

# „MATLAB Simulink Accelerator“

*Petr PFEIFER*

Katedra měření, Fakulta elektrotechnická,  
České vysoké učení v Praze, Technická 2, 166 27 Praha 6

*Abstrakt:* Příspěvek představuje vyvinutou aplikaci, která umožňuje zvýšit rychlost simulací určitého druhu v systému MATLAB Simulink až o několik řádů. Těchto výsledků se docílí převodem některých velmi výpočetně náročných úloh z oblastí simulací především komunikací do vyvinutého rychlého simulátoru datových komunikací. Pomocí toolboxu pro univerzální výměnu dat je zprostředkována pouze výměna vstupních dat, parametrů simulace a dílčích výsledků simulace.

## 1. Úvod

Již před několika lety jsme na naší katedře začali s větším a systematickým převodem jednotlivých částí různých přenosových kanálů a datových komunikací do simulačního nástroje. Jako nejvýhodnější se ukázalo použití MATLABu a jeho simulačního nástroje Simulinku. MATLAB si s jednoduchostí sobě vlastní „poradil“ s grafickou vizualizací dat, jeho používání je velmi jednoduché a intuitivní. Bohužel v případě simulací již nepodával dostačující výkony, při požadavcích na použití větších datových souborů pro výsledky simulací a simulační protokoly jeho simulační výkon nadále výrazně klesal. Ani nejnovější akcelerátory, použité například v MATLABu 6.1, apod. ve formě optimalizovaných prekompilací nepřinesly požadované zvýšení simulačního výkonu celého simulátoru. Proto bylo započato s vývojem externího simulátoru, který by umožnil zvýšit simulační výkon celého systému alespoň v některých kritických částech.

## 2. Princip použití akcelerátoru

Vyvinutý akcelerátor se skládá ze dvou základních částí. První částí je konvertor, především zdrojových souborů původně z MATLABu do formy, vhodné k použití v další části, kterou je rychlý simulátor. Tento konvertor provádí zároveň prvotní optimalizaci. Rychlý simulátor byl vyvinut především pro účely simulací datových komunikací, kde dosahuje velmi vysokých simulačních výkonů oproti použití jiných i komerčně dodávaných systémů, ovšem při zachování velké modularity celého systému, použití knihoven apod. Znamená to tedy, že můžeme dosáhnout velmi pěkných výsledků ve velmi krátké době i při použití méně výkonného počítače.

Tajemství rychlého simulátoru spočívá především ve vysoké optimalizaci jádra simulátoru, dále v optimalizaci paměti i průběžné optimalizaci, která je prováděna na cílovém kódu již za běhu programu. I tento proces má ale svá omezení, proto nemusí být použití rychlého simulátoru vždy tak o tolik výhodnější, přestože tomu v naprosté většině případů je.

Využití rychlého simulátoru s jinými systémy lze zobecnit, ačkoliv je využíván především ve spojení s MATLABem. Vlastní spojení akcelerátoru s původním systémem může probíhat standardními i nestandardními prostředky operačního systému. Může být využito i služeb Univerzálního Datového Serveru, který je podrobněji popsán v dalším příspěvku této konference MATLAB2002. V případě jeho použití v součinnosti

s MATLABem lze využít již vyvinutého PDES Toolboxu a začít ihned využívat všech výhod nového systému.

### **3. Způsoby akcelerace jednotlivých typů souborů**

Jak již bylo uvedeno, vlastní simulátor díky své struktuře nemusí nutně dokázat efektivně využívat programy, napsané na jiných platformách. Pravděpodobně by zde byla jistá možnost doplnit optimalizační algoritmy i o takovouto schopnost, otázkou ale je, zda bude taková přímá optimalizace dostatečně efektivní, především z hlediska vynaložené práce, nebo dokonce vůbec možná (ať z jakéhokoliv důvodu, včetně důvodů licencí, ochranných známek nebo velmi špatné dokumentaci formátů souborů nebo funkcí). Další otázkou zůstává, zda nezabere tato přímá optimalizace více času, než simulace vlastní, resp. simulace v původním systému. Proto bylo použito systému, kdy se nejdříve provádí jistý převod původního formátu souboru do předkompilovaného souboru nového systému, přičemž se již zde provádí prvotní optimalizace. Tyto soubory ve standardizovaném formátu lze ale i nadále upravovat a poté již přímo použít v rychlém simulátoru.

V případě MATLABu přichází aplikace konvertoru v úvahu na níže uvedené formáty souborů.

#### **3.1 Akcelerace m-souborů**

Akcelerace m-souborů byla prvotní snahou celého projektu. Tyto soubory lze velmi jednoduše zpracovávat a i optimalizace může probíhat zcela jednoduše v režii optimalizátoru.

Pro převod m-souborů byl vytvořen program, který přeloží textový soubor tohoto formátu do formátu optimalizovaného pro rychlý simulátor (jistá obdoba kompilace). Bohužel tento program zatím není nikterak dokonalý, některé optimalizace může provádět nad rámec „použitelnosti“. Proto je často nutný konečný zásah programátora, resp. minimálně revize překladu a konečné úpravy z hlediska optimálních typů proměnných, apod. Další problémy mohou vznikat při využití některých knihoven MATLABu. Běžné jednoduché autonomní úlohy je však program schopen konvertovat takřka bez problémů.

#### **3.2 Akcelerace C knihoven**

Nové verze MATLABu dovolují velmi efektivně pracovat i s jazykem C. Převod těchto částí do nového simulátoru je minimálně na úrovni použití m-souborů, ale zpravidla prakticky bezproblémový. Příslušné primární optimalizace provádí překladač sám, další optimalizace se dosahuje v simulátoru samotném. Největšího stupně akcelerace simulací se dosahuje především díky optimalizovanému jádru simulátoru a sekundárním optimalizacím, vlastní optimalizace v cílovém kódu oproti MATLABu nemusí být již tak významné. Nicméně i při použití těchto knihoven existují určitá omezení, především v případě volání dalších DLL knihoven, které již nejsou dostupné ve své zdrojové podobě.

#### **3.3 Akcelerace DLL knihoven**

DLL knihovny lze v zásadě do nového simulačního jádra převádět jednodušeji. Pracuje se zde přímo s jednotlivými instrukcemi, resp. bloky instrukcí procesoru, které lze již považovat za částečně optimalizované. Zde ale nastává druhý velký problém, který není zatím vhodně vyřešen, a tím je volání dalších částí, původně MATLABu, mimo vlastní DLL

knihovnu. V případě, kdy daný soubor využívá mnoho dalších souborů, je sestaven z nestandardních částí nebo je použito nestandardních postupů, není vlastní převod do nového simulátoru příliš efektivní, neboť je nutné násilně ukončovat přidělený proces a přenechat další (částečný) výpočet další úloze nebo dokonce MATLABu samotnému. K akceleraci simulací pak dochází velmi málo nebo vůbec.

### 3.4 Shrnutí jednotlivých způsobů akcelerace

Jak plyne z výše uvedených možností, nejjednodušším a nejúčinnějším způsobem využití akcelerátoru je použití již zdrojových kódů nejlépe v m-souborech nebo jazyce C.

## 4. Závěr

Popisovaná aplikace umožňuje zvýšit rychlost simulací určitého druhu v systému MATLAB Simulink až o několik řádů. Těchto výsledků se docílí převodem některých především výpočetně náročných úloh z MATLABu do vyvinutého rychlého simulátoru datových komunikací. V praxi bylo dosaženo zrychlení simulací až o tři řády při použití stejného m-souboru v MATLABu a konvertoru do rychlého simulátoru. Pomocí toolboxu pro univerzální výměnu dat může být zprostředkována již pouze výměna vstupních dat, parametrů simulace a dílčích nebo konečných výsledků simulace.

Popisované programové vybavení je již delší dobu úspěšně používáno. Jádro simulátoru existuje ve verzi pro DOS i pro Windows. Windows verze může být u některých simulací ve výjimečných případech nepatrně pomalejší, ale její předností je možnost použití velkého množství proměnných, procesů, velkých datových polí a jisté „pohodlí“ při vnitřní správě celého tohoto systému. Další výraznou předností je uživatelsky velmi příjemné rozhraní, které umožňuje detailní sledování celých simulací včetně struktur procesů, proměnných, modulů, apod. Toto grafické uživatelské rozhraní bylo napsáno s využitím vývojového prostředí DELPHI firmy Borland.

### *Poděkování*

Tento výsledek byl získán za finančního přispění Ministerstva školství, mládeže a tělovýchovy v rámci podpory projektu výzkumu a vývoje LN00B073.

## 5. Literatura

- [1] Internet (<http://www.mathworks.com/>)
- [2] Pfeifer, Petr: “Fast Simulation System for Simulation and Improvement of Digital Communication Systems in Transportation“, SCI2002, Orlando, Florida, USA,2002

---

**Ing. Petr PFEIFER**

Czech Technical University in Prague, Faculty of Electrical Engineering,  
Department of Measurement, Technická 2, 166 27 Praha 6, Czech Republic  
*Telefon: (420) 2 2435 2807, e-mail: xpfeifer@fel.cvut.cz*