

MODEL BASED DESIGN FOR EC MOTOR CONTROL

V. Hubík*, T. Kerlin, J. Toman

Summary: *Model Based Design is quick and effective method of any dynamic system development. The ground of a design process is a mathematical-physical model of a system in MATLAB/Simulink environment, which is during development still improved and its quality is tested in the real hardware simulations. This paper shows the possibilities of MBD during control algorithms development for brushless motor direct current motors (EC motor) in sensor and sensorless working mode. The simulation results are directly applied in the real dSPACE hardware with custom electronic connected to that. MDB offers automatic code generating from the Simulink model. This method is not only for dramatic development time reduction, but eliminates usual and often mistakes done during manual firmware writing.*

1. Úvod

Vývoj kritických aplikací, např. v leteckém průmyslu, podobně jako v průmyslu automobilovém, je ovlivněn moderními vědeckými trendy a postupy. Uplatňují se zde s výhodou metody, známé z odborných literatur jako *Model Based Design* a *Rapid Control Prototyping* [1]. S tím také velmi úzce souvisí pojem vývojových cyklus „V“, který bude v následujících odstavcích blíže popsán.

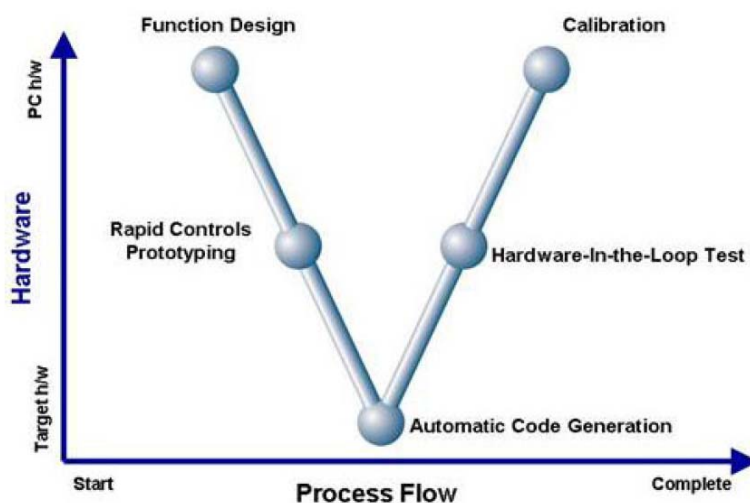
„V“ cyklus je obrazec, viz Obr. 1, který slouží ke grafickému vyjádření postupu softwarových i hardwarových vývojových prací složitých řídicích systémů. Je založen na detailním matematickém modelu celého systému v příslušném počítačovém prostředí, např. MATLAB. Jeho výhodou je intuitivní přirozenost, jednoduché použití a přenositelnost na libovolné cílové platformy. Vývoj založený na modelování (*Model Based Design*) přináší značnou úsporu časových i finančních nákladů. Vývojovým pracovníkům dále nabízí kompletní přehled a kontrolu celého vytvářeného systému v integrovaném grafickém prostředí.

Kompletní návrh systému se skládá z několika kroků – funkční návrh, *Rapid Control Prototyping*, automatické generování kódu do cílové platformy, *Hardware-In-the-Loop* (HIL) testování a finální kalibraci. Výčet jednotlivých kroků odpovídá pořadí prací při vývoji.

Během první etapy funkčního návrhu je vytvoření modelu a počítačová simulace uzavřené řídicí smyčky, tzn. model řízeného systému (*plant model*) a řídicího systému. Takto vytvořené modely jsou zatím pouze symbolické, ve tvaru daným použitým simulačním prostředím. Pokud je systém vytvořený a výsledky simulace odpovídají definovanému zadání, začnou se

* Ing. Vladimír Hubík, Institute of Production Machines, Systems and Robotics, Brno University of Technology, Technická 2896/2; 616 69 Brno; tel. +420 541 515 531; e-mail: vhubik@nbox.cz

jednotlivé algoritmy ověřovat v reálném čase a v reálném hardware. Tato fáze je nazývána *Rapid Control Prototyping* (RCP).



Obrázek 1 Vývojový „V“ cyklus

RCP je proces, při kterém se ověřují vytvořené řídicí bloky na real time počítači a na reálném nebo opět modelovaném systému. Počítačový model je použitý v případě, že chybná funkce řídicího systému může způsobit vážné poškození řízeného zařízení nebo ztráty na životech. Největší výhodou použití integrovaného prostředí pro modelování, simulaci a tvorbu prototypů je, že vývojový pracovník nepotřebuje znát C kód a nemusí mít dostatek zkušeností s implementací softwaru do reálného cíle. RPC prostředí to udělá za něj. V následující fázi je z matematického modelu automaticky generován a kompilován zdrojový kód, který představuje chování celého systému. Dochází tak ke dramatickému zkrácení vývojového času oproti klasickému vývojovému cyklu, při stálém zachování soudržnosti mezi specifikační a realizační etapou.

Jakmile je řídicí systém naprogramovaný, nastává čas na *Hardware-In-the-Loop* (HIL) etapu vývoje. HIL je forma real time simulace, která se odlišuje přidáním reálných prvků do řídicí smyčky. Tato prvky mohou být samotný řídicí systém nebo reálný řízený systém. Velmi často používaná konfigurace v průmyslu je, že *plant* model (prostředí, stejný jako v RCP fázi) je simulován a řídicí systém je reálný.

Posledním krokem „V“ cyklu je kalibrace. Je to proces optimalizace a ladění reálného řídicího algoritmu za účelem dosažení požadované odezvy systému podle technického zadání. Kalibrační přípravky jsou kombinací hardwarového rozhraní a softwarové aplikace, která umožňuje přístup a změny kalibračních proměnných v řídicím systému. Typické části, které vyžadují kalibraci, jsou různé *look-up* tabulky, násobící koeficienty, konstanty, atd. Struktura řídicích algoritmů zůstává během kalibračního procesu nezměněna.

Výhody nastíněného vývojového procesu jsou především:

- jednotné vývojové prostředí pro všechny skupiny vývojářů a všechny fáze vývoje aplikace či prototypu
- rychlá reakce na změny požadavků a parametrů aplikace

- průběžné testování, opravy chyb a optimalizace modelu během celého vývojového cyklu
- automatické generování kódu pro cílovou platformu, snadný vývoj testovacích procedur

a v neposlední řadě i automatické generování dokumentace pro certifikační fázi vývoje. Komerčně dostupné softwarové a hardwarové vybavení, např. MATLAB se systémem dSPACE, splňuje parametry dle norem RTCA DO-178, 254 a je tudíž vhodné i pro vývoj kritických aplikací v leteckém průmyslu.

2. dSPACE systém

dSPACE je systém pro řízení a simulace v reálném čase. Umožňuje rychlý vývoj řídicích systémů a simulace HIL. Je tvořen výkonným hardware s možností modularity podle požadavků spotřebitele, dále pak kompatibilní software RTI s prostředím MATLAB/Simulink, který slouží k implementaci, nastavení a obsluze vstupně/výstupních karet. Grafické prostředí ControlDesk pro ovládání, nastavení parametrů a následný monitoring simulace, je rovněž součástí dSPACE. Poslední část celého systému je tzv. *TargetLink*, který slouží k definici cílové platformy a k úpravě zdrojového kódu.

2.1 dSPACE hardware

Firma dSPACE nabízí dva druhy hardwaru:

- jednodeskové systémy pro méně náročné aplikace
- modulární systémy pro náročné aplikace s možností nakonfigurování podle požadavků uživatele

Pro požadavky řízení elektricky komutovaných motorů byl zvolen modulární systém, jehož jádrem je procesorová deska DS1005, osazená procesorem PowerPC 750GX. K této procesorové desce jsou pomocí sběrnice PHS++ připojeny rozšiřující desky s periferiemi. S ohledem na množství a druh periferií, které jsou nutné k řízení EC motoru, byly vybrány desky DS5101 a DS2004. Všechny tyto desky jsou vloženy do speciálního boxu s označením PX20, Obr. 2.



Obrázek 2 Varianty boxů s označením PX

2.1.1 PX20 desktop box

Jedná se o desktopovou skříň podobnou osobnímu počítači. Má cca. dvacet 16-bit ISA slotů, do kterých lze vložit až 19 dSPACE rozšiřujících desek. Slot č. 1 je vždy rezervovaný pro připojení uživatelského PC, což je realizováno pomocí speciálního linkového připojení nebo Ethernetu. Pro vytvoření kompaktního samostatného pracoviště, je vhodná volba použít linkové připojení s uživatelským počítačem typu PC. Napájení je v rozsahu 90-264 V AC s automatickým přepínáním.

2.1.2 Procesorová deska

Procesorová deska DS1005 je osazena procesorem od firmy IBM s označením PowerPC 750GX s frekvencí 1 GHz. Tato deska provádí výpočty modelu v reálném čase a zpracovává veškerou komunikaci mezi ostatními rozšiřujícími deskami a uživatelským PC. Tato deska může pracovat samostatně nebo jako multi-procesorový systém. Komunikace mezi deskami probíhá, jak už bylo zmíněno výše, prostřednictvím sběrnice PHS nebo PHS++. Každá deska má svoji přednastavenou fyzickou adresu. Jsou-li v systému přítomny dvě desky stejného typu, je potřeba nastavit odlišné fyzické adresy pomocí integrovaných přepínačů. Fyzická realizace a vzhled desky je na Obr. 3.



Obrázek 3 Realizace desky DS1005

Vybrané parametry řídicí desky DS1005:

Paměť

- 1 MB L2 cache
- 128 MB SDRAM globální paměť pro uživatelská data
- 16 MB flash paměť pro boot firmware a pro uživatelskou aplikaci

Rozhraní

- RS-232 s přenosovou rychlostí do 115,2 Kbit/s
- PHS++
- Linková sběrnice / Ethernet pro komunikaci s PC

2.1.3 High-Speed A/D karta DS2004

Tato karta je určena především pro měření vysokofrekvenčních signálů. Je osazena šestnácti A/D kanály s diferenčními vstupy, čtyřmi externími spouštěcími vstupy. Definovatelné spouštění A/D převodů pomocí externích vstupů umožňuje přesné a rychle měření s následným přenosem signálů procesorové karty. Vyrovnávací paměť desky zaznamená až 16 384 hodnot na jeden kanál. Karta je vhodná pro rychlé snímání průběhů indukovaných napětí při rotaci EC motoru a dalších provozních analogových veličin.

2.1.3 Digital výstupní deska DS5101

DS5101 byla rovněž vybrána s ohledem na požadavky řízení EC motorů. Dokáže generovat synchronní jedno a tří fázové pulzně-šířkově modulované signály (tří kanálová PWM).

2.2 dSPACE software

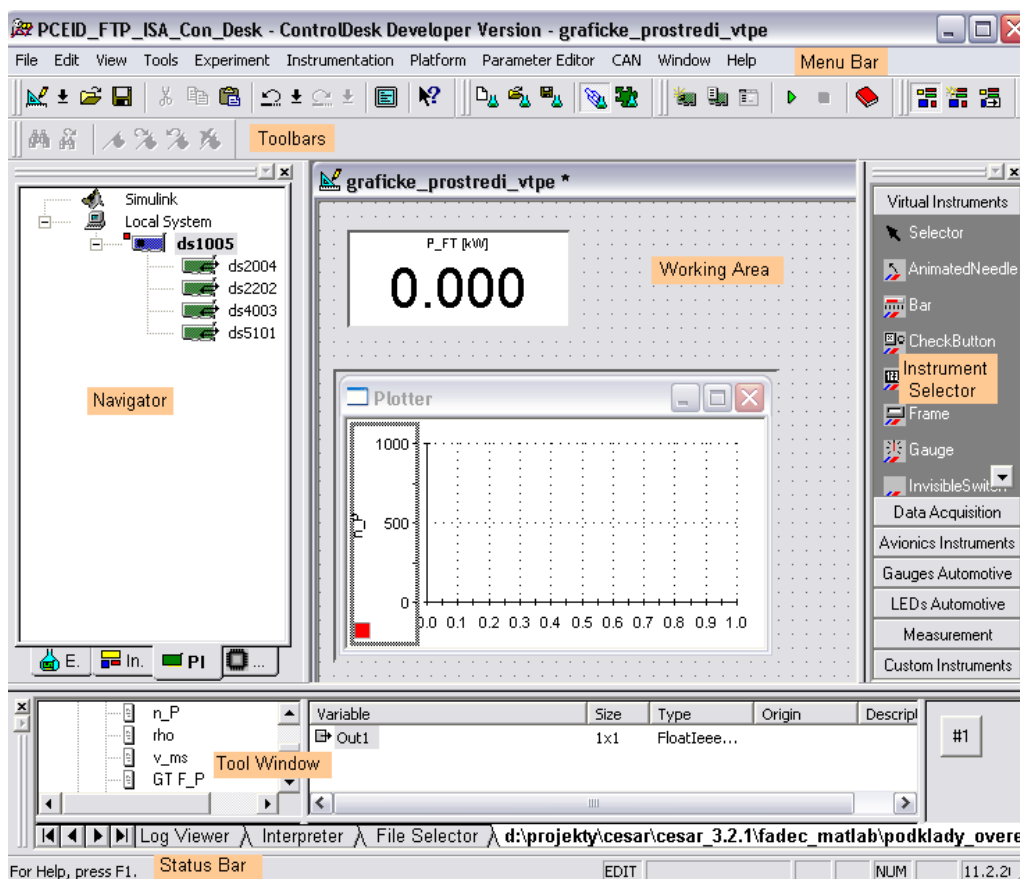
Základní software dodávaný k dSPACE systémům lze rozdělit do dvou hlavních skupin – ovládací software a realizační. Jako realizační software je označován RTI a ovládací ControlDesk. Další skupinou softwaru je již zmíněný Target Link, který umožňuje generování zdrojového kódu pro cílové platformy.

2.2.1 Real-Time Interface

RTI je spojení mezi dSPACE hardwarem a vývojovým softwarem Simulink. RTI zajišťuje konfiguraci I/O karet pomocí knihovny, která je umístěná v prostředí Simulink. Součástí této knihovny jsou předdefinované bloky I/O jednotlivých karet, ale také bloky, které je možné konfigurovat podle požadavků uživatele a možností karty. Tato knihovna obsahuje obslužné bloky pro všechny typy I/O karet. Každý blok jednotlivého typu karty obsahuje podskupiny bloků podle druhu periférií, které daná karta obsahuje. Práce s jednotlivými bloky knihoven je stejná jako v Simulinku.

2.2.1 ControlDesk

ControlDesk je experimentální software pro kontrolu a ovládání aplikací běžících v reálném čase na hardwaru dSPACE. Umožňuje interaktivní ovládání aplikace pomocí virtuálních měřících přístrojů, sběr a zobrazení naměřených dat a automatizaci experimentů pomocí skriptů a maker. Ke každé aplikaci je možné vytvořit grafické ovládací prostředí s možností interaktivní změny jednotlivých řídicích parametrů a s následným záznamem chování modelu na tuto změnu. ControlDesk umožňuje práci ve dvou režimech. První režim se nazývá vývojový mód. V tomto módu můžeme vytvořit grafické prostředí, měnit parametry a tvořit makra. Druhý mód se nazývá operátorský a je zabezpečen proti neoprávněným změnám. Prostor ControlDesku je na Obr. 4.



Obrázek 4 Prostředí ControlDesk

3. Real-Time Workshop

Real-Time Workshop překladač je rozšířením vývojového prostředí Simulink. Pro aplikace dSPACE systému je nezbytný, protože umožňuje přechod modelu v Simulinku na dSPACE hardware. Tato knihovna umí vygenerovat z modelu prototypový zdrojový kód v jazyce C. Po vygenerování real-time kódu ho zavede do procesorové karty a spustí.

Real-Time Workshop překladač má jistá omezení, neumí vygenerovat C kód ze všech bloků, které jsou k dispozici v knihovně Simulinku. Tento překladač nepodporuje algebraické smyčky, neumí pracovat s proměnným krokem *solveru*, nepodporuje S-funkce napsané v matlabovském jazyce M a nepodporuje některé optimalizace, které umožňuje Simulink.

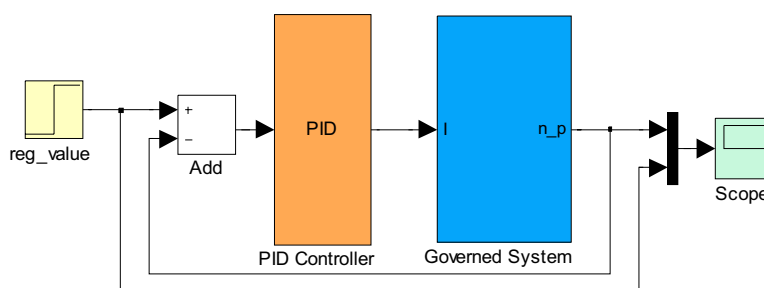
Real-Time Workshop neumí generovat zdrojový kód C takový, aby byl použitelný pro cílovou platformu. Proto tento kód výrobce nazývá prototypový.

4. Postup generování Real-Time aplikace

Postup generování Real-Time aplikace můžeme rozdělit do dvou etap. První etapa probíhá ve vývojovém prostředí Simulink. Druhá spočívá v obsluze a nastavení ovládacího prostředí v ControlDesk.

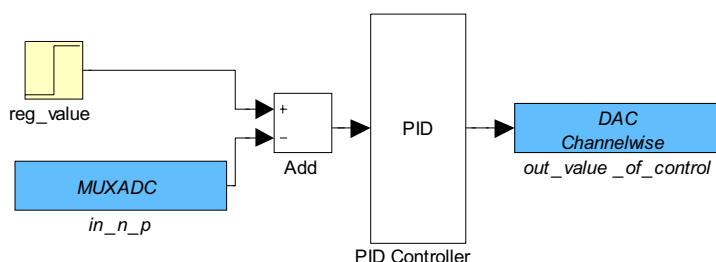
4.1 Simulink

Ve vývojovém prostředí Simulink je zapotřebí nejdříve vytvořit model aplikace Obr. 5., kterou chceme otestovat a řídit v reálném čase. Při tvorbě modelu je třeba mít na paměti fyzikální realizovatelnost, a také brát v úvahu i dostupný hardware dSPACE a jeho parametry např. frekvenci, na které model bude provozován. Základem modelu jsou jednotlivé bloky z knihovny např. Integrator, Constant, PID atd. Při sestavování blokového schématu nesmíme zapomenout na již zmíněná omezení dané kompilátorem RTW. Musíme sestavit model tak, aby se v něm nevyskytovaly algebraické smyčky, nesmíme používat S-funkce psané v M-File a hned nastavit *solver* s fixním krokem. Pokud nenajdeme způsob jak se vyhnout těmto omezením, lze použít S-funkce napsané v jazyce C nebo v C++. Pokud máme již vytvořené některé funkce pomocí skriptovacího jazyka Matlab M, lze použít z knihovny blok s názvem Embedded Matlab Function a do tohoto bloku tyto funkce implementovat. Takto napsanou funkci jazykem M již RTW překladač umí přeložit. Po sestavení modelu je zapotřebí tento model simulovat a tím otestovat funkčnost a požadavky na jeho chování.



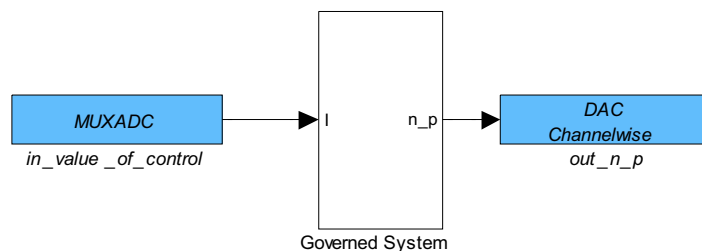
Obrázek 5 Model v Simulinku

Po ověření chování řízeného systému mohou nastat dvě situace. Do dSPACE hardware můžeme nahrát řídicí část modelu nebo řízený systém. Obě dvě možnosti se v praxi užívají. Jen záleží na tom, jestli chceme testovat v reálném čase řízení nebo řízený systém. Pokud chceme otestovat regulátor ponecháme v modelu pouze PID Controller, *reg_value* a *Add* blok ostatní odstraníme. Po odstranění nadbytečných bloků použijeme knihovnu RTI a do modelu přidáme předdefinované bloky I/O karet, které jsou potřebné pro zajištění funkčnosti viz Obr. 6.



Obrázek 6 Model regulátoru s I/O pro dSPACE

Podobný postup můžeme aplikovat pro testování řízeného systému jen s tím rozdílem, že ponecháme jen blok *Governed System* a přidáme I/O bloky viz Obr. 7.



Obrázek 7: Model řízeného systému s I/O pro dSPACE.

Po dokončení těchto úprav, nastavíme *solver* a kompilátor RTW. K nastavení těchto dvou věcí je zapotřebí otevřít v okně Simulinku záložku Tools a tam zvolit Real-Time Workshop a pak záložku Option. Tím se otevře příslušné dialogové okno.

V této chvíli je již model připraven ke kompilaci. Pro vygenerování C kódu zmáčkneme tlačítko Build nebo klávesovou zkratku ctrl+B. Tímto příkazem RTW provede vygenerování zdrojového kódu v jazyce C, slinkování, zavedení do procesorové karty a spuštění. Tímto končí první etapa Generování Real-Time Application.

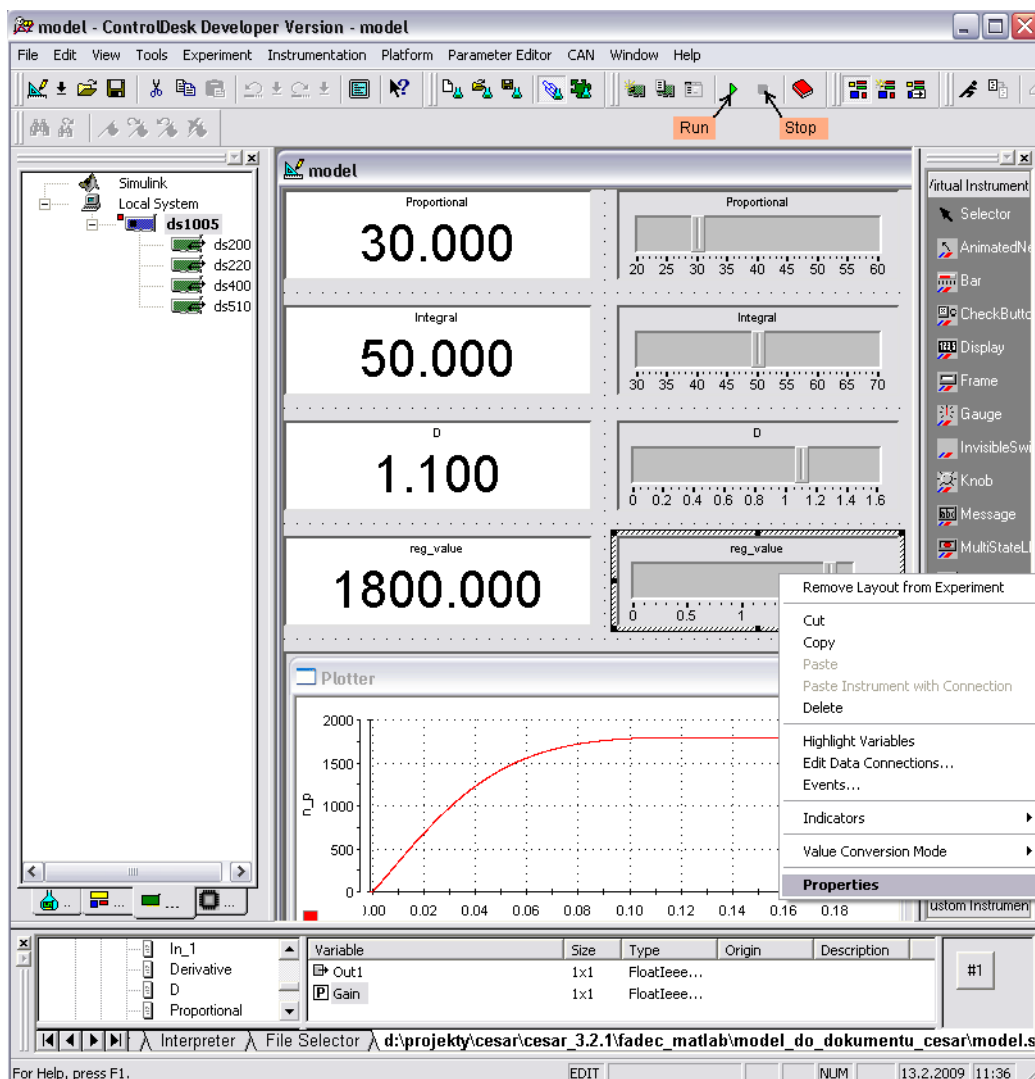
5. ControlDesk

Nyní již model běží v reálném čase na dSPACE a pomocí ControlDesku jej můžeme ovládat, měnit parametry atd. Po spuštění ControlDesk se objeví ovládací okno popsané v předchozí kapitole. Po otevření okna zvolíme záložku File a dáme vytvořit New Experiment.

Zde zadáme název experimentu, zvolíme pracovní adresář a potvrdíme tlačítkem OK. Poté v okně Navigator zvolíme záložku PI se zeleným obrázkem karty. Následně v okně Tool Window najdeme aktuální adresář, v kterém je zkompileovaný náš model a vněm soubor s příponou SDF. Tento soubor uchopíme levým tlačítkem myši a přetáhneme do okna Navigátoru na větev desky procesoru s označením DS1005 a potvrdíme. Tím se nám k tomuto experimentu nalinkuje náš model.

Nyní lze přikročit k tvorbě grafického prostředí tím, že zvolíme záložku File-New-Layout. Po zadání se objeví dvě nová okna a to Working Area a Instruments Selector.

Poté co jsme dokončili základní nastavení můžeme začít nastavovat grafické a ovládací prostředí. Nejdříve v záložkách u okna Tool Window vybereme záložku se jménem aktuálního adresáře. Tím se nám v tomto okně zobrazí struktura našeho modelu s nadefinovanými parametry, které můžeme sledovat nebo měnit. V levém okně Tool Window vybereme složku s názvem Model Root a otevřeme. Po otevření vidíme strukturu modelu s názvy jednotlivých bloků. Jakmile si zvolíme některý blok, tak se nám zobrazí v pravém okně parametry a proměnné, které tento blok má. Abychom mohli tyto parametry měnit nebo sledovat, musíme z okna Instruments Selector nejdříve vybrat a nadefinovat ovládací a měřící prvky viz Obr 8.



Obr. 8 Nastavení ovládacího prostředí

To provedeme tak, že si zvolíme v okně Instruments Selector námi požadovaný prvek a výběr potvrdíme levým tlačítkem myši. Následně přejedeme do okna Working Area a tam, kde chceme tento prvek umístit, zmáčkneme levé tlačítko myši. Tím se nám objeví okno vybraného prvku. Nakonec nastavíme parametry tohoto prvku zmáčknutím pravého tlačítka myši a vybráním kolonky Properties. Zde můžeme podle vybraného prvku nastavovat různé parametry.

Propojení mezi zvoleným ovládacím prvkem a parametrem z modelu provedeme tak, že v okně Tool Window zvolíme parametr nebo proměnnou, kterou chceme ovládat a přetáhneme jí na ovládací nebo měřící prvek viz Obr. 8.

Pro zobrazení zvoleného parametru v ovládacím prvku, musíme zapnout animační mód. To provedeme např. stisknutím klávesy F5. K spuštění nebo zastavení modelu běžícího na hardwaru dSPACE, slouží tlačítko v Toolbars panelu Run a Stop viz obr. 8.

Tímto je ukončen celý proces generování Real-Time aplikace a nyní již můžeme optimalizovat a ladit řídicí parametry.

6. Závěr

Uvedený text krátkou formou seznamuje s technologií vývoje nazývanou *Model Based Design*, která se stává stále více populární ve všech oblastech technického inženýrství. Při zvládnutí nabízených možností simulačního prostředí a systému dSPACE, dochází ke dramatickému snížení vývojového času a zátěže konstruktéra. Nesporně největší výhodou tohoto postupu řešení je snadné promítnutí veškerých změn v zadání na cílový produkt. Pokud probíhají vývojové práce v týmu čítajícím více členů, je navíc systém dSPACE a prostředí Simulink velmi vhodným prostředkem pro vzájemné předávání informací a vzájemnou kooperaci jednotlivých týmů.

Předložený text je pouhým úvodem do této problematiky, která je ze své podstaty značně rozsáhlá. Cílovým projektem jednotlivého zkoušení a seznamování se s tímto postupem řešení, je řízení chodu elektricky komutovaného motoru v bezsenzorovém režimu. Jednotlivé práce probíhají ve spolupráci s firmou UNIS, a.s., sekci Mechatronických systémů, a budou v budoucnu dále rozvíjeny nejen po stránce vývojové, ale i publikační.

7. Poděkování

Vývoj řízení EC motorů v kritických aplikacích je podporován projektem MSM 0021630518 „Simulační modelování mechatronických systémů“ a je řešen na Fakultě strojního inženýrství, Ústavu výrobních strojů, systémů a robotiky, Vysoké učení technické v Brně.

8. Reference

Karban, P. Výpočty a simulace v programech Matlab a Simulink. *První vydání. Computer Press: Praha, 2007. ISBN 978-80-251-1448-3.*

Švéda, M., Opluštil, V., Hubík, V. Design and development of COTS based engine control system for aerospace application. *In Conference Proceedings, SIES 2008.*