

pojitelné k počítači pomocí USB, které jsou vhodné zejména pro mobilní zařízení, a rovněž moduly v průmyslovém provedení komunikující rozhraním RS-485 nebo LAN (standardními protokoly Modbus RTU a Modbus TCP). V závislosti na typu jsou tato zařízení vybavena až šesti kanály SSI, zpravidla doplněnými dalšími I/O funkcemi s možností synchronní softwarové obsluhy.

Společnost se však věnuje i zakázkovému vývoji a výrobě elektroniky, takže skutečný rozsah nabízených řešení je mnohem širší.

Závěr

Komponenty TEDIA patří mezi technicky vyspělé a cenově atraktivní prostředky průmyslové automatizace a v praxi je využívá mno-

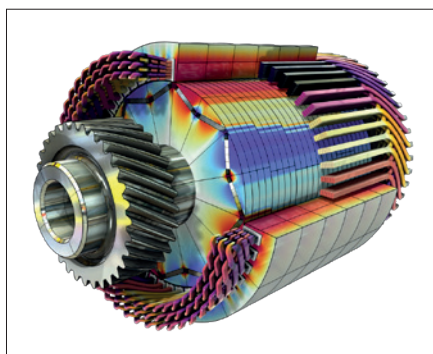
ho významných českých podniků. Kvalita produkce je garantovaná zavedeným systémem ISO 9001, zahrnujícím všechny činnosti firmy od počátečního návrhu přes vývoj a výrobu až po prodej a servis. Technické a obchodní informace o celém sortimentu výrobků lze získat např. na domovské stránce www.tedia.cz.

(TEDIA spol. s r. o.)

Jak se programuje elektrický pohon

Elektrické motory dominují oblasti malých pohonů ve spotřební elektronice a pohonů v průmyslové technice, jako jsou nejrůznější výrobní stroje a zařízení, roboty a manipulátory atd. Mílovými kroky se ale rozšiřují dále do oborů, kde dosud vládly motory spalovací. Jedním z klíčových technických témat současné doby je zejména elektromobilita. Ruku v ruce s moderní technikou pohonů se musí rozvíjet i nástroje nezbytné pro jejich vývoj. Při něm se stále více uplatňují techniky počítačového modelování a simulace, konektivita i umělá inteligence.

Současné moderní elektrické pohony využívají množství elektroniky. Vedle výkonových prvků pro napájení pohonu jde také o mikroprocesory a řídicí systémy s odpovídajícím softwarem, které se starají o jejich chod. Proč? Protože řízený pohon může pracovat efektivněji (obr. 1).



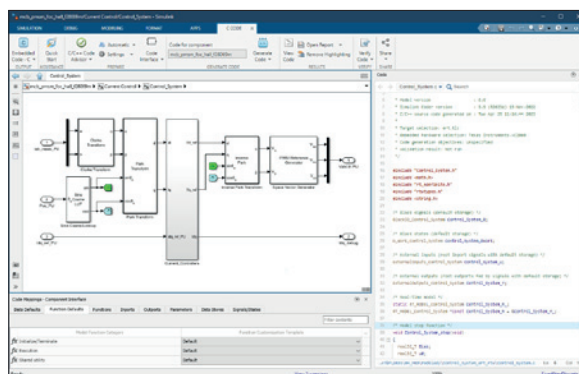
Obr. 1. Současné elektrické motory jsou konstrukčně složité struktury, které k optimální činnosti vyžadují patřičně výkonné řídicí prvky a jednotky

Jak tedy vývojové týmy tyto řídicí prvky a jednotky programují? Základní odlišností oproti klasické představě programování je skutečnost, že programy se „neprogramují“, programy se modelují. Zní to možná zvláštně. Vysvětlíme si, v čem je smysl a přínos takového přístupu.

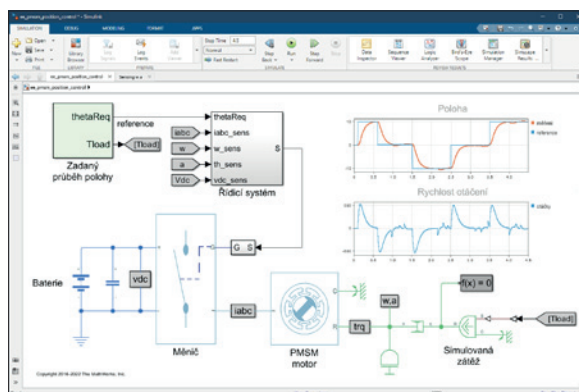
Model programu

Co je to model programu? Model programu je spustitelný matematický popis algoritmu ve vyšší úrovni abstrakce, než jakou poskytuje klasický program. Často má formu

přehledného grafického schématu (obr. 2). Takový model, stejně jako program, obsahuje postup, který bere vstupní údaje a přetvoří je na výstupy. Například může vzít údaje o aktuálním natočení rotoru elektromotoru,



Obr. 2. Model vektorového řízení motoru (Field Oriented Control – FOC) v prostředí Matlab & Simulink a generovaný zdrojový kód (vpravo)



Obr. 3. Ukázkový model pohonného systému v prostředí Simulink

požadavky na rychlost jeho otáčení a potřebnou sílu (točivý moment) a určit chování výkonové elektroniky tak, aby pohon splnil, co se od něj očekává, při zachování ekonomiky provozu a bezpečnosti. Model však, na rozdíl od programu, nemusí obsahovat formální náležitosti programovacího jazyka, jako jsou deklarace funkcí či datových proměnných, a hlavně, nemusí být „ušitý na míru“ konkrétnímu procesoru, který bude nakonec v řídicí elektronice pohonu danou úlohu vykonávat.

Právě abstrakce, tedy oddělení algoritmu od konkrétního hardwaru, je jednou z hlavních předností programování pomocí modelů označovaného *Model-Based Design*. V době, kdy není koncový hardware předem znám nebo dojde k jeho změnám, je pak snadné přenést výslednou funkci z jedné procesorové platformy na jinou, aniž by byly nutné dramatické úpravy modelu.

Co může být obsahem modelu programu? V zásadě cokoli, co do něj vývojář zanesl. Modely mohou zahrnovat jak běžné a léty prověřené algoritmy řízení systémů, tak i moderní nástroje umělé inteligence, jako jsou hluboké neuronové sítě, schopné řešit složité rozhodovací a prediktivní úlohy.

Generování kódu

Jak se pak model programu dostane do procesoru v řídicí elektronice pohonu? Využívá se metoda automatického generování zdrojového kódu, obvykle v jazyce C. V modelu se nastaví, jakým způsobem má být kód generován, lze vybrat i typ cílového procesoru pro využití optimálních funk-

cí a struktury programového kódu, a poté se „na stisknutí tlačítka“ vygeneruje zdrojový kód. Vygenerovaný kód, který lze certifikovat (např. podle ISO 26262), se pak použije přímo v cílovém procesoru.

Vedle zdrojového kódu v jazyce C, který je obvyklý pro mikroprocesory, je možné generovat kód v jazyce HDL (*Hardware Description Language*), určeném pro výkonná programovatelná hradlová pole (FPGA), nebo kód v jazyce CUDA pro grafické výpočetní jednotky.

Ověřování a simulace

Dříve, než je možné z modelu programu (algoritmu) vygenerovat cílový kód, je třeba model podrobně vyzkoušet. K tomu se využívá počítačová simulace. Model se spustí, předávají se mu různé vstupy, odpovídající scénářům z reálného provozu, a sleduje se, jak model reaguje a zda jsou vkládané údaje vyhodnocovány správně. Simulace je výhodná tím, že umožňuje rychle prověřit velké množství situací, aniž je nutné „točit“ reálným pohonem. Oporou jsou přitom další modely – tentokrát matematické modely chování motoru a výkonové elektroniky. Modely pohonů se vytvářejí ve stejných simulačních nástrojích jako modely algoritmů. K dispozici jsou připravené knihovny jednotlivých prvků s různou úrovní detailů, které se použijí v závislosti na tom, zda je středem zájmu celková simulace chování na úrovni systému, nebo nějaký konkrétní dílčí děj. Simulační model pohonu je v podstatě digitálním sourozencem reálného zařízení (obr. 3).

Simulace a ověřování (testování) se nemusí odehrávat pouze ve virtuálním prostoru vývojářova počítače. Při použití specializovaných počítačů s množstvím vstupních a výstupních rozhraní, tzv. simulátorů, je možné

simulovat chování pohonu v reálném čase. K simulátoru se připojí reálný naprogramovaný procesor či řídicí jednotka, vyzkouší se jeho výsledné chování, jako by byl připojen ke skutečnému motoru – jenomže není. V simulátoru lze snadno vytvářet extrémní scénáře a ověřit při nich v podstatě celou elektroniku vyvíjeného pohonu, aniž by hrozilo reálné nebezpečí poškození, či dokonce zničení nákladného zařízení (obr. 4). Do skutečného

ré je výhodné opětovně využít, nebo si vývojář prostě přeje danou část kódu vytvořit ručně. Modely podporují integraci existujících částí kódu, takže je možné model s ručně psaným kódem zkombinovat. Výsledný vygenerovaný kód pak obsahuje tyto ručně psané části jako součást celého programu. Nejde tedy o volbu mezi programováním a modelováním, ale o vhodném skloubení obou přístupů.



Obr. 4. Simulátory dSpace pro ověřování řídicích programů pohonů v reálném čase

pohonu se tak dostane již odladěný, detailně prozkoušený řídicí program.

A co klasické programování

Vývoj moderních pohonů je proces, při kterém je často třeba skloubit různá zařízení a postupy. Nejinak je tomu i v případě programování jejich elektroniky. Proto je nezbytné mít možnost kombinovat modelování programů s jejich klasickou tvorbou, ručním vytvářením programů v jazyce C. Může jít o programy vytvořené v minulosti, kte-

ré je výhodné opětovně využít, nebo si vývojář prostě přeje danou část kódu vytvořit ručně. Modely podporují integraci existujících částí kódu, takže je možné model s ručně psaným kódem zkombinovat. Výsledný vygenerovaný kód pak obsahuje tyto ručně psané části jako součást celého programu. Nejde tedy o volbu mezi programováním a modelováním, ale o vhodném skloubení obou přístupů.

Závěr

V článku jsou stručně přiblíženy postupy, jakými se v současnosti „programují“ elektrické pohony, jak se vytváří software pro jejich řízení a jak se vše ověřuje před reálným použitím. Stejný přístup k programování se uplatňuje i v mnoha dalších oborech. Může jít o systémy pro správu akumu-

látorových baterií, asistenční systémy řidiče automobilu, leteckou techniku nebo jakékoli jiné odvětví, kde jsou metody modelování a simulace využívány ke zvýšení kvality a spolehlivosti výsledného zařízení. Zájemci o tyto pokročilé vývojové metody mohou najít další informace na webových stránkách <https://www.humusoft.cz/matlab/simulink/a> <https://www.mathworks.com/solutions/electrification/motor-drives-traction-motors.html>.

(Humusoft)

MATLAB® & SIMULINK®

NOVÁ VERZE R2023b

POLYSPACE TEST **SIMULINK FAULT ANALYZER**

NATIVNÍ PODPORA PRO APPLE SILICON **LIVE EDITOR – TASK PRO PIVOT TABLE, GENEROVÁNÍ ULASTNÍHO TASKU A EXPORT DO JUPYTER NOTEBOOKU**

PŘEDÁVEJTE DATA MEZI EXCELEM A SIGNAL EDITOREM U
SIMULINKU ZKOPÍROVÁNÍM A ULOŽENÍM

<https://www.humusoft.cz/matlab/new>

Humusoft s.r.o.
Pobřežní 20, Praha 8

E-mail: info@humusoft.cz
Tel: +420 284 011 730