

Obr.4 Výsledný redukovaný systém

kde

$$D_1 = e^{-\frac{B_{em}T}{J_{em}}}$$

Kompenzačný člen G_{comp} eliminoval všetky časti otvorenej slučky okrem emulovanej záťaže tak, že výsledný redukovaný systém zodpovedá systému na obr. 4, kde $G_c(z)$ je regulátor rýchlosti, ktorého vlastnosti chceme skúmať.

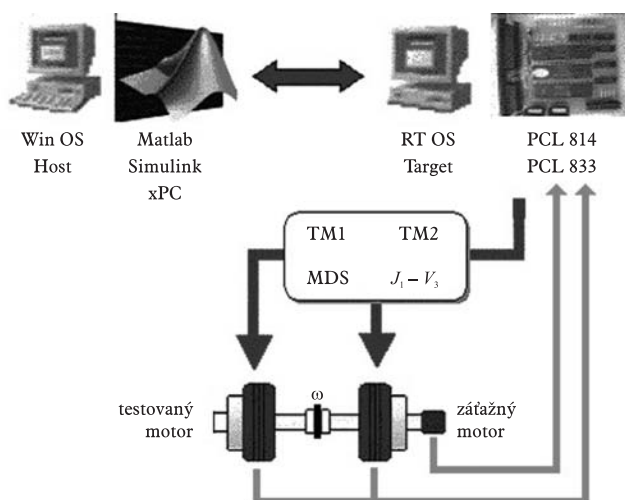
Teraz možno aplikovať rýchlostné alebo polohové riadenie na overenie kvality emulácie zvolených záťaží.

2. Realizácia

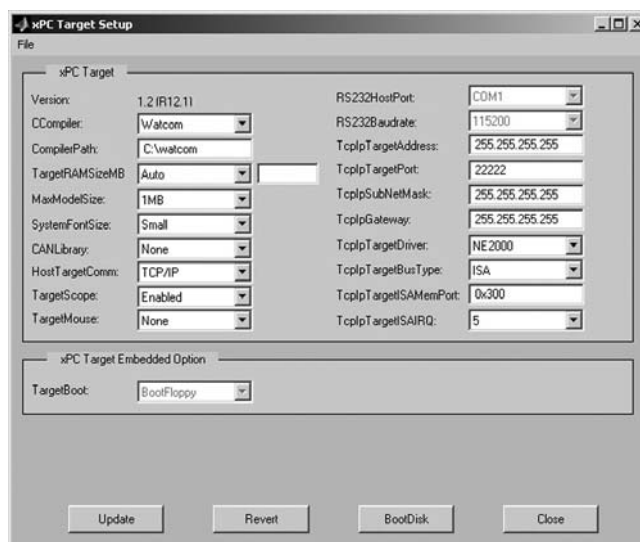
Emulátor je postavený ako systém dvoch jednosmerných motorov HSM 150 spojených pevnou spojkou. Oba motory sú napájané tranzistorovými meničmi. Schéma zapojenia je na obr. 5.

Na riadenie emulátora v reálnom čase sme použili programový balík Matlab 6.1 a prídavné knižnice Real-Time Workshop a xPC Target. Menič je riadený pomocou riadiaceho počítača (Target PC), ktorý beží na xPC operačnom systéme OS v reálnom čase a vytvorí ho priamo Matlab. Riadiaci počítač je na komunikáciu s meničom vybavený A/D a D/A kartami PCL814 a PCL833 firmy Advantech. Operačný systém sa do počítača zavádza z bootovacej diskety. Uholová rýchlosť hriadeľov motorov sa vyhodnocuje nepriamo z inkrementálneho snímača polohy IRC 205, 2500 imp./ot. Rozlišovacia schopnosť IRC je softvérovou zvýšená na 10 000 imp./ot. Riadiaci počítač je ethernetovými sieťovými kartami pripojený cez lokálnu sieť k vývojovému počítaču (Host PC). Vývojový počítač je vybavený programovým prostriedkom Matlab 6.1 s knižnicou xPC. Aby dokázal Matlab spolupracovať s A/D a D/A kartami, treba pridať a uložiť do jeho ciest prídavnú knižnicu „Karty“, ktorá obsahuje ovládače použitých kariet.

Rozširujúca knižnica xPC Target je určená na vytváranie špecifického typu real-time aplikácií, ktoré sa spúšťajú na samostatnom riadiacom počítači. Návrh a vývoj aplikácie prebieha na vývojovom počítači v Matlab Simulinku. Prenos vytvorenej aplikácie na riadiaci počítač je automatizovaný a takisto aj prenos nameraných údajov z riadiaceho počítača späť do vývojového. Na to, aby bolo možné vytvárať samostatne spustiteľné aplikácie, musíme mať na vývojovom počítači okrem nainštalovaných súčastí Matla-



Obr.5 Principiálne zapojenie emulátora



Obr.6 Nastavenie xPC Target

bu k dispozícii aj kompilátor jazyka C/C++. Spoluprácu kompilátora s Matlabom nastavíme v Matlabe príkazom mex-setup. Ďalej treba nakonfigurovať Matlab na zbiehanie xPC Target aplikácií; slúži na to príkaz xpcsetup. Po jeho spustení sa otvorí okno xPC Target Setup.

V tomto okne sa formou výberu nastaví kompilátor jazyka C/C++, cesta ku kompilátoru, pamäť RAM riadiaceho počítača. Dôležité je nastaviť správne parametre pre sieťovú komunikáciu podľa obr. 6. Ak často meníme Target PC, alebo pracujeme s viacerými, možno si uložiť nastavenia pre konkrétny prípad cez záložku File a potom ich podľa potreby nahráť do počítača a aktualizovať tlačidlom Update v tomto okne.

V tomto okne sa tlačidlom BootDisk tiež vytvára bootovacia disketa so spomínaným xPC operačným systémom. Po správnom nastavení xPC Target parametrov a vytvorení štartovacej diskety môžeme zaviesť OS do riadiaceho počítača. Po vložení štartovacej diskety a zapnutí sa nahrá BIOS, potom xPC Target a zobrazí sa používateľské rozhranie, ktoré slúži na zobrazenie parametrov riadiaceho počítača, aplikácie a niektorých stavových údajov počas jej behu. Na otestovanie nastavenia a komunikácie Host a Target PC slúži príkaz xptest, po ktorom sa začne testovanie v niekoľkých krokoch. V okne Matlabu sa zobrazujú práve vykonávané príkazy. Postupne sa zobrazujú výsledky jednotlivých testov, ktorých výsledkom môže byť:

- OK – test prebehol úspešne,
- FAILED – nastala chyba,
- SKIPPED – test sa preskočil.

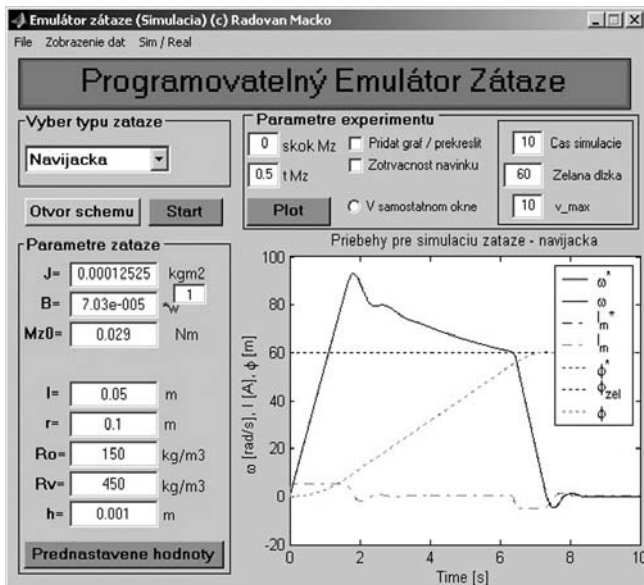
V poradí tretím testom je reštart Target PC, čo spomaľuje test. Preto ho môžeme vynechať zadáním xptest noreboot. Podrobne je práca s real-time toolboxmi opísaná v [5].

Výhodou realizácie pomocou xPC Targetu je fakt, že aplikácia beží na samostatnom počítači, čo umožňuje dosahovať vyšší výkon (menšiu periódu vzorkovania).

3. Aplikácia emulátor

Ovládanie, spúšťanie a zastavovanie emulácie riadenej xPC Targetom v reálnom čase a ukladanie nameraných údajov z fyzikálneho experimentu nie je úplne triviálne. Navrhnutý emulátor má slúžiť aj ďalším študentom pri experimentálnom overovaní nimi navrhnutých riadiacich algoritmov.

S cieľom zjednodušiť prácu s navrhnutým emulátorom od zadania typu a parametrov záťaže, cez simuláciu, spustenie fyzikálneho experimentu až po spracovanie experimentálnych údajov získaných emuláciou sme vytvorili aplikáciu „emulator“ v prostredí



Obr.7 Hlavná figúra aplikácie emulátor

Matlab 6.1. Táto aplikácia je spustiteľná z príkazového riadku Matlabu zadaním emulátor.

Po spustení sa otvorí figúra pre simulačnú časť najjednoduchšej - lineárnej zotrvačnej zátaze. V základnom menu možno zvoliť jeden zo 4 typov zátaze (lineárna, ventilátorová, navijacka a dvojhmotový pružný systém). Hlavné okno je rozdelené na 4 časti podľa obr. 7. V časti „Vyber typu zátaze“ si používateľ formou výberu z ponuky zvolí typ vyžadovanej zátaze. Po označení políčka „*J“ sa moment zotrvačnosti nezadáva číselne, ale ako násobok zotrvačnosti systému. V časti „Parametre zátaze“ sa k zvolenému typu zátaze objavia editovateľné políčka, v ktorých možno parametrizovať konkrétnu záťaž. V časti „Parametre experimentu“ sa nastavujú parametre simulácie, napr. čas simulácie, želaná hodnota riadenej veličiny alebo perióda, s ktorou sa bude pokus bude opakovať. Je tu tiež možnosť nastaviť veľkosť a čas externého skoku momentu zátaze. Ďalej sa dá nastaviť spôsob zobrazenia mera- ných údajov.

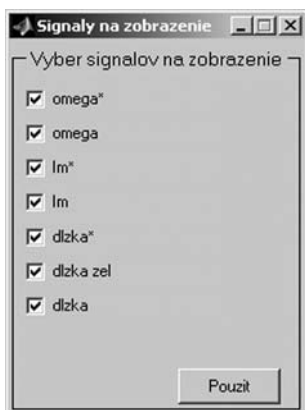
Zaškrtnutím políčkami „Pridať graf/prekresliť“ si môžeme zvoliť, či chceme viac experimentov zobrazit do jedného grafu (na porovnanie) alebo nie. Po označení políčka „V samostatnom okne“ sa budú namerané údaje zobrazovať v samostatných grafoch (figúrach). V poslednej časti je priestor na orientačné zobrazenie všetkých získaných priebehov.

Tlačidlom „Otvor schému“ otvoríme príslušnú simulačnú schému, v ktorej môžeme zmeniť typ regulátora (pri testovaní navrhnutých algoritmov), parametre štandardného (nami dodaného) regulátora a pod. Tlačidlom „Prednastavenie hodnoty“ sa vrátíme k pôvodným parametrom zátaze a experimentu. Tlačidlo

„Plot“ slúži na vykreslenie nameraných priebehov (zvoleným spôsobom). Tlačidlo je aktívne až po skončení pokusu, ktorý spúšťame tlačidlom „Štart“.

Ďalšie možnosti zobrazenia výsledkov pokusu sú prístupné takisto po skončení pokusu v položke „Zobrazenie dát“ v menu na hornej lište aplikácie. Tu si používateľ môže vykresliť chybové funkcie príslušných veličín (obr. 9).

Pri voľbe „Vyber signálov“ sa otvorí figúra obr. 8, v ktorej si



Obr.8 Výber zobrazenia údajov

používateľ vyberie signály, ktoré chce zobraziť. Má teda možnosť výberu len tých priebehov, ktoré ho zaujímajú. Ponuka signálov na zobrazenie, samozrejme, zodpovedá typu zvolenej zátaze (poloha pri navijacke atď.).

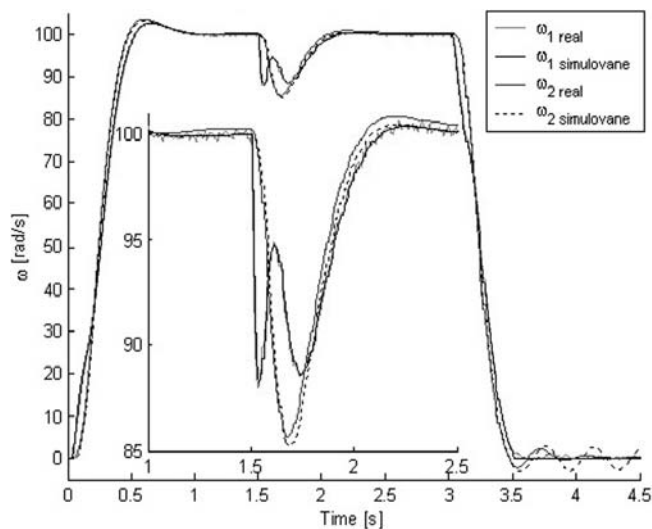


Obr.9 Výber zobrazených signálov

Ak uvážime, že simulácia pokusu so zvolenou záťažou a jej výsledky sú v poriadku a že sme pripravení uskutočniť experiment na fyzikálnom emulátore, výberom „Realizácia“ v položke „Sim/Real“ sa prepne do realizačnej časti aplikácie. Tá vyzerá rovnako ako táto s tým, že pri výbere signálov na zobrazenie je ich v ponuke menej (len experimentálne). Po prepnutí sa zachovávajú všetky nastavenia, ktoré sme urobili (porovnanie experimentu a simulácie) a tlačidlom „Štart“ sa spustí vytváranie spustiteľného kódu, jeho nahrávanie do Target PC, spustenie a zastavenie experimentu. Ak sme zaškrtnuli políčko „Pridať graf/prekresliť“, máme možnosť porovnať experiment so simuláciou v jednom grafe. Výberom „Exit“ v položke „File“ v menu sa skončí práca s aplikáciou.

4. Experimentálne výsledky

Pre názornosť uvedieme príklad zložitejšieho experimentu a porovnáme ho so simuláciou na overenie správneho fungovania emulátora. V experimente sme emulovali pružný systém bez tlmenia (najproblematickejší prípad) s konštantou torznej tuhosti $C_t = 0,1 \text{ Nm} \cdot \text{s/rad}$ a pomerom zotrvačností $R = 3$. V čase 1,5 s sme privedli (poruchu) skok momentu zátaze 0,1 Nm (asi 50 % nominálneho momentu). Na riadenie servopohonu s emulátorom pružného systému sme použili regulátor navrhnutý metódou umiestňovania pólov pomocou identických koeficientov tlmenia [7].



Obr.10 Porovnanie simulačných a experimentálnych rýchlostí dvojhmotového pružného systému

Na obr. 10 vidíme, že emulované veličiny sa zhodujú so vzáajnými simulačnými priebehmi rýchlosti a prúdu. To aj napriek tomu, že sme emulovali pružný systém s pomerom momentu zotrvačnosti zátaze k momentu zotrvačnosti motora rovným 3, konštantou torznej tuhosti len $0,1 \text{ Nm} \cdot \text{s/rad}$ a nulovým tlmením. K tomu sme navyše testovali aj vplyv poruchy na reguláciu rýchlosti servopohonu s emulátorom. Regulácia uhlovej rýchlosti zátaze ω_2 bola nepriama, teda v uzavretej slučke sme riadili len uhlovú rýchlosť motora ω_1 . Napriek takýmto podmienkam sa emulovaný pružný systém len s malým prekritom rýchlosti ustálil na vyžadovanej hodnote. V okne (obr. 10) si môžeme všimnúť, že priamo riadená rýchlosť motora sa so vzáajnou zhoduje takmer presne a nepriamo riadená rýchlosť zátaze má len malé odchýlky, ktoré sú však menšie ako 1 %.

Záver

Emulátor realizovaný s využitím real time toolboxu programového balíka Matlab a xPC Target knižnice poskytne vynikajúcu pomoc pri návrhu, vývoji a skúšaní rôznych algoritmov riadenia. Použitie xPC Targetu tiež umožnilo zbierať aplikácie s vysokými nárokmi na výpočtový výkon, ktoré by si inak vyžadovali výkonnejšie PC. Využitím možností (figúry), ktoré ponúka Matlab, sa podarilo sprístupniť emulátor širšiemu okruhu ľudí a podstatne zjednodušiť prácu s ním. Realizované riešenie emulátora je univerzálne, teda emulátor možno postaviť na ľubovoľnej dvojici motorov, čo otvára ďalšie možnosti skúmania. Emulátorom možno testovať napríklad pohony výťahov, obrábacích strojov, navíjačiek, ventilátorov, čerpadiel a v podstate akýchkoľvek záťaží, ktoré vieme matematicky popísať v tvare $\omega(s)/M_m(s)$, čo môže zjednodušiť návrh a vývoj riadiacich štruktúr a pomôcť pri overovaní vyžadovaných vlastností motorov výskumným pracoviskám aj súkromným firmám.

Literatúra

- [1] COLLINS, E. R., HUANG, Y.: A programmable Dynamometer for Testing Rotating Machinery Using Three-Phase Induction Machine. IEEE Transactions on Energy Conversion, 9, 3, 1994, str. 521 – 527.
- [2] SANDHOLD, P., RITCHIE, E., PEDERSON, J. K., BETZ, R. E.: A Dynamometer Performing Dynamical Emulation of Loads with Non-Linear Friction. IEEE Int. Symposium on Industrial Electronics, 2, 1996, str. 873 – 878.
- [3] BETZ, R. E., PENFOLD, H. B., NEWTON, R. W.: Local Vector of an AC Drive System Load Simulator. IEEE Conference on Control Applications-Proceedings. 1, 1994, str. 721 – 726.

[4] NEWTON, R. W., BETZ, R. E., PENFOLD, H. B.: Emulating Dynamical Load Characteristics Using a Dynamic Dynamometer. Proceedings of Int. Conference on Power Electronics and Drive Systems, 1, 1995, str. 465 – 470.

[5] MASÁR, I., IVANOV, I.: Aplikácie reálneho času v programovom prostriedku MATLAB/SIMULINK. Bratislava: Vydavateľstvo STU, 2001.

[6] AKPOLAT, H. Z., ASHER, M. G., CLARE, C. J.: Emulation of high bandwidth mechanical loads using vector controlled AC dynamometer. Zborník príspevkov z 8. medzinárodnej konferencie Power Electronics and Motion Control (PEMC '98), 5 diel, Praha, 1998, str. 133 – 138.

[7] ZHANG, G., FURUSHO, J.: Speed Control of Two-Inertia System by PI/PID Control. IEEE, Transactions on industrial electronics, vol. 47, No. 3, 2000.

[8] HEWSON, C. R., ASHER, M. G., SUMMER, M.: Practical emulation of the dynamic characteristics of mechanical loads. Zborník príspevkov z 8. medzinárodnej konferencie Power Electronics and Motion Control (PEMC '98), 6 diel, Praha, 1998, str. 52 – 57.

Ing. Radovan Macko
prof. Ing. Milan Žalman, PhD.
Ing. Marian Uriček

Slovenská technická univerzita
Fakulta elektrotechniky a informatiky
Ilkovičova 3
812 19 Bratislava
e-mail: radovan.macko@stuba.sk
milan.zalman@stuba.sk
marian.uricek@stuba.sk

39