

Tvorba virtuálneho modelu polohového servosystému pomocou Matlabu a EJS

Triaška Stanislav · Elektrotechnika

29.12.2010



Článok je zameraný na tvorbu virtuálneho modelu polohového servosystému typu HSM 150. Navrhnuté je Master-Slave riadenie s predkorekciou a PID regulátorom polohy. Vytvorený je model Luenbergerovho pozorovateľa uhlovej rýchlosti a aplikovaný v spätnej väzbe do systému. Vytvorená je vizualizačná animácia pomocou EJS, správnosť modelu je overená na reálnom systéme.

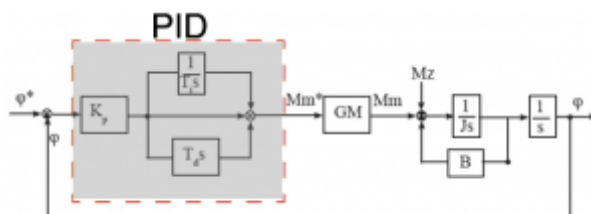
1. Úvod

Prostredníctvom animovaného pohybu a vizualizácie dokážeme priblížiť a interpretovať chovanie sa reálnych modelov a dynamických systémov. Ovládanie a skúmanie takýchto systémov prostredníctvom internetu je čoraz aktívnejšie, to nám umožňujú prostriedky multimediálnej techniky ktoré sa v tomto smere zdokonaľujú. Hlavným účelom takéhoto spôsobu prezentovania je vzdelávanie, vývoj a výskum.

Naším cieľom tohto článku je zamerať na tvorbu virtuálneho modelu servopohonu jednosmerného motora typu HSM 150, k nemu navrhnuť spôsob riadenia polohy s PID regulátorom a navrhnuť pozorovateľa uhlovej rýchlosti. Na simulačný model servosystému ktorý vytvoríme v programe Matlab/Simulink vytvoríme vizualizáciu a ovládače. Program pre riadenie systému v programe Easy Java Simulation a sprístupníme ho prostredníctvom internetu. Správnosť navrhnutého modelu budeme overovať na základe merania na reálnom modeli.

2. Polohové riadenie HSM 150

Tvorbu samotného modelu servosystému HSM 150 budeme realizovať v programe Matlab/Simulink a to tak, že navrhne polohový PID regulátor. Parametre regulátora navrhne z uzatvoreného regulačného obvodu polohy metódou pole-placement. Bloková schéma je znázornená na Obr. č.:1. Vstupom do systému predstavuje želaná poloha hriadeľa motora, výstupnú polohu snímame IRC snímačom polohy s rozlišovacou schopnosťou 10000 impulzov/otáčku. Spätná väzba polohy do systému nám vstupuje z IRC člena.



Obr.č.: 1 Bloková schéma systému

Prenos uzavretého regulačného obvodu je daný:

$$G(s) = \frac{M(s)}{N(s)} = \frac{\frac{K_p}{T_i J} + \frac{K_p}{J} s + \frac{K_p T_d}{J} s^2}{\frac{K_p}{T_i J} + \frac{K_p}{J} s + \frac{K_p T_d + B}{J} s^2 + s^3} \quad (1)$$

Pre návrh parametrov regulátora aplikujeme metódu rozmiestňovania pólov. Metóda spočíva v tom že na základe želankej dynamiky systému, prenosu si volím póly charakteristického polynómu. Porovnaním charakteristickej rovnice uzatvoreného regulačného obvodu skutočného prenosu (1) s prenosom želaného polynómu (2) pri rovnakých mocninách dostaneme parametre regulátora (4).

Prenos želankej dynamiky systému:

$$Gz(s) = \frac{M_0(s)}{N_0(s)} = \frac{k\omega_0^3}{(s^2 + 2\xi\omega_0 s + \omega_0^2)(s + k\omega_0)} \quad (2)$$

$$N_0(s) = (s^2 + 2\xi\omega_0 s + \omega_0^2)(s + k\omega_0)$$

$$N(s) = \frac{K_p}{T_i J} + \frac{K_p}{J} s + \frac{K_p T_d + B}{J} s^2 + s^3$$

Porovnaním menovateľa želaného prenosu a skutočného prenosu dostávame parametre regulátora

$$N(s) = N_0(s) \quad (3)$$

Parametre regulátora:

$$K_p = J\omega_0^2(2\xi k + 1)$$

$$T_i = \frac{2\xi k + 1}{k\omega_0} \quad (4)$$

$$T_d = \frac{J\omega_0(2\xi + k) - B}{J\omega_0(2\xi k + 1)}$$

Viskózna zložka momentu záťaže B má vplyv len na koeficient Td. Pre výpočet PID regulátora sú potrebné tri riadiace parametre:

- Pásmo priepustnosti $\omega_0 = 2\pi f_0$
- Koeficient tlmenia $0 \leq \xi \leq 1$
- k posun pólu po reálnej osi

Pásmo priepustnosti vplýva na dynamiku sústavy a dobu regulácie. Čím väčšie pásmo priepustnosti volíme, tým má pri konštantnom tlmení systém vyššiu dynamiku.

Koeficient tlmenia vplýva hlavne na preregulovanie, a v menšej miere na dynamiku systému. Čím je menší koeficient tlmenia, tým je pri konštantnom pásme priepustnosti väčšie preregulovanie a mierne rýchlejšia doba nábehu[2]. Pri syntéze návrhu PID regulátora polohy sme prenos generátora momentu skrze jeho rýchlejšiu dynamiku zanedbali, ako prenos generátora momentu sme brali $GM = 1$.

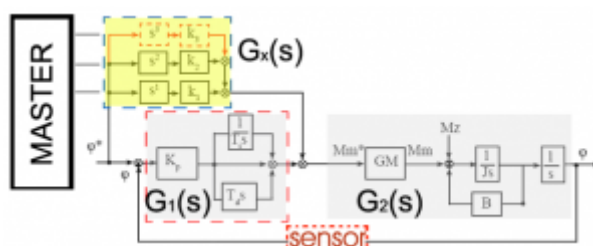
2.1 Master-Slave a dopredné riadenie

Master-slave riadenie môžeme zaradiť medzi stavové riadenie alebo riadenie pomocou modelu. Master-Slave štruktúra s dobre nastaveným generátorom stavových veličín a dobre zvolenou a naladenou štruktúrou riadenia prináša vysokú kvalitu riadenia pohybu aj pri zmene niektorých parametrov sústavy (napr. momentu zotrvačnosti) a obmedzuje vplyv razových momentov na mechanickú časť strojov aj pri použití silne nelineárnych prenosových mechanizmov[1], [2].

Master:

- Generátor riadiacej stavovej veličiny
- Vektor riadenia môže mať väčší počet zložiek ako je počet merateľných veličín
- Generátor vektora riadenia sa realizuje na princípe spätnoväzbového algoritmu
- Pre realizáciu treba poznať parametre „hrubého“ modelu riadenej sústavy[1]

Úlohou Master generátora je generovať priebehy želaných stavových veličín.



Obr.č.: 2 Master - Slave riadenie HSM 150

Dopredné riadenie sa uvádza z hľadiska kvality riadenia. Má vplyv predovšetkým na regulačnú odchýlku. Pri dobre navrhnutom doprednom riadení by sa odchýlka mala znížiť. Podstata dopredného riadenia spočíva v tom že prenos regulačnej odchýlky by mal byť nulový (5). Bloková schéma s dopredným riadením je znázornená na obr. č. 2.

$$G_e(s) = \frac{e(s)}{w(s)} = \frac{1 - G_x(s)G_2(s)}{1 + G_1(s)G_2(s)} = \frac{M(s)}{N(s)} = 0 \quad (5)$$

Riadenie je realizované štvorrozmerným master generátorom, stavové veličiny sú poloha, rýchlosť, zrýchlenie a trh. Pri návrhu parametrov predkorekcie realizáciou s 4D master generátorom, dynamiku generátora momentu nahradíme prenosom prvého rádu:

$$G_M(s) = \frac{1}{T_m(s)+1}$$

kde T_m predstavuje časovú konštantu veľkosti 1ms. Uvažovaním že prenos regulačnej odchýlky je nulový (6) dostávame:

$$\begin{aligned}
 G_x(s) &= \frac{1}{G_2(s)} \\
 G_2(s) &= \frac{1}{T_m s + 1} \cdot \frac{1}{J s + B} \cdot \frac{1}{s} \\
 G_x(s) &= k_1 s + k_2 s^2 + k_3 s^3
 \end{aligned} \tag{6}$$

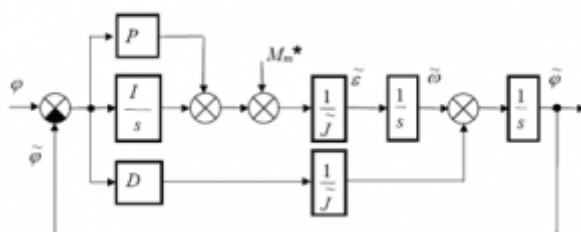
Po roznásobení, recipročná hodnota prenosu $G_2(s)$ a porovnaní pri rovnakých mocninách dostávame koeficienty predkorekcie (7):

$$\begin{aligned}
 k_1 &= B \\
 k_2 &= J + T_m B \\
 k_3 &= T_m J
 \end{aligned} \tag{7}$$

2.2 Pozorovateľ uhlovej rýchlosti

Priame pozorovatele stavových veličín využívajú dynamický model daného systému. Potrebné stavové veličiny sú vypočítané z rovníc dynamického systému a zo vstupných veličín a následne sú použité v riadiacich algoritmoch. Luenbergerov pozorovateľ patrí do skupiny pozorovateľov so spätnou väzbou[4].

Úlohou pozorovateľa uhlovej rýchlosti a zrýchlenia je zo známeho priebehu polohy (uhla natočenia rotora) numericky vyhodnotiť uhlovú rýchlosť (prvú deriváciu polohy) a uhlové zrýchlenie (druhú deriváciu polohy). Do pozorovateľa vstupuje signál z IRC snímača, ktorý nesie informáciu o uhle natočenia rotora. Ďalším vstupom je M_m^* (je to želaná hodnota momentu motora, ktorá vstupuje do GM (generátora momentu), z neho ako výstup berieme uhlovú rýchlosť resp. zrýchlenie.



Obr.č.: 3 Luenberger observer

$$G_{POZ} = \frac{\varphi(s)}{\varphi_{IRC}(s)} = \frac{\frac{D}{J}s^2 + \frac{P}{J}s + \frac{I}{J}}{s^3 + \frac{D}{J}s^2 + \frac{P}{J}s + \frac{I}{J}} \tag{8}$$

Porovnaním z rovnicou charakteristického polynómu:

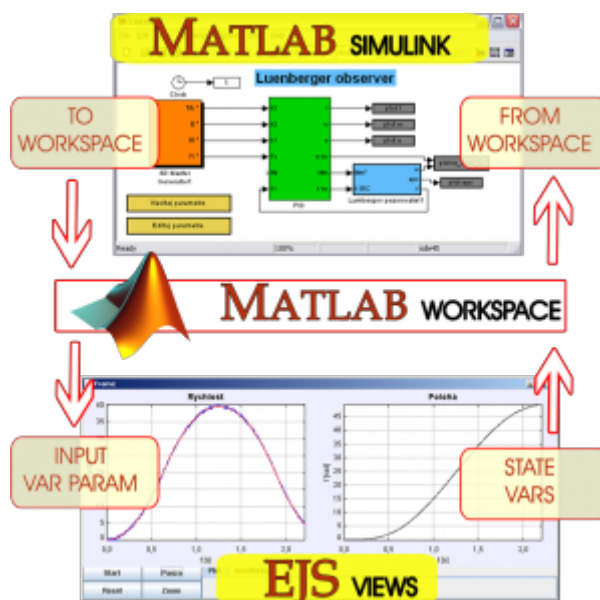
$$N_0(s) = (s^2 + 2\xi\omega_0 s + \omega_0^2)(s + k\omega_0) \tag{9}$$

Dostávame parametre PID štruktúry Luenbergerovho pozorovateľa (10).

$$\begin{aligned}
 D &= J\omega_0(2\xi + 1) \\
 I &= J\omega_0^2(2\xi + 1) \\
 P &= J\omega_0^3
 \end{aligned} \tag{10}$$

3. Virtuálny model pozorovateľa

Model Lenbergeovho pozorovateľa uhlovej rýchlosti sme vytvorili v programe Matlab/Simulink a virtuálny model sme vytvorili v programe Easy Java Simulation (EJS). Je to programový nástroj pre vytváranie interaktívnych simulácií vytvorených v programovom jazyku Java. Výsledná simulácia sa vytvorí vo formáte appletu ktorý môžeme spustiť v ľubovoľnom internetovom prehliadači. Program slúži na jednoduché vytváranie grafických prvkov vizualizácie bez potrebnej väčšej znalosti programovania v Jave. Matematický model v tvare jednoduchých algebrických či diferenciálnych rovníc sa ľahko definuje a vytvára v tomto programe.



Obr.č.: 4 EJS a Matlab/Simulink

Okrem toho, že v EJS môžeme si vytvárať model na základe matematického modelu ktorý si v ňom definujeme v EJS máme možnosť pracovať aj s „external Apps“. External Apps (ďalej iba externály) sú nezávislé zdrojové kódy vytvorené v Matlabe. Práca s externálmi je umožnená vtedy, keď v operačnom systéme je nainštalovaný Matlab. Externály, ktoré importujeme do EJS, môžu byť vo formáte „.m“- súborov alebo „.mdl“-simulinkových schém. Pri načítaní externalov je potrebné nadviazať premenné, ktoré budú v súlade s daným modelom[3].

EJS v nadviazaní s Matlabom znamená, že nám je umožnené :

- Volanie Matlabových funkcií a to:
 - priamo z EJS
 - z externých súborov
- spúšťanie Simulinkových modelov [5]

Definícia premenných resp. konštánt v externáloch je takmer rovnaká ako pri základnom vývojovom okne, rozdiel je iba v tom, že je nutné definovať pripojenie premennej na príslušné miesto do externého súboru (v prípade ak je premenná súčasťou externého súboru).

Riadenie sme realizovali tak, že sme model servosystemu JM HSM 150 s pozorovateľom uhlovej rýchlosti vytvorili v programe Matlab/Simulink-u a ovládanie modelu sme vytvorili v EJS pomocou animácie. Dokážeme monitorovať a sledovať polohu motora, rýchlosť z IRC člena a rýchlosť z pozorovateľa. Ďalšími veličinami,

ktoré sledujeme z modelu sú moment motora, regulačná odchýlka a veličiny generované z mastra. Výpočet parametrov regulátora, predkorekcie a Luenbergerovho pozorovateľa sme realizovali v samotnom EJS. Vypočítané hodnoty importujeme do Matlab workspacu, ktoré budú použité pri vizualizácii.

Vizualizácia funguje takým spôsobom, že všetky symbolické premenné ktoré sa vyskytujú v modeli vytvorenom v Simulinku, musia byť exportované do Matlab workspacu. Z workspacu s týmito premennými, pri spustení animácie model zo simulinku dokáže komunikovať (čítať premenné a zapisovať). Samotný animovaný pohyb je realizovaný po určitom výpočtovom kroku. Pri grafickom znázornení sa vykresľuje rozdiel aktuálneho stavu pozorovanej premennej a predchádzajúceho stavu na základe určitého počtu výpočtových krokov.

Výpočtový krok je definovaný ako jeden úplný cyklus ktorý ubehne od momentu vyexportovania premenných do Matlab workspacu, využitia premenných v Simulinkovom modeli na jednu periódu vzorkovania až do spätne odoslaných dát do EJS views prostredníctvom Matlab workspacu. Jeden výpočtový krok ktorý sme použili je rovný jednej perióde vzorkovania v modeli, na grafickú prezentáciu vo vizualizácii sme pre jeden animačný pohyb využili dva výpočtové kroky. Proces vymieňania dát Matlab-u a EJS je cyklický až do splnenia určitej podmienky ktorú musíme sami definovať, napríklad nejaké ovládacie tlačidlo. Principiálna schéma vzájomnej komunikácie programov je znázornená na obr.č.:4.

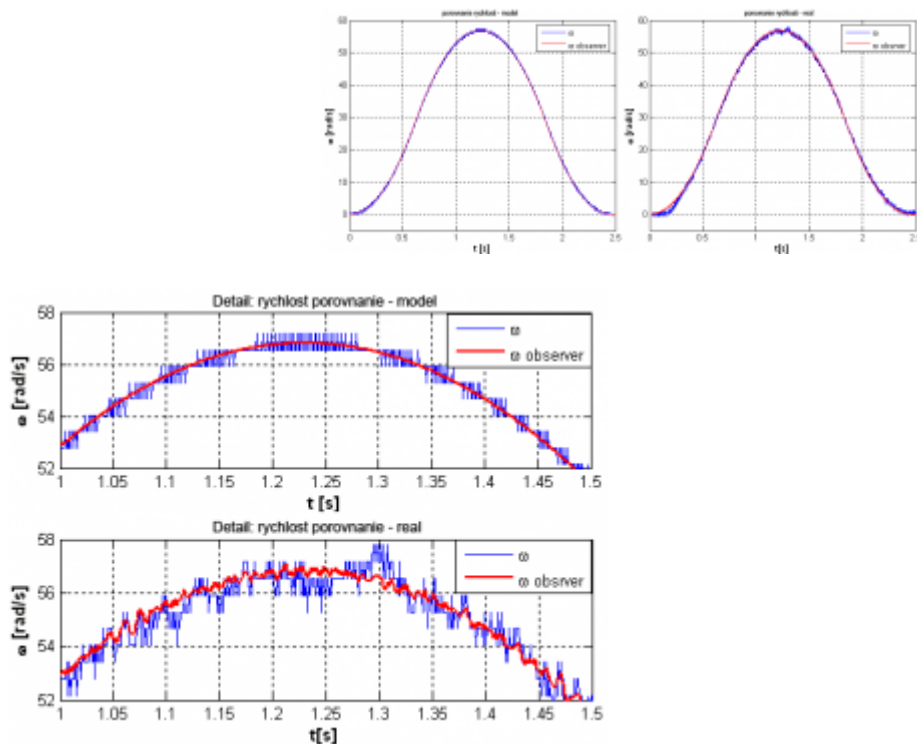
Veľkosť periódy vzorkovania výpočtového kroku a počet výpočtových krokov pre znázornenie animovaného pohybu, majú priamy vplyv na rýchlosť pohybu animácie. Pri zobrazovaní väčšieho počtu výpočtových krokov je animácia náročnejšia a pomalšia, obdobne platí aj naopak. Rýchlosť animácie sa môže nastavovať aj priamo vo vývojovom okne EJS, nastavením počtu FPS (frame per second) animácie. Tieto dva spôsoby sú vzájomne nezávislé a majú priamy vplyv na rýchlosť zobrazovania animovaného pohybu.

Simulačný model je umiestnený na servery, a koncovému užívateľovi je prezentovaná animácia prostredníctvom appletu pomocou ktorého dokážeme ovládať a manipulovať z modelom. Pre takéto riadenie na servery, či pracovnom počítači musí bežať Matlab workspace.

4. Experimentálne meranie

Správnosť navrhnutého modelu riadenia polohového servosystému JM HSM 150 v spätnej väzbe polohy z pozorovateľa, sme overovali na reálnom modeli. Na vstup sme aplikovali štvorrozmerný master generátor želanej veličiny, stavové veličiny generátora sme nastavili tak aby počas experimentu sa nedostali na obmedzenie. Experiment sme robili pri premiestnení polohy z 0 na 70 rad. Rýchlosť sme obmedzili na $\omega_{max} = 100$ [rad/s], zrýchlenie na $\varepsilon_{max} = 90$ [rad/s²] a trh sme obmedzili na hodnotu $d_{max} = 150$ [rad/s³].

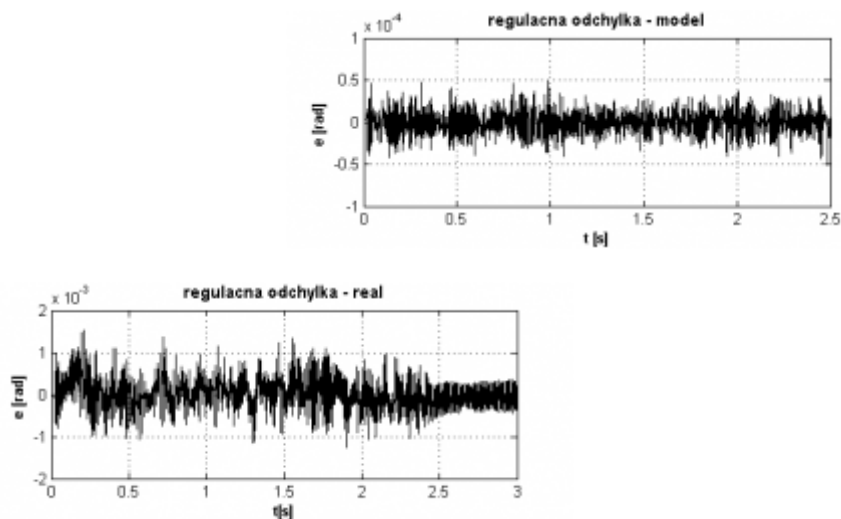
Na základe nameraných výsledkov sme vyhodnocovali simulačný model. Na obr.č.:5 je znázornené porovnanie výsledkov merania uhlovej rýchlosti.



Obr.č.: 5 Porovnanie rýchlosti z modelu a reálu

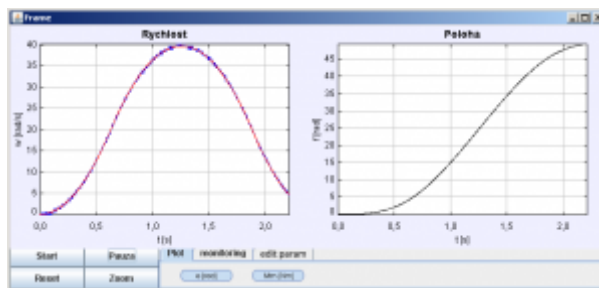
Periódou vzorkovania na oboch meraní bola rovnaká, a vynašala: $T=1$ ms.

Modrou čiarou je prezentovaná rýchlosť nameraná IRC členom, červená čiara predstavuje rýchlosť z pozorovateľa. Kvalitu regulácie vyhodnotíme z priebehu regulačnej odchýlky nameranej na reálnom modeli, ktorá je zanedbateľne malá, takmer nulová obr.č.:6.



Obr.č.: 6 Porovnanie regulačnej odchýlky

Moment motora, výstupná poloha, zidentifikované zrýchlenie, a výstupy z master generátora sú na modeli tiež merateľné veličiny, ktoré aj graficky prezentujeme vo výslednom programe.



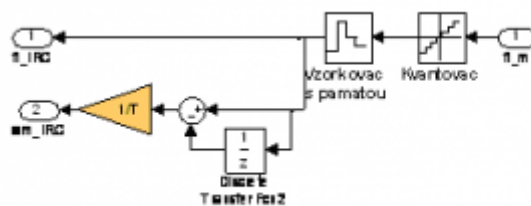
Obr.č.: 7 Program observera

Vo výslednom programe je vyriešené ovládanie modelu a modifikácia parametrov pre nastavenie zmeny obmedzenia master generátora, zmena momentu zotrvačnosti motora atď., na ľubovoľné parametre modelu.

5. Súhrn a záver

Článok je zameraný na tvorbu virtuálneho modelu polohového servosystému HSM 150 z pozorovateľom uhlovej rýchlosti. Na polohový systém sme metódou pole placement navrhli parametre PID regulátora, pri návrhu parametrov regulátora volili sme si vlastnú frekvenciu $f_0 = 20$ [Hz], tlmenie $b=1$, a posun pólu na reálnej osi $k=1$. Pre zníženie dynamickej regulačnej odchýlky sme navrhli dopredné riadenie, na vstup sme aplikovali štvorrozmerný master generátor, ktorého parametre sme nastavili tak aby pri presune želanej polohy nedošlo na obmedzenie. Pre pozorovateľa uhlovej rýchlosti sme podobným spôsobom navrhli parametre regulátora.

Voliteľným parametrom pre regulátor pozorovateľa bola vlastná frekvencia ktorú sme volili rád nižšie ako pri PID regulátore polohy: $f_p = 0.1f_0$, tlmenie a posun pólu na reálnej osi sme volili rovný 1. Na snímanie uhlovej rýchlosti sme použili IRC člen s rozlišovacou schopnosťou 10000 imp/ot., bloková schéma IRC snímača použitého v modeli je na obr.č.:8.



Obr.č.: 8 Realizácia IRC člena v Matlab/Simulink

Vstupom do IRC v modeli je výstupné pootočenie hriadeľa motora, a výstupom z IRC je diskretná rýchlosť a poloha. Model polohového servosystému z pozorovateľov v spätnej väzbe polohy sme odladili v programe Matlab/Simulink, a koncovú vizualizáciu systému v programe EJS. Výsledky sme vyhodnocovali na základe porovnania výsledkov získaných na reálnom modeli.

Referencie

1. ŽALMAN, M. prednášky „Inteligentné servosystémy“, dostupné : <http://servo.urpi.fe.i.stuba.sk>
2. ŽALMAN, M. prednášky „Servosystémy“, dostupné : <http://servo.urpi.fe.i.stuba.sk>
3. TRIAŠKA, S., ŽALMAN, M. : Virtuálne laboratórium servosystémov, Diplomová práca,

máj 2009

4. ŽALMAN M., JOVANKOVIČ J., „Inteligentné servosystémy“, STU BRATISLAVA SLOVENSKÁ E-AKADÉMIA, N.O. 2007
5. ESQUEMBRE, F., SÁNCHEZ-MORENO, J.: Using EJS to run Simulink models in an interactive way, <http://fem.um.es/Ejs>, January 2005
6. BISTÁK, P.: Matlab and Java Based Virtual and Remote Laboratories for Control Engineering, 17th Mediterranean Conference on Control & Automation, Makedonia Palace, Thessaloniki, Greece, June 24 - 26, 2009
7. Dormido, S., Vargas, H., Sanchez, J., Duro, N., Dormido, R.; Dormido-Canto, S., Esquembre, F. (2007) Using webbased laboratories for control engineering education, Proceeding of International Conference on Engineering Education, Coimbra, Portugal.

Spoluautorom článku je M.Gažo, Ústav Riadenia a Priemyselnej Informatiky, Fakulta Elektrotechniky a Informatiky, Ilkovičova 3, 812 19 Bratislava, Slovenská Technická Univerzita v Bratislave
