

# Vložené řídicí systémy Vzdálené a virtuální laboratoře

Pavel Balda  
ZČU v Plzni, FAV, KKY

## Osnova přednášky

- n Struktury virtuálních laboratoří
- n Struktury vzdálených laboratoří
- n Vzdálená a virtuální laboratoř na KKY
- n Porovnání přístupů pro budování laboratoří
- n Otevřené problémy

2

## Úvod

- n Oblast automatického řízení lze podle Fleminga (1989) charakterizovat následovně:
  - n Automatické řízení je ze své podstaty interdisciplinárním oborem
  - n Matematika hrála a v budoucnu bude ještě více hrát zásadní roli v rozvoji automatického řízení
  - n Od samého začátku je vztah mezi matematikou a automatickým řízením vztahem vzájemného „proplétání“
  - n Pokroků v oblasti automatického řízení se dosahuje prostřednictvím směsi matematiky, modelování, výpočetní techniky a experimentování
- n Nové technologie ve výpočetní technice podstatně rozšiřují možnosti praktických aplikací z automatického řízení
  - n Inteligentní čidla a akční členy, vložené řízení, pokročilé řídicí algoritmy, průmyslové komunikace, internetové řídicí systémy
- n Výjimečné postavení internetu má vliv nejen na výuku automatického řízení ale umožňuje i **experimentování na dálku**

3

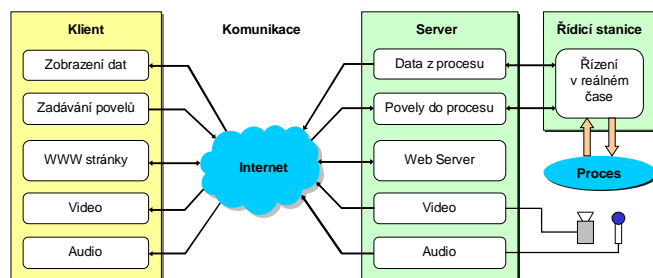
## Co je vzdálená a virtuální laboratoř?

- n **Vzdálená laboratoř** (remote laboratory)
  - n Uživatel není fyzicky přítomen, ale je ve spojení s jejím zařízením přes komunikaci, nejčastěji Internet
- n **Virtuální laboratoř** (virtual laboratory)
  - n Experimenty se neprovádějí na skutečném zařízení, ale na simulačním modelu (virtuálním zařízení)
- n Vytvoření kvalitní vzdálené nebo virtuální laboratoře je komplexní interdisciplinární úkol. Lze jej rozdělit na dvě oblasti:
  - n Didaktická oblast – metodika výuky spojená s prudkým rozvojem technologií Internetu (e-Learning)
  - n Technická oblast – formulace technických úloh a jejich řešení při návrhu laboratoří

Přístup k zařízení	Charakter zařízení	
	Skutečné	Simulované
Místní	Skutečná laboratoř	Jednouživatelská virtuální laboratoř
Vzdálený	Vzdálená laboratoř	Víceživatelská virtuální laboratoř

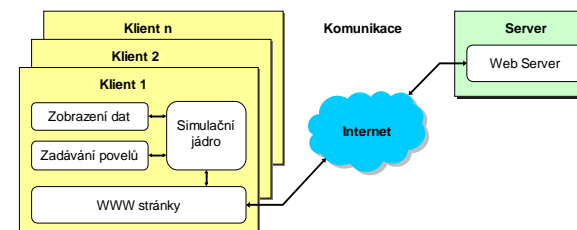
4

## Obecná struktura vzdálených laboratoří



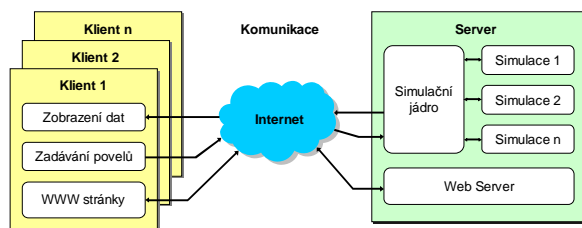
5

## Obecná struktura virtuálních laboratoří s lokálním simulačním jádrem



6

## Obecná struktura virtuálních laboratoří se vzdáleným simulačním jádrem



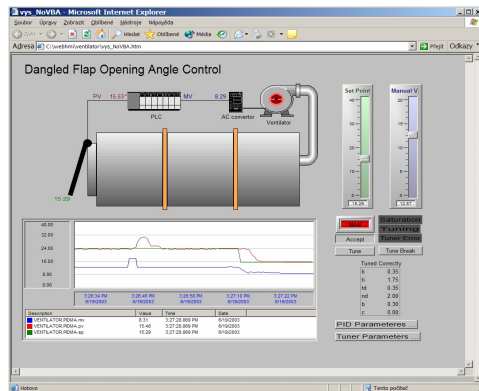
7

## Používané softwarové technologie

- n **Operační systémy reálného času** – Windows CE, Phar Lap ETS, RTX, Hyperkernel, RTLinux, QNX, VxWorks
- n **Komunikační protokoly** – TCP/IP, UDP/IP, FTP, HTTP, RTP, RPC, DCOM
- n **Softwarové komponenty** – DLL, COM, OLE, ActiveX, OPC (DA, A&E, HDA, Batch)
- n **Skripty pro klienty** – VBScript, JScript, Java applety
- n **Skripty pro servery** – CGI, Perl, PHP, ASP, ASP.NET

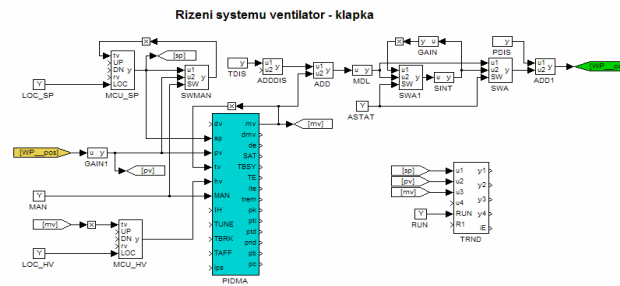
8

## Příklad: řízení systému ventilátor - klapka



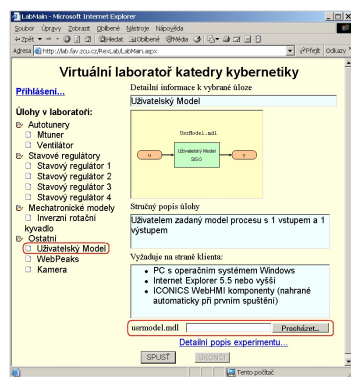
9

## Regulační schéma příkladu



10

## Příklad otevřeného experimentu (1/2)



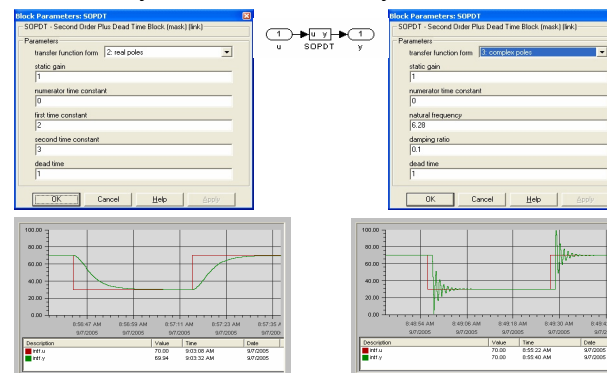
### Příklad Uživatelský model

- n nejjednodušším příkladem úlohy, ve které lze poslat řídicí algoritmus na server
- n Soubor **usrmodel.mdl** se zařadí do konfigurace a spolu s dalšími soubory se přeloží pro ŘS REX
- n Případné chyby překladu jsou vypsané uživateli přímo do horní části obrazovky.
- n Je-li překlad bez chyby, úloha se spustí
- n Tímto způsobem lze připravovat úlohy typu „pro daný model navrhněte řídicí systém“. Model může být simulační i reálný.

11

## Příklad otevřeného experimentu (2/2)

### Model systému druhého řádu realizovaný blokem SOPDT:



12



## Ad hoc vs. koncepční řešení

- n Ad hoc řešení neuvažují širší souvislosti
  - n Nedoporučeno
- n **Koncepční řešení**
  - n Systematicky do návrhu zahrnují společné rysy řešených úloh
  - n Snadné zařazování obdobných úloh – otevřenost
- n Dva typy koncepčních řešení
  - n Řešení založená na standardním firemním řešení
    - n Nejčastěji založená na systémech Labview, Matlab-Simulink, Matlab Web Server, Real-Time Workshop, ...
    - n Malé riziko nefunkčnosti, avšak nemusí poskytovat všechny žádané vlastnosti
  - n Řešení založená na vlastním výzkumu autorů
    - n Podstatně větší úsilí k dosažení dobrých výsledků
    - n Často založené na jazyku Java na serveru a Java appletech na klientech
    - n Nevýhodou je omezená kapacita autorů

13



## Offline vs. Online interaktivita

- n **Offline (batch) úlohy**
  - n Klient odešle na server např. název a parametry úlohy, server danou úlohu s těmito parametry spustí a výsledky odešle klientovi formou HTML stránky
  - n Takto funguje např. Matlab Web Server
  - n Výhoda snadná implementace, komunikace požadavek (request) – odpověď (reply) je zabudována do HTTP protokolu
  - n Zásadní nevýhoda – uživatel nemůže do probíhajícího experimentu zasahovat
- n **Online úlohy**
  - n Online interaktivita je zásadní rys určující iluzi přítomnosti v laboratoři

14



## Omezené vs. otevřené experimenty

- n Experimenty s omezenou interakcí
  - n Změna jednoho či několika parametrů
  - n Doporučeno pro počáteční fáze seznamování se s daným modelem (řízeným systémem)
- n **Otevřené experimenty**
  - n Mnohem větší rozsah možností
  - n Tvorba vlastních řídicích systémů nebo uživatelského rozhraní
  - n Zajímavější práce, větší motivace
  - n Více odkrývá podstatu skryté technologie, blíží se návrhu reálných řídicích systémů
  - n Příprava experimentů je však náročnější, nutno zabránit zničení modelu neodborným zásahem

15



## Simulace vs. řízení v reálném čase

- n Pro daný fyzikální model je vhodné mít i odpovídající simulační úlohy
- n U otevřených úloh je důležité, aby návrh algoritmů pro simulaci i řízení probíhal jednotným způsobem.
  - n Není cílem učit studenty několik různých návrhových prostředí, ale naučit je myslet v této oblasti
- n Nesplnění předchozí podmínky vede k nevýhodám:
  - n Pracný přechod od simulace k řízení v reálném čase a zpět
  - n Proto se provádějí nepromyšlené experimenty přímo na fyzikálním modelu
  - n To vede k většímu zatížení modelu a blokování dalších uživatelů
- n V naší laboratoři:
  - n Pro simulaci zvoleno známé prostředí Matlab-Simulink
  - n Pro řízení v reálném čase zvolen systém REX kompatibilní s Matlab-Simulink

16

## Měkký vs. pevný reálný čas

- Operacní systémy s měkkým reálným časem (soft real-time)
  - Nedosahují příliš dobré přesnosti periody vzorkování
  - Windows 2000/XP, Linux
  - Problematická řešení spouštění řídicích úloh z Matlabu pomocí ovladačů vstupně-výstupních karet
- OS s pevným reálným časem (hard real-time)
  - Malá tolerance periody vzorkování, vysoká opakovatelnost měření
  - VxWorks, RTLinux, Phar Lap ETS, Windows CE (od verze 3), RTX pro Windows
  - Pro řízení rychlých experimentů jsou třeba vždy
  - REX lze provozovat ve Windows CE 4.x a Phar Lap ETS

17

## Tenký vs. tlustý klient

- Tloušťka klienta závisí na množství programového vybavení nezbytného pro jeho provoz nad rámec internetového prohlížeče
- Tenký klient
  - Využívá pouze internetový prohlížeč
  - Funguje ve všech operačních systémech a běžných prohlížečích
  - Využívá průnik internetových technologií – HTML a Java applety
  - Nevýhoda – nutnost znalosti programování v Java, velká pracnost
- Tlustý klient
  - Bud' se instaluje speciální aplikace nebo zásuvné moduly (komponenty ActiveX) do prohlížeče
  - Nevýhody - bezpečnostní riziko (možnost zanesení virů), běh jen v Internet Exploreru
  - Největší výhoda – snadné vytváření kvalitní vizualizace v dlouho vyvíjených vizualizačních systémech (Labview, Genesis 32, InTouch, ...)

18

## Architektura laboratoře

- Otevřená architektura ve 3 směrech
  - Možnost přidávat další rysy (programy a komponenty)
  - Snadné přidávání dalších úloh (experimentů)
  - Snadné přidávání dalších řídicích stanic (PC, PLC)
- Důraz na využití průmyslových standardů a technologií
- Výhody využití řídicího systému REX
  - Knihovna funkčních bloků RexLib vhodných pro průmyslové aplikace
  - Možnost přenosu na libovolnou otevřenou platformu s překladčem C/C++
  - Optimalizace systému pro diskrétní (nebo diskretizované bloky)
  - Možnost provozování v operačních systémech pevného reálného času s velmi krátkou periodou vzorkování (i pod 1 ms)

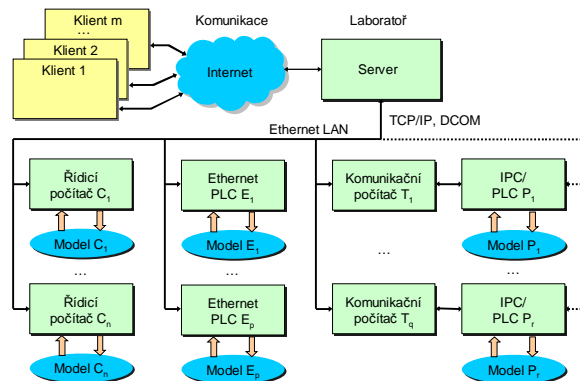
19

## Základní vlastnosti laboratoře

- Klíčové rysy navrženého řešení:
  - Otevřenost laboratoře (modularita) – snadné přidávání dalších úloh, fyzikálních modelů i řídicích stanic (PC, PLC)
  - Komplexnost experimentů – možnost vytváření více úloh pro každý model (simulační i reálný)
    - Jednoduché úlohy, umožňující specifikaci několika parametrů
    - Úlohy umožňující návrh a posílání na server (upload) vlastních řídicích algoritmů a popř. i vizualizačních snímků
  - Kvalitní simulační a výpočetní systém – Matlab-Simulink volaný pomocí Matlab Web Serveru
  - Řízení v pevném reálném čase (hard real-time) – využití řídicího systému REX (kompatibilního s Matlab-Simulink) na OS Windows CE a Phar Lap ETS
  - Použití průmyslových standardů – Microsoft Server 2003, IIS, Visual Studio .NET 2003, komunikační protokoly HTTP, TCP/IP, OPC, atd.

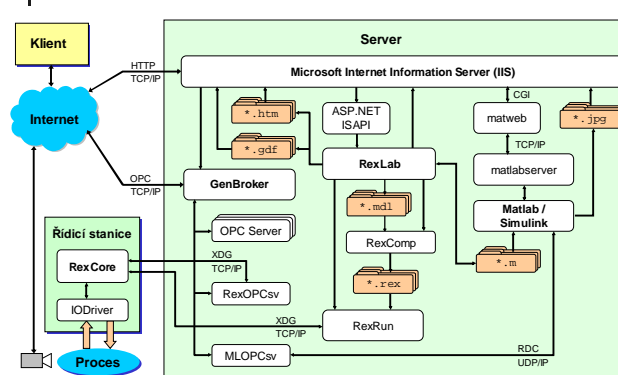
20

## Struktura technických prostředků



21

## Architektura programového vybavení



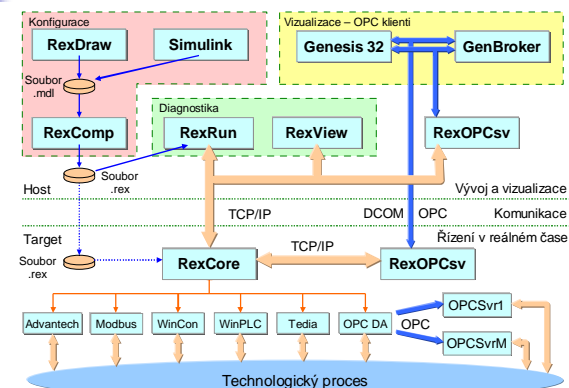
22

## Základní vlastnosti systému REX

- n Kompatibilita se systémem Matlab-Simulink
- n Bohatý sortiment funkčních bloků i otevřená architektura (možnost tvorby vlastních algoritmů a ovladačů)
- n Přenositelnost na různé platformy a operační systémy (Windows, Windows CE, PharLap ETS, apod.)
- n Podpora průmyslových standardů (TCP/IP, OPC).

23

## Integrace s řídicím systémem REX



24

## Serverová aplikace RexLab

- n **RexLab** – internetová aplikace rozšiřující IIS vytvořená v nejnovější technologii ASP.NET firmy Microsoft. RexLab provádí:
  - n Generuje hlavní stránku laboratoře ve formátu **.aspx**
  - n Ve spolupráci s IIS umožňuje přihlašování se do laboratoře
  - n Kontroluje a ukládá na server konfigurace řídicích systémů a vizualizace zaslané klienty
  - n Komunikuje s ostatními aplikacemi potřebnými pro provoz laboratoře
  - n Umožňuje spouštět a ukončovat zvolené úlohy a experimenty uživatelem
  - n Řídí omezení počtu současně přistupujících klientů k jednotlivým experimentům a omezuje maximální dobu připojení klienta
  - n Poskytuje odkazy na dokumentaci k úlohám
- n **Úlohy** spouštěné aplikací RexLab **jsou uloženy v databázi** s položkami:
  - n ID, Name/Jmeno, Description/Popis, ManualRef/ManualOdkaz, VisualRef/VisualOdkaz, Requires/Vyzaduje, Folder, Figure, ExpType, MaxUsers, Computer, RexCfgName, AnonymousAccess, UpldEnable, UpldFiles, Resources

25

## Ukázka příkladů z laboratoře

- n Adresa: **lab.fav.zcu.cz**
- n **Kamera**
  - n Internetová kamera umístěná na soustavě dvou motorků, které ji otáčejí ve dvou osách
- n **Inverzní rotační kyvadlo**
  - n Řízení rychlého modelu s periodou vzorkování 1 ms
  - n Vyhoupnutí kyvadla
  - n Stabilizace v obou rovnovážných polohách
  - n Demonstrační režim – přejezdy ramene do různých poloh při probíhající stabilizaci kyvadla

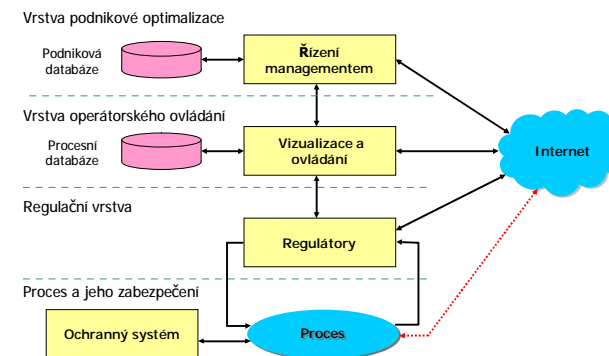
26

## Některé otevřené problémy

- n Určení vhodné šířky pásma přenášených signálů
- n Uzavírání regulačních smyček přes Internet (časově variantní dopravní zpoždění až řízení v otevřené smyčce)

27

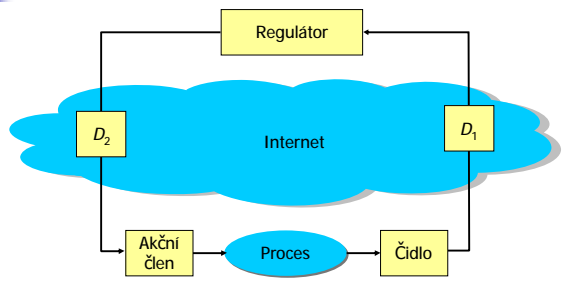
## Průmyslový řídicí systém v Internetu



28



## Regulační smyčky uzavřené přes Internet



Dopravní zpoždění  $D_1 + D_2$  způsobené komunikací v Internetu je časově variantní!