

VŠB - Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení



Návrh a realizace virtuálních laboratorních modelů

Design and Realisation of Virtual Laboratory Models

Student:

Bc. Jiří Híkl

Vedoucí diplomové práce:

doc. Ing. Renata Wagnerová, Ph.D.

Ostrava 2010

Zadání diplomové práce

Student: **Bc. Jiří Hikl**

Studijní program: N2301 Strojní inženýrství

Studijní obor: 3902T004 Automatické řízení a inženýrská informatika

Téma: **Návrh a realizace virtuálních laboratorních modelů**
Design and Realisation of Virtual Laboratory Models

Zásady pro vypracování:

1. Seznamte se s realizovanými virtuálními laboratořemi na vysokých školách, zaměřte se na použité softwarové nástroje.
2. Popište vlastnosti a funkce vybraných toolboxů programového systému Matlab (Matlab Compiler, Matlab Builder for .NET, Matlab report generátor, Virtual Reality).
3. Vytvořte virtuální model vybraného technologického procesu.
4. Popište možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet.
5. Zhodnoťte dosažené výsledky a navrhnete směr dalšího řešení.

Seznam doporučené odborné literatury:

- [1] BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.
- [2] DORF, C. R. & BISHOP, H. R. *Modern Control Systems*. London: Prentice-Hall, Inc. 2001. 831p. ISBN 0-13-030660-6.
- [3] FARANA, R. AJ. *Programová podpora simulace dynamických systémů. Sbirka řešených příkladů*. 1 vyd. Ostrava: KAKI 1996, 114 s. ISBN 80-02-01129-5.
- [4] NOSKIEVIČ, P. *Modelování a identifikace systémů*. 1. vyd. Ostrava : MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.
- [5] MATHWORKS. *Help k programům Matlab, Matlab Compiler, Matlab Builder for .NET, Matlab report generátor*. PDF formát. [DVD] Dostupný z [www: <URL: http://www.mathworks.com>](http://www.mathworks.com).
- [6] PEŠKE, M. *3D animace teplovzdušného modelu*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2008. 41 stran. Bakalářská práce, vedoucí: Smutný, L., konzultant: Babiuch, M.
- [7] VALAS, M. *Softwarová podpora výuky základů automatizace*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2006. 68 stran. Bakalářská práce, vedoucí: Wagnerová, R.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Renata Wagnerová, Ph.D.**

Datum zadání: 18.12.2009

Datum odevzdání: 21.05.2010



prof. RNDr. Lubomír Smutný, Dr.
vedoucí katedry



prof. Ing. Radim Farana, CSc.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucí diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě 21. 5. 2010

.....
podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby

V Ostravě 21. 5. 2010

.....
podpis studenta

Bc. Jiří Hikl
Budovatelů 13
Nový Jičín
PSČ 74101

ANOTACE DIPLOMOVÉ PRÁCE

HIKL, J. *Návrh a realizace virtuálních laboratorních modelů: diplomová práce*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2010. 56 s. Vedoucí práce: Wagnerová, R.

Diplomová práce se zabývá návrhem a realizací virtuálních laboratorních modelů. Po úvodním seznámení s realizovanými virtuálními laboratořemi na vysokých školách jsou popsány vlastnosti a funkce programového systému MATLAB a jeho vybraných toolboxů. Následuje popis systému „Batyskaf“, pro nějž je vytvořen matematický a simulační model, jehož funkčnost je ověřena na reálném laboratorním zařízení. V další části projektu je pomocí nástroje V-Realm Builder vytvořen virtuální model systému „Batyskaf“ a jeho grafické ovládací prostředí. Model umožňuje virtuální simulaci čtyř druhů regulace, různá nastavení parametrů regulace i regulátorů, sledování aktuálních hodnot na grafu a ukládání naměřených dat průběhu regulace do souboru k pozdějšímu využití. V závěrečné části jsou popsány možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet.

ANNOTATION OF MASTER THESIS

HIKL, J. *Design and realisation of virtual laboratory models: Master Thesis*. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2010, 56 p. Thesis head: Wagnerová, R.

Master thesis deals with design and realisation of virtual laboratory models. After opening acquaintance with realised virtual laboratories in universities there are features and functions of MATLAB system and its chosen toolboxes described. Description of “Batyskaf” system follows, mathematical and simulation models are built up and its functionality is verified on real laboratory system. In the next part virtual model of „Batyskaf” system is made using V-Realm Builder tool and its own graphical user interface is created too. Model allows virtual simulation of four simulation types, various regulation and controller variables settings, plot actual value monitoring and measured data saving into file for later use. In the last part possibilities of Internet/Intranet environment access of virtual model are described.

Obsah

| | |
|---|----|
| Seznam použitých značek a zkratk | 7 |
| 1 Úvod | 9 |
| 2 Virtuální laboratoře na vysokých školách | 10 |
| 2.1. Virtuální laboratoř (ČVUT Praha) | 10 |
| 2.2. Vzdálené a virtuální laboratoře (UK Praha) | 11 |
| 2.3. Projekt Virlab (VŠB-TU Ostrava) | 13 |
| 2.4. Model tlakové soustavy (VŠB-TU Ostrava) | 14 |
| 2.5. 3D animace teplovzdušného modelu | 15 |
| 2.6. Virtuální modely v MATLAB VRML Editoru | 16 |
| 3 Vlastnosti a funkce vybraných toolboxů programového systému MATLAB | 18 |
| 3.1. MATLAB Compiler | 20 |
| 3.2. MATLAB Builder for .NET | 21 |
| 3.3. MATLAB Report Generator | 22 |
| 3.4. MATLAB Virtual Reality | 24 |
| 4 Vytvoření virtuálního modelu systému „Batyskaf“ | 29 |
| 4.1. Popis laboratorního modelu systému „Batyskaf“ | 29 |
| 4.2. Matematický popis systému „Batyskaf“ | 31 |
| 4.3. Simulační model systému „Batyskaf“ | 34 |
| 4.4. Virtuální model systému „Batyskaf“ | 37 |
| 4.5. Uživatelské prostředí a ovládání virtuálního modelu | 42 |
| 4.6. Možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet | 51 |
| 5 Závěr | 54 |
| 6 Seznam použité literatury | 55 |

Seznam použitých značek a zkratek

| | |
|-----------|--|
| ρ | hustota kapaliny [kg/m ³] |
| a | zrychlení plováčku [m/s ²] |
| D_S | D'Alembertova setrvačná síla [N] |
| F_{VZ} | vztlková síla [N] |
| G | tíhová síla [N] |
| g | tíhové zrychlení [m/s ²] |
| k | konstanta odporu prostředí [N/m/s] |
| m | hmotnost plováčku [kg] |
| O | odpor prostředí [N] |
| p_a | atmosférický tlak [Pa] |
| p_{bat} | tlak uvnitř plováčku [Pa] |
| p_h | hydrostatický tlak kapaliny [Pa] |
| p_k | skutečný tlak v soustavě – pomocná regulovaná veličina [Pa] |
| v | rychlost plováčku [m/s] |
| V | celkový objem plováčku [m ³] |
| V_{ST} | objem pevných částí plováčku [m ³] |
| V_{VZ} | objem vzduchu v plováčku [m ³] |
| V_{VZO} | objem vzduchu v plováčku při atmosférickém tlaku [m ³] |
| y | poloha plováčku – regulovaná veličina [m] |
| y_h | vzdálenost hladin ve válci a v plováčku [m] |
| AVI | Audio Video Interleave |
| CTF | Component Technology File |
| ČVUT | České vysoké učení technické |
| DDE | Dynamic Data Exchange |
| DLL | Dynamically Linked Library |
| GUI | Graphical User Interface |
| GUIDE | Graphical User Interface Development Environment |
| HTML | Hyper Text Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IPA | Internet Protocol Address |
| IPC | Industrial PC |
| ISES | softwarová stavebnice pro podporu vzdálených experimentů |
| LAN | Local Area Network |
| MCR | MATLAB Compiler Runtime |

| | |
|-----------|--|
| PC | Personal Computer |
| RTW | Real-Time Workshop |
| SCADA/MMI | Supervisory Control and Data Acquisition/Man-Machine Interface |
| TCP/IP | Transmission Control Protocol/ Internet Protocol |
| TVM | teplovzdušný model |
| VRT | Virtual Reality Toolbox |
| VRML | Virtual Reality Modeling Language |
| VŠB-TU | Vysoká škola báňská – Technická univerzita |
| UK | Univerzita Karlova |

1 Úvod

Pojmy virtuální model či virtuální laboratoř nejsou v dnešní době ničím novým a neznámým, nýbrž jsou realizovány a používány již řadu let. Vývoj technologií spěje kupředu mílovými kroky a s tím souvisí i vývoj realizačních prostředků virtuálních laboratoří.

V klasickém pojetí laboratorní úlohy obsahuje úloha popis s teorií, pracovní úkoly (včetně návodu na zapojení jednotlivých komponent apod.) a pokyny k měření. Měření se provádí fyzicky přímo na konkrétním laboratorním modelu a výstupem jsou naměřená data, která se dále zpracovávají. V podstatě to samé nabízí a umožňuje virtuální laboratoř, jen s tím rozdílem, že celý model a jeho komponenty jsou virtuální a měření probíhá simulací reálného procesu prostřednictvím virtuálního modelu. Výstupem jsou opět naměřená data ke zpracování. Hlavní výhodou je pohodlná obsluha ze vzdáleného počítače, kdy student testuje zařízení či technologický proces aniž by byl fyzicky přítomen v klasicky pojaté laboratoři. Jedinou podmínkou je počítač s přístupem na Internet. Prostředí virtuální laboratoře poskytuje mimo jiné i unikátní možnost řešení praktických laboratorních úloh zejména studentům kombinované formy studia a zároveň umožňuje zpřístupnění jinak méně využitých a často nákladných laboratorních zařízení pro samostatnou práci v časech mimo výuku.

V úvodní části tohoto projektu se budu zabývat seznámením s již realizovanými virtuálními laboratořemi na vysokých školách především z pohledu použitého softwaru. V druhé části projektu budou popsány vlastnosti a funkce vybraných toolboxů programového systému MATLAB, konkrétně toolboxu MATLAB Compiler, MATLAB Builder for .NET, MATLAB Report Generator a MATLAB Virtual Reality. V další části projektu bude popsán systém „Batyskař“, pro nějž bude vytvořen matematický a simulační model (v prostředí MATLAB, verze R2008a), jehož funkčnost bude ověřena na reálném laboratorním zařízení tohoto systému. Poté bude vytvořen virtuální model pomocí nástroje pro tvorbu virtuální reality V-Realm Builder verze 2.0, bude vytvořeno vlastní grafické uživatelské prostředí a popsáno ovládání modelu. Vytvořený virtuální model bude umožňovat názornou simulaci čtyř druhů regulací, různá nastavení parametrů regulace i jednotlivých regulátorů, sledování aktuálních hodnot na grafu a ukládání naměřených dat průběhu regulace do souboru k pozdějšímu využití.

V závěrečné části projektu budou popsány možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet a bude provedeno zhodnocení dosažených výsledků.

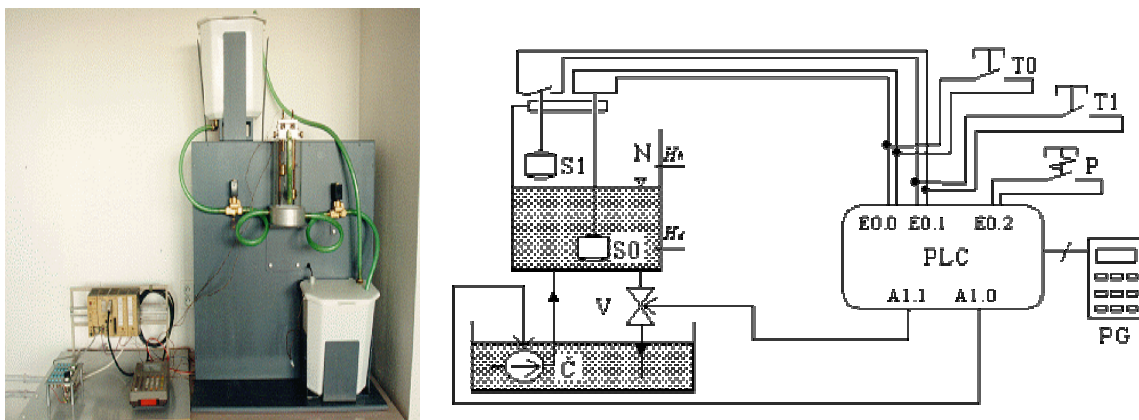
2 Virtuální laboratoře na vysokých školách

Virtuální laboratoře jsou především úspornou alternativou ke klasickým laboratořím. Prováděné experimenty umožňují nejen měření a řízení procesu, ale i přenos dat do klientských aplikací. Tyto laboratoře poskytují možnost řešení praktických laboratorních úloh zejména studentům kombinované formy studia a dále umožňují přístup k laboratorním experimentům pro samostatnou práci v čase mimo výuku. Ve virtuálních laboratořích se posluchači mohou seznámit nejen s hotovými laboratorními úlohami, ale mohou si nakonfigurovat zkoumaný proces dle vlastních potřeb. Virtuální laboratoře nejsou jen doménou vědců, ale začínají se hojně uplatňovat ve školním prostředí, kde přináší vizualizace a simulace technologických procesů pomocí Java appletů, Flash animací a podobně. Získané poznatky pak v konečném důsledku souvisí se vzdáleným řízením a kontrolou procesů nejen v průmyslové sféře, ale i v naší domácnosti (dálkové ovládání topení, monitoring okolí aj.). Speciální applety přinášejí zajímavé možnosti například z mikro i makro světa, které jsou v klasické laboratoři jen stěží (pokud vůbec) realizovatelné.

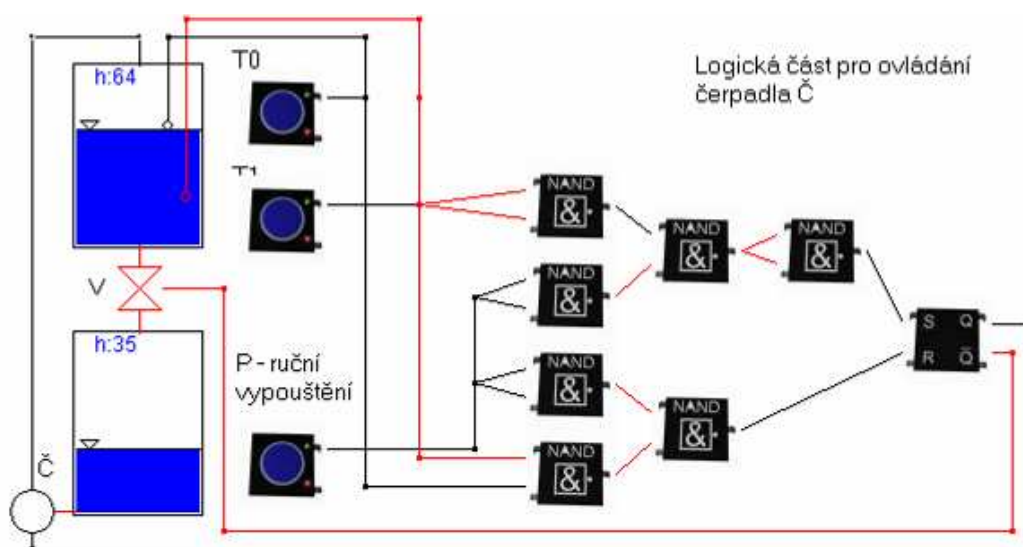
2.1 Virtuální laboratoř (ČVUT Praha)

Na Strojní fakultě ČVUT v Praze je k dispozici virtuální laboratoř, která obsahuje virtuální modely reálných úloh z oblasti automatického řízení. Tato laboratoř je volně přístupná pomocí běžného internetového prohlížeče, čímž je umožněno všem zájemcům z celého světa (tj. nejen studentům), simulačně provádět experimenty z oblasti automatického řízení pro ověření a prohloubení svých teoretických znalostí. Virtuální modely za pomoci animace a simulace znázorňují chování reálných úloh pro zvolené průběhy vstupních signálů a současně jsou i zaznamenávány průběhy všech signálů regulačního obvodu. Součástí virtuální laboratoře jsou i popisy, fotografie a video-záběry reálných úloh a návrhy možných úkolů pro experimentování ve virtuální laboratoři [Hofreiter, 2008].

K dispozici jsou zde laboratorní úlohy jak pro oblast logického řízení (vjezdová závora, kolejíště, řízení výšky hladiny – viz Obr. 1 a Obr. 2, balíková plošina apod.), tak i analogového a diskrétního řízení (vzdušník, nádrž s vodou, kulička na desce, vozík na nakloněné rovině, spojené nádoby apod.). Softwarovým nástrojem je programovací jazyk JavaTM verze 1.5, jazyk VRML a vizualizační software Sorsybat Logic.



Obr. 1 - Reálný laboratorní model (vlevo) úlohy "Řízení výšky hladiny" a schéma zapojení (vpravo) [Hofreiter, 2008]



Obr. 2 - Simulační model laboratorní úlohy "Řízení výšky hladiny" [Hofreiter, 2008]

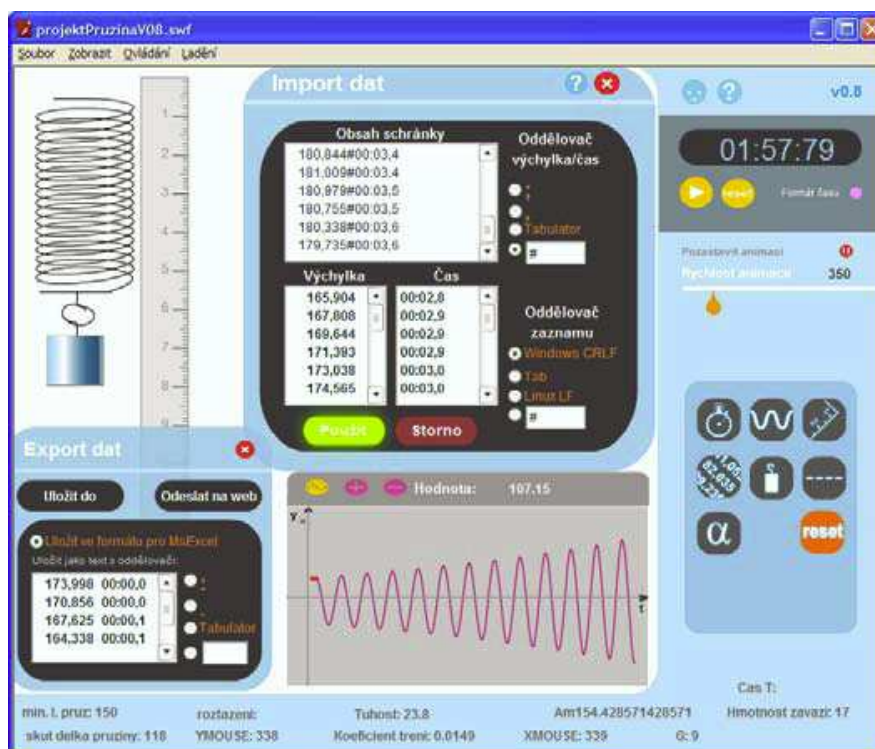
2.2 Vzdálené a virtuální laboratoře (UK Praha)

Vzdálené laboratoře začaly na Matematicko – fyzikální fakultě Univerzity Karlovy v Praze postupně vznikat od roku 2002. Současné pojetí vzdálených laboratorních úloh se blíží tradičnímu pojetí laboratorní úlohy. Úlohy obsahují popis s teorií, pracovní úkoly a pokyny k měření. Laboratorní úlohy byly vytvořeny pomocí soupravy ISES pro počítačem podporovaná měření a softwarovou stavebnicí ISES WEB Control (ukázka laboratorní úlohy na Obr. 3). ISES WEB Control je stavebnice 20 appletů a serverových komponent, ze kterých lze flexibilně sestavit i velmi komplikovaný vzdálený experiment. Experiment se připraví jako klasický reálný experiment s počítačem a poté se nad tímto lokálním hardwarem experimentu vytvoří (doslova pouze „poskládají“) applety do WWW stránky. Tímto jednoduchým způsobem se lokální experiment stane vzdáleným. Přístup do těchto vzdálených laboratoří je přes pouhý prohlížeč typu Internet Explorer, Firefox Mozilla, aj. [Lustig, 2008].



Obr. 3 - Ukázka laboratorní úlohy „Přeměna solární energie“ [Lustig, 2008]

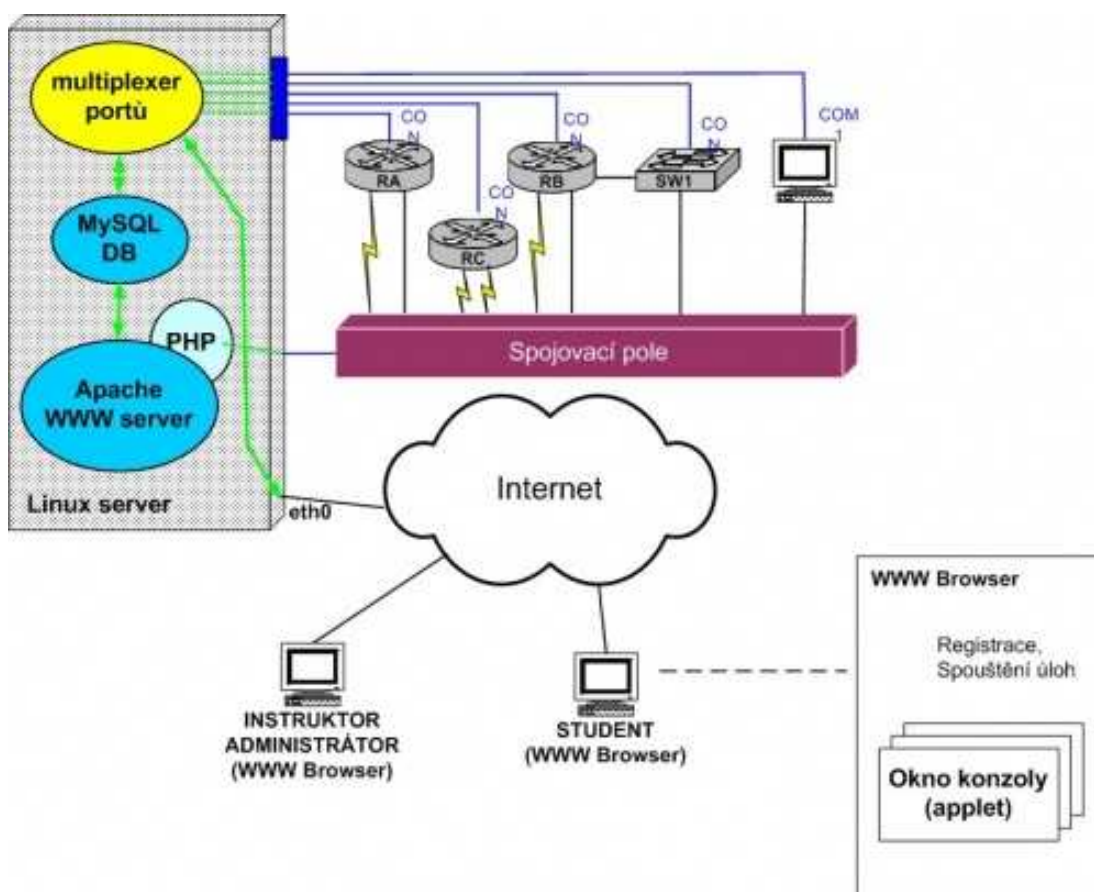
Virtuální laboratoře s applety, simulacemi a modely jsou prudce se rozvíjející technologií v přírodních vědách. Na Obr. 4 je ukázka appletu, který je částečně vhodný k vzdálené laboratorní úloze „Elektromagnetická indukce. Jedná se o applet s datovým exportem a importem s možností přenosu simulovaných dat přes clipboard (přenos dat je možný oběma směry, tedy mezi klasickou vzdálenou a virtuální laboratoří a můžeme porovnávat reálná data s virtuálními a nebo obráceně virtuální - teoretická data můžeme importovat do software aparatury měřící reálná data).



Obr. 4 - Ukázka Flash appletu „Kmity na pružině“ s datovým importem a exportem a interaktivními vstupy (tuhost pružiny, hmotnost závaží, tlumení, gravitace aj.) [Lustig, 2008]

2.3 Projekt Virlab (VŠB-TU Ostrava)

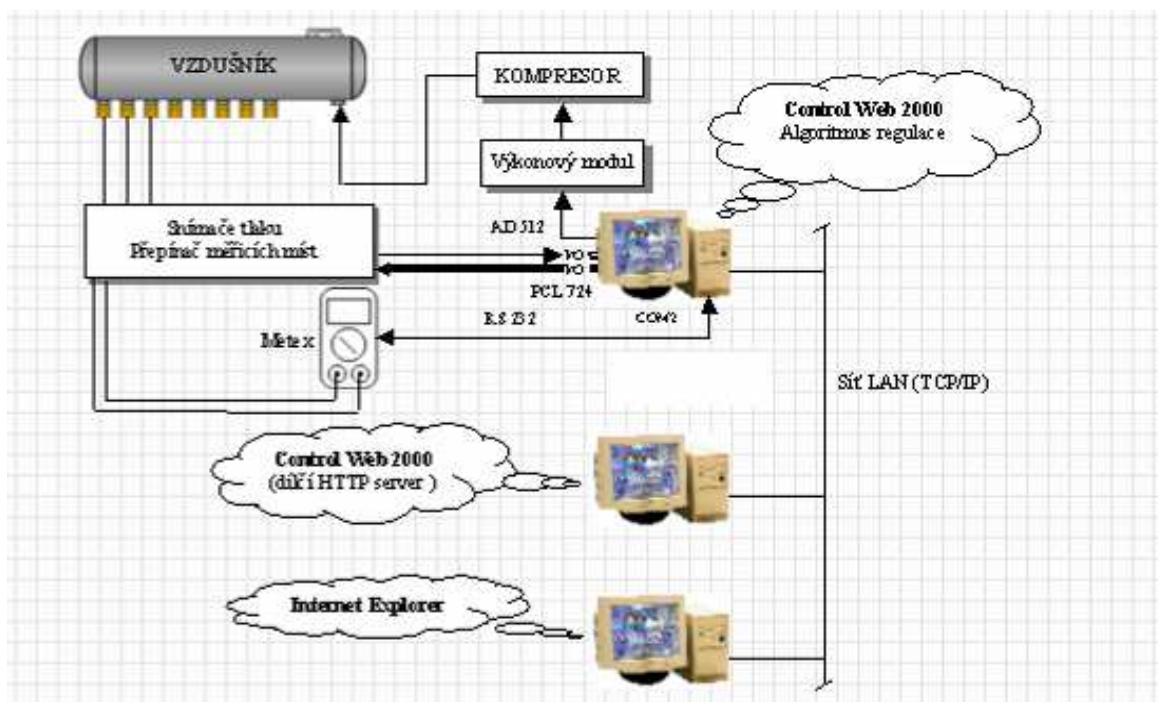
Pomocí projektu Virlab jsou na Fakultě elektrotechniky a informatiky na VŠB-TU v Ostravě zpřístupněny laboratorní prvky pro praktickou výuku počítačových sítí vzdáleně prostřednictvím Internetu (viz Obr. 5). Studenti si mohou pomocí WWW rozhraní rezervovat laboratorní prvky na určitý časový interval a následně k nim přistupovat pomocí běžného WWW prohlížeče s podporou Java appletů. Propojení laboratorních prvků se uskutečňuje automaticky podle výběru konkrétní úlohy ze souboru nabízených laboratorních úloh nebo si student může zadat svou vlastní topologii. Systém dovoluje spolupráci více lokalit vzájemně sdílejících síťové prvky a realizaci virtuálních síťových topologií přes Internet. Fyzické síťové prvky nutné pro vytvoření studentem vybrané topologie jsou v době rezervace vyhledávány dynamicky ve všech lokalitách [Grygárek, 2008].



Obr. 5 - Základní architektura projektu Virlab [Grygárek, 2008]

2.4 Model tlakové soustavy (VŠB-TU Ostrava)

Tento virtuální model byl vytvořen na Fakultě strojní VŠB-TU v Ostravě za účelem seznámení studentů s laboratorními úlohami pracujícími se SCADA/MMI systémy. Jako SCADA/MMI systém je použit systém Control Web 2000 umožňující konfigurovat a nastavovat řídicí systémy na nižší úrovni řízení. Těmito řídicími systémy jsou programovatelné logické automaty, jednočipové mikrokontroléry nebo řídicím systémem na nejnižší úrovni může být přímo IPC.



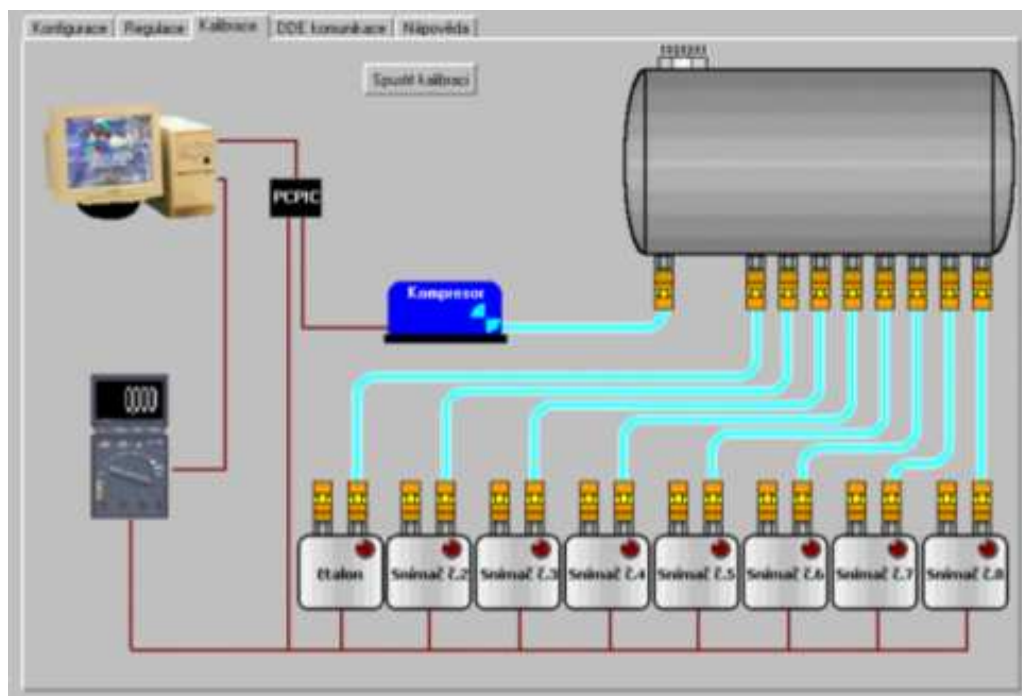
Obr. 6 - Blokové schéma zapojení tlakové soustavy [Škuta, 2006]

Model tlakové soustavy obsahuje tři snímače tlaku (viz schéma na Obr. 6). Jeden z nich lze volit jako referenční. Jeho naměřené hodnoty budou posílány do regulátoru. Pomocí regulátoru lze udržovat dle žádané hodnoty konstantní tlak v soustavě.

Zbylé dva snímače využijeme v druhé části měření nazvané ověření měřicích přístrojů. Výstupní napětí ze snímačů je měřeno přístrojem Metex. Z něho pomocí sériové linky se posílají data do počítače, na kterém je spuštěna aplikace vytvořená v systému Control Web 2000 (viz Obr. 7). Pomocí DDE komunikace lze data načíst do programu Microsoft Excel, kde se provede jejich vyhodnocení. Úkolem je zjistit odchylky naměřených hodnot testovaných snímačů od snímače referenčního. Výstup tvoří tabulky naměřených hodnot a grafy statických charakteristik.

Vytvořená aplikace v systému Control Web se skládá ze dvou modulů. Komunikace mezi nimi je zajištěna po LAN s využitím TCP/IP protokolu. Protože moduly nejsou

na sobě závislé, jedná se o spolupracující aplikaci. První modul zajišťuje sběr, řízení a vizualizaci dat. Na druhém modulu poběží HTTP server, pomocí kterého můžeme prezentovat data přes síť Internet. Tento modul je nadřazen prvnímu modulu. To znamená, že v rámci vybraných kanálů, které jsou zpřístupněné lze i z něho řídit modelovou úlohu. Pro prezentaci dat na síti Internet je využita služba WWW pracující čistě na principu požadavek – odpověď. Není možné, aby server sám poslal data klientovi např. při změně stavu nějaké veličiny. Vždy o ně musíme požádat v rámci HTML dokumentu.



Obr. 7 -Okno „Vizualizace“ úlohy pro kalibraci testovaných měřidel [Škuta, 2006]

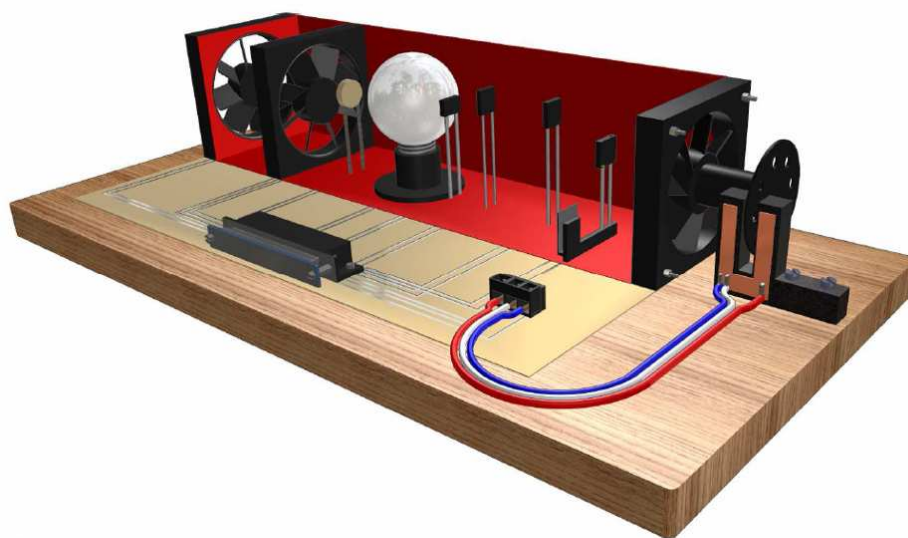
Tato úloha je také přístupná ze vzdáleného počítače, na kterém se aplikace v systému Control Web spouští jako druhá. Zároveň je zde umístěn HTTP server pro zpřístupnění úlohy přes síť Internet. Pro ovládání regulace z jakéhokoliv místa musíme v prohlížeči zadat IP adresu vzdáleného počítače [Škuta, 2006].

2.5 3D animace teplovzdušného modelu

Tento model byl vytvořen na Fakultě strojní VŠB-TU v Ostravě. Jedná se o virtuálního model, realizovaný pomocí 3D animace TVM v programu Macromedia Flash, jehož účelem je teoreticky připravit studenta tak, aby byl schopen dále pracovat samostatně na reálném TVM a programovém modulu WINGaP – CTRL. Tento model je řízen programovým modulem WINGaP – CTRL a umožňuje simulaci různých druhů regulací a poruch [Peške, 2008]. Studenti mají možnost si vyzkoušet simulaci čtyř typů regulací

TVM. Dvoupolohovou regulaci, kde je regulována teplota baňky žárovky pomocí akční veličiny napětí na žárovce, PI regulaci průtoku vzduchu, kde je regulován průtok vzduchu tunelem za sinusové poruchy a akční veličinou je napětí na hlavním ventilátoru, dále PI regulaci teploty, kde je regulována teplota baňky žárovky pomocí simulovaného PI regulátoru. Posledním typem regulace je PI regulace teploty a průtoku vzduchu.

K vytvoření animace bylo nutno nejprve vytvořit jednotlivé části 3D modelu TVM v programu Inventor, který slouží pro 3D modelování a teprve poté byly jednotlivé komponenty implementovány pomocí animací a action skriptu v programu Macromedia Flash do konečné podoby virtuálního modelu TVM (viz Obr. 8).



Obr. 8 - 3D animace teplovzdušného modelu [Peške, 2008]

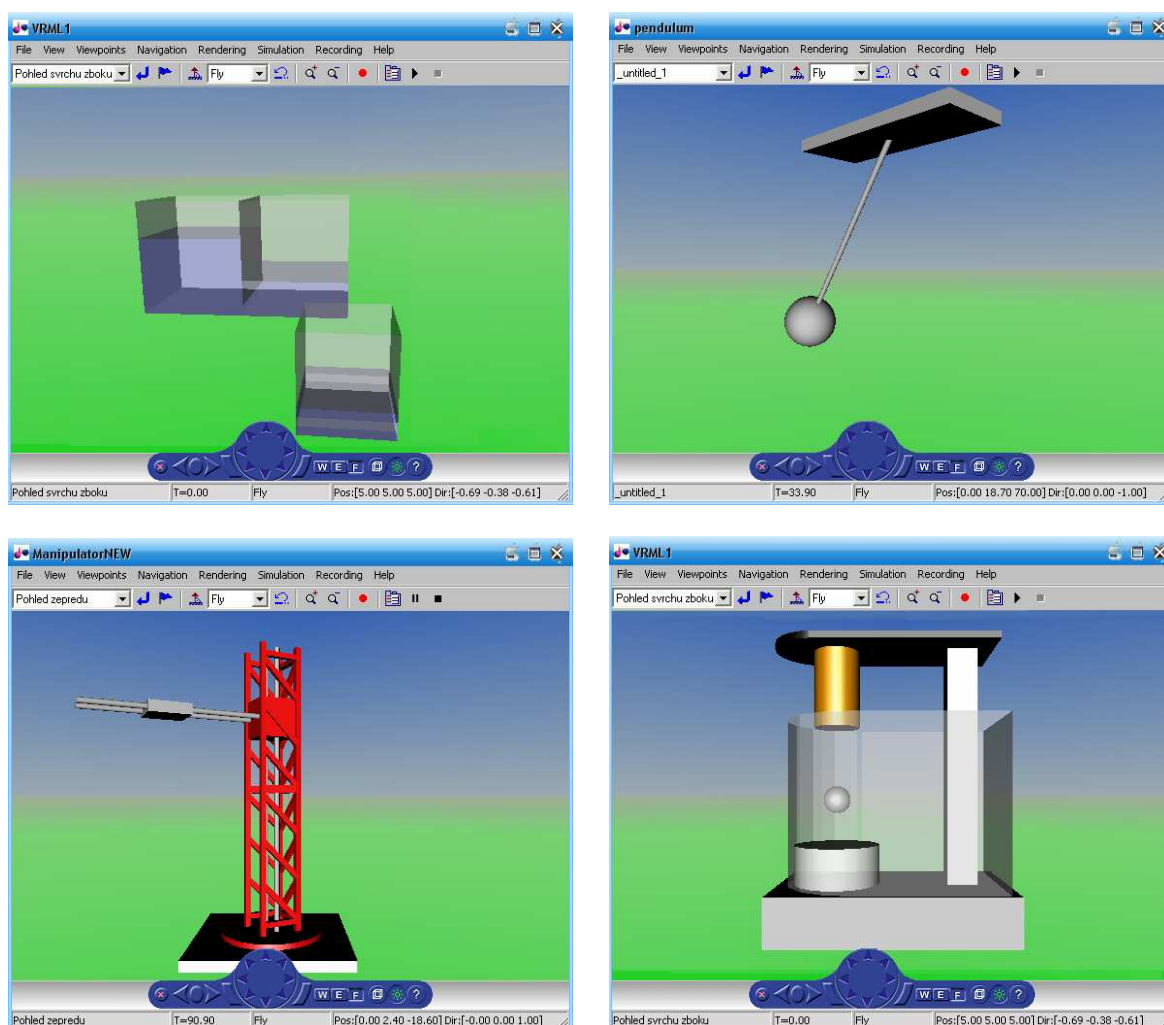
2.6 Virtuální modely v MATLAB VRML Editoru

Tyto virtuální modely byly také vytvořeny na Fakultě strojní VŠB-TU v Ostravě a mají za úkol sloužit jako názorné pomůcky ve výuce předmětu Základy automatizace. V prostředí programového systému MATLAB (verze 7.0.1) a jeho specializovaných knihoven Simulink a Virtual Reality, byly za pomoci simulačního a animačního nástroje VRML Editor vytvořeny modely hydromechanického systému, mechanického systému, modelu kyvadla, manipulátoru a modelu levitace (viz Obr. 9). Modely kyvadla a levitace jsou řízeny signály z prostředí Simulinku, pro ostatní bylo vytvořeno virtuální prostředí, které je možné ovládat interaktivně pomocí tříosého joysticku *GEMBIRD USB 2.0 RZ axis* [Valas, 2006].

Všechny výše uvedené virtuální modely byly vytvořeny stejným postupem. Nejprve byla popsána dynamika každého systému matematicky pomocí diferenciálních rovnic, poté byl sestaven simulační model pomocí simulačního nástroje Simulink v prostředí MATLAB, dále byl

vytvořen model virtuálním nástrojem VRML Editor, jenž je součástí specializovaného Toolboxu Virtual Reality a jeho nástroje pro vytváření virtuálního prostředí V-Realm Builder a nakonec spojením blokového schématu s vytvořeným virtuálním prostředím vznikla animace (funkční virtuální model), která popisuje dynamické chování daného systému.

Vytvořené virtuální modely napomáhají studentům v pochopení spojitosti diferenciálních rovnic a simulačních modelů při popisu reálných systémů. Studenti mají možnost sledovat vliv změn volitelných parametrů na chování systému nejen pomocí grafů, ale i přímo vizuálně na těchto modelech.

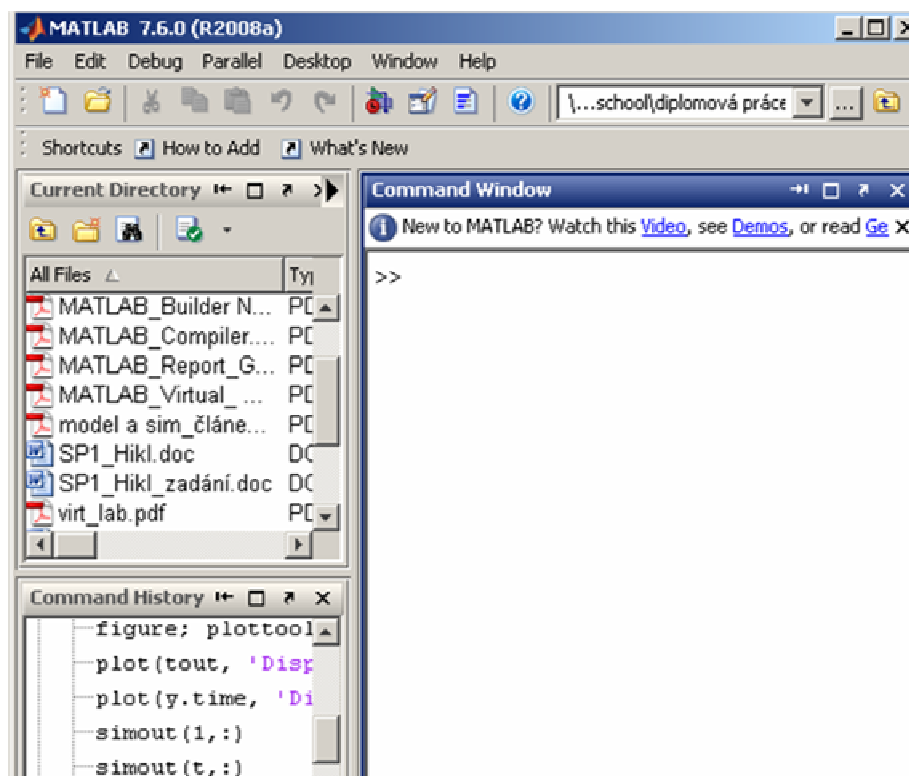


Obr. 9 – Ukázka modelů vytvořených v prostředí V-Realm Building: hydromechanický systém (vlevo nahoře), kyvadlo (vpravo nahoře), manipulátor (vlevo dole) a model levitace (vpravo dole) [Valas, 2006]

3 Vlastnosti a funkce vybraných toolboxů programového systému MATLAB

Programový systém MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. [HUMUSOFT, 2008]. Název MATLAB vznikl zkrácením z MATrix (matice) a LABoratory (laboratoř). MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Za nejsilnější stránku MATLABu je považováno mimořádně rychlé výpočetní jádro s optimálními algoritmy. MATLAB byl implementován na všech významných platformách (Windows, Linux, Solaris, Mac). Typickými oblastmi použití MATLABu jsou:

- Matematické výpočty
- Vývoj algoritmů
- Modelování a simulace
- Analýza dat a vizualizace
- Vědecká a inženýrská grafika
- Vývoj aplikací včetně vytváření uživatelského grafického rozhraní



Obr. 10 - MATLAB Desktop intuitivní rozhraní konfigurovatelné dle požadavků uživatele [Matlab, 2008]

Programový systém MATLAB se skládá z těchto hlavních částí:

- **Nástroje a vývojové prostředí** - tato sada nástrojů a prostředků představuje pomocníka, jak používat a stát se produktivnějším při práci s funkcemi a soubory v MATLABu. Mnohé z těchto nástrojů jsou součástí grafického uživatelského rozhraní, které zahrnuje pracovní plochu MATLABu a příkazové okno (viz Obr. 10), historie příkazů, editor a debugger, analyzátor kódu, prohlížeče pro shlížení nápovědy, pracovní oblasti, souborů a hledání cesty.
- **Knihovna matematických funkcí** – je to rozsáhlá sbírka výpočtových algoritmů od základních funkcí jako suma, sinus, cosinus a komplexní aritmetika, po náročnější funkce jako inverzní matice, vlastní hodnota matice, Besselova funkce a rychlá Fourierova transformace.
- **MATLAB jazyk** - je vysoko-úrovňový jazyk druhu matice/pole s kontrolovaným tokem sdělení, funkcí, datových struktur, vstupů/výstupů a objektivě orientovaných programových vlastností. To umožňuje jak „programování v malém“ k rychlému vytváření programů nanečisto, tak i „programování ve velkém“ k vytváření obsáhlých a komplexních programových aplikací.
- **MATLAB grafika** - MATLAB poskytuje rozsáhlé prostředky k zobrazení vektorů a matic jako grafy, jakož i ke značení a tištění těchto grafů. Tyto prostředky zahrnují i vysoko-úrovňové funkce k 2 a 3-rozměrné vizualizaci dat, zpracování obrazu, animaci a prezentační grafiky a zahrnují také nízko-úrovňové funkce umožňující plně-volitelné nastavení vzhledu grafiky, jakož i vytvoření úplného grafického uživatelského rozhraní vlastní aplikace MATLAB.
- **MATLAB aplikace programového rozhraní** (API – Application Program Interface) - je knihovna, která umožňuje psát programy v jazycích C a Fortran, které komunikují s MATLABem. Zahrnuje prostředky k vyvolání běžných funkcí z MATLABu (dynamické spojování), k volání MATLABu jako výpočetního stroje a pro čtení a psaní MAT-souborů.

3.1. MATLAB Compiler

MATLAB Compiler je programový nástroj systému MATLAB, jenž slouží ke kompilaci M-souborů, MEX-souborů, MATLAB objektů nebo dalších MATLAB kódů. MATLAB Compiler je možno použít k vytváření:

- samostatných aplikací na platformách UNIX, Windows a Macintosh
- sdílených knihoven jazyka C a C++ (dynamicky spojované knihovny – DLL pro Microsoft Windows).

Z hlediska popisu funkce a vlastností MATLAB Compileru je nutno uvést tři hlavní funkční bloky:

MATLAB Compiler generuje aplikaci nebo knihovnu

Balíky aplikací a knihoven vytvořené v MATLAB Compileru a určené k distribuci, zahrnují sadu knihoven MATLAB Compiler Runtime (MCR) a taktéž sadu podpůrných souborů rovněž generovaných MATLAB Compilerem. Zde se nastavují také systémové cesty na cílový stroj tak, aby byly nalezeny knihovny MCR a podpůrné soubory. MCR je samostatná sada sdílených knihoven, které podporují provedení kompilovaných M-souborů namísto matematických C/C++ a grafických knihoven a poskytuje kompletní podporu všech rysů jazyka MATLAB.

Aplikace nebo knihovna vytvořená MATLAB Compilerem se skládá ze dvou souborů: specifického binárního a archivačního, který obsahuje funkce a data MATLABu. Aplikační binární soubor je tvořen hlavní funkcí. Naproti tomu binární soubor knihovny se skládá z více funkcí pro export.

Obalové soubory

K vytvoření uživatelem specifikované binární soustavy na platformě MATLABu generuje MATLAB Compiler jeden nebo více obalových souborů. Obalové soubory poskytují rozhraní zkompilevanému M-kódu a liší se dle vykonávaného prostředí. Obalové soubory provádějí následující:

- vykonávají inicializaci a ukončení dle potřeb základního rozhraní

- definují datová pole obsahující informace o cestě, kódovacích klíčích a dalších, které potřebuje MCR
- poskytují kódy nutné k předání pokynů od funkcí rozhraní funkcím MATLABu v MCR
- pro aplikaci – obsahují hlavní funkci
- pro knihovnu – obsahují vstupní body pro každou veřejnou funkci M-souboru. Používají-li knihovny vytvořené MATLAB Compilerem, vyvolávají inicializaci a ukončení knihovny v jejich klientském kódu.

Soubor technologických komponentů (CTF-Component Technology File)

MATLAB Compiler generuje také soubor technologických komponentů, který je nezávislý na konečném cílovém typu (samostatná aplikace nebo knihovna) a je specifický ke každému operačnímu systému platformy. Tento soubor, který je pojmenován koncovkou .ctf, obsahuje funkce MATLABu a data, která definují aplikaci nebo knihovnu a je zakomponován samostatně v základním nastavení binárních funkcí sdílených knihoven C/C++ [MATHWORKS, 2008].

3.2. MATLAB Builder for .NET

MATLAB Builder for .NET je nadstavba MATLAB Compileru umožňující programátorům zabalit funkce MATLABu tak, aby byly přístupné z jakéhokoliv jazyka podporovaného specifikací CLS (Common Language Specification). Tato specifikace je sada jazykových vlastností podporována programovým nástrojem .NET Common Language Runtime (CLR). Specifikace CLS zahrnuje vlastnosti společné několika objektově-orientovaným programovacím jazykům jako C#, VB.NET a C++ s řízenými nadstavbami. Komponenty a nástroje podporované specifikací CLS zaručují součinnost s jinými komponenty a nástroji podporujícími CLS

Při distribuci balíku aplikace uživateli je nutno zahrnout nejen podpůrné soubory generované MATLAB Builderem, ale i sadu MCR.

MATLAB Builder for .NET funguje tak, že převádí funkce MATLABu na metody .NET, které zapouzdří kód MATLABu. Každá komponenta MATLAB Builderu obsahuje jednu či více tříd a každá třída poskytuje rozhraní M-funcím, jenž jsou přidávány do tříd během vytváření. Komponenty .NET poskytují sadu metod podporujících specifikaci CLS, která zabalí M-kód (všechny M-kódy musí být ve formě funkce).

Builder poskytuje robustní konverzi dat, indexování a schopnosti formátování pole, aby zachoval flexibilitu MATLABu při vyvolání řízeným kódem. Na podporu typů dat MATLABu poskytuje MATLAB Builder třídy MWArray ke konverzi dat, které jsou definovány v sestavě MATLAB Builder NE MWArray assembly. Na tuto sestavu je třeba se odkázat v řízené aplikaci k převedení původních polí na pole MATLABu a naopak.

Builder poskytuje i správu vstupních chyb, takže chyby pocházející z funkcí MATLABu jsou hlášeny jako standardní řízené výjimky. Popis chyby obsahuje specifický odkaz na kód MATLABu, čímž zjednodušuje proces ladění.

Vytváření komponentů COM

Builder lze rovněž použít k vytváření komponentů COM (Component Object Model), což je softwarová architektura vyvinutá firmou Microsoft k vytváření aplikací založených na komponentech. Objekty COM vystavují rozhraní, které umožňují aplikacím a jiným komponentům přístup k vlastnostem těchto objektů. Objekty COM jsou přístupné v Microsoft Visual Basicu, C++ či v jakémkoliv jazyku podporujícím objekty COM.

Nepodporované typy dat MATLABu

MATLAB Builder for .NET nepodporuje typy objektových dat MATLAB (např. Time Series Objects) a většinu neoznačených číselných typů [MATHWORKS, 2008].

3.3. MATLAB Report Generator

MATLAB Report Generator je programový software, který je možno použít k:

- dokumentaci úloh softwaru MATLAB jako jsou analyzování a vizualizace dat a vývoj algoritmů
- vytváření znovu-použitelných šablon zápisu, dle svých vlastních stylů a standardů
- aktuálnímu udržování dokumentace a specifikací v průběhu práce
- vytváření uživatelských návodů, které přesně vystihují schopnosti a specifikace tvořené aplikace

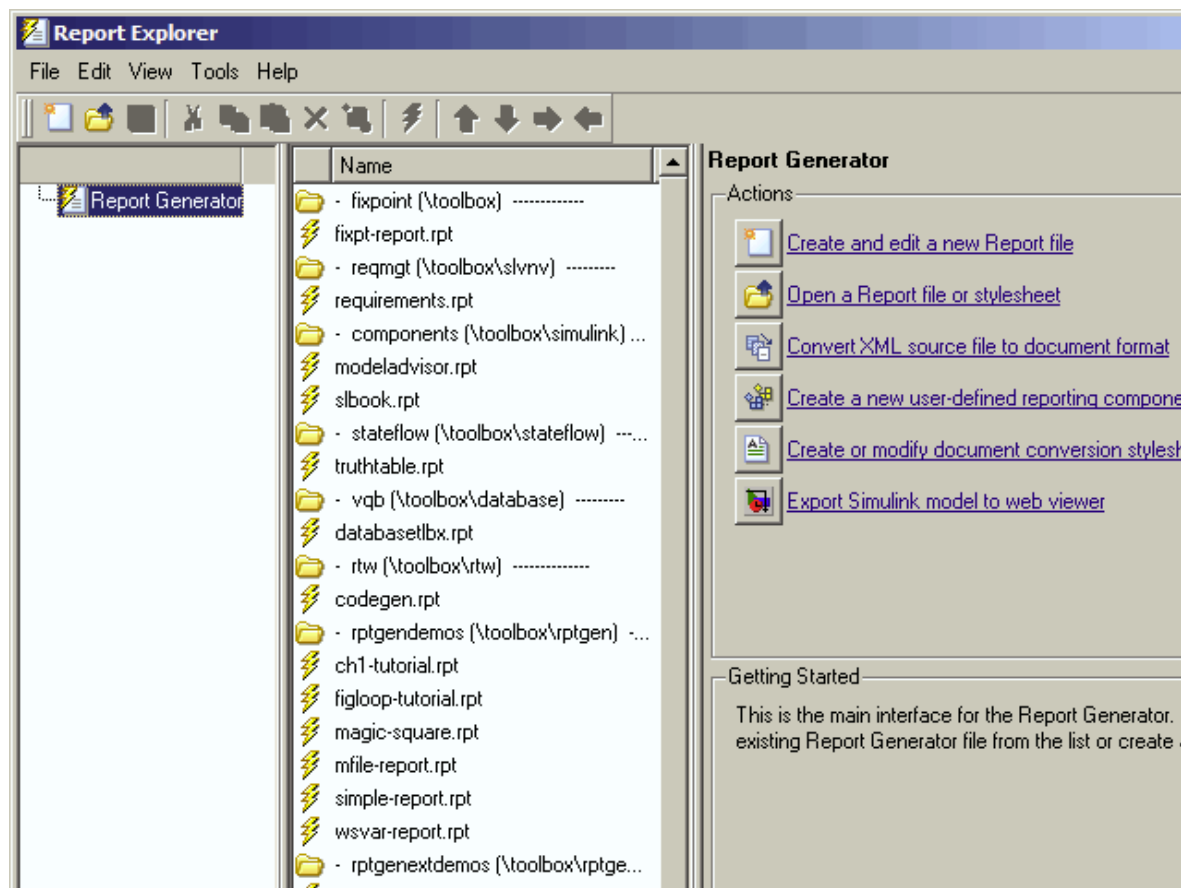
Proces tvorby záznamu vytváří dokumentaci (nazývanou záznamy) o datech MATLABu a pracovních postupech. Šablony záznamů je možno přizpůsobit, aby zahrnovaly:

- kód MATLABu a jeho výstup
- data pracovní oblasti
- grafiku MATLABu
- prvky logiky jako IF, THEN, ELSE a WHILE, které umožní podmíněné vytvoření záznamu
- smyčky, které vykonávají série operací na mnohonásobných objektech
- uživatelsky definovaný text, styly a komponenty

Report Explorer

Report Explorer je grafické uživatelské rozhraní (graphical user interface - GUI) MATLAB Report Generatoru a Simulink Report Generatoru, které je přístupné z příkazové řádky MATLABu a umožňuje:

- vytvářet a modifikovat šablony záznamu
- použít listy stylů k formátování generovaného záznamu
- specifikovat formát záznamového souboru
- generovat záznamy [MATHWORKS, 2008]

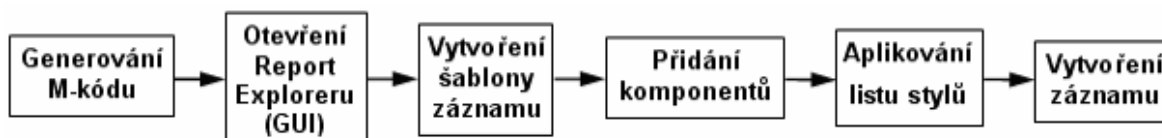


Obr. 11 - Okno grafického uživatelského rozhraní (GUI) [MATHWORKS, 2008]

Report Explorer má tři okna (viz Obr. 11):

- okno přehledu (nalevo) ukazuje hierarchii komponentů v aktuálně otevřených šablonách záznamu
- okno voleb (uprostřed), kde je možno listovat možnostmi dostupnými v kontextu okna přehledu. Není-li otevřena žádná šablona záznamu, okno voleb zobrazí přehled dostupných záznamů. Je-li šablona záznamu otevřena, okno voleb zobrazuje nabídku komponentů k vložení. Je-li otevřen list stylů, okno zobrazuje dostupné atributy.
- Není-li otevřena žádná šablona záznamu, okno vlastností (napravo) zobrazí úkoly, které může Report Explorer vykonat. Je-li šablona záznamu otevřena, okno vlastností zobrazuje vlastnosti položky aktuálně zvolené v oknu voleb

Součinností MATLAB Report Generátoru a softwaru MATLAB se vytváří záznamy. Typický pracovní postup je následující (viz schéma na Obr. 12):



Obr. 12 - Blokové schéma pracovního postupu MATLAB Report Generátoru

3.4. MATLAB Virtual Reality

Virtual Reality Toolbox (VRT) je nástrojem pro řešení součinnosti modelů virtuální reality dynamických systémů v časové oblasti. Rozšiřuje schopnosti MATLABu a Simulinku o grafiku virtuální reality [MATHWORKS, 2008].

- **Virtuální prostředí** – k vytváření virtuálního prostředí nebo trojrozměrných scén používá standardní technologie jazyka VRML.
- **Dynamické systémy** - vytváří a definuje dynamické systémy pomocí MATLABu a Simulinku.
- **Animace** - umožňuje zhlédnutí pohybujících se trojrozměrných scén řízených signály z prostředí Simulinku.
- **Manipulace** - umožňuje měnit pozici a vlastnosti objektů ve virtuálním prostředí během simulace.

K poskytnutí úplného pracovního prostředí, jsou součástí VRT tyto komponenty:

- **VRML prohlížeč** – pro platformy Windows, UNIX, Apple Mac OS X a Linux je možno použít buď výchozí VRT prohlížeč nebo připojení Blaxxun Contact Plug-in pro webové prohlížeče k zobrazení virtuálních prostředí [MATHWORKS, 2008].
- **VRML editor** – k vytvoření VRML prostředí je možno použít jakýkoliv VRML editor. K vytvoření a editaci VRML kódu pro potřeby platformy Windows se používá software Ligos® V-Realm Builder. K psaní VRML kódu pro vytvoření virtuálního prostředí pro platformy UNIX nebo Linux se používá textový editor MATLAB [MATHWORKS, 2008].

Mezi nejdůležitější softwarové vlastnosti VRT patří:

Podpora jazyka VRML

Jazyk VRML je ISO norma, která je otevřená, textově založená a používá WWW-orientovaný formát. Jazyk VRML se používá k definování virtuálního prostředí, které může být zobrazeno pomocí prohlížeče VRML a připojeno k modelu vytvořeného v Simulinku.

Software VRT využívá mnoho pokročilých funkcí, které jsou definovány v současné specifikaci VRML97. Výraz VRML definuje norma VRML97 ISO/IEC 14772-1:1997, dostupná z <http://www.web3d.org>. Tento formát zahrnuje popis 3-D scén, zvuků, vnitřních akcí a pilířů WWW [MATHWORKS, 2008].

VRT analyzuje strukturu virtuálního prostředí, určuje dostupné signály a umožňuje jejich použitelnost v prostředí MATLAB a Simulink.

VRT prohlížeč podporuje většinu uzlů normy VRML97 a umožňuje téměř úplnou kontrolu přidružených virtuálních prostředí. Blaxxun Contact plug-in podporuje všechny uzly normy VRML97.

VRT zajišťuje, že se změny uskutečněné ve virtuálním prostředí projeví v rozhraní MATLAB a Simulink. Změníme-li bod pohledu ve virtuálním prostředí, projeví se tato změna ve vlastnostech *vrworld* objektu v rozhraní MATLAB a Simulink [MATHWORKS, 2008].

Rozhraní MATLAB

Díky softwaru VRT je pružné rozhraní MATLABu poskytnuto prostředí virtuální reality. Po vytvoření objektů MATLABu a jejich propojení s prostředím virtuální reality, je možno řídit virtuální prostředí pomocí funkcí a metod.

Prostřednictvím softwaru MATLAB je možno nastavit polohu a vlastnosti objektů VRML, vytvořit zpětné vazby z grafického rozhraní a mapovat data k virtuálním objektům. Rovněž je možno zhlédnout prostředí prostřednictvím prohlížeče VRML, určit jeho strukturu a zavést nové hodnoty všem možným uzlům a jejich polím. VRT zahrnuje funkce k získání a změně vlastností virtuálního prostředí a pro ukládání souborů VRML dle aktuální struktury [MATHWORKS, 2008].

Software MATLAB poskytuje komunikaci pro řízení a manipulaci objektů virtuální reality použitím objektů MATLABu.

Rozhraní Simulink

Prostřednictvím modelu v Simulinku je možno pozorovat simulaci dynamického systému v čase na 2-D vizuálně reálném modelu.

Sada VRT poskytuje bloky k přímému zapojení signálů Simulinku s virtuálním prostředím. Toto spojení vizualizuje model jako trojrozměrnou animaci.

Většinu vlastností knihovny je možno implementovat s bloky Simulinku. Jsou-li jednou tyto bloky zahrnuty v diagramu Simulinku, je možno vybrat virtuální prostředí a propojit s ním signály Simulinku. Knihovna automaticky skenuje virtuální prostředí pro dostupné uzly VRML, které může software Simulinku řídit.

Všechny vlastnosti uzlů VRML jsou zobrazeny v hierarchickém stromu prohlížeče. Volí se stupně volnosti k řízení rozhraní Simulinku. Jakmile se zavře dialogové okno parametrů bloku, software Simulinku aktualizuje blok se vstupy a výstupy odpovídajícími vybraným uzlům virtuálního prostředí. Po propojení těchto vstupů s příslušnými signály Simulinku je možno zhlédnout simulaci v prohlížeči VRML [MATHWORKS, 2008].

Použitím bloků VRT poskytuje Simulink komunikaci k řízení a manipulaci objektů virtuální reality.

Podpora MATLAB Compileru

Díky této podpoře je možno použít kompilér a M-soubory jako vstupy a generovat znovu-rozšiřitelné, samostatné aplikace zahrnující funkcionalitu knihovny, včetně prohlížeče VRT. Samostatná aplikace má následující omezení:

- žádná podpora softwaru Simulinku, bez přístupu do knihovny VRT Simulinku (*vrlib*)
- žádný server VRT, bez možnosti vzdáleného připojení pro prohlížeče Orbisnap či Blaxxun
- není možnost záznamu animace
- není možnost editace prostředí
- následující vlastnosti prohlížeče VRT nemohou být použity v samostatné aplikaci:
 - Soubor > **Otevřít v Editoru**
 - Menu **Záznam**
 - Menu **Simulace**
 - Přístup do **Nápovědy**

Pro použití těchto vlastností je potřeba vytvořit M-soubor v rozhraní MATLABu pro VRT (kupříkladu k vytvoření, otevření a zavření objektů vrworld), poté je možno použít MATLAB Compiler.

Podpora Real-Time Workshop

Rozhraní Simulinku ve VRT podporuje Real-Time Windows Target™. Za použití externího režimu Simulinku je možno působit na kód Real-time, který je generován nástrojem Real-Time Workshop a kompilován s odjinud pocházejícím C/C++ kompilérem v prostředí Real-Time Windows Target.

Podpora SimMechanics

VRT je možno použít k zobrazení chování modelu vytvořených pomocí softwaru SimMechanics. Nejprve je vytvořen model stroje v rozhraní Simulinku s využitím bloků SimMechanics. Potom je vytvořen detailní obraz stroje ve virtuálním prostředí, jenž je následně připojeno k výstupům senzorů tělesa SimMechanics a je možno sledovat chování těles v prohlížeči VRML [MATHWORKS, 2008].

Hardwarová podpora

Nástrojová sada VRT obsahuje funkce pro připojení k hardwarovým zařízením, zahrnující joysticky a space mice, za použití bloků Simulinku.

Architektura Klient-Server

Software VRT spojuje MATLAB a Simulink s webovými prohlížeči na bázi jazyka VRML, využívající protokoly TCP/IP [MATHWORKS, 2008]. Tuto knihovnu je možno využít ve dvou konfiguracích:

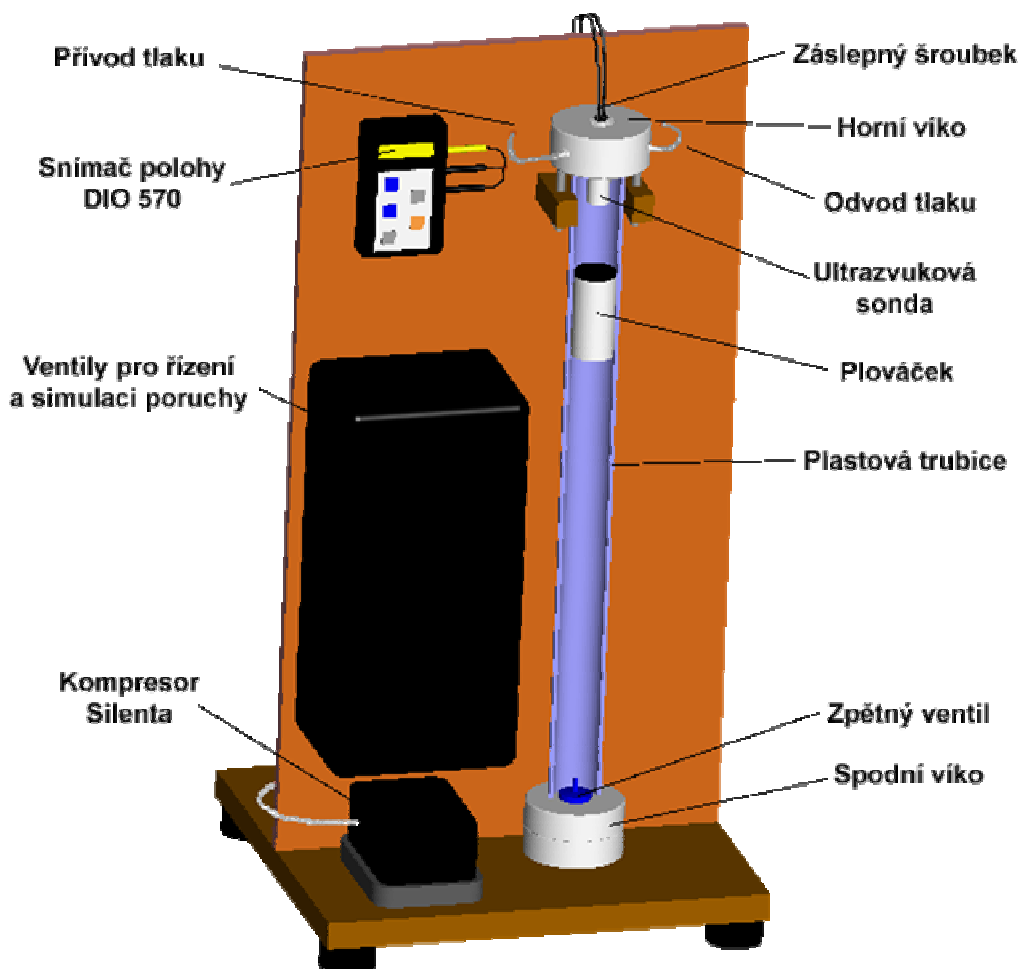
- **Jednotlivý počítač** – MATLAB, Simulink i prostředí virtuální reality běží na stejném hostitelském počítači.
- **Síťový počítač** – animované virtuální prostředí je možno sledovat na odděleném počítači, bez nainstalované knihovny VRT. Víceru klientů může být připojeno k jednomu serveru.

4 Vytvoření virtuálního modelu systému „Batyskaf“

Pro vytvoření virtuálního modelu bylo nutno se nejprve seznámit s reálným laboratorním zařízením systému „Batyskaf“, vytvořit matematický popis této soustavy, dále vytvořit pomocí programu MATLAB Simulink simulační modely pro zvolené typy regulací, poté vytvořit grafickou podobu virtuálního modelu, propojit ji se simulačními modely, vytvořit grafické uživatelské prostředí a zvolit způsob ovládání virtuálního modelu.

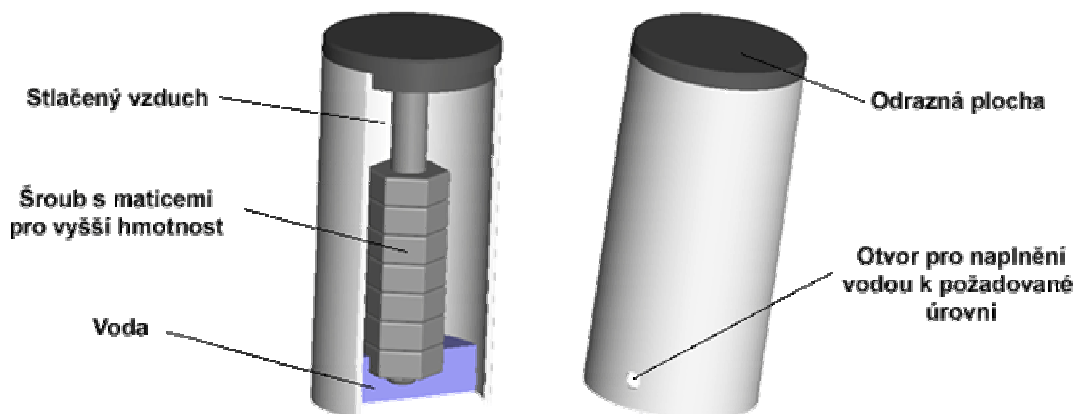
4.1. Popis laboratorního modelu systému „Batyskaf“

Laboratorní model systému „Batyskaf“ slouží k řízení polohy plováčku ve vodním sloupci. Hlavním účelem tohoto laboratorního zařízení je poskytnout možnosti prakticky odzkoušet většinu typů regulací, které se v rámci výuky předmětu Základy automatizace probírají [Hofreiter, 2005]. Názorné schéma a popis jednotlivých komponentů laboratorního modelu batyskaf je na Obr. 13.



Obr. 13 -Schéma jednotlivých komponentů systému "Batyskaf" [Hofreiter, 2005]

Soustavu tvoří plastová průhledná trubice ve vertikální poloze naplněná destilovanou vodou. Horní i spodní část jsou uzavřeny a utěsněny. K hornímu uzávěru je připevněno odměřovací zařízení a přívod tlakového vzduchu. Ve válci je umístěn plováček (viz Obr. 14), který je reprezentován kalíškem otočeným dnem vzhůru a v jeho ose je připevněn šroub s maticemi pro zvýšení hmotnosti. Do boku plováčku je u spodního okraje vyvrtán otvor. Při ponoření se naplní plováček až po otvor vodou a ve zbylém objemu je uzavřen vzduch.



Obr. 14 - Schéma plováčku [Hofreiter, 2005]

Princip pohybu plováčku spočívá v tom, že ve vyváženém stavu je tíhová síla rovna síle vztakové. Stoupne-li tlak, zmenší se objem vzduchu uvnitř plováčku (neboť vzduch je stlačitelný), tím se také zmenší objem vytlačené vody a zároveň se zmenší i vztaková síla. Gravitační síla převládne a plováček začne klesat. Přetlak v soustavě vůči atmosférickému tlaku se mění v rozmezí (0÷35) kPa. Hmotnost plováčku a objem v něm uzavřeného vzduchu je stanoven tak, aby začal klesat přibližně v polovině tlakového rozsahu, tj. při přetlaku kolem 20kPa [Hofreiter, 2005].

Pro řízení přetlaku je k válci připojen kompresor Silenta a dva ventily typu RFO-352. Tlak jdoucí z kompresoru je konstantní a výsledný tlak ve válci je závislý na otevření ventilů. Jeden je zde pro vykonávání akčního zásahu a druhý pro simulaci poruchy. Oba jsou umístěny na výstupu tlaku z válce a svým zavíráním kladou odpor vzduchu vypouštěnému do atmosféry. Tím stoupá tlak v celém systému. Jejich otvírání a zavírání je řešeno servomotory HS-785HB (umožňují otočení o $7,5 \times 360^\circ$). Tato serva jsou k ventilům připojena přes spojovací člen, který umožňuje posuv stopky ventilu a jsou ovládány z počítače programem MATLAB – Simulink přes sériovou komunikaci.

Měření polohy plováčku je provedeno bezdotykově vysílací/přijímací ultrazvukovou sondou v horní ucpávce trubice. Sonda vysílá ultrazvuk, který prochází vodou, dopadá na horní plochu plováčku a od ní se odráží zpět, kde ho přijímací část sondy přijme. Dle doby mezi vysláním a přijetím ultrazvukového signálu se měří vzdálenost plováčku od

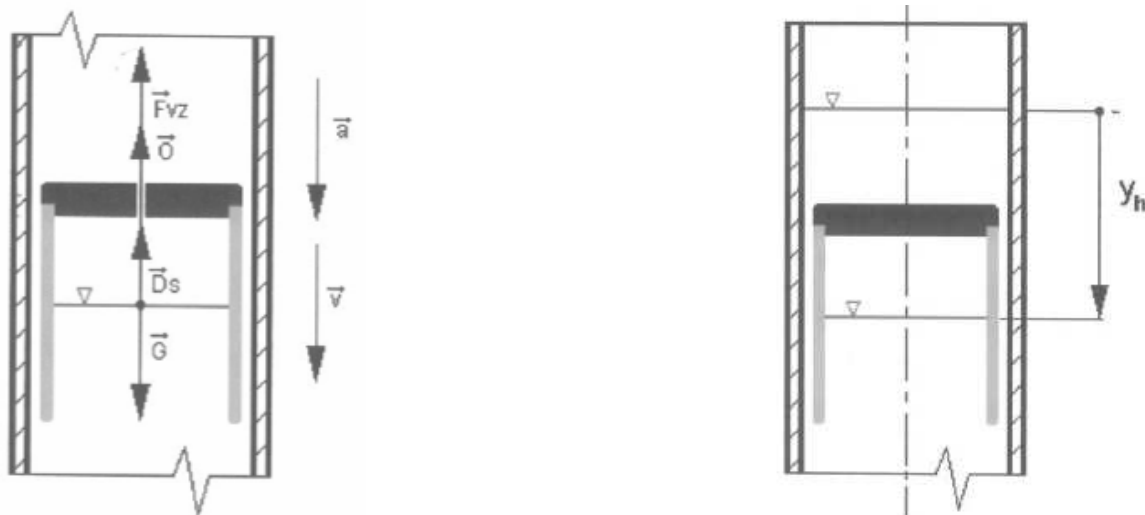
sondy. Měření času a jeho přepočítání na hodnotu vzdálenosti zajišťuje měřidlo DIO 570. Hlavním parametrem potřebným k výpočtu vzdálenosti je rychlost šíření ultrazvuku ve vodě. Ta je nastavena na 390 m/s tak, aby údaj o vzdálenosti plováčku od sondy odpovídal realitě. Přístroj je přes sériový kabel propojen s počítačem pro zpracování údajů v MATLABu.

System „Batyskař“ obsahuje dále tlakové čidlo, které je umístěno na zadní straně stojanu spolu s řídicí elektronikou pro serva. Tlak je k němu přiváděn odbočkou mezi výstupem tlaku z válce a ventily. Údaj z čidla je opět přes sériový port předáván do programu Matlab – Simulink [Hofreiter, 2005].

Laboratorní soustava „Batyskař“ je ovládána pomocí PC přes dva sériové porty COM: první sériový port slouží k výstupu žádané polohy serva řízení a ke vstupu údaje o poloze plováčku. Druhý sériový port slouží k výstupu požadované polohy serva poruchy a ke vstupu údaje o tlaku v systému. Sériové porty jsou ovládány přes bloky Serial in a Serial out z Real Time Toolboxu, který je nástavbou aplikace Matlab.

4.2. Matematický popis systému „Batyskař“

Při odvození diferenciální rovnice popisující dynamiku této soustavy bude použito základních fyzikálních vztahů a schématu na Obr. 15.



Obr. 15 - Dynamické schéma plováčku (vlevo) a význam souřadnice y_h (vpravo) [Hofreiter, 2005]

Tíhová síla G [N] působící na plováček je

$$G = m \cdot g, \tag{1}$$

kde je: m - hmotnost plováčku [kg], g - tíhové zrychlení [m/s^2], $g = 9,81 \text{ m/s}^2$.

Zjednodušený odpor O [N] prostředí při obtékání se spočte jako

$$O = k \cdot v, \quad (2)$$

kde je: k - konstanta odporu při obtékání plováčku [N/m/s], v - rychlost plováčku [m/s].

D'Alembertova setrvačná síla D_s [N] působící na plováček je dána vztahem

$$D_s = m \cdot a, \quad (3)$$

kde je: a - zrychlení plováčku [m/s²].

Vztlaková síla F_{VZ} [N] působící na plováček se určí jako

$$F_{VZ} = V \cdot \rho \cdot g, \quad (4)$$

kde je: V - objem plováčkem vytlačené kapaliny [m³], ρ - hustota této kapaliny [kg/m³].

Celkový objem plováčku V [m³] je dán součtem objemu pevných částí plováčku V_{ST} [m³] a objemu uzavřeného vzduchu V_{VZ} [m³]

$$V = V_{ST} + V_{VZ}. \quad (5)$$

Dosazením pak dostaneme

$$F_{VZ} = (V_{ST} + V_{VZ}) \cdot \rho \cdot g. \quad (6)$$

Rovnice rovnováhy sil je

$$\sum F_i = 0 \quad (7)$$

a pro plováček je možno ji zapsat v podobě

$$F_{VZ} + O + D_s - G = 0. \quad (8)$$

Po dosazení pak dostáváme výchozí tvar pro matematický model plováčku

$$(V_{ST} + V_{VZ}) \cdot \rho \cdot g + k \cdot v + m \cdot a - m \cdot g = 0. \quad (9)$$

Rovnice izotermie ideálního plynu má tvar

$$p_1 \cdot V_1 = p_2 \cdot V_2 \quad (10)$$

a pro případ vzduchu uzavřeného v plováčku vypadá

$$p_a \cdot V_{VZO} = p_{bat} \cdot V_{VZ}. \quad (11)$$

Tlak v plováčku se vypočítá dle vztahu

$$p_{bat} = p_a + p_k + p_h, \quad (12)$$

kde je: p_a - atmosférický tlak [Pa], p_k - tlak od kompresoru [Pa] a p_h - hydrostatický tlak vodního sloupce nad plováčkem [Pa], který se spočte

$$p_h = y_h \cdot \rho \cdot g, \quad (13)$$

kde je: y_h - vzdálenost plováčku pod hladinou [m].

Objem vzduchu uzavřeného v plováčku získáme úpravou z rovnic (11) a (12)

$$V_{VZ} = \frac{V_{VZ0} \cdot p_a}{p_a + p_k + p_h}. \quad (14)$$

Po dosazení do rovnovážné rovnice (3.9) získáme vztah

$$\left(V_{ST} + \frac{V_{VZ0} \cdot p_a}{p_a + p_k + p_h} \right) \cdot \rho \cdot g + k \cdot v + m \cdot a - m \cdot g = 0. \quad (15)$$

Dosadíme-li do rovnice (3.15) za

$$a = \ddot{y}_h, \quad (16)$$

$$v = \dot{y}_h, \quad (17)$$

a dle (3.13) za p_h , dostáváme výslednou **dynamickou rovnici plováčku**

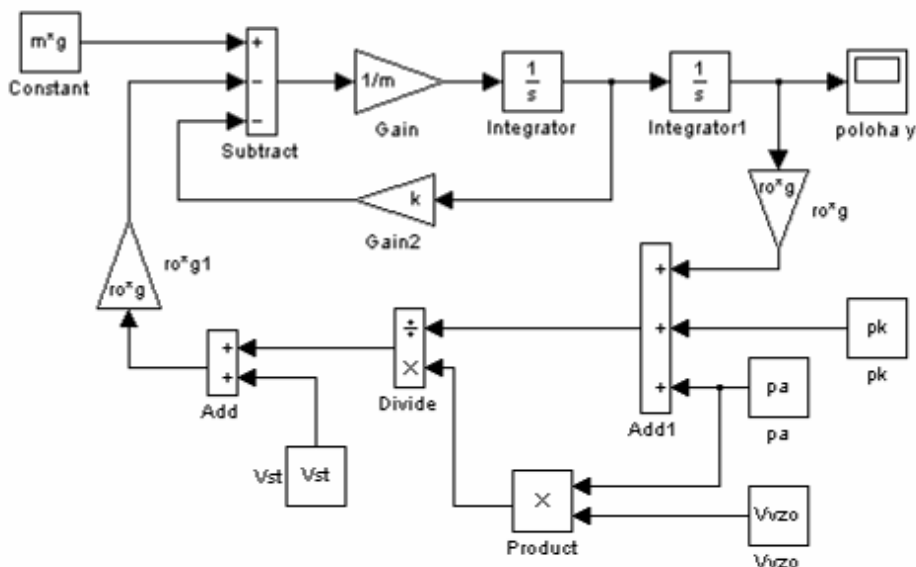
$$m \cdot \ddot{y}_h + k \cdot \dot{y}_h + \left(V_{ST} + \frac{V_{VZ0} \cdot p_a}{p_a + p_k + \rho \cdot g \cdot y_h} \right) \cdot \rho \cdot g - m \cdot g = 0. \quad (18)$$

4.3. Simulační model systému „Batyskaf“

Na základě matematického modelu byl vytvořen výchozí simulační model systému „Batyskaf“ v programu MATLAB R2008a pomocí jeho simulačního nástroje Simulink (viz Obr. 16). Pro simulaci byly použity hodnoty konstant dle Tab. 1.

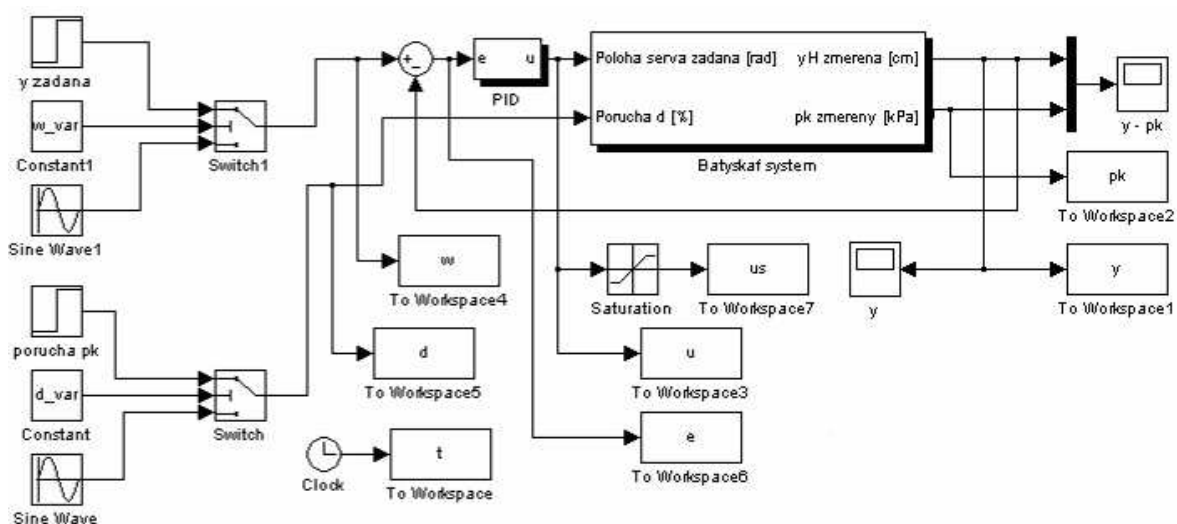
Tab. 1 - Hodnoty konstant pro simulaci pohybu plováčku

| | | | |
|-------------------------------------|-----------|-------------|----------|
| Konstanta odporu při obtékání | k | 2,8 | N/m/s |
| Hmotnost plováčku | m | 0,039 203 | kg |
| Objem pevných částí plováčku | V_{ST} | 0,000 019 5 | m^3 |
| Objem uzavřeného vzduchu v plováčku | V_{VZO} | 0,000 023 2 | m^3 |
| Atmosférický tlak | p_a | 101 325 | Pa |
| Hustota kapaliny | ρ | 998 | kg/m^3 |
| Tíhové zrychlení | g | 9,81 | m/s^2 |



Obr. 16 – Výchozí simulační model systému „Batyskaf“

Pro ověření správné funkce simulačního modelu bylo třeba odzkoušet chování simulované a reálné soustavy na reálném laboratorním zařízení. Jelikož základem pro vytvoření výchozího simulačního modelu systému „Batyskaf“ byl pouze teoretický matematický model dynamiky plováčku, bylo nutno vytvořit přesnější simulační model, aby se co nejvíce přiblížil chování reálné soustavy. Výchozí simulační model byl rozšířen o dílčí modely tlakování, servomotorů, ventilů, tlakového čidla, snímače polohy, odporu při obtékání, poruchy a byly zavedeny nelinearity. Blokové schéma výsledného simulačního modelu systému „Batyskaf“ pro PID regulaci je na Obr. 17.



Obr. 17 - Simulační model systému „Batyskař“ pro PID regulaci

Ověření funkčnosti simulačního modelu na reálném laboratorním zařízení

Funkčnost simulačního modelu pro PID regulaci soustavy byla ověřena na reálném laboratorním zařízení systému „Batyskař“ na Obr. 18. K reálnému řízení a simulaci této soustavy slouží výukový program **SORSYBAT**, jenž umožňuje čtyři druhy regulace (dvoupolohová, čtyřpolohová, PID a rozvětvená regulace). Umožňuje nastavovat běžné průběhy žádané hodnoty regulované veličiny, poruchy a příslušné parametry regulátorů. Grafické výsledky pak lze uložit do MATLABovského souboru typu *Figure* nebo vkládat do programu Word apod. přes schránku [Hofreiter, 2005].

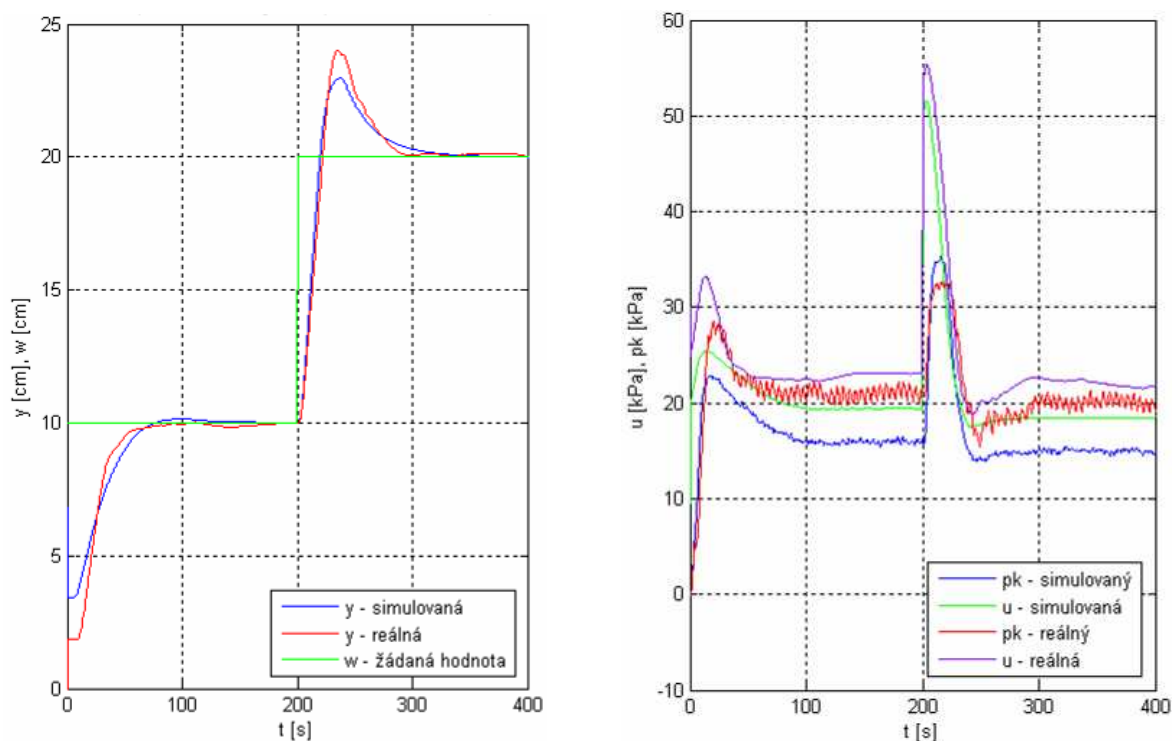


Obr. 18 - Pohled na reálné laboratorní zařízení systému "Batyskař"

Pro ověření funkčnosti simulačního modelu byla nejprve odsimulována regulace PID a poté proběhlo reálné měření se stejným nastavením parametrů regulace na laboratorním modelu. Hodnoty parametrů jsou uvedeny v Tab. 2. Výsledné grafy srovnání průběhů reálné a simulované PID regulace jsou na Obr. 19.

Tab. 2 - Hodnoty parametrů pro PID regulaci

| Parametry regulace: | | Parametry regulátoru | |
|-------------------------------------|-----|----------------------|-----|
| žádaná hodnota w_1 [cm] | 10 | P složka [-] | 3 |
| žádaná hodnota w_2 [cm] | 20 | I složka [1/s] | 0,1 |
| čas skoku $w_1 \rightarrow w_2$ [s] | 200 | D složka [s] | 0 |
| doba regulace [s] | 400 | | |



Obr. 19 - Grafy průběhů PID regulace (reálné a simulované)



Z grafů srovnání průběhů reálné a simulované PID regulace je patrné, že chování simulované soustavy, tzn. dynamika soustavy i doba reakce na změnu žádané veličiny (polohy plováčku) jsou téměř shodné s chováním reálné soustavy laboratorního systému „Batyskaf“.

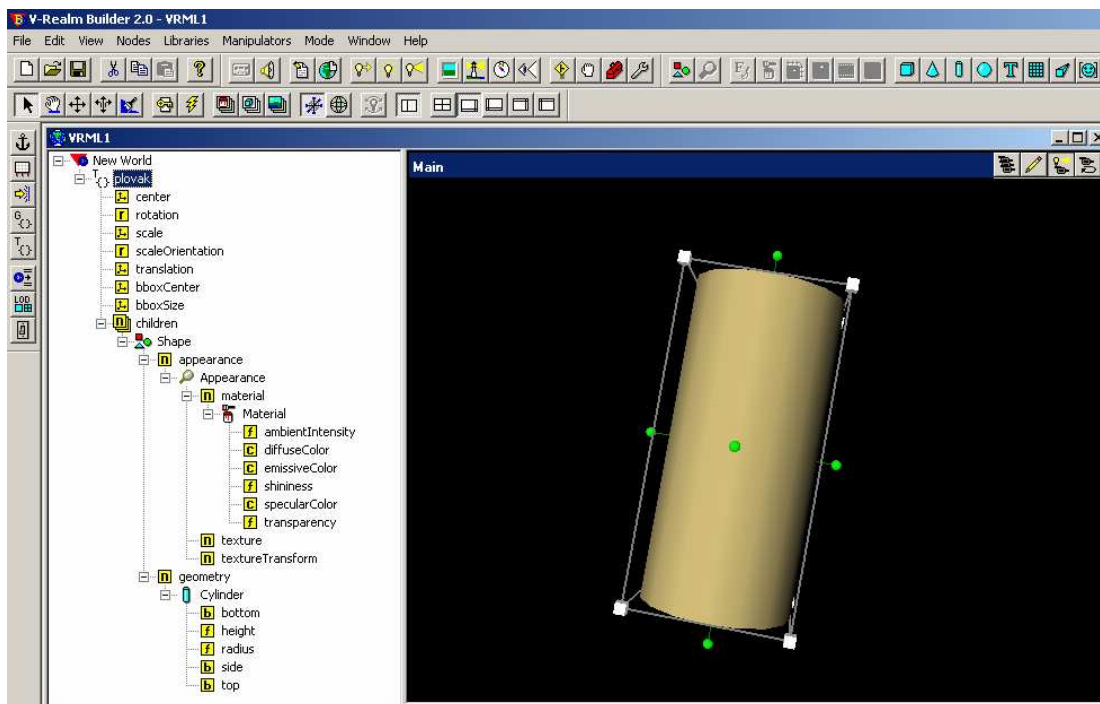
4.4. Virtuální model systému „Batyskaf“

K vytvoření úplného virtuálního modelu systému „Batyskaf“ bylo potřeba kromě již výše popsaného simulačního modelu, který definuje dynamiku soustavy, vytvořit také grafiku (vizuální podobu) virtuálního modelu, tyto navzájem propojit, aby bylo dosaženo požadované funkčnosti celého virtuálního modelu a zvolit způsob jeho ovládání.





Vytvoření grafiky (vizuální podoby) modelu

Vizuální podoba virtuálního modelu byla vytvořena v prostředí VRML Editoru, pomocí jeho nástroje V-Realm Builder 2.0. Model je tvořen několika komponenty: plovákem, trubicí s vodou, horní a spodní zátkou s ultrazvukovou sondou a zpětným ventilem, základnou s podstavcem, zadním čelem, kompresorem, boxem ventilů, snímačem polohy s displejem a tlačítky, propojovacími kabely a hadičkami.




Po spuštění aplikace V-Realm Builder 2.0 dvojklikem na programovou ikonu  *vrbuild2.exe*, umístěném ve zdrojovém adresáři aplikace MATLAB, se otevře hlavní okno programu se všemi nástrojovými lištami a jejich menu, pracovním oknem a hierarchicky uspořádaným oknem vytvářených uzlů (viz Obr. 20) a lze založit nový projekt s názvem **batyskaf_virtual.wrl**. Nový projekt je možno založit buď kliknutím ve *File menu* na položku *New* nebo v panelech nástrojů na ikonu . Poté byly namodelovány jednotlivé komponenty modelu:




Obr. 20 – Hlavní okno programu V-Realm Builder 2.0

- **Plováček** - z nástrojové lišty s předdefinovanými tvary byl vybrán válec kliknutím na ikonu . V pravém pracovním (černém) okně se zobrazí krychle souřadného systému, vyplněná požadovaným tvarem a zároveň se v levém okně zobrazí hierarchický strom obsahující hlavní položky *Transform*, *Scale*, *Translation*, *Shape*, *Appearance*, *Material* a *Geometry*. V položce *Geometry* byla zvolena výška *Height* = 5, poloměr *Radius* = 1.9, v položce *Material* byla nastavena barva *DiffuseColor*, položka *Transform* byla přejmenována na *plovak* a u položky umístění *Translation* byly zvoleny souřadnice [0;0;0], jelikož plováček v horní poloze reprezentuje regulovanou veličinu – výchozí polohu $y = 0$. K plováčku byl připojen *vrsek_plovaku* kliknutím na položku potomci *Children* ve stromu, poté byl z nabídky tvarů vybrán opět válec , nastavena *Height* = 0.2, *Radius* = 1.91, barva *DiffuseColor* na černou a posunutí položkou *Translation* na souřadnice $[x = 0; y = 2.4; z = 0]$.
- **Trubice batyskafu** – v levém okně bylo nutno nejprve ve stromu označit kliknutím myši položku *New World*, aby nedošlo nedopatřením k vytvoření dalšího potomka již vytvořené položky *plovak*. V nástrojové liště byl vybrán válec  a položka byla pojmenována jako *trubice*. Výška byla nastavena na *Height* = 47, *Radius* = 2, barva *DiffuseColor* na modrou, ale v položce *Transparency* byla nastavena průhlednost na hodnotu 0.8, aby imitovala původní materiál – průhledné sklo. Trubice byla posunuta na souřadnice $[x = 0; y = -19; z = 0]$. Vodní náplň trubice byla vytvořena jako potomek *trubice* ve tvaru válce s názvem *voda*, o výšce 46.5, poloměru 1.95, barva *DiffuseColor* modrá, průhlednost byla nastavena na hodnotu 0.7 a souřadnice umístění $[x = 0; y = -0.7; z = 0]$.
- **Horní zátka s ultrazvukovou sondou** – je tvořena válcem s označením *horni_viko*, o výšce 1.5, poloměru 3.55, barvy krémové, s umístěním na souřadnice $[x = 0; y = 5.25; z = 0]$. Sondu tvoří válec nazvaný *sonda*, o výšce 1.5, poloměru 0.5, barvy fialové, s umístěním na souřadnice $[x = 0; y = 3.75; z = 0]$. K horní zátce byly přidány přívodní hadičky a kabel s označením *hv_left*, *hv_right* a *hv_top*, které jsou tvořeny soustavou válců *hvl_1* až 4, *hvr_1* až 4 a *hvt_1* až 9 o výšce 1, poloměru 0.3, krémové a černé barvy a byly spojeny klouby, které vznikly použitím ikony koule  s poloměrem 0.32. K polohování těchto dílčích částí trubiček byla kromě položky *Translation* použita i položka *Rotation*, která slouží k prostorovému otáčení zvolené položky kolem zvolených os (x, y nebo z) v rozmezí +/- 180°.
- **Spodní zátka se zpětným ventilem** – byla vytvořena obdobným postupem jako horní, tzn. nový válec byl pojmenován *spodni_viko*, výška nastavena na 3, poloměr 3.5, barva krémová, s umístěním na souřadnice $[x = 0; y = -44; z = 0]$. Zpětný ventil tvoří dva válce modré barvy, nazvané *zpet_ventil_1* (o výšce 0.15, poloměru 1.2

a umístěn na souřadnice $[x = 0; y = -42.425; z = 0]$ a *zpet_ventil_2* (o výšce 1 a poloměru 0.1, na souřadnicích $[x = 0; y = -42; z = 0]$).

- **Základna s podstavcem a kompresorem** – základnu tvoří deska s názvem *podstava*, která byla vytvořena pomocí ikony krychle . V položce *Box – Size* byly definovány její rozměry $[x = 25; y = 1; z = 20]$, dále byla zvolena barva a umístění desky do souřadnic $[x = -5; y = -46; z = -3]$. Podstavec základny byl vytvořen v rámci položky *Children* a tvoří jej čtyři válce nazvané *noha_1* až *4*, které mají výšku 2, poloměr 2 a jsou umístěny v jednotlivých rozích základny pomocí nastavených souřadnic v položce *Translation*. Kompresor je tvořen dvěma krychlemi s názvy *kompresor_podstava* (velikost *Size* nastavena na $[x = 6; y = 0.5; z = 7.6]$ a umístění na souřadnicích $[x = -12; y = -45.3; z = 1.5]$) a *kompresor* (o velikosti $[x = 5.5; y = 4; z = 7.2]$ a souřadnicích $[x = -12; y = -43.2; z = 1.5]$). Hadička kompresoru je namodelována v rámci položky *Children* obdobným způsobem jako u horní zátky trubice, tzn. soustavou válců a kloubů pod názvy *komp_had_1* až *7*. Výšky jsou nastaveny na hodnotu 2, poloměry na 0.3 (poloměry koulí jsou 0.32) a rotace individuálně.
- **Zadní čelo s boxem ventilů** – je tvořeno vertikální deskou (ikona krychle ) pojmenované *zadni_celo*, o velikosti $[x = 25; y = 57; z = 1]$, umístěné na pozici $[x = -5; y = -17; z = -4]$, shodné barvy se spodní základnou. K zadnímu čelu je položkou *Children* připojena schránka ventilů *box_ventily* (kvádr o rozměrech $[x = 8; y = 21; z = 7]$, pozice $[x = -12; y = -24; z = 0]$), a *zadni_kryt* (kvádr o rozměrech $[x = 20; y = 53; z = 1.5]$, pozice $[x = 0; y = -1.2; z = -1.2]$).
- **Snímač polohy s displejem a tlačítky** – snímač polohy je tvořen kvádrem černé barvy s názvem *snimac_polohy* o velikosti $[x = 5; y = 7; z = 2]$, umístěném na souřadnicích $[x = -11; y = 0; z = -2.5]$. K němu je pod položkou *Children* připojen plochý kvádr *display* šedé barvy o velikosti $[x = 4; y = 1.7; z = 1]$ a souřadnicích $[x = 0; y = 2.2; z = 0.5]$. Dále je vytvořeno *podlozi_tlacitek* plochým kvádrem šedé barvy o velikosti $[x = 4; y = 4; z = 1]$ na souřadnicích $[x = 0; y = -1.1; z = 0.5]$. Tlačítka jsou namodelována opět v rámci položky *Children* plochými kvádry bílé, žluté a modré barvy s názvy *tlacitko_1* až *5* o velikosti $[x = 1.2; y = 0.8; z = 1]$ a individuálně definovaných souřadnicích. Přívodní kabeláž snímače je namodelována v rámci položky *Children* soustavou válců a kloubů pod názvy *kabel_1* až *3*. Výšky válců jsou nastaveny na hodnotu 1.5, poloměry na 0.15 (poloměry koulí jsou 0.17, u položky *slucovac* je poloměr 0.33) a rotace jsou nastaveny individuálně.
- **Pozadí a pohledy** – struktura a barevná kombinace pozadí modelu byly upraveny po rozkliknutí ikony  *Insert Background* z nástrojové lišty. Dále byly zvoleny


a vloženy pohledy s názvy *prední_pohled*, *pohled_pravobok* a *pohled_levobok* aktivací ikony  *Access/Edit View Point*. *Viewpoint* určuje bod, ze kterého je vizualizace sledována, proto je potřeba u jeho položky *Position* nastavit souřadnice. *Prední_pohled* má souřadnice $[x = -6; y = -20; z = 78]$, *pohled_pravobok* $[x = 43; y = -19.6; z = 67]$ a *pohled_levobok* je umístěn na $[x = -52; y = -19.6; z = 68]$. Změnu natočení lze provést v položce *Orientation*. Vytvořený pohled je aktivní po zaškrtnutí políčka *True* v položce *Set-Bind*.

Výsledná podoba grafiky virtuálního modelu se zobrazeným stromem je na Obr. 21.



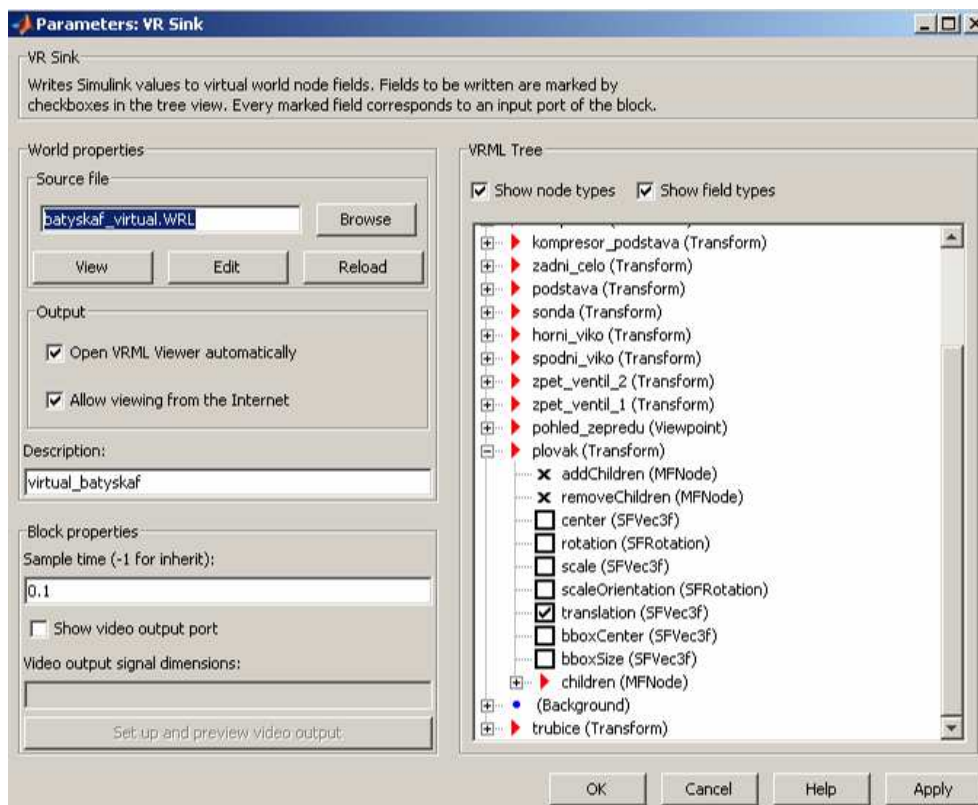
Obr. 21 - Hierarchický strom s uzly a vizuální podoba vytvořeného modelu

V prostředí V-Realm Builder 2.0 je k dispozici několik funkčních nástrojů, které v průběhu vytváření grafické podoby modelu napomáhají k přehlednosti a uspořádání pracovního okna. Pod tlačítkem *View* v nástrojové liště se rozbalí menu, kde je možno volit pohledy předo/zadní (*Front/Back*), shora/zespodu (*Top/Bottom*), boční zleva/zprava (*Left/Right*) či kombinovaný (*Equalize View*). Pravým klikem na plochu pracovního okna lze aktivovat režimy dynamické: položkou *Pick* lze označit, uchopit a otáčet jednotlivé komponenty modelu, režimem *Model* lze s modelem manipulovat, režim *Pan* slouží

k rotaci modelu kolem pevného bodu, v režimu *Navigation* lze model posouvat v rámci zorného úhlu obrazovky. V horní nástrojové liště je například nabídka knihovny materiálů (ikona ) , kde je možno vybrat typ a barvu povrchu z nabídky *Get From Library*, upravenou plochu pak přidat jako novou položku do knihovny materiálů kliknutím na *Add To Library* nebo zpětně vyvolat příkazem *Get from Library*.

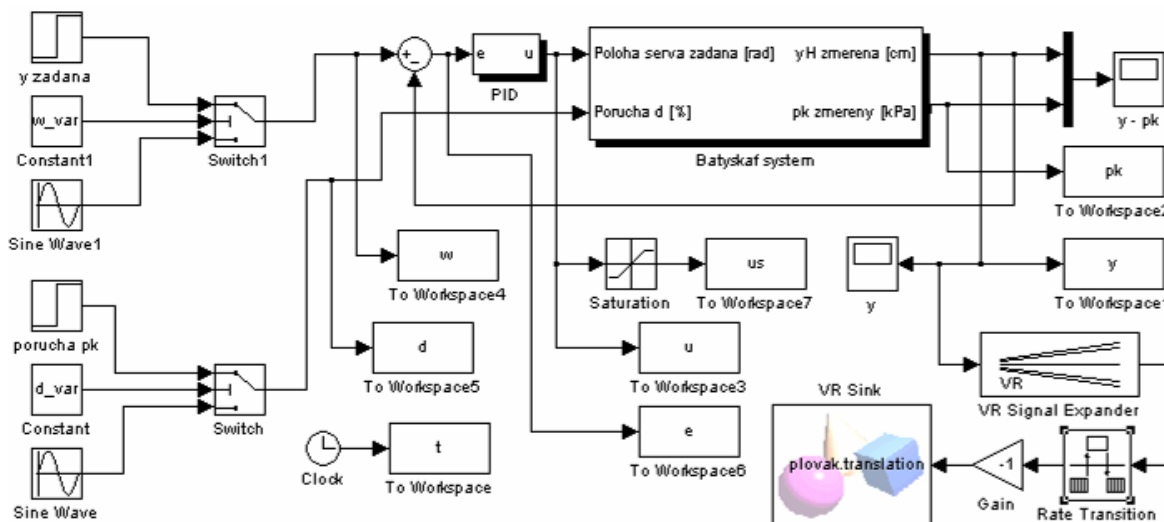
Propojení grafiky se simulačním modelem

Po vytvoření vizuální podoby modelu bylo možno provést jeho „oživení“, tzn. napojení na dynamickou část - simulační model. Do simulačních modelů pro všechny čtyři druhy regulace byly nejprve z knihovny VRT Simulinku přidány bloky **VR Signal Expander**, které redukuje vstupní vektor (jako *Output signal* byla zadána hodnota 2 - pro osu *y*, jako *Output width* byla zadána hodnota 3) a bloky **VR Sink**, které reprezentují spojení mezi signály simulačních modelů a vizuální podobou modelu. Po otevření bloku dvojklikem bylo pomocí položky *Browse* přiřazeno k modelu virtuální prostředí *batyskaf_virtual.wrl* uložené ve zdrojovém adresáři na místním disku. Pro automatické otevření vizuální části virtuálního modelu bylo třeba v okně *Output* zaškrtnout políčko *Open VRML Viewer automatically*. Dále byla v pravém okně tohoto bloku zaškrtnuta položka *Translation* u uzlu *plovak* (řídí pohyb plováčku). Do okna *Description* byl zadán název souboru a napojení bylo dokončeno příkazem *Apply* (viz Obr. 22).



Obr. 22 - Nastavení parametrů bloku *VR Sink*

Do simulačních modelů byly ještě zakomponovány bloky *Rate Transition*, které zprostředkovávají přenos dat mezi dvěma porty s rozdílnou rychlostí, a dále bloky zesílení *Gain* s hodnotou -1 , čímž bylo dosaženo převrácení kladné orientace osy y , která představuje pracovní osu pohybu plováčku. Výsledné schéma simulačního modelu s PID regulátorem pro virtuální model systému „Batyskař“ je na Obr. 23.



Obr. 23 – Blokové schéma simulačního modelu pro virtuální model systému „Batyskař“

4.5. Uživatelské prostředí a ovládání virtuálního modelu

Pro obsluhu virtuálního modelu systému „Batyskař“ bylo vytvořeno přehledné grafické uživatelské prostředí s názvem **VIRTBAT**, které vzniklo kombinací programovacích nástrojů MATLABu: *M-souborů* (se zdrojovými skripty, příkazy a funkcemi), systému *Handle Graphics* (pro vytváření grafických objektů) a nástroje *GUIDE* (pro interaktivní tvorbu grafického rozhraní).

- **M-soubory** – jsou soubory s příponou **.m*. Obsahují posloupnost příkazů a povelů, se kterými umí systém MATLAB pracovat. Existují dva typy: *scripty* (prosté seznamy příkazů a povelů) a *funkce* (jsou tvořeny také příkazy a povely, ale mohou obsahovat i jeden či více vstupních a výstupních parametrů) [Zaplatílek, 2004]. Jeden M-soubor může obsahovat i více funkcí navzájem provázaných nebo mohou být volány další M-soubory. Soubory lze editovat a ladit (*debugging*), vykonání (spuštění skriptu) se provádí klávesou *F5*.
- **Handle Graphics** – je grafický systém pro práci s grafickými objekty (obrázky, grafy, tlačítka, pole pro text, rolovací menu, posuvníky, zatržítka a podobně), které jsou podřízeny následující hierarchii: *Root* ⇨ *Figure* ⇨ *Axes* (*Uicontrol*, *Uimenu*,

Uicontextmenu) ⇒ *Image* (*Light*, *Line*, *Patch*, *Text*, atd.). Každý grafický objekt má přiděleno své číslo, prostřednictvím kterého se s ním pracuje (vyvolání, úprava parametrů, atd.).

- **GUIDE** – funguje v podstatě jako automatický generátor grafického uživatelského prostředí *GUI*. Po spuštění vede uživatele interaktivním způsobem krok po kroku při vytváření grafických objektů, jejich umístění, zadávání parametrů, atd. Výhodou je jednoduchost a nenáročnost při tvorbě objektů, nevýhodou je vytváření delšího a méně přehledného zdrojového kódu s odlišnou strukturou, který nemusí být vždy pro následné úpravy optimální.

Vytvoření grafického uživatelského prostředí

Grafické uživatelské prostředí je koncipováno jako posloupnost postupně se rozbalujících ovládacích oken s možnostmi nastavení volitelných parametrů pro virtuální simulaci modelu systému „Batyskaf“ a funkčními ovládacími tlačítky. Celé uživatelské ovládací prostředí je složeno z hlavního okna aplikace (*Úvodní okno*), čtyř podřízených oken pro jednotlivé druhy regulací (*Dvoupolohová regulace*, *Čtyřpolohová regulace*, *PID regulace* a *Rozvětvená regulace*) a dvou doplňkových oken (*Přepočítání PID parametrů* a *Zobrazení vstupů*).

- **Úvodní okno aplikace** – je definováno ve spouštěcím M-souboru aplikace *virtbat.m*.
 - Okno je vytvořeno objektem *Figure* „*fi_main=figure*“ s pozicí *Position* a rozměry [5 0.03*Monitor(4) 0.992*Monitor(3) 0.947*Monitor(4)], barvou *color* [0.6 0.2 0], s názvem *name* „*VIRTBAT - ÚVODNÍ OKNO*“. Přizpůsobení velikosti hlavního okna aplikace aktuální velikosti monitoru je ošetřeno příkazem *resize on*, před kterým je nejprve zjištěna velikost aktuálního monitoru příkazem *Monitor=get(0,'ScreenSize')*.
 - V okně jsou dále definována čtyři ovládací tlačítka pro otvírání podřízených oken jednotlivých typů regulací. Tlačítka jsou tvořena grafickými objekty *uicontrol* s přiřazeným stylem *pushbutton*, umístěním a rozměry *position* [50 845 150 80], zobrazeným textem *string* („*Dvoupolohová regulace*“, „*Čtyřpolohová regulace*“, „*PID regulace*“ a „*Rozvětvená regulace*“). Tlačítkům jsou přiřazeny funkce *callback*, které provedou aktivaci dalších M-souborů (*m_dvapolreg*, *m_ctyropolreg*, *m_pidreg* a *m_rozvetvenareg*) a kterými se otevře ovládací okno pro daný typ regulace. Popisek (upoutávka) tlačítka je vytvořen syntaxí *tooltipstring* „*Simulace*

řízení pomocí *dvoupolohové regulace*“ (a obdobně pro zbývající tři typy regulací), s barvou pozadí *backgroundcolor* [1 0.7 0] a tloušťkou písma *fontweight* „*bold*“.

- Hlavní text okna je vytvořen objektem *uicontrol* se stylem *text*, na pozici [25 625 270 60], textem *string* „*VIRTUAL BATYSKAF*“, barvou pozadí *backgroundcolor* [0.6 0.2 0], barvou popředí *foregroundcolor* [1 0.7 0], vodorovným zarovnáním *horizontalalignment*, velikostí písma *fontsize* 20 a stylem písma *fontangle*, *italic*, *fontweight*, *bold*. V úvodním okně je vložen obrázek příkazem *X1=imread('obr1.bmp')* s umístěním a rozměry *position* [0.55 0.05 0.4 0.9].
- **Okno regulace** – je definováno v pěti M-souborech a je tvořeno vždy částí společnou (definovanou v M-souboru *m_spolecne.m*) a částmi pro jednotlivé typy regulace zvlášť (M-soubory *m_dvapolreg.m*, *m_ctyropolreg.m*, *m_pidreg.m* a *m_rozvetvenareg.m*).
 - V části společné je vytvořen rám okna objektem *uicontrol* se stylem *style* „*frame*“, s umístěním a rozměry *position* [10 10 320 400], barvou pozadí *backgroundcolor* [1 0.7 0] a popředí *foregroundcolor* [1 0.7 0].
 - V okně jsou definovány posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení žádané veličiny *w1*, *w2*, skoku z *w1* na *w2*, poruchy *d1*, *d2*, skoku z *d1* na *d2* a nastavení doby regulace *t*. Dále jsou zde vytvořeny radiobuttony pro přepínání mezi skokovou změnou průběhu regulované veličiny i poruchy a sinovým průběhem, tlačítka *Graf*, „*w*“ a „*d*“, *ULOŽIT DATA* a *KONEC*. Posuvníky jsou tvořeny objekty *uicontrol* se stylem *style* „*slider*“ na pozici *position* [150 355 80 20] a obdobně, s definovaným rozsahem *max* 38 a *min* 4 (a obdobně), s krokem *sliderstep* [0.0029 0.0029], udávající hodnoty *value* *w1* (*w2*, *tw*, *d1*, *d2* a *dt*), aktivující funkcí *callback* M-soubory *m_zadana_hodnota1.m* (*m_zadana_hodnota2.m*, *m_zadana_hodnotat.m*, *m_porucha1.m*, *m_porucha2.m* a *m_poruchat.m*). Obdobným způsobem jsou vytvořeny posuvníky pro ovládání hodnoty skoku poruchy a u volby sinového průběhu poruchy nastavení amplitudy, středu a periody kmitů poruchy. Hodnoty posuvníků jsou tvořeny objekty *uicontrol* se stylem *text*, umístěním *position* [250 355 60 20], odvolávkou na řetězec *string* „*num2str*“ (*w1* a obdobně) a barvou pozadí *backgroundcolor* [1 0.7 0]. Popisy posuvníků jsou tvořeny objekty *uicontrol* se stylem *text*, umístěním *position* [20 355 122 20] a obdobně, zobrazeným textem *string* „*Žádaná hodnota w1 [cm]*“ a obdobně, barvou pozadí *backgroundcolor* [1 0.7 0] a zarovnáním *horizontalalignment* „*left*“.
 - Radiobuttony pro volbu průběhu žádané veličiny a poruchy jsou tvořeny objekty *uicontrol* se stylem *radiobutton*, hodnotou *value* 1, na pozici [20 380 60 20],

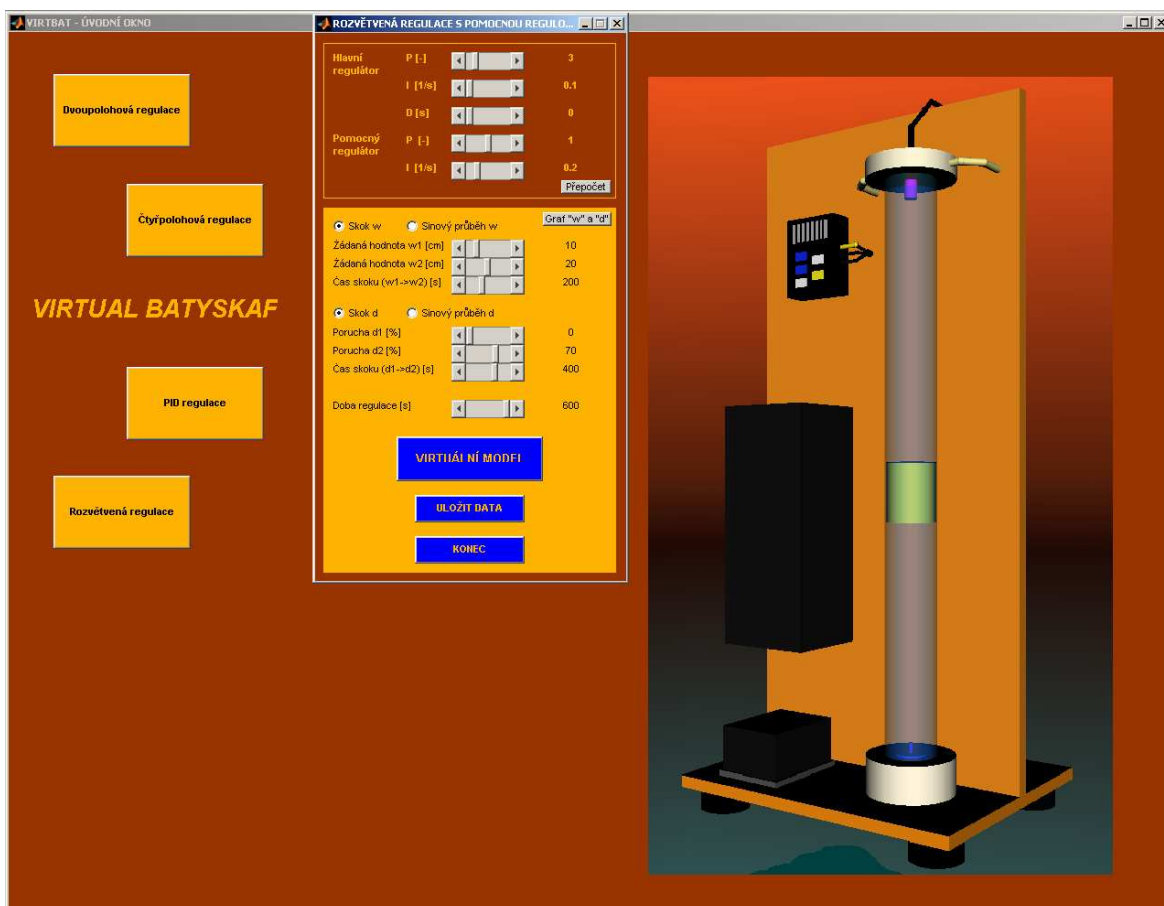
minimem $Min\ 0$ a maximem $Max\ 1$, textovým řetězcem *string* „Skok w “, s funkcí *callback*, která aktivuje M-soubor *m_wskok.m*, barva pozadí *backgroundcolor* je [1 0.7 0].

- Pro zobrazení průběhu žádané hodnoty w [cm] a poruchy d [kPa] je definováno objektem *uicontrol* se stylem *pushbutton*, na pozici [250 390 75 15], tlačítko s textem *string* „Graf " w " a " d “, s funkcí *callback* se zdrojovými daty v M-souboru *m_zadani.m*.
- Pro uložení dat je definováno tlačítko objektem *uicontrol* na pozici [110 65 120 30], s textem *ULOŽIT DATA*, s přiřazenou funkcí *callback*, která načte M-soubor *m_save1.m* a uloží data z grafu do souboru *s1.mat*.
- Tlačítko *KONEC* je tvořeno objektem *uicontrol* se stylem *pushbutton*, na pozici a rozměru [110 20 120 30], s funkcí *callback*, která spustí M-soubor *m_konec.m*, který zavře ovládací okno a ukončí zvolenou regulaci. Tlačítko má nedefinovanou barvu pozadí [0 0 1] a popředí [1 0.7 0] a zvýrazněné písmo *fontweight* „bold“.
- V podřízeném okně pro dvupolohovou regulaci jsou definovány již výše popsaným způsobem posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení hystereze a radiobuttony pro maximální a poloviční akční zásah s přiřazenou funkcí *callback*, která aktivuje odpovídající M-soubory *m_dvapolreg_plnau.m* a *m_dvapolreg_polovinau.m*, ve kterých je interním skriptem ošetřeno provedení požadované akce.
- Podobným způsobem je pro každý typ regulace vytvořeno tlačítko *VIRTUÁLNÍ MODEL* objektem *uicontrol* se stylem *pushbutton*, s pozicí a rozměrem [90 110 160 50], s přiřazenou funkcí *callback*, která aktivuje příslušný zdrojový M-soubor pro otevření virtuálního modelu (*m_dvapolreg_simulace*, *m_ctyropolreg_simulace*, *m_pidreg_simulace* a *m_rozvetvenareg_simulace*) a dojde k otevření virtuálního modelu Batyskaf s navolenými parametry.
- V podřízeném okně pro čtyřpolohovou regulaci jsou definovány posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení mezí $e1$ a $e2$ pomocí objektů *uicontrol* se stylem *slider*, o umístění a rozměru [150 510 80 20], maximu $max\ 10$ a minimu $min\ 0$, s krokem *sliderstep* [0.01 0.01], hodnotou *value* $abs(E1)$, s přiřazenou funkcí *callback* pro načtení zdrojových souborů *m_ctyropolreg_e1* a *e2*.
- V podřízeném okně pro PID regulaci jsou definovány posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení jednotlivých složek P , I a D

regulátoru PID a radiobuttony pro volbu typu regulátoru (P, I, D, PI, PD a PID). Vše je opět propojeno *callbacky* na příslušné zdrojové M-soubory (*m_pidreg_p*, *m_pidreg_i*, *m_pidreg_d*, *m_pidreg_preg*, *m_pidreg_ireg*, *m_pidreg_dreg*, *m_pidreg_pireg*, *m_pidreg_pdreg*, *m_pidreg_pidreg*). V tomto okně je také definováno tlačítko pro přepočítání parametrů PID regulátoru $r0-Ti-Td$ na $P-I-D$ objektem *uicontrol* se stylem *pushbutton*, o pozici a rozměru [270 425 55 15], s funkcí *callback*, která aktivuje otevření nového okna definovaného v M-souboru *m_prepocet_z_PID.m*.

- V podřízeném okně pro rozvětvenou regulaci jsou obdobným způsobem definovány posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení jednotlivých složek P , I a D hlavního regulátoru PID a složek P a I pomocného PI regulátoru. Všechny posuvníky opět využívají přiřazených funkcí *callback* s odkazem na příslušné M-soubory *m_rozvetvenareg_hp*, *m_rozvetvenareg_hi*, *m_rozvetvenareg_hd*, *m_rozvetvenareg_pp* a *m_rozvetvenareg_pi*.

Vytvořené grafické uživatelské prostředí virtuálního modelu systému „Batyskař“ s aktivovaným podřízeným ovládacím oknem pro rozvětvenou regulaci s pomocnou regulovanou veličinou lze vidět na Obr. 24.



Obr. 24 – Vytvořené grafické uživatelské prostředí virtuálního modelu systému „Batyskař“

- **Okno Přepočít PID parametrů** – je definováno v M-souboru *m_prepecet.m*.
 - Okno je tvořeno dialogovým oknem *d_prepecet=dialog* s umístěním a rozměry [686 590 410 230], barvou [0 0.4 0.2] a pojmenováním *PŘEPOČET PID PARAMETRŮ*.
 - V okně jsou vytvořeny čtyři rámy (*PID*, *r0TiTd*, *prPID* a *pr0TiTd*) pomocí objektů *uicontrol* se stylem *frame*, na pozici [20 20 170 80], barvou pozadí [1 0.7 0] a popředí [0 0 0].
 - Dále jsou v okně definovány posuvníky, hodnoty posuvníků a popisy posuvníků pro nastavení jednotlivých složek *P*, *I* a *D* regulátoru PID s přiřazenými funkcemi *callback*, které aktivují M-soubory *m_prepecet_p*, *m_prepecet_i*, *m_prepecet_d*, *m_prepecet_r0*, *m_prepecet_Ti* a *m_prepecet_Td*.
 - V M-souboru jsou definovány vztahy pro přepočít jednotlivých parametrů a výsledné zobrazení je provedeno objekty *uicontrol* se stylem *text* s odkazem na jednotlivé stringy *num2str (pr0)* atd.
 - Dále jsou zde definována tlačítka *Zadej*, které změní parametry regulátoru na zadané. Jsou tvořena objekty *uicontrol* se stylem *pushbutton*, umístěním a rozměry [150 105 40 15], s funkcemi *callback*, které aktivují M-soubory *m_prepecet_zadej_PID* a *m_prepecet_zadej_r0TiTd* s interním skriptem pro nastavení zvolené přepočtené hodnoty posuvníku do ovládacího okna zvoleného typu regulace.
 - Dále je zde vytvořeno tlačítko *MIMO ROZSAH*, které se zobrazí v případě zadání hodnot složek *P*, *I* a *D* mimo rozsah PID regulátoru. *Callbackem* na M-soubor *m_prepecet_mimo* se aktivuje vytvoření dialogového okna se zobrazenými povolenými rozsahy jednotlivých složek *P*, *I* a *D*.

Pro názornost jsou uvedeny vztahy (19) a (20) pro vyjádření konstant regulátoru PID a jejich analogie ($r_0-T_i-T_d \Leftrightarrow P-I-D$):

- v klasickém vyjádření konstant regulátoru:

$$u = r_0 e + \frac{r_0}{T_i} \int_0^t e(\tau) d\tau + r_0 T_D \frac{de}{dt} \quad G_{PID}(s) = r_0 + \frac{r_0}{T_i s} + r_0 T_D s \quad (19)$$



- ve vyjádření konstant regulátoru dle MATLABu:


$$u = P e + I \int_0^t e(\tau) d\tau + D \frac{de}{dt} \quad G_{PID}(s) = P + \frac{I}{s} + D s \quad (20)$$

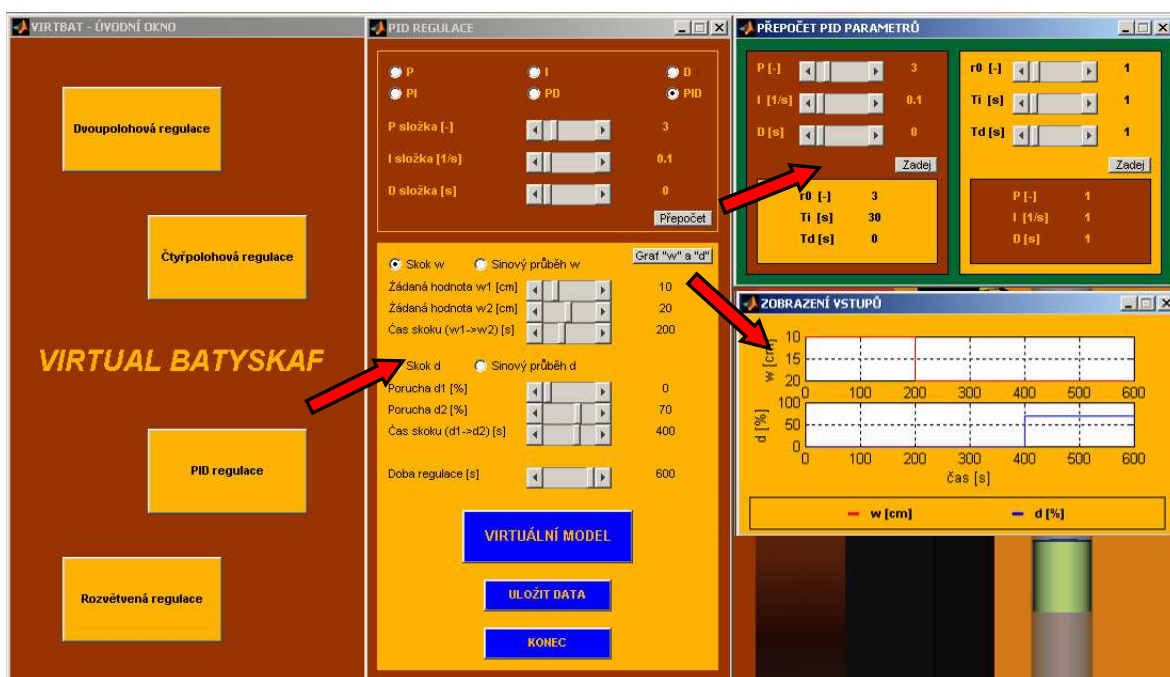
- **Okno Zobrazení vstupů** – je definováno v M-souboru *m_zadani.m*.
 - Okno je tvořeno dialogovým oknem *d_zadani=dialog* s umístěním a rozměry [686 590 400 200], barvou [1 0.7 0] a pojmenováním *ZOBRAZENÍ VSTUPŮ*.
 - V okně je nedefinováno zobrazení grafů žádané veličiny *w* a poruchy *d* pomocí objektů *axes* s pozicí a rozměry [0.15 0.7 0.75 0.2]). Data jsou do grafu načítána vyvolaným interním skriptem MATLABu a vykreslena pomocí příkazů *plot(zt,zw,'r')* a *plot(zt,zd,'b')*, s popisky os *xlabel('čas [s]')*, *ylabel('w [cm]'* a *'d[%]')* a se zapnutou mřížkou grafu *grid on*.
 - Pod grafy je nedefinováno zobrazení legendy s příslušnými popisky *w [cm]* a *d [%]* jednotlivých veličin pomocí objektů *uicontrol* se stylem *text* a odpovídajícími barvami pozadí [1 0.7 0] a popředí [0 0 0].

Ovládání virtuálního modelu

Pro spuštění virtuálního modelu je nutno nejprve spustit aplikaci MATLAB a do pracovního adresáře namapovat cestu, kde jsou uloženy všechny zdrojové soubory virtuálního modelu systému „Batyskař“. Poté postupujeme následujícími kroky:


- **Úvodní okno aplikace** – spuštění aplikace se provede zadáním příkazu *virtbat* do příkazového řádku a potvrzením klávesou *Enter*. Program MATLAB tímto provede načtení zdrojového M-souboru *virtbat.m*, který obsahuje výchozí vstupní parametry pro simulační modely a otevře hlavní okno grafického uživatelského prostředí s obrázkem grafické podoby virtuálního modelu, názvem *VIRTUAL BATYSKAF* a s úvodním menu čtyř tlačítek (*Dvoupolohová regulace*, *Čtyřpolohová regulace*, *PID regulace* a *Rozvětvená regulace*), které umožňují volbu typu regulace.
- **Okno zvolené regulace** - po aktivaci zvoleného tlačítka typu regulace (např. *PID regulace*) dojde k otevření ovládacího okna zvoleného typu regulace s menu pro volbu jednotlivých typů regulátorů (*P*, *I*, *D*, *PI*, *PD* a *PID*) a jejich parametrů (tlačítkem **Přepočít** se aktivuje okno pro přepočít konstant PID regulátoru z modelu ***P-I-D*** do modelu $r_0-T_r-T_d$ a naopak), posuvníky  lze nastavit hodnoty jednotlivých složek *P*, *I* a *D*. Pro dvoupolohovou regulaci lze posuvníky nastavit hodnota hysterese a radiobutony  zvolit maximální či poloviční akční zásah. Pro čtyřpolohovou regulaci lze posuvníky nastavit meze e_1 a e_2 . Pro rozvětvenou regulaci s pomocnou regulovanou veličinou lze posuvníky nastavit konstanty *P*, *I* a *D* hlavního PID regulátoru a konstanta *P* a *I* pomocného PI regulátoru.

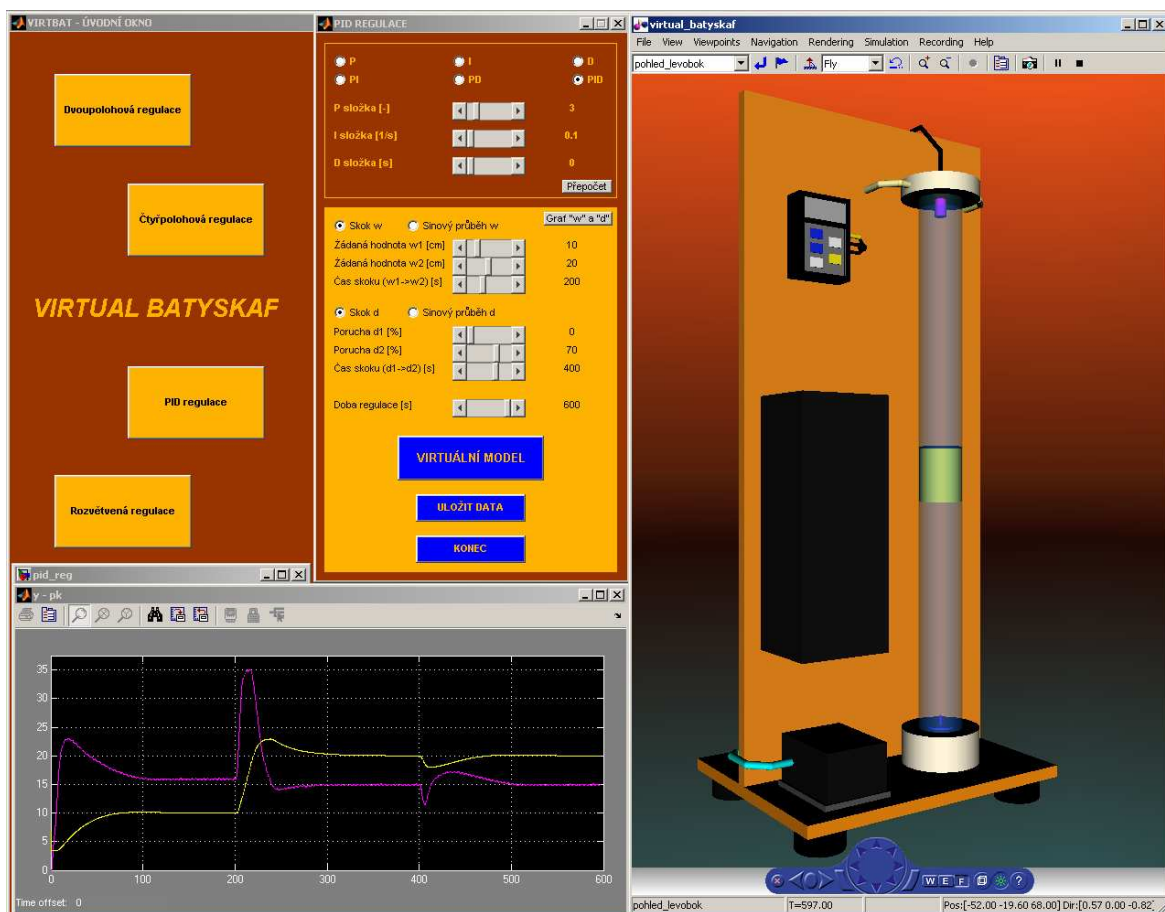
- **Nastavení parametrů regulace** – v oblasti nastavení žádané hodnoty regulované veličiny w lze radiobuttony  vybrat mezi skokovou změnou a periodickým sinovým průběhem a posuvníky lze nastavit parametry skoku i sinové křivky. Pro skok se nastavuje počáteční hodnota w_1 , konečná hodnota w_2 a čas skoku (přechodu mezi nimi). Pro periodický průběh se nastavuje amplituda, střední hodnota a perioda kmitů. Oblast nastavení poruchové veličiny d je obdobou nastavení žádané hodnoty regulované veličiny (jsou jen rozdílné rozsahy hodnot). Tlačítkem **Graf "w" a "d"** se otevře nové okno s vykreslením předvolených průběhů žádané hodnoty a poruchy do grafu. Posledním v pořadí je posuvník pro nastavování doby simulace (viz Obr. 25).



Obr. 25 - Postup volby typu regulace a zadání parametrů

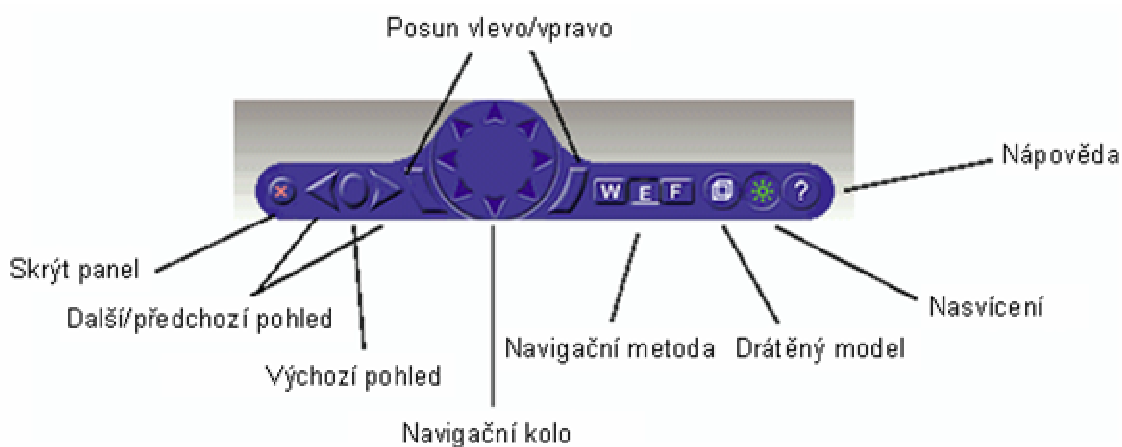
- **Otevření a spuštění virtuálního modelu** – tlačítkem **VIRTUÁLNÍ MODEL** se provede příkaz k načtení a otevření simulačního modelu pro zvolený typ regulace, automaticky také dojde k otevření virtuálního modelu v prostředí VRML Viewer a grafu, který průběžně zobrazuje aktuální hodnoty regulované veličiny (polohy plováčku). Vzhled uživatelského prostředí s virtuální simulací je na Obr. 26.

Spuštění virtuálního modelu se provede ikonou trojúhelníku *Start* na horní ovládací liště vpravo . Po jeho aktivaci se tlačítko automaticky přepne do režimu *Pause* k pozastavení simulace (opětovné spuštění se provádí aktivací tlačítka *Start*) a dále je v liště čtvercové tlačítko *Stop* k ukončení simulace.



Obr. 26 - Uživatelské prostředí virtuálního modelu systému "Batyskaf" v průběhu simulace

V průběhu simulace je možno měnit předdefinované pohledy zobrazení modelu v okně s menu pohledů vlevo nahoře a spodním ovládacím panelem modré barvy na Obr. 27 lze model přemísťovat, dynamicky otáčet či přibližovat a oddalovat, dále je možno vypnout nasvícení objektů či úplně vypnout textury anebo simulaci nahrát aktivací tlačítka *Recording* v horní nástrojové liště.



Obr. 27 - Navigační panel s popisky ovladačů [MATHWORKS, 2008]

- **Uložení dat** – v průběhu simulace nebo po jejím ukončení lze tlačítkem **ULOŽIT DATA** uložit aktuálně naměřená data průběhů simulací z grafu do souboru *s1.mat*, kde jsou k dispozici pro následné zpracování.
- **Ukončení zvoleného typu regulace** – tlačítkem **KONEC** je možno uzavřít najednou ovládací okno zvoleného typu regulace, graf průběhu simulace a virtuální prostředí modelu (zůstane otevřeno pouze úvodní okno aplikace *VIRTBAT*). Pokud je v aktuálním okamžiku spuštěna simulace ve virtuálním prostředí, je nejprve nutno simulaci ukončit ovládacím tlačítkem *Stop* v prostředí virtuálního modelu a poté aktivací tlačítka *KONEC* dokončit požadovanou akci.

4.6. Možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet

Vytvořený virtuální model je možno zhlédnout buď přímo v prohlížeči *VRT Viewer* nebo vzdáleně prostřednictvím webového prohlížeče, který formát VRML podporuje. Existuje i možnost zpřístupnění modelu v prostředí Internet/Intranet bez podpory formátu VRML prostřednictvím samostatné *stand-alone* aplikace *Orbisnap*, popřípadě je možno sledovat záznam činnosti virtuálního modelu ve formátu *.avi*.

Zpřístupnění na hostitelském serveru ve formátu VRML

Pro platformy *PC*, *Internet Explorer*, *Netscape Navigator*, *Opera* či *Mozilla* je produkt *VRT* vybaven přípojkou *VRML plug-in (Blaxxun Contact)*, které lze použít jako alternativní prohlížeč virtuálního prostředí za podpory *MATLABu* a *Simulinku*. *Blaxxun Contact VRML* je k dispozici ve dvou základních konfiguracích za použití ovladačů *OpenGL* a *Direct3D*. Pokud grafická karta a systémový ovladač umožní stereo režim celé obrazovky a k dispozici je zároveň odpovídající hardware pro stereo vidění (stereoskopické brýle), *Blaxxun Contact* zpřístupní také stereoskopický obraz.

Orbisnap – samostatná aplikace

Orbisnap je samostatná aplikace prohlížeče *VRT*, která umožňuje vizualizaci virtuálních modelů nebo předem nahraných animovaných souborů bez nutnosti prostředí *MATLAB* nebo produktů *VRT*. Vzdáleným připojením prostřednictvím internetové sítě

umožňuje *Orbisnap* zhlédnutí, předvedení a navigaci virtuálních modelů (neumožňuje simulaci).

Orbisnap je multiplatforma fungující na všech platformách, které VRT podporuje a není k ní potřeba licence Mathworks. Struktura zahrnuje spouštěcí, vykonávací a podpůrné soubory *Orbisnap*. Systémové nároky (hardware a software) jsou stejné jako pro aplikaci MATLAB a *Orbisnap* lze spustit na PC s operačním systémem Windows nebo Linux, funguje i s hardwarem Solaris se systémem UNIX, dále s hardwarem Apple Power Macintosh se systémem Mac OS X a s hardwarem Hewlett-Packard™ se systémem HP-UX.

Ke zhlédnutí virtuálního modelu z VRT serveru v aplikaci *Orbisnap* je potřeba:

- spuštěnou aplikaci MATLAB na VRT serveru
- verze VRT serveru, ke kterému se připojujeme, musí být kompatibilní s verzí aplikace *Orbisnap*, kterou máme spuštěnou
- Internet/Intranet spojení mezi uživatelským PC (na kterém je spuštěna aplikace *Orbisnap*) a hostitelským PC (kde je spuštěn MATLAB a VRT server). Pro zpřístupnění virtuálního modelu přes Internet je zároveň nutno ve VRT prohlížeči povolit prohlížení v nastavení modelu (*Simulation* ⇒ *Block Parameters* vybrat *Allow viewing from the Internet* a potvrdit *OK*).

Pro použití aplikace *Orbisnap* vzdáleně je nutno zmínit, že ačkoliv tento způsob zpřístupnění virtuálního modelu umožňuje jeho vizualizaci, žádná navigace ani interpretace v jednom prohlížeči nemá zpětnou vazbu v prohlížeči druhém (a naopak).

Aplikací *Orbisnap* není možno spustit či zastavit simulaci. Lze ji pouze zhlédnout (simulace virtuálního modelu probíhá na VRT serveru). Vzdáleným připojením aplikace *Orbisnap* na VRT server může dojít ke zpomalení simulace.

Off-line animace ve formátu .avi

VRT umožňuje zpřístupnění záznamů virtuálních scén řízených ze Simulinku či MATLABu pro pozdější zhlédnutí těchto animací *off-line* (tedy bez VRT prohlížeče). Tyto záznamy mohou sloužit k prezentacím, distribuci či archivaci výsledků. *Off-line* data animací mohou být uchována v následujících formátech:

- **Soubor 3-D VRML** – VRT kopíruje pohyby objektů a tato data uloží do souboru VRML použitím standardních interpolátorů VRML97. Žádné navigační pohyby, které jsou prováděny během záznamu animace, nejsou ukládány. Tyto soubory je možno zhlédnout v prostředí virtuální reality a zaberou mnohem méně prostoru na disku než soubory .avi.
- **Soubor 2-D Audio Video Interleave (AVI)** – VRT zaznamenává data animace do souboru .avi 2-D animace za použití objektů *vfigure*. Záznam přesně zachytí veškeré činnosti zobrazené během nahrávání.

5 Závěr

Prostudováním doporučené literatury a dostupných zdrojů (tzn. internetové odkazy se zadanou tematikou) jsem se při vytváření první části projektu seznámil s různými druhy realizovaných virtuálních laboratoří a modelů na vysokých školách. Při vytváření virtuálních komponent má tvůrce možnost použít různé druhy softwarových aplikací, které mu jsou nápomocny k dokonalému zpracování konkrétního virtuálního modelu dle zadaných kritérií. Realizace laboratoří a modelů jsou v řadě případů podobné (Java applety, Flash animace) a mezi nimi jsou použita odlišná softwarová prostředí (ISES WEB Control, MATLAB VRML Editor).

V druhé části projektu jsem nastudoval anglické manuály k toolboxům programového systému MATLAB, konkrétně k toolboxům MATLAB Compiler, MATLAB Builder for .NET, MATLAB Report Generator, MATLAB Virtual Reality a popsal vlastnosti a funkce programového systému MATLAB i jeho vybraných toolboxů.

V další části projektu jsem se seznámil s laboratorním systémem „Batyskař“ a popsal jeho strukturu a způsob ovládání. Poté jsem provedl matematický popis této soustavy a pomocí nástroje MATLAB Simulink vytvořil simulační modely pro čtyři vybrané druhy regulace. Funkčnost simulačního modelu byla ověřena porovnáním chování soustavy při simulované a reálné PID regulaci na laboratorním zařízení. Následně jsem v prostředí VRML Editoru, pomocí jeho nástroje V-Realm Builder 2.0, vytvořil vizuální podobu modelu soustavy, provedl její propojení se simulačními modely, vytvořil grafické uživatelské prostředí a popsal systém ovládání pro nově vzniklý virtuální model systému „Batyskař“, který umožňuje názornou simulaci čtyř druhů regulací: dvoupolohovou, čtyřpolohovou a PID regulaci a rozvětvenou regulaci s pomocnou regulovanou veličinou. Virtuální model umožňuje různá nastavení jednotlivých parametrů regulace i regulátorů a sledování vlivu změny jejich nastavení na regulační pochod. V průběhu simulace model umožňuje sledování aktuálních hodnot na grafu a ukládání naměřených dat průběhu regulace do souboru k pozdějšímu využití. V závěrečné fázi projektu jsem popsal možnosti zpřístupnění virtuálního modelu v prostředí Internet/Intranet.

Z pohledu uživatele jsou virtuální laboratoře a modely názornými a funkčními pomůckami pro snazší pochopení spojitosti diferenciálních rovnic a simulačních modelů při popisu reálných systémů nejen studentům, ale i široké laické veřejnosti, jelikož umožňují nahlédnout a zkoumat princip technologických procesů či zařízení, která jsou za běžných podmínek těžko přístupná nebo zcela nedostupná. A to vše v pohodlí z domácího počítače s internetovým připojením.

6 Seznam použité literatury

BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.

DORF, C. R. & BISHOP, H. R. *Modern Control Systems*. London: Prentice-Hall, Inc. 2001. 831p. ISBN 0-13-030660-6.

FARANA, R. AJ. *Programová podpora simulace dynamických systémů. Sběrka řešených příkladů*. 1 vyd. Ostrava: KAKI 1996, 114 s. ISBN 80-02-01129-5.

GRYGÁREK, P. *Virtlab*. [cit. 2009-09-05]. Dostupné z www <URL: http://www.cs.vsb.cz/vl-wiki/index.php/Virtu%C3%A1ln%C3%AD_laborato%C5%99_po%C4%8D%C3%ADta%C4%8Dov%C3%BDch_s%C3%ADt%C3%AD>

HOFREITER, M. *Laboratorní úloha Batyskaf* [cit. 2009-09-26]. Dostupné z www <URL: <http://www.ar-batyskaf.wz.cz>>

HOFREITER, M. *Virtuální laboratoř*. [cit. 2009-08-20]. Dostupné z www <URL: <http://www.fsid.cvut.cz/cz/u12110/ar/>>

HUMUSOFT. *MATLAB*. [cit. 2009-09-15]. Dostupné z www <URL: <http://www.humusoft.cz/>>

LUSTIG, F. *Projekt e-laboratoř*. [cit. 2009-08-21]. Dostupné z www <URL: <http://www.ises.info/index.php/cs/>>

MATHWORKS. *User's Guide - MATLAB Compiler 4, MATLAB Builder NE 3, MATLAB Report Generator 3, Virtual Reality Toolbox 4*. [cit. 2010-03-15]. PDF format. [DVD] Available from WWW: <URL: <http://www.mathworks.com>>.

NOSKIEVIČ, P. *Modelování a identifikace systémů*. 1. vyd. Ostrava : MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.

PEŠKE, M. *3D animace teplotovzdušného modelu*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2008. 41 stran. Bakalářská práce, vedoucí: Smutný, L., konzultant: Babiuch, M.

ŠKUTA, J. *Návody do cvičení H303*. [cit. 2009-09-05]. Dostupné z www <URL: <http://352lab.vsb.cz/ServerFinalVer/Laboratore.htm>>

VALAS, M. *Softwarová podpora výuky základů automatizace*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2006. 68 stran. Bakalářská práce, vedoucí: Wagnerová, R.

WAGNEROVÁ, R. *Using Matlab for Laboratory Model Control*. In Proceedings of the International Workshop Modelling and Simulation in Management, Informatics and Control MOSMIC '2001. Ed. M. Alexik. Suľov, Slovak Republic : University of Žilina, October 9-10, 2001, pp. 57-62. ISBN 80-7100-883-4.

WAGNEROVÁ, R. & DRLÍK, M. *Řízení technologických procesů v reálném čase*. Ostrava: FS VŠB-TUO, 2004. Available from web: <URL: http://www.352.vsb.cz/uc_texty/RealCasMatlab/default.htm>

WAGNEROVÁ, R. & HÉDL, L. *Hypertextová podpora výuky v oblasti automatického řízení*. Ostrava: FS VŠB-TUO, 2003. Available from web: <URL: http://www.352.vsb.cz/uc_texty/AutoRizMatlab/index.html>

ZAPLATÍLEK, K. DOŇAR, B. *MATLAB - tvorba uživatelských aplikací*. 1. vyd. Praha: BEN, 2004. 215 s. ISBN 80-7300-133-0.