

COMMUNICATION MATLAB–ETHERNET

Vratislav Kořínek, Master Degree Programme (5)
Dept. of Control and Instrumentation, FEEC, BUT
E-mail: ante@ante.cz

Supervised by: Prof. Petr Pivoňka

ABSTRACT

This paper shows how to implement control algorithms in real-time from simulation environment MATLAB-Simulink into programmable logic controller B&R 2005.

1 ÚVOD

Při implementaci řídicích algoritmů ze simulačního prostředí do průmyslových regulátorů (PLC) je potřebné efektivně minimalizovat množství implementačních chyb a čas, potřebný pro implementaci algoritmů. Dále je vhodné dodržovat přenositelnost vyvíjených řídicích algoritmů mezi simulačním prostředím a průmyslovým regulátorem.

Obecně lze přímou implementaci řídicích algoritmů rozdělit do tří následujících fází:

- Vytvoření a ověření algoritmu na matematickém modelu v simulačním prostředí.
- Odladění algoritmu v simulačním prostředí propojeném se skutečným fyzikálním modelem soustavy
- Implementace algoritmu do programovatelného automatu, simulační prostředí slouží pro monitorování regulačního děje a nastavování žádané hodnoty či jiných potřebných parametrů.

Z druhé a třetí fáze vyplývá, že musí existovat klient umožňující komunikaci mezi PLC a MATLABem.

2 SOFTWAREVÉ PROSTŘEDKY - MATLAB-SIMULINK A PVI

Prostředí simulačního programu MATLAB-Simulink obsahuje nástroj na tvorbu vlastních uživatelských bloků. Tyto uživatelské bloky mají přesně danou strukturu a nazývají se S-funkce. Pro tvorbu S-funkcí lze použít tyto programovací jazyky: MATLAB® (m-files), C, C++, Fortran, nebo Ada. V našem případě je vhodné z důvodu zachování přenositelnosti implementovat řídicí algoritmy v jazyce ANSI C, který podporuje nejen MATLAB-Simulink, ale právě i samotný programovatelný automat B&R 2005. S-funkce napsané v jazyce ANSI C lze pomocí interního nebo externího překladače snadno zkompileovat do Microsoft Windows

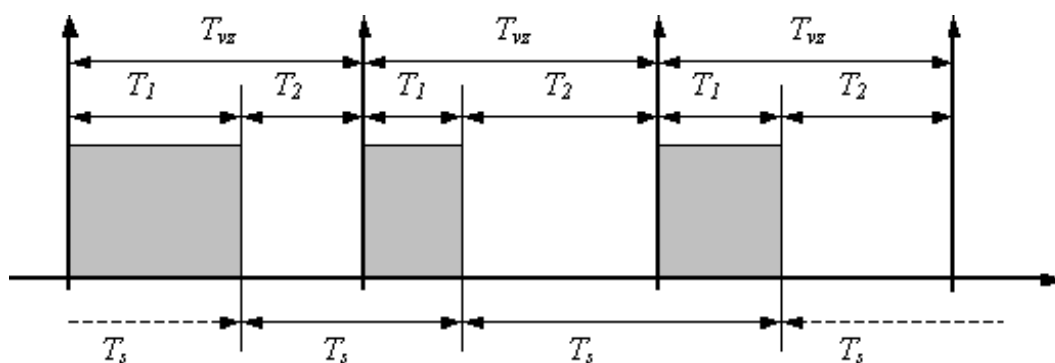
32-bitové dynamické knihovny (DLL). Takto vzniklé DLL jsou pak při prvním spuštění nahrány do paměti a tím odpadá interpretace programového kódu a jeho překlad do spustitelného programu, čímž dochází ke značnému urychlení běhu simulace.

Program PVI (Process Visualization Interface) zastřešuje různé komunikační rozhraní s jejich protokoly a uživatel nemusí znát specifikace sériové linky, CANu, Ethernetu, atd., jelikož komunikaci na nižší úrovni zajišťuje PVI. Uživatel použitím PVI tak získá standardizovaný přístup k různým zařízením a přenosovým protokolům s plnou podporou zpracování kontrolních a řídicích dat (proměnné, status, atd.). Knihovny PVI umožňují za pomoci standardních programovacích nástrojů (MS Visual C++, Borland C++, Delphi, gcc) vytvářet vlastní nástroje pro komunikaci s automatem – tzv. klienty. Díky tomu lze urychlit a zjednodušit vývoj uživatelských aplikací pro vizualizaci a aplikací určených k nastavení nebo řízení programovatelného automatu.

3 KOMUNIKACE V REÁLNÉM ČASE MEZI MATLAB-SIMULINKEM A PLC

Příspěvek řeší komunikaci po Ethernetu a sériové lince RS232 použitím rozhraní PVI mezi MATLAB-Simulinkem a PLC. Pro uskutečnění spojení byl navržen vlastní postup synchronizace běhu MATLAB-Simulinku s reálným časem. Vlastní řešení bylo zvoleno zejména z důvodu nedostatečné robustnosti a složitosti implementace pomocí toolboxů Real-Time WorkShop nebo Real-Time Windows Target.

Komunikace v reálném čase mezi MATLAB-Simulinkem a PLC musí dodržet průběh zobrazený na obr. 1.



Obr. 1: Časový průběh komunikace v reálném čase.

Čas T_1 je tzv. reakční doba a skládá se ze tří časových úseků: času potřebného pro vyčtení výstupu soustavy, času pro vypočítání akčního zásahu a času potřebného pro zápis akčního zásahu. Pro třetí fázi implementace řídicího algoritmu odpadá čas nutný pro výpočet akčního zásahu. Čas T_2 je čas po kterou je S-funkce v nečinnosti. T_s je celkový čas strávený v S-funkci. Je zřejmé, že vyjma periody vzorkování T_{vz} délku ostatních časů neznáme. Celá synchronizace je řízena hodinovým taktém CPU z PC a je založena na deaktivaci procesu na zadanou dobu a na znovu aktivaci tohoto procesu v potřebný okamžik operačním systémem.

Z výše uvedeného postupu je patrné, že jde o soft-realtimové řešení. Výhodou je, že nedodržení vzorkovací periody nezpůsobí zhroucení operačního systému a soft-realtimové řešení je pro potřeby testování řídicích algoritmů dostačující.

Noví komunikační klienti pro druhou a třetí fázi využívají výše popsanou realtimovú

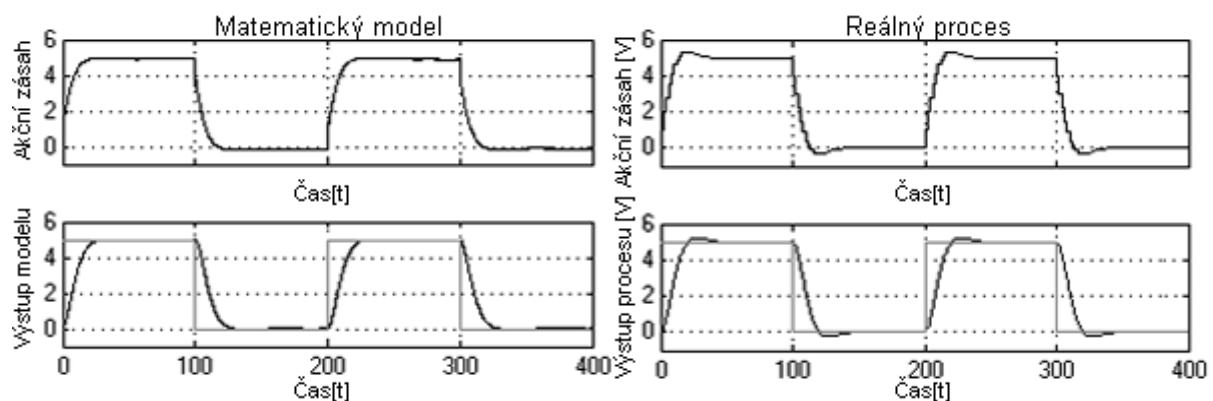
komunikaci a všechny potřebné funkce pro spojení, které poskytuje samotné rozhraní PVI.

4 EXPERIMENT S PŘÍMOU IMPLEMENTACÍ PSD REGULÁTORU

Pro experiment ve školní laboratoři byl použit polohový PSD regulátor a lokální síť typu Ethernetu a byl regulován fyzikální model soustavy s přenosovou funkcí:

$$F(s) = \frac{1}{(3s+1)(2s+1)}$$

Při experimentu byla zjištěna běžná kritická doba reakce $T_r = 200$ ms. Časový průběh regulace s modelem soustavy a se skutečným fyzikálním modelem je zobrazen na obr. 2.



Obr. 2: *Regulace matematického modelu soustavy a reálného procesu pomocí programu MATLAB-Simulink polohovým PSD regulátorem.*

5 ZÁVĚR

Článek ukázal řešení přímé implementace řídicích algoritmů ze simulačního prostředí do programovatelného automatu. Předkládané řešení efektivně zkracuje dobu nutnou pro implementaci řídicích algoritmů. Vyvinutí klienti jsou snadno použitelní, jsou taktéž snadno srozumitelní pro uživatele, robustní a vhodné především pro ověřování a testování složitých řídicích algoritmů pro řízení technologických procesů.

LITERATURA

- [1] The MathWorks Inc. Documentation. MATLAB 6.1, Simulink, 2001, <http://www.mathworks.com>
- [2] B&R Documentation. Automation Net/PVI Online Help, AS240Ee, 1997-2003, <http://www.br-automation.com>
- [3] Kořínek, V.: Komunikace MATLAB-Ethernet. Semestrální práce 2, ÚAMT FEKT VUT, Brno, 2003.
- [4] Švancara, K., Pivoňka, P.: Přímá implementace řídicích algoritmů z prostředí MATLAB/Simulink do PLC. Automatizace, 46 (2003), č. 12, s. 790-793.