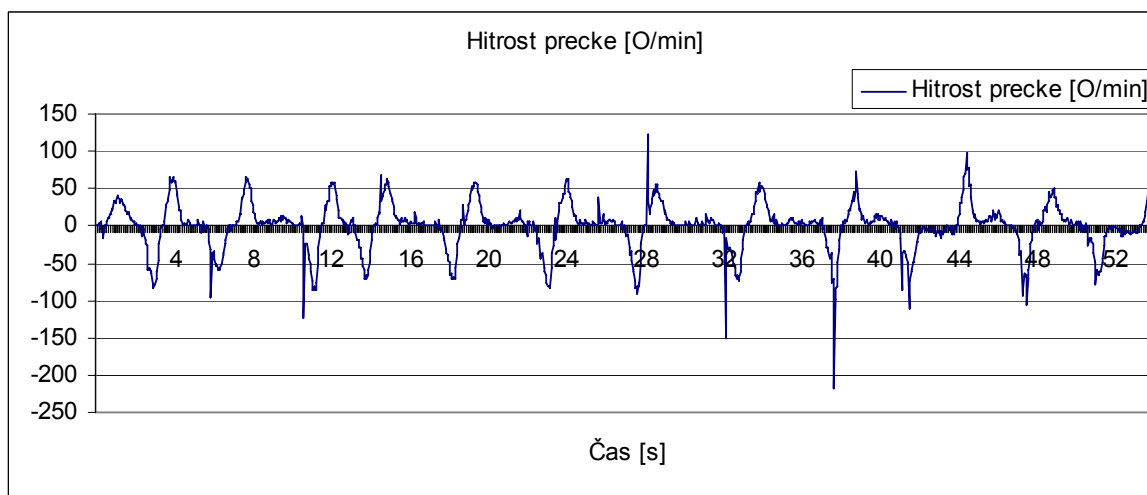
$$K_1 = -0.6063 \quad K_2 = -0.6166 \quad K_3 = 15.7254 \quad K_4 = 2.4974$$

Zaprozančno ojačanje:

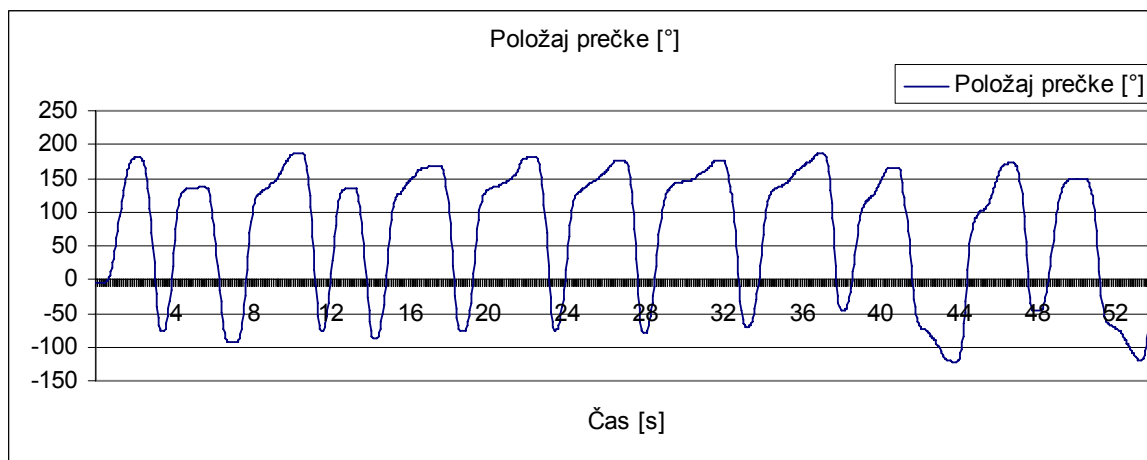
$$k = 1800$$



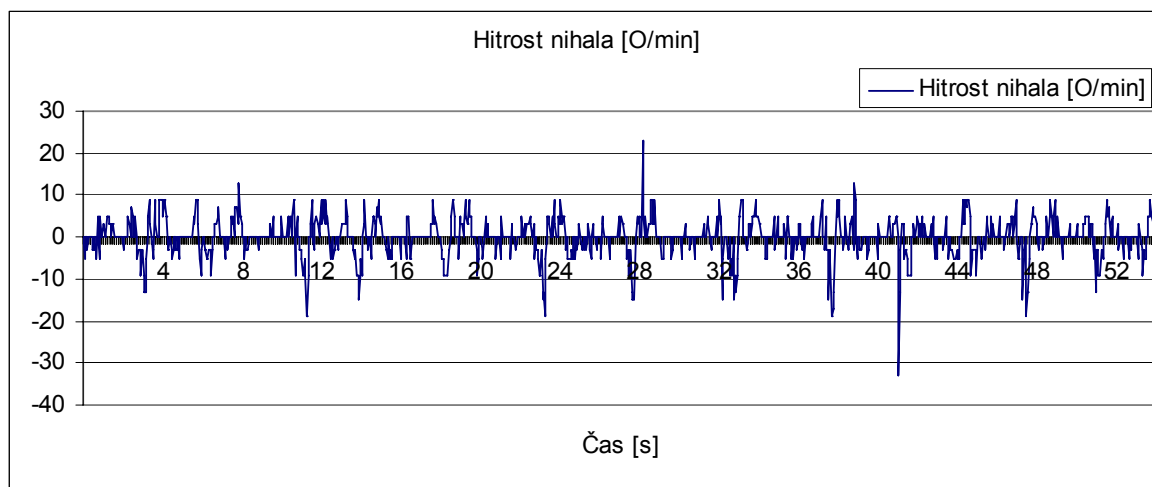
Slika 6.17: Merite sistema četrtega reda



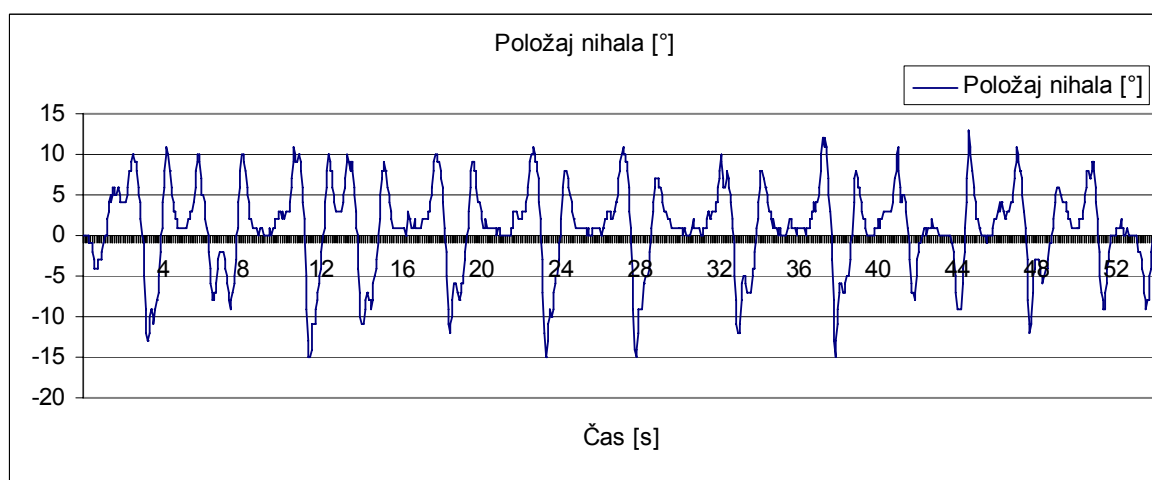
Slika 6.18: Merite sistema četrtega reda



Slika 6.19: Merite sistema četrtega reda



Slika 6.20: Merite sistema četrtega reda



Slika 6.21: Merite sistema četrtega reda

Izbrani parametri in s tem zaprtizančni poli sistema:

$$d = 0.5$$

$$\omega_0 = 8$$

$$s_1 = -4.0000 + 6.9282i$$

$$s_2 = -4.0000 - 6.9282i$$

$$s_3 = -5.0000$$

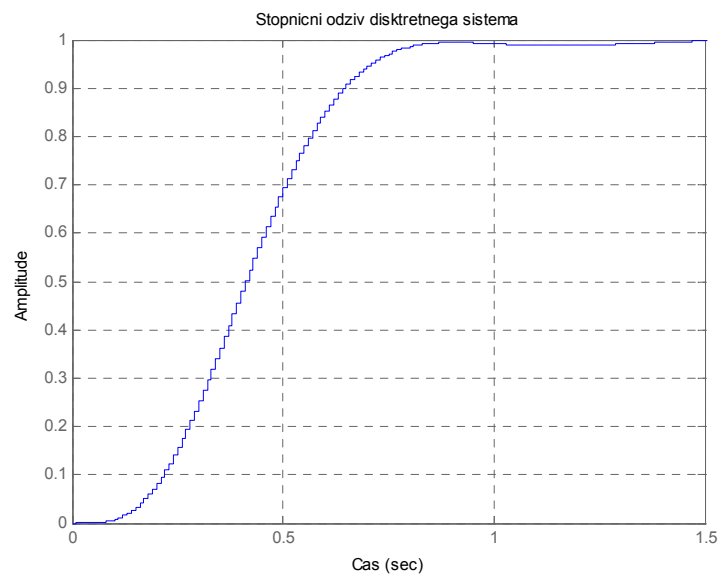
$$s_4 = -10.0000$$

$$z_1 = 0.9585 + 0.0665i$$

$$z_2 = 0.9585 - 0.0665i$$

$$z_3 = 0.9512$$

$$z_4 = 0.9048$$



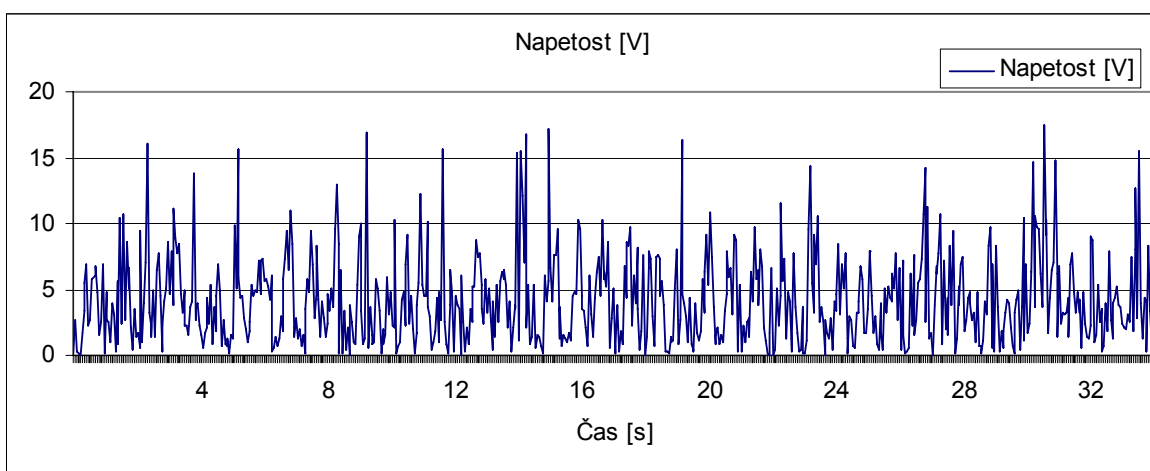
Slika 6.22: Stopnični odziv izbranega sistema

Izračunana ojačanja:

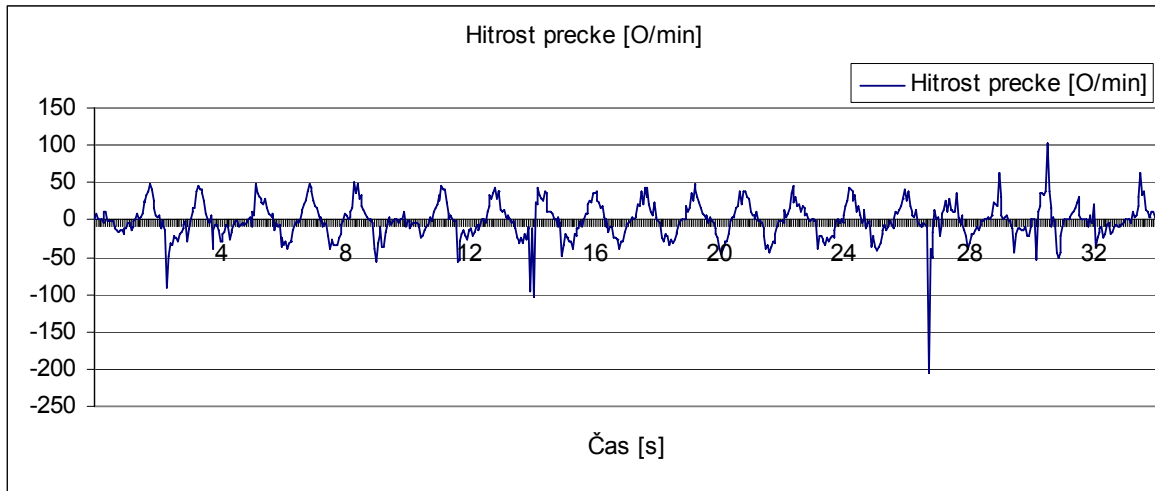
$$K_1 = -6.5499 \quad K_2 = -2.9910 \quad K_3 = 53.8479 \quad K_4 = 7.7033$$

Zaprozančno ojačanje:

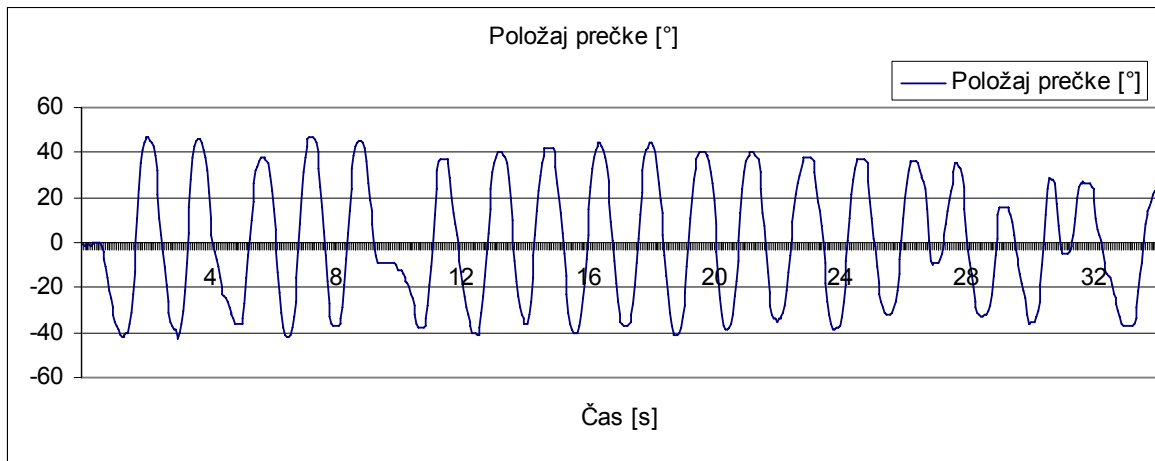
$$k = 750$$



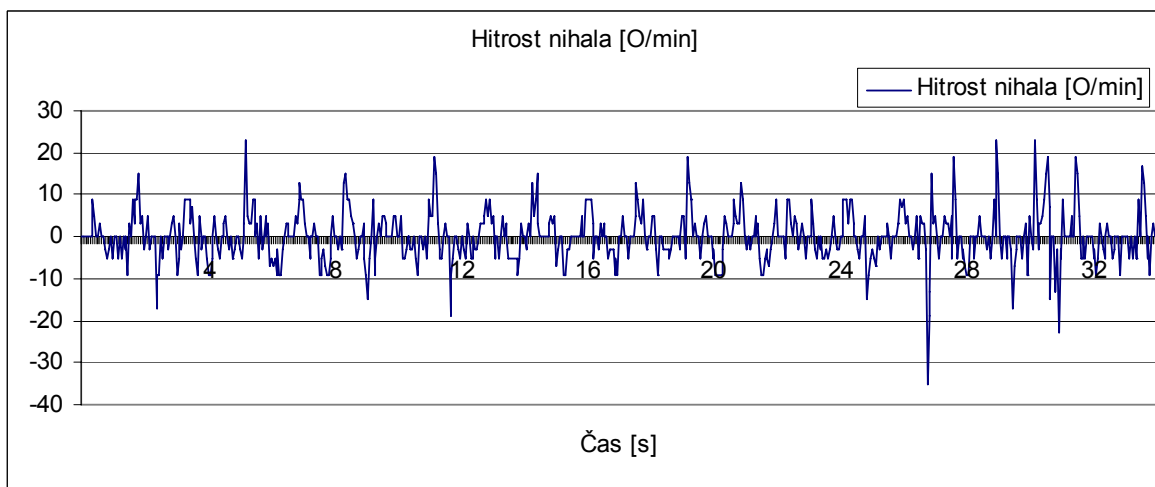
Slika 6.23: Merite sistema četrtega reda



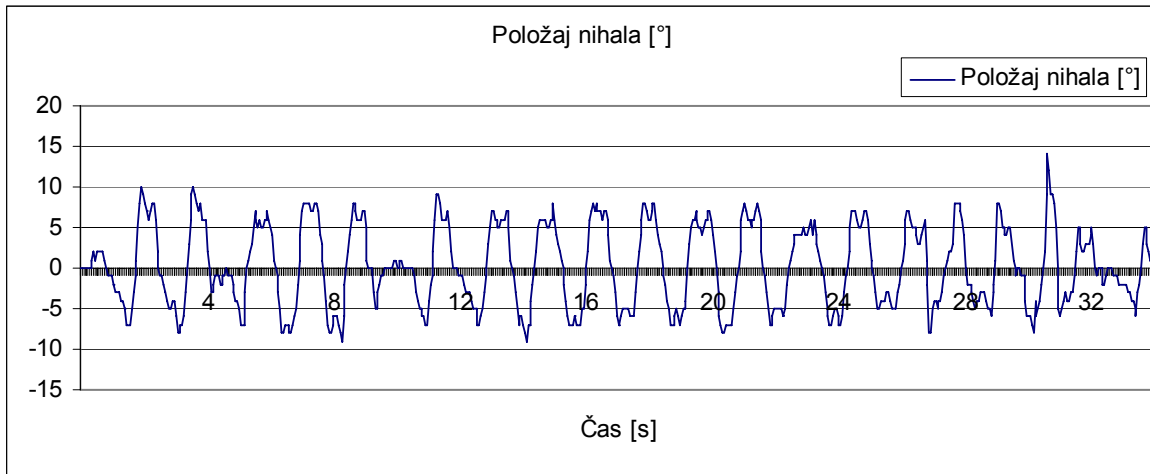
Slika 6.24: Merite sistema četrtega reda



Slika 6.25: Merite sistema četrtega reda



Slika 6.26: Merite sistema četrtega reda



Slika 6.27: Merite sistema četrtega reda

Kot referenčno stanje smo izbrali položaj prečke. Tako je regulator izračunaval vhodno napetost motorja na podlagi vseh stanj. Kot vidimo iz meritev, so razlike pri teh dveh regulatorjih več kot očitne. Namen regulatorjev je popolna stabilizacija nihala. Kot vidimo nihalo ves čas niha okrog navpične lega. Podobno se dogaja s prečko. Že na prvi pogled so ti nihaji veliko večji pri prvem regulatorju. Prav tako so večji prenehaji nihala. Smiselna je primerjava zaprtozančnih ojačanj. V tem primeru smo morali obema regulatorjema nekoliko zmanjšati zaprtozančno ojačanje k . Prvemu za slabih dvajset odstotkov, drugemu pa nekoliko več, za okrog šestdeset odstotkov. Če primerjamo izračunana zaprtozančna ojačanja regulatorjev, opazimo večje razlike. Ojačanje položaja prečke je skoraj štiri krat večje, ojačanje hitrosti prečke pa je skoraj dva krat večje. Ojačanja hitrosti in položaja nihala pa sta le nekoliko višji. To je povsem skladno z meritvami. Drugemu regulatorju smo namreč povečali občutljivost na položaj in hitrost nihala.

6.3 Izvedba LQR regulatorja

Kot smo že omenili smo ustrezna zaprtozanka ojačanja izračunali s pomočjo Matlaba z ukazom »lqr«. Zahteve ukaza pa so zraven posredovanja sistemskih matrik, tudi utežitveni matriki. Matrika Q predstavlja utežitev notranjih stanj. Izbira le diagonalnih elementov matrike je smiselna, saj tako utežimo vsako stanje posebej. Matrika R pa utežitev vhoda. Najprej bomo podali več različnih utežitvenih matrik, in izračunane vrednosti regulatorjev. Kot smo že ugotovili iz prejšnjih meritev je smiselno najbolj utežiti položaj nihala. Meritve bomo prikazali, le za dva regulatorja.

$$Q_4 = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]$$

$$R_4 = [0.1]$$

$$K_1 = -2.5083 \quad K_2 = -3.5163 \quad K_3 = 79.6607 \quad K_4 = 11.4929$$

$$Q_4 = [10 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1]$$

$$R_4 = [0.1]$$

$$K_1 = -7.8547 \quad K_2 = -5.2902 \quad K_3 = 100.2101 \quad K_4 = 14.3438$$

$$Q_4 = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 0 \ 0 \ 1]$$

$$R_4 = [0.1]$$

$$K_1 = -3.1623 \quad K_2 = -4,3972 \quad K_3 = 98,5505 \quad K_4 = 14,0358$$

$$Q_4 = [1 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 5]$$

$$R_4 = [0.1]$$

$$K_1 = -2.2433 \quad K_2 = -4.1063 \quad K_3 = 96.7711 \quad K_4 = 14.4718$$

Kot lahko razberemo iz zgornjih izračunov vidimo, da se kljub zelo različnim nastavitvam utežitvenih matrik, rezultati regulatorjev ne razlikujejo bistveno. Vidimo, da je stanje

položaja nihala močno uteženo. Nekoliko lahko vplivamo tudi na občutljivost položaja in hitrosti prečke. To pomeni, da lahko regulatorju nekoliko povečamo občutljivost položaja prečke v primerjavi z občutljivostjo hitrosti prečke, ali obratno.

Za sistem tretjega reda, je izbira utežitvene matrike zelo enostavna.

$$Q_3 = [2 \ 0 \ 0 \ 0 \ 8 \ 0 \ 0 \ 0 \ 4]$$

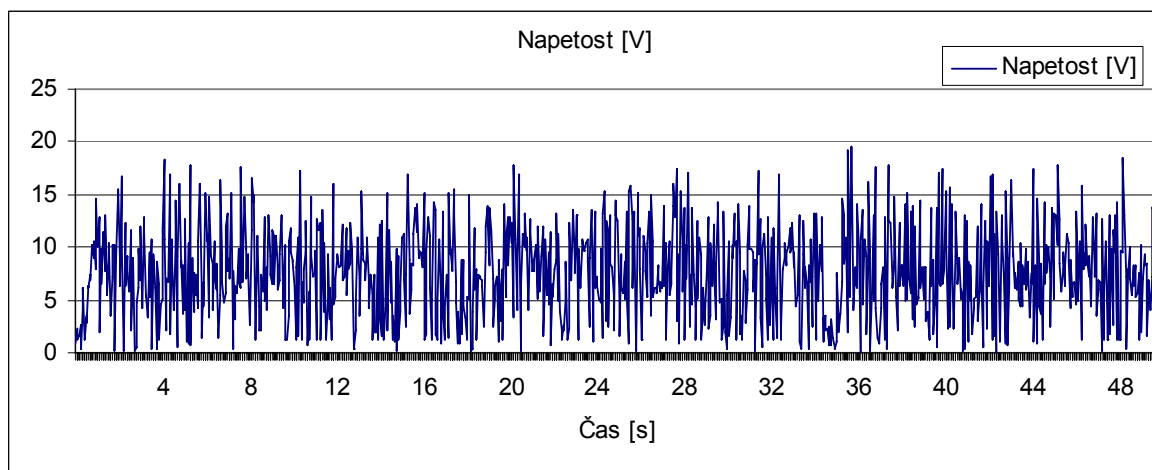
$$R_3 = [0.1]$$

Dobili smo naslednji regulator:

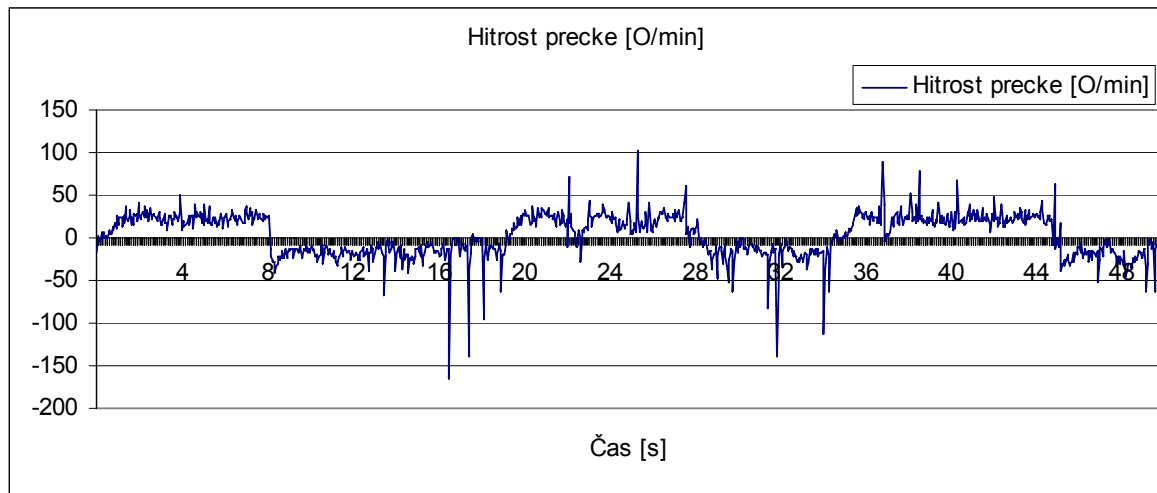
$$K_1 = -3.3826 \quad K_2 = 87.3203 \quad K_3 = 13.0964$$

Povratno zančno ojačanje k :

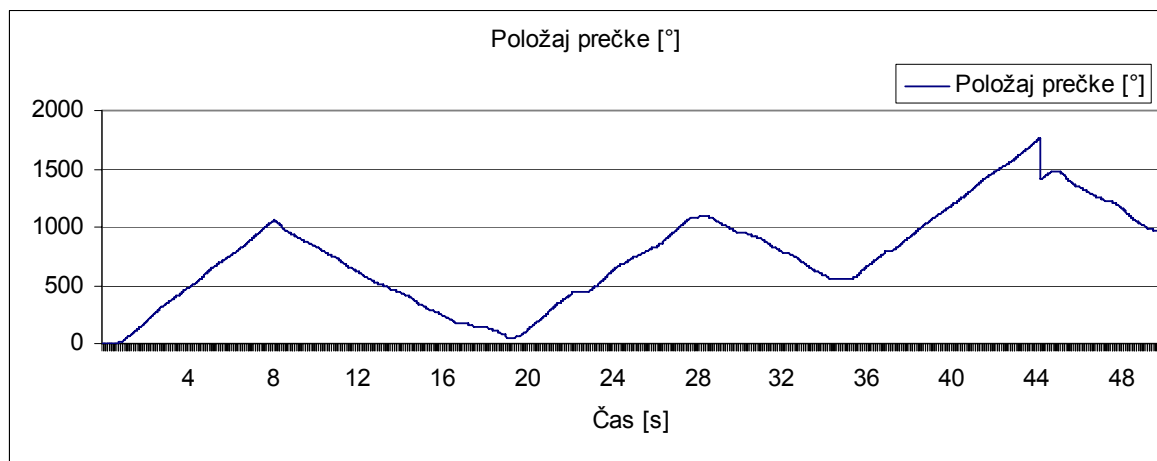
$$k = 400$$



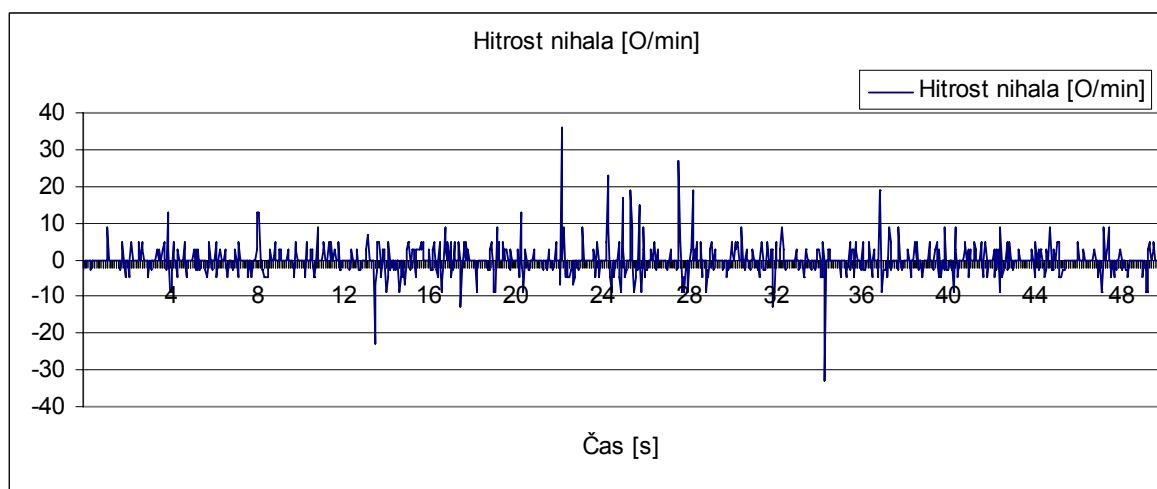
Slika 6.28: Meritev sistema tretjega reda



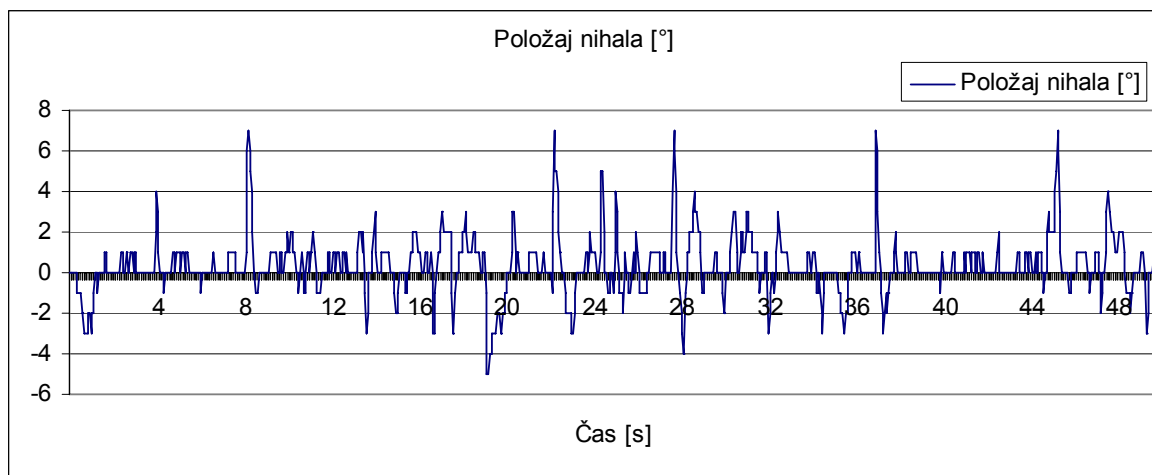
Slika 6.29: Meritev sistema tretjega reda



Slika 6.30: Meritev sistema tretjega reda



Slika 6.31: Meritev sistema tretjega reda



Slika 6.32: Meritev sistema tretjega reda

Za načrtovanje sistema četrtega reda, kjer smo izbrali položaj prečke kot referenčno, smo izbrali:

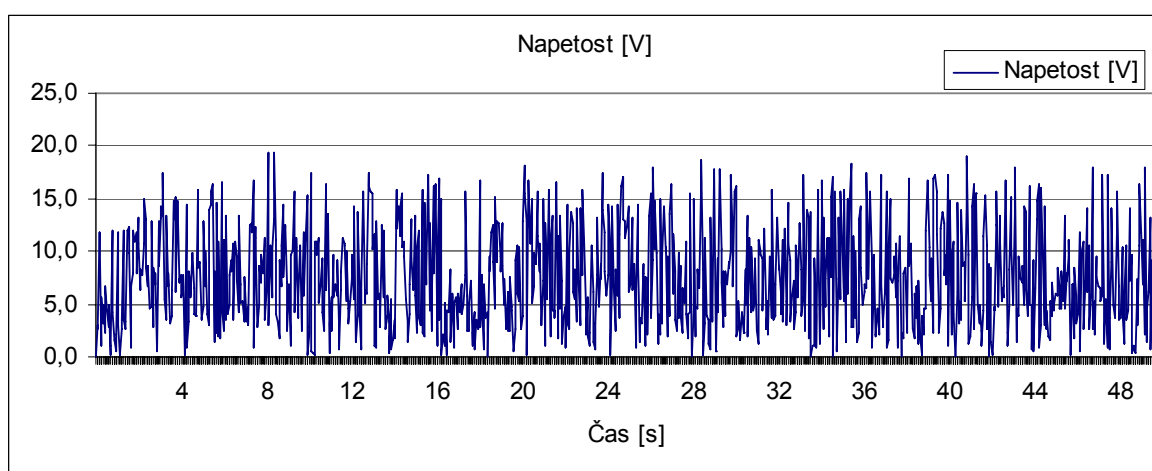
$$Q_4 = [2 \ 0 \ 0 \ 0 \ 0 \ 4 \ 0 \ 0 \ 0 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \ 8]$$

$$R_4 = [0.1]$$

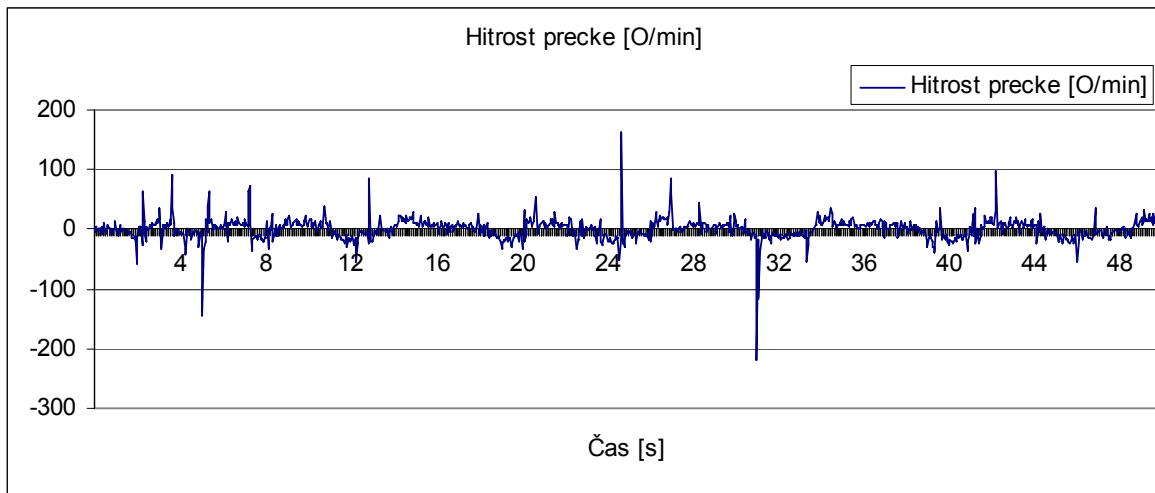
Dobili smo:

$$K_1 = -2.8847 \quad K_2 = -5.2015 \quad K_3 = 117.7250 \quad K_4 = 17.5764$$

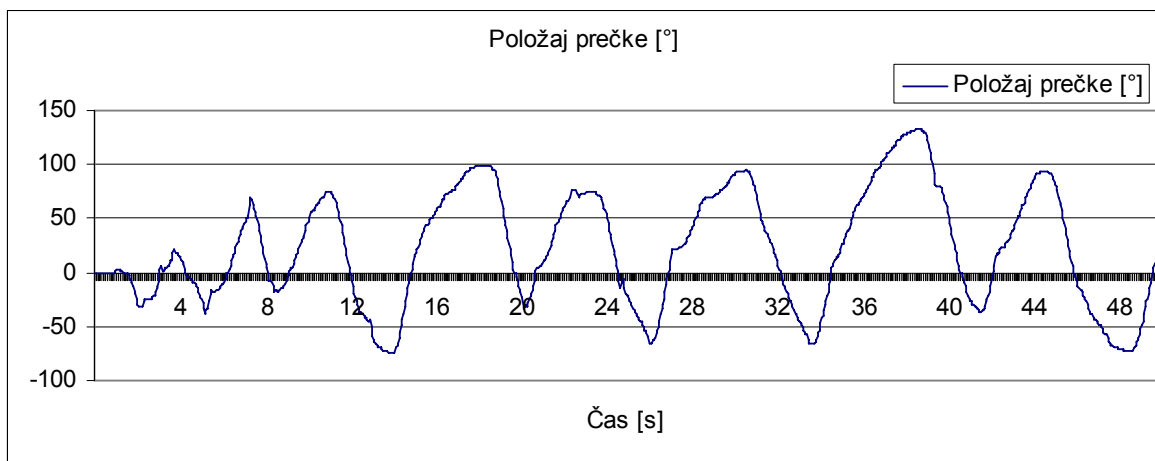
Povratno zanjno ojačanje k : $k = 400$



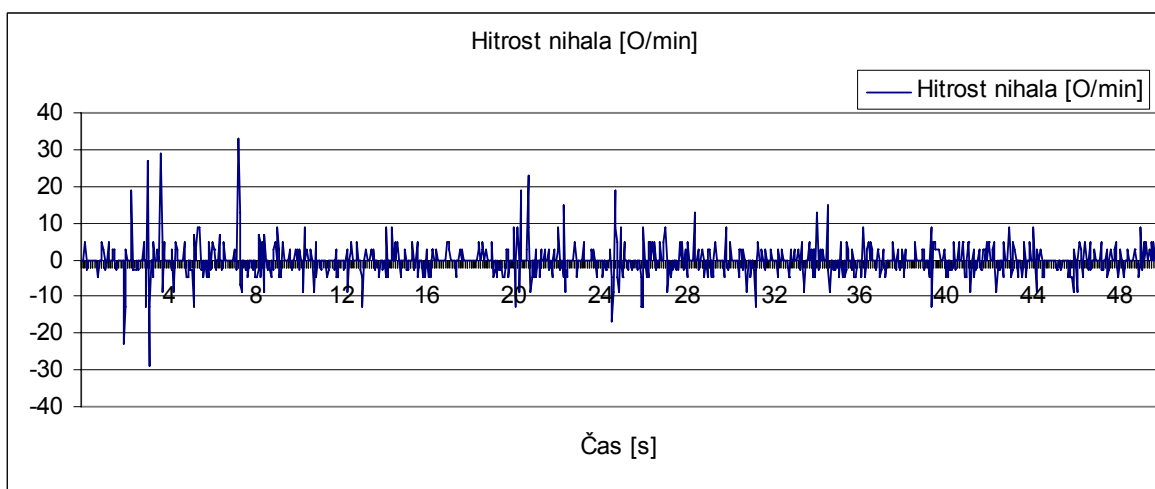
Slika 6.33: Meritev sistema četrtega reda



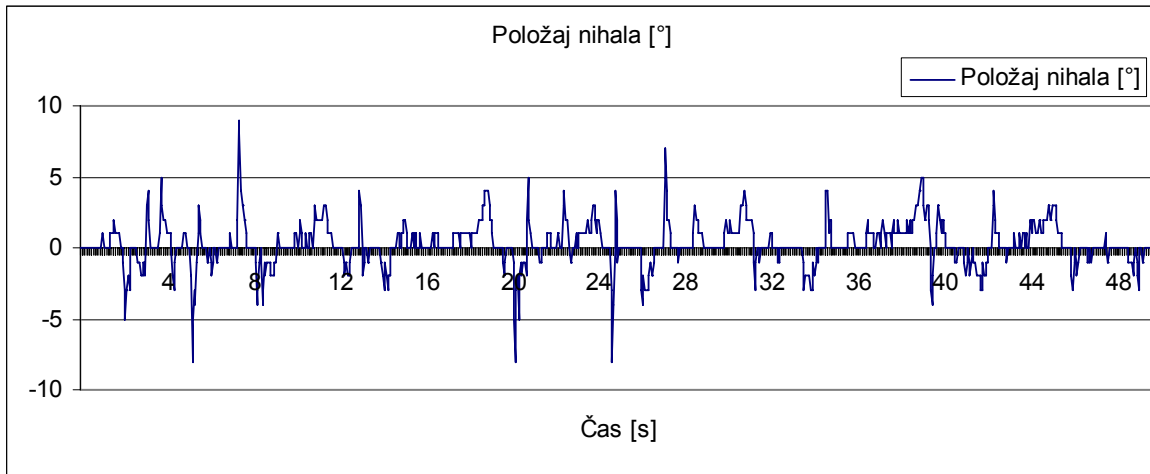
Slika 6.34: Meritev sistema četrtega reda



Slika 6.35: Meritev sistema četrtega reda



Slika 6.36: Meritev sistema četrtega reda



Slika 6.37: Meritev sistema četrtega reda

Izračunali smo dva linearna kvadratična regulatorja. Pri prvem smo za načrtovanje uporabili sistem tretjega reda. Pri drugem pa sistem četrtega reda in položaj prečke kot referenčno stanje. Tudi tukaj smo morali zmanjšati zaprtozančno ojačanje k , saj imata oba regulatorja visoko ojačanje položaja nihala. Če meritve primerjamo s prej omenjenimi regulatorji, že na prvi pogled vidimo, da sta položaja nihala veliko mirnejša. Hitrosti nihala pa veliko nižje. Če primerjamo ta dva regulatorja s prejšnjimi ugotovimo, da imata višjo občutljivost na položaj nihala. S povečevanjem občutljivosti na položaj nihala pa se povečuje tudi občutljivost na hitrost nihala. Visoka občutljivost regulatorja na hitrost nihala pa ni zaželena. Že uvodoma smo namreč ugotovili, da k hitrosti priintegriramo tudi majhne pogreške AD pretvornika, ki pa zdaj več niso zanemarljivi.

7 SKLEP

V tem delu nam je uspelo realizirati vse potrebne komponente procesorja, ki smo jih uspešno uporabili pri vodenju IRN. Samo zasnovo programskih gradnikov smo izvedli v nekoliko širšem smislu, predvsem zaradi zahteve po nadaljnji uporabi DSP-ja. Tako so posamezne komponente enostavno prilagodljive za morebitne druge objekte vodenja. Podobno so izvedeni komunikacijski moduli DSP-ja. Osnovni gradniki oz. funkcije, ki omogočajo komunikacijo z zunanjimi napravami (*SPI*) ali osebnim računalnikom (*SCI*) so izvedene v celoti. Tako je prilagoditev sistema enostavna in učinkovita.

Po ustrezni prilagoditvi napetostnih nivojev, smo izvedli uspešno migracijo procesorja v sistem IRN. Merili smo vse notranje spremenljivke sistema razen hitrosti nihala. Hitrost nihala smo izračunali na podlagi trenutne in prejšnje izmerjene vrednosti položaja nihala. Predvidevali smo dokaj velike pogoške pri hitrosti nihala, ampak meritve so pokazale, da temu ni tako. Pogoške hitrosti bi lahko zelo zmanjšali z implementacijo enkoderja za merjenje hitrosti. Tako bi merili vsa štiri stanja.

Z linearnim kvadratičnim regulatorjem smo dosegli veliko boljše rezultate, saj je bilo nihalo veliko mirnejše. Vendar je prečka veliko bolj nihala okrog svoje ravnovesne lege. Tako, da tudi regulatorja za pomikanje polov ne gre zanemariti.

Nadaljnje raziskave bi lahko potekale v smislu iskanja še ustrežnejšega razmerja med ojačanju regulatorja. Vendar je zaradi relativno visoke procesorske moči DSP-ja smiselno razmisliti tudi o implementaciji regulatorja na osnovi mehke logike ali nevronske mreže.

“Če se na poti proti cilju ustavljaš, da bi vrgel kamen v vsakega psa, ki laja nate, ne boš nikdar prispel do cilja.”

Fjodor Mihajlovič Dostojevski

8 VIRI, LITERATURA

- [1] R. Svečko, *Diskretni regulacijski sistemi*, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, 2005.
- [2] Texas Instruments, Incorporated, *Reference Guide, TMS320x2833x, 2823x Enhanced Quadrature Encoder Pulse (eQEP) Module*, SPRUG05A, avgust 2008.
- [3] Texas Instruments, Incorporated, *Data Manual, TMS320F28335, TMS320F28334, TMS320F28332, TMS320F28235, TMS320F28234, TMS320F28232, Digital Signal Controllers (DSCs)*, SPRS439F, junij 2007.
- [4] R. Svečko, *Teorija sistemov*, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, 2005.
- [5] Tan Kok Chye, Teo Chun Sang, *Rotary inverted pendulum*, School of electrical and electronic engineering, 1999.
- [6] Texas Instruments, Incorporated, *Application Report, An Overview of Designing Analog Interface With TMS320F28xx/28xxx DSCs*, SPRAAP6, maj 2005
- [7] Texas Instruments, Incorporated, *Reference Guide, TMS320x2833x Analog-to-Digital Converter (ADC) Module*, SPRU812, september 2007.
- [8] Burr Brown Corporation, *ADS1212, ADS1213, 22Bit analog to digital converter*, USA, april 1998.
- [9] Texas Instruments, Incorporated, *Reference Guide, TMS320x2833x, 2823x DSC Serial Peripheral Interface (SPI)*, SPRUEU3, avgust 2008.
- [10] Texas Instruments, Incorporated, *Reference Guide, TMS320F2833x, 2823x Serial Communications Interface (SCI)*, SPRUFZ5, avgust 2008.
- [11] Texas Instruments, Incorporated, *Reference Guide, TMS320x2833x, 2823x Enhanced Pulse Width Modulator (ePWM) Module*, SPRUG04, oktober 2008.
- [12] Texas Instruments, Incorporated, *Reference Guide, TMS320C28x DSP CPU and Instruction Set*, SPRU430, avgust 2001.
- [13] Texas Instruments, Incorporated, *Reference Guide, TMS320x2833x, 2823x System Control and Interrupts*, SPRUFB0, september 2007.
- [14] [http://en.wikipedia.org/wiki/State_space_\(controls\)](http://en.wikipedia.org/wiki/State_space_(controls)).
- [15] http://en.wikipedia.org/wiki/Full_state_feedback.
- [16] http://en.wikipedia.org/wiki/Linear-quadratic_regulator.
- [17] http://en.wikipedia.org/wiki/Algebraic_Riccati_equation.

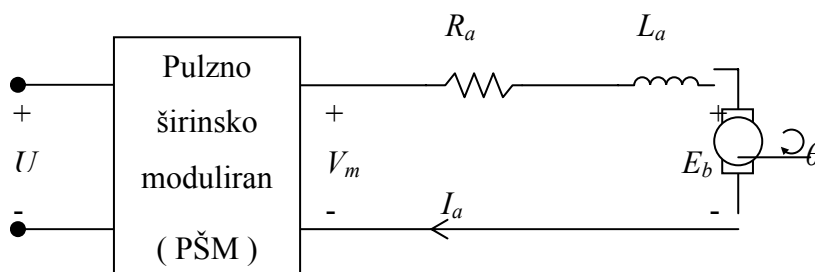
9 PRILOGE

9.1 Matematični model inverznega nihala [5]

Za izračun regulatorjev oz. algoritmov vodenja potrebujemo matematični model, ki kar se da natančno opisuje realni fizikalni objekt. Matematični model IRN je bil izpeljan s pomočjo Lagrange-ovih enačb gibanja. IRN velja za nelinearen sistem. Algoritme vodenja pa lahko izračunamo le za linearne sisteme, tako bo potrebno kompleksen sistem nelinearnih enačb linearizirati.

9.1.1 Model enosmernega motorja

Električni motor vodimo s procesorjem s pomočjo PWM. Motor je del našega sistema vodenja. Tako je nujno, da poznamo njegov matematični model.



Kjer so:

Slika 9.1: enosmerni motor

U – napajalna napetost motorja v obliki PWM

V_m – enosmerna napajalna napetost motorja

R_a – upornost kotve

L_a – induktivnost navitja

I_a – tok skozi navitje

E_b – inducirana napetost

Po Kirchoff-ovem zakonu dobimo napetostno enačbo:

$$V_m = I_a \cdot R_a + L_a \cdot \frac{dI_a}{dt} + E_b \quad (9.1)$$

Inducirana napetost je odvisna od spremembe magnetnega sklepa v času in tako od hitrosti rotorja.

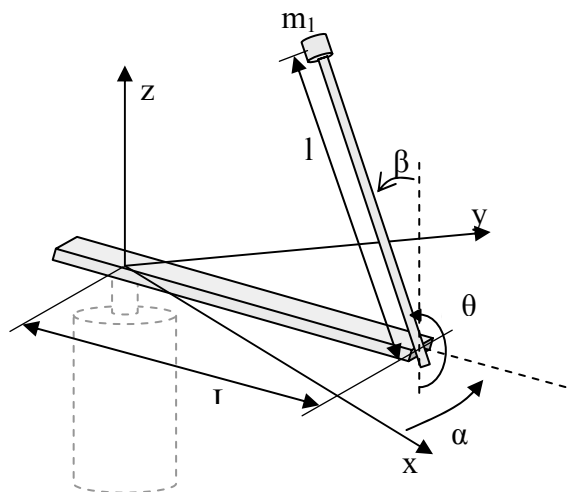
$$E_b = K_b \cdot \frac{d\theta}{dt} = K_b \cdot \dot{\theta} \quad (9.2)$$

Navor motorja je sorazmeren s tokom skozi navitje. Če predpostavimo, da je vpliv induktivnosti zanemarljiv, lahko navor predstavimo kot:

$$M_m = K_t \cdot I_a$$

$$M_m = \frac{K_t \cdot (V_m - E_b)}{R_a} = \frac{K_t \cdot V_m - K_t \cdot K_b \cdot \dot{\theta}}{R_a} \quad (9.3)$$

9.1.2 Model inverznega nihala



Slika 9.2: Koordinatni sistem IRN

Matematični model je osnovan na desno sučnem kartezijskem koordinatnem sistemu. Izhodišče kota β je v navpični legi nihala, kota θ pa v spodnji legi.

Model je bil izpeljan s pomočjo Lagrange-evih enačb:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Theta}_i} \right) - \frac{\partial L}{\partial \Theta_i} = Q_i \quad (9.4)$$

$$L = W_k - W_p, \quad (9.5)$$

Kjer je:

W_k – kinetična energija sistema,

W_p – potencialna energija sistema,

Q_i – doveden navor na i-ti osi,

Θ_i – kot telesa glede na i-to os,

L – Lagrange

Za prečko lahko kinetično in potencialno energijo zapišemo kot:

$$W_{kp} = \frac{J_0 \cdot \dot{\alpha}^2}{2} \quad (9.6)$$

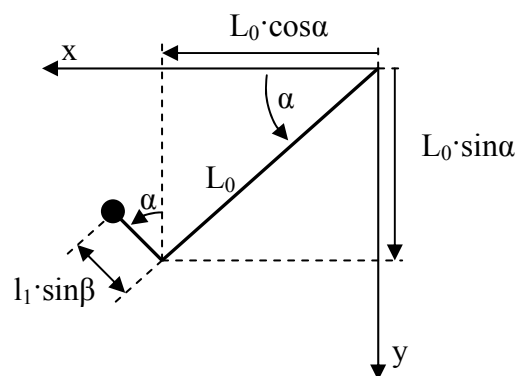
$$W_{pp} = 0 \quad (9.7)$$

Kjer je:

W_{kp} – kinetična energija prečke,

W_{pp} – potencialna energija prečke

Na sliki 5.3 vidimo tloris nihala. Iz tega lahko izpeljemo enačbe kinetične in potencialne energije nihala.



Slika 9.3: Tloris inverznega nihala

Enačbe položaja:

$$x_1 = L_0 \cdot \cos \alpha + l_1 \cdot \sin \alpha \cdot \sin \beta \quad (9.8)$$

$$y_1 = L_0 \cdot \sin \alpha - l_1 \cdot \cos \alpha \cdot \sin \beta \quad (9.9)$$

$$z_1 = l_1 \cdot \cos \beta \quad (9.10)$$

Po odvajanju dobimo enačbe hitrosti:

$$\dot{x}_1 = -L_0 \cdot \dot{\alpha} \cdot \sin \alpha + l_1 \cdot \dot{\alpha} \cdot \cos \alpha \cdot \sin \beta + l_1 \cdot \dot{\beta} \cdot \sin \alpha \cdot \cos \beta \quad (9.11)$$

$$\dot{y}_1 = L_0 \cdot \dot{\alpha} \cdot \cos \alpha + l_1 \cdot \dot{\alpha} \cdot \sin \alpha \cdot \sin \beta - l_1 \cdot \dot{\beta} \cdot \cos \alpha \cdot \cos \beta \quad (9.12)$$

$$\dot{z}_1 = -l_1 \cdot \dot{\beta} \cdot \sin \beta \quad (9.13)$$

Za izračun kinetične energije bomo potrebovali kvadrate hitrosti

$$\begin{aligned} \dot{x}_1^2 &= L_0^2 \cdot \dot{\alpha}^2 \cdot \sin^2 \alpha + l_1^2 \cdot \dot{\alpha}^2 \cdot \cos^2 \alpha \cdot \sin^2 \beta + l_1^2 \cdot \dot{\beta}^2 \cdot \sin^2 \alpha \cdot \cos^2 \beta \\ &\quad - 2 \cdot L_0 \cdot l_1 \cdot \dot{\alpha}^2 \cdot \sin \alpha \cdot \cos \alpha \cdot \sin \beta - 2 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin^2 \alpha \cdot \cos \beta \\ &\quad + 2 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin \alpha \cdot \cos \alpha \cdot \sin \beta \cdot \cos \beta \end{aligned} \quad (9.14)$$

$$\begin{aligned} \dot{y}_1^2 &= L_0^2 \cdot \dot{\alpha}^2 \cdot \cos^2 \alpha + l_1^2 \cdot \dot{\alpha}^2 \cdot \sin^2 \alpha \cdot \sin^2 \beta + l_1^2 \cdot \dot{\beta}^2 \cdot \cos^2 \alpha \cdot \cos^2 \beta \\ &\quad + 2 \cdot L_0 \cdot l_1 \cdot \dot{\alpha}^2 \cdot \sin \alpha \cdot \cos \alpha \cdot \sin \beta - 2 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \cos^2 \alpha \cdot \cos \beta \\ &\quad - 2 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin \alpha \cdot \cos \alpha \cdot \sin \beta \cdot \cos \beta \end{aligned} \quad (9.15)$$

$$\dot{z}_1^2 = l_1^2 \cdot \dot{\beta}^2 \cdot \sin^2 \beta \quad (9.16)$$

Vsota kvadratov hitrosti v posameznih smereh je:

$$\begin{aligned} \dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2 &= L_0^2 \cdot \dot{\alpha}^2 + l_1^2 \cdot \dot{\alpha}^2 \cdot \sin^2 \beta + l_1^2 \cdot \dot{\beta}^2 \\ &\quad - 2 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \cos \beta \end{aligned} \quad (9.17)$$

Sedaj lahko zapišemo še enačbi za kinetično in potencialno energijo nihala:

$$\begin{aligned} W_{kn} &= \frac{J_1 \cdot \dot{\beta}^2}{2} + \frac{m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2)}{2} \\ &= \frac{m_1 \cdot L_0^2}{2} \cdot \dot{\alpha}^2 + \frac{(J_1 + m_1 \cdot l_1^2)}{2} \cdot \dot{\beta}^2 + \frac{m_1 \cdot l_1^2}{2} \cdot \dot{\alpha}^2 \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \cos \beta \end{aligned} \quad (9.18)$$

$$W_{pn} = m_1 \cdot g \cdot l_1 \cdot \cos \beta \quad (9.19)$$

Zgornje enačbe vstavimo v enačbo (9.5):

$$L = W_{kp} + W_{kn} - W_{pp} - W_{pn} \quad (9.20)$$

$$L = \frac{(J_0 + m_1 \cdot L_0^2)}{2} \cdot \dot{\alpha}^2 + \frac{(J_1 + m_1 \cdot l_1^2)}{2} \cdot \dot{\beta}^2 + \frac{m_1 \cdot l_1^2}{2} \cdot \dot{\alpha}^2 \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \cos \beta - m_1 \cdot g \cdot l_1 \cdot \cos \beta \quad (9.21)$$

Lagrange-ian odvajamo po spremenljivki α .

$$\begin{aligned} \frac{\partial L}{\partial \alpha} &= 0 \\ \frac{\partial L}{\partial \dot{\alpha}} &= (J_0 + m_1 \cdot L_0^2) \cdot \dot{\alpha} + m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta} \cdot \cos \beta \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}} &= (J_0 + m_1 \cdot L_0^2) \cdot \ddot{\alpha} + m_1 \cdot l_1^2 \cdot \ddot{\alpha} \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\beta} \cdot \cos \beta \\ &\quad + m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin 2\beta + m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta}^2 \cdot \sin \beta \end{aligned} \quad (9.22)$$

Po enačbi (9.4) dobimo:

$$\begin{aligned} (J_0 + m_1 \cdot L_0^2) \cdot \ddot{\alpha} + m_1 \cdot l_1^2 \cdot \ddot{\alpha} \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\beta} \cdot \cos \beta \\ + m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin 2\beta + m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta}^2 \cdot \sin \beta = M_p \end{aligned} \quad (9.23)$$

Pri čemer je:

M_p – skupen navor na prečko

$$M_p = M_m - C_0 \cdot \dot{\alpha} \quad (9.24)$$

Kjer je:

C_0 – koeficient viskoznega trenja v ležajih prečke

Skupen navor M_p na prečko je posledica navora motorja M_m zmanjšanega za iznos navora trenja, ki je posledica trenja v ležajih motorja in prečke.

V enčabo (9.24) vstavimo (9.3), to pa nato v enačbo (9.24) in dobimo:

$$\begin{aligned} & (J_0 + m_1 \cdot L_0^2) \cdot \ddot{\alpha} + m_1 \cdot l_1^2 \cdot \ddot{\alpha} \cdot \sin^2 \beta - m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\beta} \cdot \cos \beta \\ & + (C_0 + \frac{K_t \cdot K_b}{R_a}) \cdot \dot{\alpha} + m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin 2\beta + m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta}^2 \cdot \sin \beta \\ & - \frac{K_t}{R_a} \cdot V_m = 0 \end{aligned} \quad (9.25)$$

Upoštevamo še:

$$A = J_0 + m_1 \cdot L_0^2 \quad (9.26)$$

$$B = m_1 \cdot l_1^2 \quad (9.27)$$

$$C = m_1 \cdot L_0 \cdot l_1 \quad (9.28)$$

$$D = C_0 + \frac{K_t \cdot K_b}{R_a} \quad (9.29)$$

$$E = \frac{K_t}{R_a} \quad (9.30)$$

Po vstavitvi enačb (9.26)- (9.30) v (9.25) dobimo poenostavljeno obliko:

$$\begin{aligned} & A \cdot \ddot{\alpha} + B \cdot \ddot{\alpha} \cdot \sin^2 \beta - C \cdot \ddot{\beta} \cdot \cos \beta + D \cdot \dot{\alpha} + B \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin 2\beta \\ & + C \cdot \dot{\beta}^2 \cdot \sin \beta - E \cdot V_m = 0 \end{aligned} \quad (9.31)$$

Lagrange-ian odvajamo še po spremenljivki β in dobimo:

$$\begin{aligned}
\frac{\partial L}{\partial \beta} &= \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha}^2 \cdot \sin 2\beta \cdot \cos \beta + m_1 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin \beta \\
&\quad + m_1 \cdot g \cdot l_1 \cdot \sin \beta \\
\frac{\partial L}{\partial \dot{\beta}} &= (J_1 + m_1 \cdot l_1^2) \cdot \dot{\beta} - m_1 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \cos \beta \\
\frac{d}{dt} \frac{\partial L}{\partial \dot{\beta}} &= (J_1 + m_1 \cdot l_1^2) \cdot \ddot{\beta} - m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\alpha} \cdot \cos \beta + m_1 \cdot L_0 \cdot l_1 \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin \beta
\end{aligned} \tag{9.32}$$

Po enačbi (9.4) dobimo:

$$\begin{aligned}
&-m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\alpha} \cdot \cos \beta + (J_1 + m_1 \cdot l_1^2) \cdot \ddot{\beta} - \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha}^2 \cdot \sin 2\beta \\
&-m_1 \cdot g \cdot l_1 \cdot \sin \beta = M_n
\end{aligned} \tag{9.33}$$

Pri čemer je:

M_n – skupen navor na nihalo

$$M_n = -C_1 \cdot \dot{\beta} \tag{9.34}$$

Kjer je:

C_1 – koeficient viskoznega trenja v ležaju nihala

V enačbo (9.33) vstavimo (9.34) in dobimo:

$$\begin{aligned}
&-m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\alpha} \cdot \cos \beta + (J_1 + m_1 \cdot l_1^2) \cdot \ddot{\beta} - \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha}^2 \cdot \sin 2\beta \\
&+ C_1 \cdot \dot{\beta} - m_1 \cdot g \cdot l_1 \cdot \sin \beta = 0
\end{aligned} \tag{9.35}$$

Upoštevamo še:

$$F = J_1 + m_1 \cdot l_1^2 \quad (9.36)$$

$$H = m_1 \cdot g \cdot l_1 \quad (9.37)$$

Po vstavitvi enačb (9.36), (9.37) in enačb (9.27), (9-28) v enačbo (9.35) dobimo:

$$\begin{aligned} -C \cdot \ddot{\alpha} \cdot \cos \beta + F \cdot \ddot{\beta} - \frac{1}{2} B \cdot \dot{\alpha}^2 \cdot \sin 2\beta \\ + C_1 \cdot \dot{\beta} - H \cdot \sin \beta = 0 \end{aligned} \quad (9.38)$$

Enačbi (9.25) in (9.35) lahko zapišemo v matrični obliki:

$$\begin{aligned} & \begin{bmatrix} J_0 + m_1 \cdot L_0^2 + m_1 \cdot l_1^2 \cdot \sin^2 \beta & -m_1 \cdot L_0 \cdot l_1 \cdot \cos \beta \\ -m_1 \cdot L_0 \cdot l_1 \cdot \cos \beta & J_1 + m_1 \cdot l_1^2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} \\ & \begin{bmatrix} C_0 + \frac{K_t \cdot K_b}{R_a} + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\beta} \cdot \sin 2\beta & m_1 \cdot L_0 \cdot l_1 \cdot \dot{\beta} \cdot \sin \beta + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\beta \\ -\frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\beta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (9.39) \\ & \begin{bmatrix} 0 \\ -m_1 \cdot g \cdot l_1 \cdot \sin \beta \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R_a} \\ 0 \end{bmatrix} V_m \end{aligned}$$

Z uporabo enačb (9.26)-(9-30) in (9.36), (9.37) dobimo:

$$\begin{aligned} & \begin{bmatrix} A + B \cdot \sin^2 \beta & -C \cdot \cos \beta \\ -C \cdot \cos \beta & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} \\ & \begin{bmatrix} D + \frac{1}{2} B \cdot \dot{\beta} \cdot \sin 2\beta & C \cdot \dot{\beta} \cdot \sin \beta + \frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\beta \\ -\frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\beta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (9.40) \\ & \begin{bmatrix} 0 \\ -H \cdot \sin \beta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m \end{aligned}$$

9.2 Matematični model nihala v običajni legi [5]

Gre za enak fizikalni objekt. Razlika se pojavi le pri kotu nihala, ki se spremeni za 90° . Kot β tako nadomesti kot θ .

$$\begin{aligned}
 \beta &= \theta - \pi \\
 \dot{\beta} &= \dot{\theta} \\
 \ddot{\beta} &= \ddot{\theta}
 \end{aligned}
 \tag{9.41}$$

$$\begin{aligned}
 \cos \beta &= \cos(\theta - \pi) & \sin \beta &= \sin(\theta - \pi) & \sin 2\beta &= \sin 2(\theta - \pi) \\
 &= -\cos \theta & &= -\sin \theta & &= \sin(2\theta - 2\pi) \\
 & & & & &= \sin 2\theta
 \end{aligned}$$

Tako dobimo:

$$\begin{aligned}
 &(J_0 + m_1 \cdot L_0^2) \cdot \ddot{\alpha} + m_1 \cdot l_1^2 \cdot \ddot{\alpha} \cdot \sin^2 \theta + m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\theta} \cdot \cos \theta \\
 &+ (C_0 + \frac{K_t \cdot K_b}{R_a}) \cdot \dot{\alpha} + m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \dot{\theta} \cdot \sin 2\theta - m_1 \cdot L_0 \cdot l_1 \cdot \dot{\theta}^2 \cdot \sin \theta \\
 &- \frac{K_t}{R_a} \cdot V_m = 0
 \end{aligned}
 \tag{9.42}$$

in

$$\begin{aligned}
 &m_1 \cdot L_0 \cdot l_1 \cdot \ddot{\alpha} \cdot \cos \theta + (J_1 + m_1 \cdot l_1^2) \cdot \ddot{\theta} - \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha}^2 \cdot \sin 2\theta \\
 &+ C_1 \cdot \dot{\theta} + m_1 \cdot g \cdot l_1 \cdot \sin \theta = 0
 \end{aligned}
 \tag{9.43}$$

Ali v matrični obliki:

$$\begin{aligned}
& \begin{bmatrix} J_0 + m_1 \cdot L_0^2 + m_1 \cdot l_1^2 \cdot \sin^2 \theta & m_1 \cdot L_0 \cdot l_1 \cdot \cos \theta \\ m_1 \cdot L_0 \cdot l_1 \cdot \cos \theta & J_1 + m_1 \cdot l_1^2 \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} \\
& \begin{bmatrix} C_0 + \frac{K_t \cdot K_b}{R_a} + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\theta} \cdot \sin 2\theta & -m_1 \cdot L_0 \cdot l_1 \cdot \dot{\theta} \cdot \sin \theta + \frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\theta \\ -\frac{1}{2} m_1 \cdot l_1^2 \cdot \dot{\alpha} \cdot \sin 2\theta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \end{bmatrix} \quad (9.44) \\
& \begin{bmatrix} 0 \\ m_1 \cdot g \cdot l_1 \cdot \sin \theta \end{bmatrix} = \begin{bmatrix} \frac{K_t}{R_a} \\ 0 \end{bmatrix} V_m
\end{aligned}$$

Z uporabo enačb (9.26)-(9.30) in (9.36), (9.37) dobimo:

$$\begin{aligned}
& \begin{bmatrix} A + B \cdot \sin^2 \theta & C \cdot \cos \theta \\ C \cdot \cos \theta & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} \\
& \begin{bmatrix} D + \frac{1}{2} B \cdot \dot{\theta} \cdot \sin 2\theta & -C \cdot \dot{\theta} \cdot \sin \theta + \frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\theta \\ -\frac{1}{2} B \cdot \dot{\alpha} \cdot \sin 2\theta & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\theta} \end{bmatrix} \quad (9.45) \\
& \begin{bmatrix} 0 \\ H \cdot \sin \theta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m
\end{aligned}$$

9.3 Linearizacija [5]

Linearizacija temelji na razvoju v Taylorjevo vrsto okrog delovne točke, pri kateri obdržimo le linearne člene, člene višjega reda pa zanemarimo. Za n spremenljivk lahko zapišemo splošno enačbo:

$$y = f(x_1, x_2, \dots, x_n) \quad (9.46)$$

$$\bar{y} = f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$$

$$y - \bar{y} \approx (x_1 - \bar{x}_1) \left. \frac{\partial f}{\partial x_1} \right|_{\substack{x_1 = \bar{x}_1 \\ x_2 = \bar{x}_2 \\ \dots \\ x_n = \bar{x}_n}} + (x_2 - \bar{x}_2) \left. \frac{\partial f}{\partial x_2} \right|_{\substack{x_1 = \bar{x}_1 \\ x_2 = \bar{x}_2 \\ \dots \\ x_n = \bar{x}_n}} + \dots + (x_n - \bar{x}_n) \left. \frac{\partial f}{\partial x_n} \right|_{\substack{x_1 = \bar{x}_1 \\ x_2 = \bar{x}_2 \\ \dots \\ x_n = \bar{x}_n}} \quad (9.47)$$

Linearizacija je smiselna v delovni točki:

$$\begin{aligned} T(\alpha, \beta, \dot{\beta}) &= (0, 0, 0) \\ x_1 &= \dot{\alpha} \quad x_2 = \beta \quad x_3 = \dot{\beta} \\ \bar{x}_1 &= 0 \quad \bar{x}_2 = 0 \quad \bar{x}_3 = 0 \end{aligned} \quad (9.48)$$

Zaradi večje preglednosti bomo linearizacijo izpeljali s pomočjo enačb (9.31) in (9.38).

Linearizacija enačbe (9.31):

$$\begin{aligned} y &= A \cdot \ddot{\alpha} + B \cdot \ddot{\alpha} \cdot \sin^2 \beta - C \cdot \ddot{\beta} \cdot \cos \beta + D \cdot \dot{\alpha} + B \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \sin 2\beta \\ &+ C \cdot \dot{\beta}^2 \cdot \sin \beta - E \cdot V_m \end{aligned} \quad (9.49)$$

$$\bar{y} = f(0, 0, 0) = A \cdot \ddot{\alpha} - C \cdot \ddot{\beta} - E \cdot V_m \quad (9.50)$$

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= \frac{\partial f}{\partial \dot{\alpha}} = D + B \cdot \dot{\beta} \cdot \sin 2\beta \\ \frac{\partial f}{\partial \dot{\alpha}} \Big|_{\substack{\dot{\alpha}=0 \\ \beta=0 \\ \dot{\beta}=0}} &= D \end{aligned} \quad (9.51)$$

$$\begin{aligned} \frac{\partial f}{\partial x_2} &= \frac{\partial f}{\partial \beta} = 2B \cdot \dot{\alpha} \cdot \sin \beta \cos \beta + C \cdot \ddot{\beta} \cdot \sin \beta \\ &+ 2B \cdot \dot{\alpha} \cdot \dot{\beta} \cdot \cos 2\beta + C \cdot \dot{\beta}^2 \cdot \cos \beta \end{aligned} \quad (9.52)$$

$$\frac{\partial f}{\partial \beta} \Big|_{\substack{\dot{\alpha}=0 \\ \beta=0 \\ \dot{\beta}=0}} = 0$$

$$\begin{aligned} \frac{\partial f}{\partial x_3} &= \frac{\partial f}{\partial \dot{\beta}} = B \cdot \dot{\alpha} \cdot \sin \beta + 2C \cdot \dot{\beta} \cdot \sin \beta \\ \frac{\partial f}{\partial \dot{\beta}} \Big|_{\substack{\dot{\alpha}=0 \\ \beta=0 \\ \dot{\beta}=0}} &= 0 \end{aligned} \quad (9.53)$$

$$\begin{aligned}
 y - \bar{y} &\approx (\dot{\alpha} - 0) \left. \frac{\partial f}{\partial \dot{\alpha}} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\beta - 0) \left. \frac{\partial f}{\partial \beta} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\dot{\beta} - 0) \left. \frac{\partial f}{\partial \dot{\beta}} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} \\
 y &\approx \bar{y} + (\dot{\alpha} - 0) \left. \frac{\partial f}{\partial \dot{\alpha}} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\beta - 0) \left. \frac{\partial f}{\partial \beta} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\dot{\beta} - 0) \left. \frac{\partial f}{\partial \dot{\beta}} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}}
 \end{aligned} \tag{9.54}$$

$$y = A \cdot \ddot{\alpha} - C \cdot \ddot{\beta} + D \cdot \dot{\alpha} - E \cdot V_m$$

Tako pridemo do linearizirane oblike enačbe:

$$A \cdot \ddot{\alpha} - C \cdot \ddot{\beta} + D \cdot \dot{\alpha} - E \cdot V_m = 0 \tag{9.55}$$

Linearizacija enačbe (9.38):

$$\begin{aligned}
 y &= -C \cdot \ddot{\alpha} \cdot \cos \beta + F \cdot \ddot{\beta} - \frac{1}{2} B \cdot \dot{\alpha}^2 \cdot \sin 2\beta \\
 &+ C_1 \cdot \dot{\beta} - H \cdot \sin \beta
 \end{aligned}$$

$$\bar{y} = f(0, 0, 0) = -C \cdot \ddot{\alpha} + F \cdot \ddot{\beta} \tag{9.56}$$

$$\begin{aligned}
 \frac{\partial f}{\partial x_1} &= \frac{\partial f}{\partial \dot{\alpha}} = -B \cdot \dot{\alpha} \cdot \sin 2\beta \\
 \left. \frac{\partial f}{\partial \dot{\alpha}} \right|_{\substack{\alpha_1=0 \\ \beta=0 \\ \dot{\beta}=0}} &= 0
 \end{aligned} \tag{9.57}$$

$$\frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial \beta} = -C \cdot \ddot{\alpha} \cdot \sin \beta - B \cdot \dot{\alpha}^2 \cos 2\beta - H \cdot \cos \beta$$

$$\left. \frac{\partial f}{\partial \beta} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} = -H \quad (9.58)$$

$$\frac{\partial f}{\partial x_3} = \frac{\partial f}{\partial \dot{\beta}} = C_1$$

$$\left. \frac{\partial f}{\partial \dot{\beta}} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} = C_1 \quad (9.59)$$

$$y - \bar{y} \approx (\dot{\alpha} - 0) \left. \frac{\partial f}{\partial \dot{\alpha}} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\beta - 0) \left. \frac{\partial f}{\partial \beta} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\dot{\beta} - 0) \left. \frac{\partial f}{\partial \dot{\beta}} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}}$$

$$y \approx \bar{y} + (\dot{\alpha} - 0) \left. \frac{\partial f}{\partial \dot{\alpha}} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\beta - 0) \left. \frac{\partial f}{\partial \beta} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} + (\dot{\beta} - 0) \left. \frac{\partial f}{\partial \dot{\beta}} \right|_{\substack{\dot{\alpha}_1=0 \\ \beta=0 \\ \dot{\beta}=0}} \quad (9.60)$$

$$y = -C \cdot \ddot{\alpha} + F \cdot \ddot{\beta} - H \cdot \beta + C_1 \cdot \dot{\beta}$$

Torej je linearizirana oblika enačbe:

$$-C \cdot \ddot{\alpha} + F \cdot \ddot{\beta} + C_1 \cdot \dot{\beta} - H \cdot \beta = 0 \quad (9.61)$$

Enačbi (9.55) in (9.61) v matrični obliki:

$$\begin{bmatrix} A & -C \\ -C & F \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + \begin{bmatrix} D & 0 \\ 0 & C_1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -H \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix} V_m \quad (9.62)$$

9.4 Meritve ojačanja napetostnega pretvornika

Tabela 7 Meritve ojačanja

| Prevaljalno razmerje [%] | Vrednost števnika | Napetost na izhodu DSP-ja [V] | Napetost na vhodu motorja [V] | Vrednost števnika na volt | Ojačanje napajalnika |
|--------------------------|-------------------|-------------------------------|-------------------------------|---------------------------|----------------------|
| 10,67 | 4000 | 0,32 | 1,5 | 2667 | 4,6875 |
| 13,33 | 5000 | 0,4 | 2 | 2500 | 5 |
| 16,00 | 6000 | 0,48 | 2,4 | 2500 | 5 |
| 18,67 | 7000 | 0,56 | 2,9 | 2414 | 5,1785714 |
| 21,33 | 8000 | 0,64 | 3,4 | 2353 | 5,3125 |
| 24,00 | 9000 | 0,72 | 3,85 | 2338 | 5,3472222 |
| 26,67 | 10000 | 0,8 | 4,3 | 2326 | 5,375 |
| 29,33 | 11000 | 0,88 | 4,7 | 2340 | 5,3409091 |
| 32,00 | 12000 | 0,96 | 5,4 | 2222 | 5,625 |
| 34,67 | 13000 | 1,04 | 5,8 | 2241 | 5,5769231 |
| 37,33 | 14000 | 1,12 | 6,2 | 2258 | 5,5357143 |
| 40,00 | 15000 | 1,2 | 6,6 | 2273 | 5,5 |
| 42,67 | 16000 | 1,28 | 7 | 2286 | 5,46875 |
| 45,33 | 17000 | 1,36 | 7,5 | 2267 | 5,5147059 |
| 48,00 | 18000 | 1,44 | 7,9 | 2278 | 5,4861111 |
| 50,67 | 19000 | 1,52 | 8,3 | 2289 | 5,4605263 |
| 53,33 | 20000 | 1,6 | 8,8 | 2273 | 5,5 |
| 56,00 | 21000 | 1,68 | 9,2 | 2283 | 5,4761905 |
| 58,67 | 22000 | 1,76 | 9,6 | 2292 | 5,4545455 |
| 61,33 | 23000 | 1,84 | 10 | 2300 | 5,4347826 |
| 64,00 | 24000 | 1,92 | 10,4 | 2308 | 5,4166667 |
| 66,67 | 25000 | 2 | 10,8 | 2315 | 5,4 |
| 69,33 | 26000 | 2,08 | 11,2 | 2321 | 5,3846154 |
| 72,00 | 27000 | 2,16 | 11,6 | 2328 | 5,3703704 |
| 74,67 | 28000 | 2,24 | 12 | 2333 | 5,3571429 |
| 77,33 | 29000 | 2,32 | 12,4 | 2339 | 5,3448276 |
| 80,00 | 30000 | 2,4 | 12,8 | 2344 | 5,3333333 |
| 82,67 | 31000 | 2,48 | 13,2 | 2348 | 5,3225806 |
| 85,33 | 32000 | 2,56 | 13,7 | 2336 | 5,3515625 |
| 88,00 | 33000 | 2,64 | 14 | 2357 | 5,3030303 |
| 90,67 | 34000 | 2,72 | 14,5 | 2345 | 5,3308824 |
| 93,33 | 35000 | 2,8 | 14,8 | 2365 | 5,2857143 |
| 96,00 | 36000 | 2,88 | 15,2 | 2368 | 5,2777778 |
| 98,67 | 37000 | 2,96 | 15,6 | 2372 | 5,2702703 |
| Povprečna vednost: | | | | 2338 | 5,353639 |

9.5 Seznam slik

| | |
|--|----|
| Slika 2.1: Inverzno rotacijsko nihalo..... | 4 |
| Slika 2.2: Potenciometer za merjenje položajhja nihala..... | 4 |
| Slika 2.3: Prenos signalov je izveden preko ščetk..... | 4 |
| Slika 2.4: Izhodna signala iz enkoderja [2] | 5 |
| Slika 2.5: Napajalnik..... | 5 |
| Slika 2.6:Razvojna plošča za pogon motorja..... | 5 |
| Slika 2.7: Razvojna plošča z DSP TMS320F28335 | 7 |
| Slika 2.8: JTAG XDS510 | 8 |
| Slika 3.1: Model nihala..... | 9 |
| Slika 3.2: Krivulja lege korenov odprtozančnega sistema | 17 |
| Slika 3.3: Odprtozančni odziv nelinearnega modela..... | 18 |
| Slika 3.4: Odprtozančni odziv linearnega modela..... | 18 |
| Slika 4.1: Optični enkoder [2] | 20 |
| Slika 4.2: Posamezne enote modula [2]..... | 22 |
| Slika 4.3: Shema merjenja položaja in časa [2]..... | 24 |
| Slika 4.4: Pogreški algoritmov merjenja hitrosti motorja | 26 |
| Slika 4.5: QPOSMAX = 4 [2] | 28 |
| Slika 4.6: Shema AD pretvornika [6]..... | 29 |
| Slika 4.7: Prilagoditveno vezje za AD pretvorbo [6] | 30 |
| Slika 4.8: Shema vgrajenega AD pretvornika [7] | 31 |
| Slika 4.9: Shema zgradbe AD pretvornika [8] | 32 |
| Slika 4.10: Potek komunikacije [8] | 34 |
| Slika 4.11: Shema AD pretvornika [8]..... | 35 |
| Slika 4.12: Način komunikacije gospodar ali suženj [9]..... | 37 |

| | |
|---|----|
| Slika 4.13: Prikaz pošiljanja podatkov | 38 |
| Slika 4.14: Shema SCI Modula [10] | 40 |
| Slika 4.15: Osnovna okvirja protokolov [10]..... | 41 |
| Slika 4.16: <i>Idle-line</i> protokol [10]..... | 42 |
| Slika 4.17: <i>Address-bit</i> protokol [10]..... | 42 |
| Slika 4.18: format komunikacije [10] | 43 |
| Slika 4.19: Lastnosti posameznega PWM modula [11] | 45 |
| Slika 4.20: PIE [13]..... | 47 |
| Slika 4.21: Zgradba programa..... | 48 |
| Slika 5.1: Primer modela v prostoru stanj [4] | 51 |
| Slika 5.2: Primer sistema s povrežno povezavo [4] | 52 |
| Slika 5.3: Sledenje referenčnemu vhodu [4] | 54 |
| Slika 6.1: Model IRN v Simulinku..... | 57 |
| Slika 6.2: Odziv na pravokotni vhodni signal | 57 |
| Slika 6.3: Odziv na pravokotni vhodni signal | 58 |
| Slika 6.4: Stopnični odziv izbranega sistema..... | 60 |
| Slika 6.5: Meritev sistema tretjega reda | 60 |
| Slika 6.6: Meritev sistema tretjega reda | 61 |
| Slika 6.7: Meritev sistema tretjega reda | 61 |
| Slika 6.8: Meritev sistema tretjega reda | 61 |
| Slika 6.9: Meritev sistema tretjega reda | 62 |
| Slika 6.10: Stopnični odziv izbranega sistema..... | 63 |
| Slika 6.11: Meritev sistema tretjega reda | 63 |
| Slika 6.12: Meritev sistema tretjega reda | 63 |
| Slika 6.13: Meritev sistema tretjega reda | 63 |

| | |
|--|----|
| Slika 6.14: Meritev sistema tretjega reda | 64 |
| Slika 6.15: Meritev sistema tretjega reda | 64 |
| Slika 6.16: Stopnični odziv izbranega sistema..... | 65 |
| Slika 6.17: Merite sistema četrtega reda..... | 66 |
| Slika 6.18: Merite sistema četrtega reda..... | 66 |
| Slika 6.19: Merite sistema četrtega reda..... | 66 |
| Slika 6.20: Merite sistema četrtega reda..... | 67 |
| Slika 6.21: Merite sistema četrtega reda..... | 67 |
| Slika 6.22: Stopnični odziv izbranega sistema..... | 68 |
| Slika 6.23: Merite sistema četrtega reda..... | 68 |
| Slika 6.24: Merite sistema četrtega reda..... | 69 |
| Slika 6.25: Merite sistema četrtega reda..... | 69 |
| Slika 6.26: Merite sistema četrtega reda..... | 69 |
| Slika 6.27: Merite sistema četrtega reda..... | 70 |
| Slika 6.28: Meritev sistema tretjega reda | 72 |
| Slika 6.29: Meritev sistema tretjega reda | 73 |
| Slika 6.30: Meritev sistema tretjega reda | 73 |
| Slika 6.31: Meritev sistema tretjega reda | 73 |
| Slika 6.32: Meritev sistema tretjega reda | 74 |
| Slika 6.33: Meritev sistema četrtega reda..... | 74 |
| Slika 6.34: Meritev sistema četrtega reda..... | 75 |
| Slika 6.35: Meritev sistema četrtega reda..... | 75 |
| Slika 6.36: Meritev sistema četrtega reda..... | 75 |
| Slika 6.37: Meritev sistema četrtega reda..... | 76 |

9.6 Seznam preglednic

| | |
|---|----|
| Tabela 1: Lastnosti procesorja [3]..... | 8 |
| Tabela 2: Parametri sistema | 16 |
| Tabela 3: Prikaz pogreškov algoritmov | 26 |
| Tabela 4: Resolucija AD [8] | 33 |
| Tabela 5: Šum AD [8]..... | 33 |
| Tabela 6: Periferne prekinitve [13] | 48 |
| Tabela 7 Meritve ojačanja | 92 |

9.7 Naslov študenta

Osebni podatki

| | |
|---------------------|---|
| Priimek / Ime | Semenič Nikolaj |
| Naslov | Rošpoh 74, 2351 Kamnica |
| Telefonska številka | 040 627 543 |
| e-mail | semenic@gmail.com |
| Državljanstvo | Slovensko |
| Datum rojstva | 3. april 1985 |
| Fakulteta | Fakulteta za elektrotehniko, računalništvo in informatiko |

9.8 Kratek življenjepis

Delovne izkušnje

| | |
|------------------|--|
| Obdobje | 2007 - 2009 |
| Delovno mesto | Margento R&D |
| Glavne naloge | Programiranje v ANSI c |
| Vrsta dejavnosti | Telekomunikacije, mikroprocesorski sistemi |

Izobraževanje in usposabljanje

| | |
|---|--|
| Obdobje | 2000-2004 |
| Naziv izobrazbe | Gimnazijski maturant |
| Naziv in status ustanove, ki je podelila diplomo, spričevalo ali certifikat | Prva gimnazija Maribor Trg generala Maistra 1, 2000 Maribor |
| Stopnja izobrazbe po nacionalni klasifikacijski lestvici | V. stopnja |
| Obdobje | 1992-2000 |
| Naziv izobrazbe | Končana osnovna šola |
| Naziv in status ustanove, ki je podelila diplomo, spričevalo ali certifikat | Osnovna šola Kamnica Vrbanska c. 93, 2351, Kamnica |
| Stopnja izobrazbe po nacionalni klasifikacijski lestvici | II. stopnja |



IZJAVA

Podpisani Nikolaj Semenič, rojen 3. aprila 1985, študent Fakultete za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, smer avtomatika – univerzitetni program, izjavljam, da je diplomsko delo z naslovom Stabilizacija inverznega rotacijskega nihala z DSP pri mentorju izr. prof. dr. Rajko Svečko, avtorsko delo.

V diplomskem delu so uporabljeni viri in literatura korektno navedeni; teksti niso prepisani brez navedbe avtorjev.

(podpis študenta)

Maribor, junij 2009