



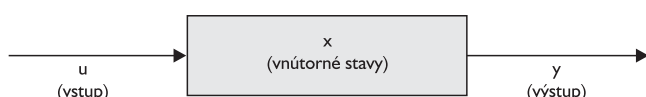
MATLAB

SIMULINK – simulačné prostredie pre Matlab (2)

Matematické modelovanie na počítačoch sa stalo účinnou metódou riešenia vedecko-technických problémov. Počítačový model zostavený na základe matematickej analýzy predstavuje náhradnú vzorku skúmaného objektu. S modelom môžeme vykonávať experimenty podobné tým na reálnom objekte. Výhodou sú nižšie náklady na experimenty, možnosť viacnásobného opakovania experimentu bez rizika poškodenia, prípadne experimenty, ktoré by reálny objekt zlikvidovali. Experiment vykonaný na modeli sa nazýva simulácia. Snažíme sa navodiť podmienky podobné reálnym a model podrobujeme testovaniu v rôznych situáciách. Pomocou moderných simulačných prostriedkov sa vykonávajú komplikované výpočty, ktoré dávajú odpoveď na dôsledky nášho konania. Dá sa teda predikovať dosah klimatických zmien, dôsledky premnoženia dravcov, ale aj správanie motora pri určitej záťaži. Vierohodnosť výsledkov závisí od kvality modelu, ako aj od správneho vykonania simulácie [1].

Nástrojov na počítačové modelovanie a simulovanie je dnes na trhu dostatok, ale len málokto sa môže chváliť takým širokým zameraním a podporou ako Simulink od spoločnosti The MathWorks. Simulink je grafickou nadstavbou programového balíka Matlab. Pri simuláciách využíva jeho výpočtové jadro a dokáže pristupovať k celej škále rozširovacích modulov – toolboxov. V novších verziách sa stretávame už aj so špecializovanými nadstavbami pre Simulink – blocksets. Tie dokážu rozšíriť základné knižnice Simulinku o bloky so špeciálnym zameraním napr. na energetiku, hydrauliku alebo mechanické systémy. Zákazník si tak poskladá konfiguráciu softvéru podľa svojich potrieb. Už základná inštalácia obsahuje bloky na modelovanie spojitých, diskretných, lineárnych alebo nelineárnych systémov. Nájdeme tu aj základné logické a matematické operácie. Samozrejmosťou sú aj rôzne možnosti zobrazovania simulovaných veličín.

Simulink je vhodný nástroj na tvorbu dynamických modelov a ich simuláciu. Na tvorbu modelov netreba programovať zložité diferenciálne alebo diferenčné rovnice, ale pomocou blokových schém sa skladajú jednotlivé funkčné bloky, ktoré sa vzájomne prepájajú a tým tvoria medzi sebou väzby. Modelované systémy môžu byť spojité, diskretné, prípadne ich kombináciou sa vytvárajú modely hybridných systémov. Simulačná schéma pozostáva z blokov a signálnych väzieb (obr. 1).



Obr.1 Blok simulačnej schémy

Do bloku vstupuje vstupný signál u . V bloku sa spracuje jeho hodnota a generuje sa výstupný signál y . Ak blok modeluje určitú dynamiku, tak výstup y sa počíta na základe stavov x a vstupu u . Medzi bloky s dynamikou patria napr. integrátor alebo prenosová funkcia. Existujú aj bloky, ktoré nemajú svoje vnútorné stavy, napr. zosilnenie alebo sumačný blok. V takýchto blokoch sa výstup ráta len na základe aktuálnej hodnoty vstupného signálu. Na jednoduchom príklade môžeme uviesť, ako na základe opisu systému sústavou diferenciálnych rovníc vytvoríme funkčnú schému. Nech je jednosmerný motor opísaný diferenciálnou rovnicou (1) [2].

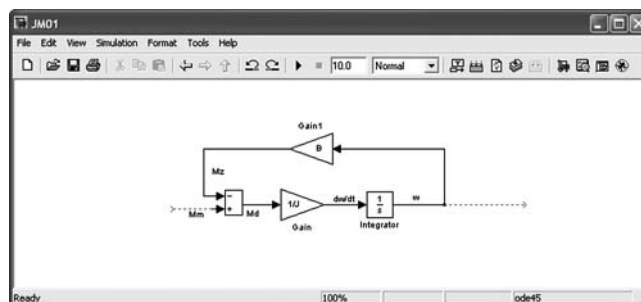
$$M_m - M_z = J \frac{d\omega}{dt} \quad (1)$$

Kde M_m je moment motora, M_z moment záťaže, J moment zotrvačnosti a ω uhlová rýchlosť. Moment záťaže opíšeme (2)

$$M_z = B\omega \quad (2)$$

Rovnicu (1) upravíme na tvar (3).

$$\frac{d\omega}{dt} = \frac{1}{J} (M_m - M_z) \quad (3)$$



Obr.2 Schéma mechanickej časti jednosmerného motora

Rovnice (2) a (3) už prekreslíme jednoducho do schémy v Simulinku (obr. 2), čím sme vytvorili model mechanickej časti jednosmerného motora.

Elektrická časť je opísaná rovnicami (4), (5) a (6).

$$M_m = C_u \cdot i \quad (4)$$

$$u_i = C_u \cdot \omega \quad (5)$$

$$u = R \cdot i + L \frac{di}{dt} + u_i \quad (6)$$

Rovnicu (6) upravíme na tvar (7)

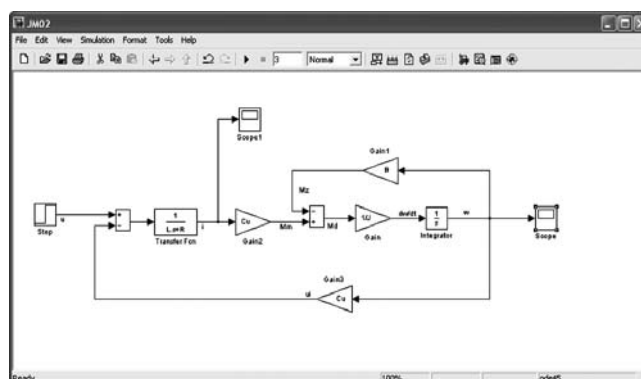
$$L \frac{di}{dt} + R \cdot i = u - u_i \quad (7)$$

Na základe uvedených rovníc vytvoríme schému jednosmerného motora. Veličiny, ktorých vývoj v čase potrebujeme sledovať (i , ω), pripojíme na blok Scope (obr. 3).

Aby bolo možné vykonať experimenty s opísaným motorom, treba nastaviť parametre (tab. 1).

Nakolko je Matlab plne previazaný so Simulinkom, parametre priradíme do premenných v Matlabe (obr. 4).

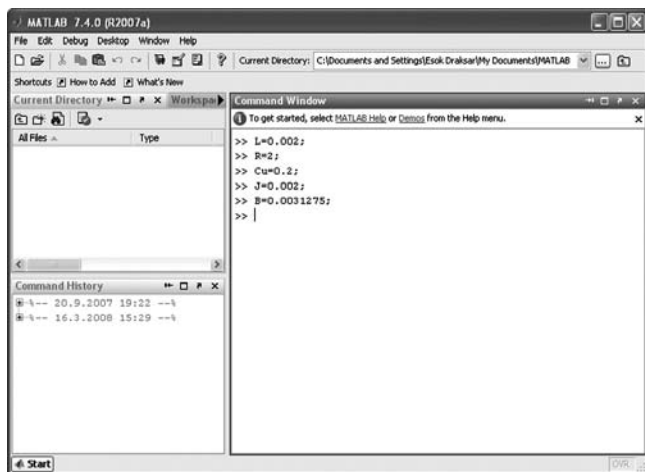
V menu Simulinku nastavíme parametre simulácie, ako sú čas simulácie, periódu vzorkovania, prípadne niektorú z ponúkaných integračných metód. V tomto okamihu je model pripravený na vykonanie simulácie. Tá sa spúšťa buď priamo v okne Simulinku, alebo v Matlabe zadáním príkazu sim. Ak sú všetky nastavenia správne, dôjde k samotnej simulácii. Výsledkom sú priebehy monitorovaných veličín. V našom prípade



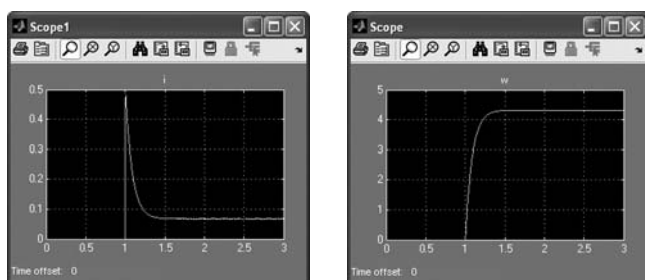
Obr.3 Schéma jednosmerného motora

| | | |
|-------|------------------------|---------------------------|
| L | indukčnosť motora | 0,002 H |
| R | odpor vinutia motora | 2 W |
| C_u | konštanta motora | 0,2 Vs |
| J | moment zotrvačnosti | 0,002 Ws ³ |
| B | viskózná zložka záťaže | 0,0031275 Ws ³ |

Tab.1 Parametre jednosmerného motora



Obr.4 Parametrizácia veličín pre simulačnú schému v Matlabe



Obr.5 Priebeh prúdu a uhlovej rýchlosti po pripojení budiaceho napätia

je to priebeh prúdu a uhlovej rýchlosti v čase. Veličiny sa zobrazujú v Simulinku po otvorení príslušného Scope (obr. 5).

Druhou možnosťou je zobraziť priebehy prúdu a uhlovej rýchlosti v Matlabe, do ktorého sa zo Simulinku dostali ako vektory. V Matlabe sa môžu vykonať aj ďalšie analýzy priebehov so zapojením celého výpočtového aparátu.

Uvedený príklad sa zaoberá elektromechanickým modelom jednosmerného motora. Pre takto vytvorený model možno jednoducho a efektívne navrhovať riadenie, pričom sledujeme, či špička prúdu nepresahuje povolenú úroveň, alebo či priebeh regulovanej veličiny spĺňa naše požiadavky. Ak je takýto motor len istou časťou komplexnej schémy, ktorá zahŕňa komplikovaný systém pozostávajúci z množstva ďalších objektov, úspešne sa využíva zoskupovanie blokov. V tomto prípade by sme celý model jednosmerného motora uzavreli do jedného bloku, ktorý by mal ako vstup budiace napätia a výstupom by bola uhlová rýchlosť. Pomocou hierarchického vnárania funkčných blokov sa celá schéma sprehľadňuje. Ďalším pozitívom je, že na komplikovanej schéme môže pracovať súčasne niekoľko vývojárov a vzájomne si odovzdávajú čiastkové bloky, ktoré sa neskôr integrujú do komplexnej schémy.

Simulačné experimenty v reálnom čase sú pre vývoj riadiacich systémov veľmi významné. V tomto prípade treba zabezpečiť dodržanie požadovanej periódy vzorkovania a komunikáciu s okolím prostredníctvom unifikovaných signálov. Simulink prináša dve možnosti riešenia tohto problému. Prvé sa opiera o nastavbový produkt Real Time Workshop [3]. V tomto prípade sa schéma prekladá do kódu jazyka C. Takto vytvorený kód sa následne prekladá na spustiteľnú aplikáciu pre cieľový (target) hardvér. Týmto hardvérom môže byť riadiaci počítač typu PC alebo jednočipový mikroprocesor. Je zrejmé, že Real Time Workshop rozširuje možnosti Simulinku aj o komunikáciu s okolitým svetom prostredníctvom vstupno-výstupných kariet. Škála podporovaných kariet je široká a zahŕňa produkty významných spoločností, ako Advantech alebo National Instruments. Druhou možnosťou je využiť produkt spoločnosti Humusoft, s. r. o., Real Time Toolbox [4]. Ide o knižnicu blokov, ktorá rozširuje Simulink o možnosť pripojenia niektorých z veľkej množiny podporovaných vstupno-výstupných kariet. Priamo do schémy tak pripájame komunikačný kanál karty, ktorý

v schéme vystupuje ako signál. Bez problémov tak vysielame riadiace signály a spracúvame namerané hodnoty charakterizujúce stav riadeného objektu. V tomto prípade nedochádza k žiadnej kompilácii. Tento fakt spôsobuje na jednej strane výrazné zjednodušenie nasadenia riadiacich algoritmov na reálne zariadenie. Na druhej strane simulácia prechádza cez všetky vrstvy operačného systému, až kým sa dostane na najnižšiu hardvérovú úroveň. To má za dôsledok výrazne dlhšie minimálne časy vzorkovania v porovnaní so vzorkovaním kompilátom z Real Time Workshopu. Ak navrhujeme riadenie pre pomalšie systémy, kde stačí perióda vzorkovania na úrovni 10 ms, môžeme sa spoľahnúť na službu Real Time Toolboxu. Ide o lacnejšiu a jednoduchšiu možnosť. Podobne ako Matlab je aj Simulink portovaný na viacero operačných systémov. V tomto prípade sa však dizajn Simulinku pre Windows výraznejšie líši od verzií pre Linux a Unix systémy. Simulink pre Windows využíva stromovú štruktúru, kde sú uložené knižničné bloky. Unixová verzia Simulinku má bloky uložené v akýchsi podskupinách pripomínajúcich subsystémy (obr. 6).

Simulink sa stal obľúbeným nástrojom v situáciách, kde treba pomocou blokových schém vytvárať modely určitých systémov. Ide teda o univerzálny modelovací a simulačný nástroj, ktorý sa uplatňuje v rôznych vedných a technických oblastiach. V súčasnosti existuje niekoľko špecializovaných knižníc, ktoré rozširujú jeho možnosti v danom vednom odbore. Významnými knižnicami sú toolboxy a blocksety zamerané na návrh riadiacich systémov, modelovanie energetických sústav, mechanických objektov alebo hydraulických systémov. Komunikácia v reálnom čase len podčiarkuje široké možnosti Simulinku vo vedeckej aj v komerčnej sfére. Podrobné informácie o produkte Simulink nájdete na stránkach výrobcu [5] alebo vo webovom seminári [6].

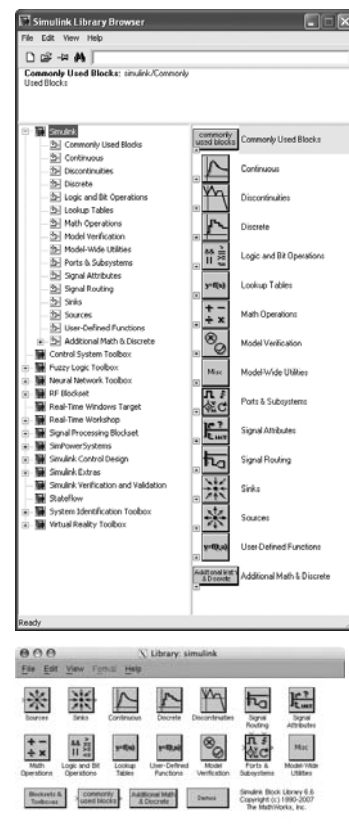
Článok vznikol s podporou APVV-99-045805.

Literatúra

- [1] PAZOUŘEK, J.: Simulace biologických systémů. Praha: Grada 1992.
- [2] ŽALMAN, M.: Akčné členy. Bratislava: STU, 2003.
- [3] <http://www.mathworks.com/products/rtw/>
- [4] <http://www.humusoft.cz/rt/indexcz.htm>
- [5] <http://www.mathworks.com/products/simulink/>
- [6] <http://www.mathworks.com/cmspro/req15027.html?eventid=48972>

Ing. Martin Foltin, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3, 812 19 Bratislava
e-mail: martin.foltin@syprin.sk



Obr.6 Simulink pre Windows a Mac OS X