

# Sodobna orodja in postopki za načrtovanje algoritmov vodenja servopogonov

Miran Rodič, Karel Jezernik

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

Smetanova ul. 17, SI-2000 Maribor

miran.rodic@uni-mb.si, karel.jezernik@uni-mb.si

*State-of-the-art tools and procedures for the design of the control algorithms for servodrives*

*Abstract: Advanced tools and procedures for the design and analysis of the electromechanical servodrives' control algorithms are presented in the paper together with the classical ones. It is shown that due to the development in the last years many advanced tools enabling the shorter time-to-market and improved validation have been made available. Some of those methods are presented and compared among each other.*

## 1 Uvod

Možnost izvajanja eksperimentov na sistemih, ki uporabljajo pogone z električnimi motorji je pogosto omejena. Najpomembnejša vzroka za to sta cena sistema in težavna ali celo nemogoča namestitvev senzorjev.

Prvi način reševanja problema je uporaba simulacij, kjer potrebujemo natančne in preverjene modele sistema in njegovega elektromehanskega pogona. S takšnim pristopom lahko razvijemo osnovne algoritme vodenja, ne moremo pa preizkusiti njihovega delovanja na uporabljeni strojni opremi.

Drugi pristop je uporaba eksperimentalnega sistema in statičnega bremena. Na ta način je mogoče preizkusiti delovanje programske opreme, vendar le za statično obremenitev. Z uporabo elektromehanskih zavora je do določene mere mogoče emulirati tudi spremenljiva bremena, vendar le njihovo statično karakteristiko.

Nov pristop k reševanju problema je dinamična emulacija mehanskih bremen. Za razliko od klasičnih simulacij, ki tečejo samo na računalniku, imamo tukaj na razpolago celoten pogonski sistem, ki pa je namesto na pogonjano progo mehansko priključen na sistem za aktivno obremenjevanje. Tako je mogoče namesto simulacijskega modela v simulacijsko zanko vključiti elektromehanski sistem, ki ga nameravamo uporabiti v realni aplikaciji, pogonjani mehanizem pa nadomestiti z aktivnim bremenom, ki ga predstavlja elektromehanski sistem z izvedeno dinamično emulacijo mehanskega bremena. Na ta način je mogoče že v fazi načrtovanja izdelka izvajati določene teste, za katere bi sicer potrebovali prototip. S tem je mogoče bistveno skrajšati čas, ki preteče od zamisli do končnega izdelka, pri tem pa povečati produktivnost razvojne ekipe. Pomembno dejstvo je tudi, da je mogoče isti sistem uporabiti za razvoj večjega števila izdelkov, kar je še posebno pomembno, ko gre za unikatne ali maloserijske izdelke.

Pri načrtovanju algoritmov vodenja igrajo pomembno vlogo tudi postopki programiranja. Klasični pristop je pisanje programske kode v zbirnem jeziku ali v kakšnem od množice višjih programskih jezikov, danes najpogosteje v programskem jeziku C.

Zbirni jezik omogoča najboljšo kontrolo nad programom, vendar je izredno nepregleden, kar povzroča velike težave pri iskanju napak. Boljšo rešitev predstavlja programski jezik C. V preteklosti je uporaba višjih programskih jezikov pomenila počasnejše izvajanje programa, z vpeljavo optimizatorjev pa tudi to ne drži več, saj le-ti pogosto v optimizaciji

prekašajo manj izkušene programerje.

Prihodnost na tem področju predstavlja uporaba grafičnih programskih orodij. Njihova največja prednost je velika preglednost, saj z nekaj bloki ponazorimo tudi precej kompleksne postopke. S tem se bistveno zmanjša možnost napak. Tudi čas pisanja programske opreme je bistveno krajši. Takšen način programiranja omogočajo danes tudi splošno razširjena programska orodja, kakršen je MATLAB/Simulink.

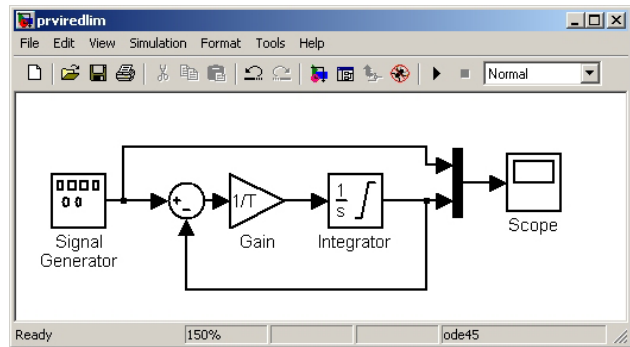
## 2 MATLAB/Simulink

MATLAB je programsko orodje, ki se uporablja za matematično računanje, ki ga povezuje z vizualizacijo in močnim programskim jezikom. Sam proizvajalec ga predstavlja kot programski jezik za tehnično računanje. Ena njegovih najpomembnejših lastnosti je odprta struktura, ki uporabniku omogoča preprosto delo in z uporabo obstoječih funkcij razvoj novih, ki jih sestavi v skladu s svojimi potrebami [1].

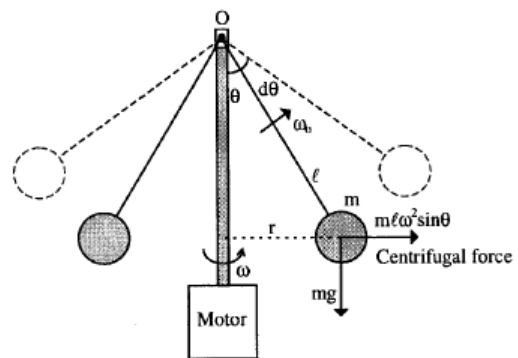
Simulink je razširitev MATLABa, ki ponuja modeliranje, simulacije in analizo dinamičnih sistemov v grafičnem (GUI) okolju. Model sestavimo s pomočjo miške, s katero izberemo ustrezeni blok in ga postavimo na željeno mesto. Na voljo so knjižnice elementov, ki omogočajo linearno in nelinearno analizo sistema. Ker je Simulink del MATLABa, je mogoče med uporabniškima vmesnikoma prosto prehajati, kar omogoča uporabniku izkoriščanje prednosti obeh okolij.

Primer preproste analize kaže Slika 1, kjer lahko izvedemo simulacijo časovnega odziva člena 1. reda z omejitvijo na različne vhodne signale.

Seveda pa analiza preprostih sistemov ne opravičuje uporabe tako kompleksnih in dragih orodij kot je Simulink. Bolj smiselno jih je uporabiti za kompleksnejše sisteme, kakršen je npr. Wattov regulator, ki ga kaže Slika 2.



Slika 1: Člen 1. reda z omejitvijo v Simulinku



Slika 2: Wattov regulator

Sam mehanizem sistema je preprost, vendar pa je njegov matematični model dokaj kompleksen in ga je potrebno zapisati s sistemom treh nelinearnih diferencialnih enačb:

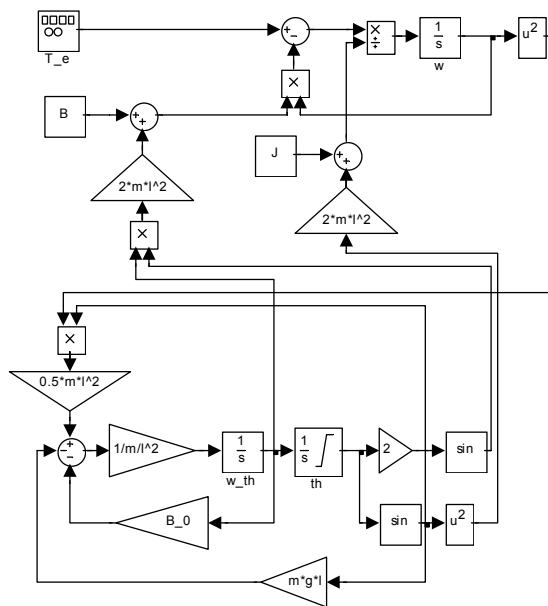
$$\frac{d\omega}{dt} = \frac{-(B + 2ml^2\omega_\theta \sin(2\theta))\omega + T_e}{J + 2ml^2 \sin^2(\theta)} \quad (1)$$

$$\frac{d\omega_\theta}{dt} = -\frac{B_0}{ml^2}\omega_\theta + \frac{1}{2}\omega^2 \sin(2\theta) - \frac{g}{l}\sin(\theta) \quad (2)$$

$$\frac{d\theta}{dt} = \omega_\theta \quad (3)$$

Model v Simulinku kaže Slika 3. Na prvi pogled deluje zapleteno, vendar je bistveno bolj pregleden kot zapis v programskem jeziku C. Poleg tega ni potrebno programirati numeričnih metod reševanja diferencialnih enačb, pa tudi prikaz rezultatov je bistveno enostavnejši.

Na vsak način dosežemo bistveno večjo preglednost, kar zmanjša možnost napak. Sistem lahko združimo v en sam podsistem, kar omogoči njegovo nadaljnjo obravnavo kot en sam blok.



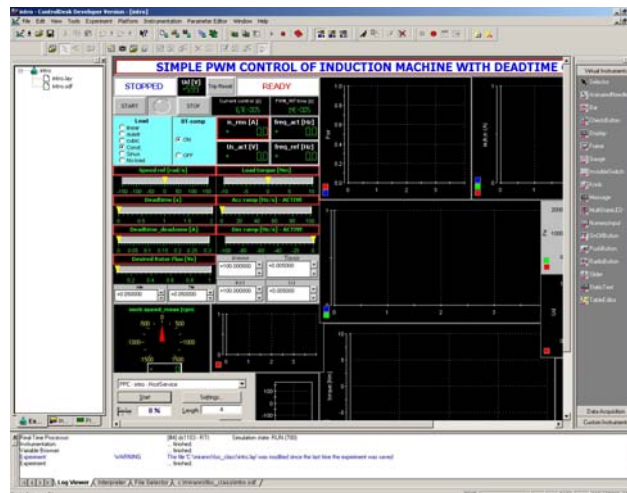
Slika 3: Model Wattovega regulatorja v Simulinku

Z uporabo standardnih orodij, kakršno je MATLAB/Simulink, je mogoče v delovno okolje vključiti tudi modele drugih raziskovalnih skupin. Primer za to bi lahko bil razvoj komponent za avtomobil, kjer bi proizvajalec avtomobila sestavil matematično-fizikalni model le-tega in vplivov okolja ter ga dal na razpolago proizvajalcem posameznih komponent, ki bi s svoje strani prav tako prispevali modele svojih izdelkov.

## 2.1 dSPACE

V zadnjih letih se pojavljajo postopki kot so "Hardware-in-the-loop" in "Rapid prototyping", na tržišču so se pojavila tudi razvojna okolja, ki omogočajo relativno enostavno gradnjo razvojnih in testnih aplikacij, pri čemer računalnik nadomesti večje število instrumentov. Eno od takih orodij je dSPACE [2], ki deluje v povezavi s programskim orodjem MATLAB/Simulink. Ponuja preprosto, popolnoma integrirano razvojno orodje, ki ga je mogoče uporabiti v vseh fazah projekta.

Njegov uporabniški vmesnik za primer aplikacije vodenja asinhronskega motorja kaže Slika 4.



Slika 4: Uporabniški vmesnik dSPACE

Algoritem vodenja razvijemo v Simulinku in ga prevedemo v kodo, ki jo lahko naložimo na ciljni sistem. Poleg opazovanja več veličin hkrati nam je omogočeno tudi sprotno spreminjanje parametrov sistema.

Ena od najpomembnejših dobrih lastnosti orodja dSPACE je možnost uporabe na širokem področju aplikacij. V avtomobilski industriji (Volvo, DaimlerCrysler, Audi, Opel) ga uporabljajo za razvoj pogonov, vzmetenja, zavornih sistemov in elektromehanskih komponent, v letalski industriji je primer uporabe vzdrževanje tlaka v kabini, Siemens ga uporablja za testiranje regulatorjev plinskih turbin, pogosto pa se razna podjetja zanj odločajo tudi pri razvoju raznih mehatronskih izdelkov.

## 3 Simulacijski pristop

V prvi fazi načrtovanja algoritmov se pogosto poslužujemo simulacij, s katerimi izvajamo analizo algoritmov vodenja še preden imamo na razpolago sam sistem. Sodobna simulacijska orodja nam omogočajo tudi načrtovanje zelo kompleksnih izdelkov, za kar pa potrebujemo ustrezno kompleksne modele. Pogosto nam le-ti niso na razpolago, zato uporabljamo njihove bolj ali manj natančne približke.

Klasične simulacije tečejo samo na osebnih računalnikih, vedno pogosteje pa se uporabljajo tudi metode, ki namesto modela uporabljajo kar

dejanski sistem. Ker so takšne simulacije bistveno zanesljivejše, je mogoče pri načrtovanju upoštevati manjše rezerve in s tem povečati izkoristek.

Takšen pristop je simulacijski postopek "Hardware-in-the-loop", kjer je v simulacijsko zanko vključena celotna oprema, ki je namenjena za uporabo v končnem izdelku. Testno mesto sestavljajo tri komponente:

- preizkušani sistem,
- računalnik, na katerem se v realnem času računajo simulacijski modeli in
- delovna postaja (PC), ki služi kot uporabniški vmesnik.

V primeru testnega mesta za elektromehanske pogone predstavlja preizkušani sistem električni motor s pretvornikom in senzorji.

#### 4 Statično obremenjevanje z uporabo elektromehanskega sistema

Statična emulacija bremen predstavlja klasičen postopek meritev električnih motorjev. Standardno orodje zanjo je dinamometer, ki se tradicionalno uporablja za meritev karakteristik (parametrov, izkoristka, ...) motorjev [1],[4],[5]. V teh aplikacijah preizkušamo motor pod stacionarno obremenitvijo ali pri počasi spremenljivih pogojih. Takšni postopki so zelo primerni za preizkušanje pogonov, ki delujejo s konstantno obremenitvijo, manj pa nam povedo o vplivih spreminjanja bremena in parametrov. Poleg tega so tudi manj primerni za preizkušanje reguliranih pogonov.

Novejši postopki, ki se ukvarjajo z dinamično emulacijo bremen z višjo dinamiko [5], [6], [7], [8], izvajajo emulacijo simuliranega bremena v pogojih odprte zanke, kar pomeni, da emulirano breme ni del zaprte hitrostne ali pozicijske zanke. V teh postopkih je uporabljen po navoru voden motor, kar omogoča spremembe obremenitve, v nekaterih primerih tudi z zelo visoko dinamiko.

Osnovni princip postopka je podan z

naslednjo enačbo, ki predstavlja znano enačbo mehanskega gibanja:

$$J \frac{d\omega}{dt} = T_e - T_L \quad (4)$$

$J$  je vztrajnostni moment mehanizma,  $\omega$  predstavlja hitrost vrtenja osi,  $T_e$  je navor, ki ga proizvaja pogonski motor,  $T_L$  pa navor, ki ga proizvaja aktivno breme.

Navor  $T_L$  proizvodimo z zavoro. Simuliramo lahko obnašanje pod pogoji linearnega trenja:

$$T_L = B_v \omega, \quad (5)$$

suhega trenja:

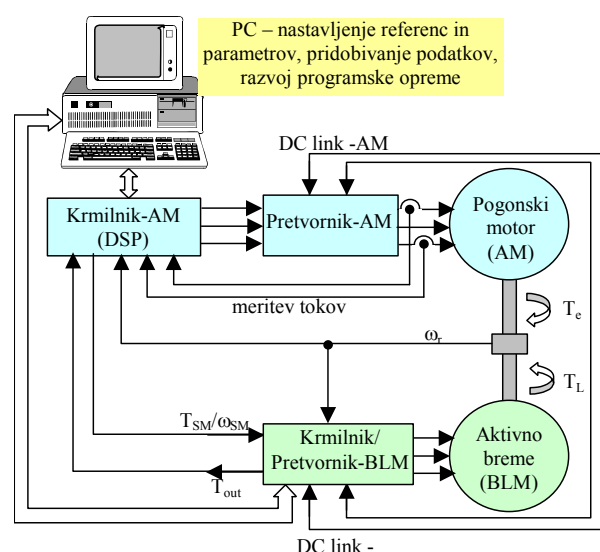
$$T_L = B_{st} \text{sign}(\omega), \quad (6)$$

zračnega upora:

$$T_L = B_{zu} \omega^2 \quad (7)$$

in podobnih stacionarnih vplivov.

Primer sistema za dinamično obremenjevanje kaže Slika 5. Sestavljen je iz dveh mehansko sklopljenih motorjev, ki imata v idealnem primeru skupno os (v praksi je uporabljena sklopka), pri čemer je eden od njiju pogonski motor, drugi pa služi za aktivno obremenjevanje (Slika 6).



Slika 5: Primer eksperimentalnega sistema

Oba sta vodena s pomočjo istega sistema

osebnega računalnika s programskim orodjem dSPACE, kar poleg sočasnega nastavljanja mehanskih in električnih vrednosti omogoča tudi njihovo sočasno opazovanje. Na ta način je omogočeno tudi ustrezno ovrednotenje pretvornika, ki ga je sicer v klasičnih simulacijah težko zajeti. Algoritmi so izvedeni z grafičnim opisnim jezikom MATLAB/Simulink. V celotnem procesu (načrtovanje, simulacija in implementacija) bo uporabljena ista strojna in programska oprema.



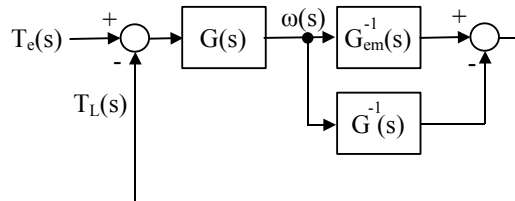
Slika 6: Motorja eksperimentalnega sistema

## 5 Dinamična emulacija mehanskih bremen

Klasična (statična) emulacija mehanskih bremen ne zadošča več potrebam razvoja, vrednotenja in testiranja za preiskovanje postopkov vodenja servomehanizmov s spremenljivim navorom in hitrostjo. Za vrednotenje algoritmov vodenja se je pokazala za potrebno uporaba dinamične emulacije bremen v zaprti zanki [9], [10], [11], [12]. V teh metodah je podan postopek, ki omogoča direktno vključevanje modela mehanizma v algoritem vodenja in hkrati zagotavlja zaprtoznančno obnašanje, kakršno naj bi imel realni mehanizem.

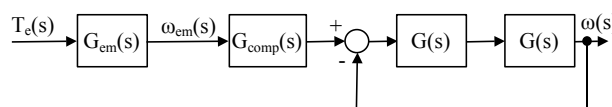
Osnovni princip postopka [9] je nekoliko bolj zapleten kot v primeru stacionarne emulacije. Kaže ga Slika 7.  $G(s)$  predstavlja prenosno funkcijo dejanskega sistema,  $G_{em}(s)$  pa prenosno funkcijo emuliranega sistema.

Postopka ni mogoče uporabiti direktno, saj bi bilo zelo nepraktično računati inverzne modele kompleksnih nelinearnih sistemov. Bolj smiselno je vpeljati izpeljane metode.



Slika 7: Princip dinamične emulacije mehanskih bremen

Ena od metod, ki jih je možno uporabiti, je emulacija bremen z uporabo sledenja hitrosti modela, ki jo kaže Slika 8.



Slika 8: Princip dinamične emulacije mehanskih bremen z uporabo sledenja hitrosti

$G_i(s)$  predstavlja prenosno funkcijo regulatorja navora,  $G_{comp}(s)$  pa prenosno funkcijo kompenzatorja, ki odpravlja vpliv dejanskega sistema.

Opisani postopki dinamične emulacije bremen omogočajo dobro emulacijo bremen, vendar le pod pogojem, da dovolj dobro poznamo navor, ki ga proizvaja pogonski motor. To v mnogih primerih ni mogoče [13], pogosto pa je ta veličina že sama predmet meritev.

Namen dinamične emulacije mehanskih bremen je emulacija mehanskih karakteristik aplikacije, ne da bi bilo potrebno imeti to aplikacijo dejansko na razpolago. Na ta način je mogoče na istem razvojnem orodju emulirati več sistemov z različnimi karakteristikami (linearna in nelinearna bremenja, časovno variabilni sistemi, ...).

Dinamična emulacija mehanskih bremen je nova metoda načrtovanja izdelkov, ki se uporablja za skrajšanje časa, ki preteče med zamisljivo in izvedbo končnega izdelka. Za razliko od klasičnih simulacij, ki tečejo samo na računalniku, imamo tukaj na razpolago celoten pogonski sistem, ki pa je namesto na poganjano progo mehansko priključen na sistem za aktivno obremenjevanje. Tako je mogoče nemesto simulacijskega modela v simulacijsko zanko vključiti elektromehanski sistem, ki ga nameravamo uporabiti v realni aplikaciji,



poganjani mehanizem pa nadomestiti z aktivnim bremenom, ki ga predstavlja elektromehanski sistem z izvedeno dinamično emulacijo mehanskega bremena. Na ta način je mogoče že v fazi načrtovanja izdelka izvajati določene teste, za katere bi sicer potrebovali prototip.

## 6 Zaključek

Klasični simulacijski postopki danes ne zadoščajo več za uspešno načrtovanje algoritmov vodenja servopogonov, saj je z njihovo uporabo nemogoče kvalitetno preiskusiti uporabljeni pogonski sistem.

Uporaba simulacijskih postopkov "Hardware-in-the-loop" omogoča prihranek časa, opreme in denarja, saj precej skrajša čase, ki pretečejo med zamisljivo in izdelkom. Poleg tega ni več potrebe po gradnji posebnega testnega sistema. Pri iskanju napak v algoritmu je mogoče zagotoviti zadovoljivo ponovljivost postopka, še posebno v primeru izvajanja ponavljajočih se kritičnih testov, kar je mogoče izvesti samo z ustrežno opremo na primerno avtomatiziranem delovnem mestu.

Tudi klasična (statična) emulacija mehanskih bremen ne zadošča več potrebam razvoja, vrednotenja in testiranja. V prihodnosti jo bo nadomestila dinamična emulacija mehanskih bremen, ki bo omogočila tudi analizo dinamičnih vplivov mehanizma na pogon, pri tem pa sam mehanizem ne bo potreben. Tako se bodo zmanjšale potrebe po gradnji prototipov, kar bo pospešilo in pocenilo razvoj kompleksnih izdelkov.

Strojna oprema, potrebna za opisane postopke, je že na voljo, potrebno pa je še veliko dela za povečanje njene uporabnosti, kar bo mogoče doseči z razvojem novih metod in tehnik.

## 7 Reference

- [1] <http://www.mathworks.com/>, Spletna stran podjetja Mathworks.
- [2] <http://www.dspaceinc.com/>, Spletna stran podjetja dSPACE.
- [3] C.R. Wasko, *500HP, 120 Hz Current-fed Field Oriented Control Inverter for Fuel Pump Test*

- Stands*, v Conf. Rec. 1986 21st Annual Meeting IEEE Ind. App. Soc., str. 314-320.
- [4] A.C. Wiliamson, *An Improved Engine-Testing Dynamometer*, v Fourth International Conference on Electrical Machines and Drives, IEE Conference Publication 310, 1989, str. 374-378.
  - [5] R.W. Newton, R.E. Betz, H.B. Penfold, *Emulating Dynamic Load Characteristic Using a Dynamic Dyamometer*, v Proceedings of the IEEE Power Electronics and Drive Systems (PEDS), Singapore, 1995, vol. 1, str. 465-470.
  - [6] E.R. Collins, Y. Huang, *A programmable dynamometer for testing rotating machinery using three-phase induction machine*, IEEE Trans. on Energy Conversion, vol. 9, 1994, str. 521-527.
  - [7] R.E. Betz, H.B. Penfold, R.W. Newton, *Local vector control of an AC drive system load simulator*, v Proc. IEEE Conf. Control Applications, 1994, vol. 1, str. 721-726
  - [8] P. Sandholdt, E. Ritchie, J.K. Pedersen, R.E. Betz, *A Dynamometer Performing Dynamical Emulation of Loads with Non-linear Friction*, v Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'96), Varšava, Poljska, 1996, Part 2, str. 873-878.
  - [9] Z.H. Akpolat, G.M. Asher, J.C. Clare, *Dynamic Emulation of Mechanical Loads Using a Vector Controlled Induction Motor-Generator Set*, IEEE Trans. on Industrial Electronics, vol. 46, no. 2, 1999, str. 370-379.
  - [10] Z.H. Akpolat, G.M. Asher, J.C. Clare, *Experimental dynamometer emulation of non-linear mechanical loads*, IEEE Trans. on Industry Applications, vol. 35, no. 6, 1999, str. 1367-1273.
  - [11] Z.H. Akpolat, G.M. Asher, J.C. Clare, *A practical approach to the design of robust speed controllers for machine drives*, IEEE trans. on Industrial Electronics, vol. 47, no. 2, 2000, str. 315-324.
  - [12] H. Baibanou, P. Sicard, A. Ba-Razzouk, *Solutions to Typical Motor Load Emulation Control Problems*, [http://www.opal-rt.com/downloads/pdfs/138H\\_Baibanou.pdf](http://www.opal-rt.com/downloads/pdfs/138H_Baibanou.pdf)
  - [13] M. Rodič, K. Jezernik, *Speed-Sensorless Sliding Mode Torque Control of an Induction Motor*, IEEE Transactions on Industrial Electronics, vol 49, no. 1., 2001, str. 87-95.

## Zahvala

Zahvaljujemo se podjetjema Danfoss in Control Techniques ter Univerzi v Aalborgu, Danska, za donacije, ki so nam omogočile raziskave na področju aktivnega obremenjevanja servopogonov in dinamične emulacije mehanskih bremen.