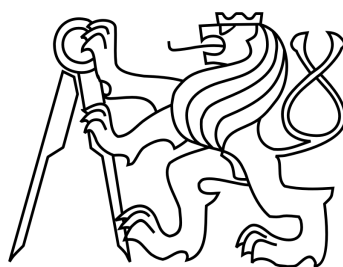


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY



BAKALÁŘSKÁ PRÁCE

Laboratorní modely a Real-Time Windows
Target

Autor: Radek Soukup

Praha, 2013

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne

.....

Podpis

Poděkování

Rád bych poděkoval především svému vedoucímu panu Ing. Fukovi za vyčerpávající množství informací, které mi předával v průběhu práce. Také bych rád poděkoval společnosti Humusoft za kvalitní technickou podporu.

Název práce: Laboratorní modely a Real-Time Windows Target

Autor: Radek Soukup

Katedra: Katedra řídicí techniky FEL ČVUT v Praze

Vedoucí bakalářské práce: Ing. Jindřich Fuka

e-mail vedoucího: fuka@fel.cvut.cz

Abstrakt

Cílem práce je přepracování počítačových modelů, které slouží k ovládání reálných systémů v laboratořích Katedry řídicí techniky FEL ČVUT v Praze. Počítačové modely byly vytvořeny v prostředí Matlab/Simulink za použití Real-Time (RT) toolboxu, vyvinutého společností Humusoft. Je-li s novou verzí Matlabu R2012b jsou funkce RT toolboxu součástí Real-Time Windows Target (RTWT) toolboxu, společnost Humusoft ukončila podporu RT toolboxu a dále nebude vyvíjen. Pro přechod na nový operační systém Windows 7 64bit bylo nutné ovládací simulinkové modely upravit pro RTWT toolbox a prakticky vyzkoušet. Součástí práce je rovněž návod na vytvoření jednoduchého modelu v RTWT toolboxu a seznámení s měřicími kartami pro PC.

Klíčová slova: Model systému, řídicí systém, Matlab, Simulink, řízení v reálném čase

Title: Laboratory models and Real-Time Windows Target

Author: Radek Soukup

Department: Department of Control Engineering FEL CTU in Prague

Supervisor: Ing. Jindřich Fuka

Supervisor's e-mail address: fuka@fel.cvut.cz

Abstract

The main goal of the work is remake of computer's models used for controlling real-time systems at laboratory of Department of Control Engineering FEL CTU in Prague. Computer's models were created in Matlab/Simulink environment using Real-Time (RT) toolbox developed by Humusoft. The reason of remaking the models was that the Real-Time Windows Target (RTWT) toolbox includes RT toolbox functionality in new version of Matlab R2012b which means that Humusoft doesn't continue in developing. New models were created and tested in RTWT toolbox to be able to run on Windows 7 64bit platform. The work also contains tutorial of making easy model in RTWT toolbox and introduction of Data Acquisition (DAQ) card for PC.

Keywords: System model, control system, Matlab, Simulink, real-time controlling

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Radek Soukup**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Laboratorní modely a Real-Time Windows Target**

Pokyny pro vypracování:

1. Seznamte se s určenými výukovými laboratorními modely pro automatické řízení v laboratořích katedry řídicí techniky, které byly dlouhodobě monitorovány a řízeny z prostředí programu Simulink s využitím Real-Time Toolboxu (dále RTT) Matlabu.
2. Jelikož od verze Matlabu R2012b není již RTT podporován, vytvořte a vyzkoušejte nové vzorové ovládací modely v Simulinku s jeho doporučenou náhradou Real-Time Windows Target (dále RTWT).
3. Popište stručně postup vytvoření nového ovládacího modelu pomocí RTWT. Zpracujte převodní tabulky mapování portů používaných IO karet od firem Advantech a Humusoft a ověřte plně jejich funkčnost.
4. Vytvořte popisy nových ovládacích Simulinkových modelů

Seznam odborné literatury:

firemní literatura www.mathworks.com; www.humusoft.cz
Pavel Karban: Výpočty a simulace v programech Matlab a Simulink, Computer Press Praha, 2010

Vedoucí: Ing. Jindřich Fuka

Platnost zadání: do letního semestru 2012/2013

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 14. 5. 2013

Obsah

Abstrakt	iii
Zadání práce	iv
Seznam obrázků	vii
Seznam tabulek	ix
1 Úvod	1
2 Vývojové prostředí Matlab	2
2.1 Simulink a Real Time Windows Target	2
3 Měřicí karty pro PC	4
3.1 Humusoft MF614	5
3.2 Advantech PCI-1711	7
4 Vytvoření modelu s použitím RTWT	9
5 GUI pro ovládací modely	14
5.1 Interaktivní nástroj GUIDE	14
5.1.1 Callback funkce	16
5.1.2 Ovládání Simulinkového modelu	17
5.1.3 Event listener funkce	17
6 Popis laboratorních modelů	19
6.1 Vodárny	20
6.2 Servo Amira	22
6.3 Kyvadlo	23
6.4 Kulička na tyči	24
6.5 Model vznášení	26
6.6 Tepelný systém	28
6.7 Servo S2	29
6.8 Vodní elektrárna	30
6.9 Rotační kyvadlo	32
7 Testování modelů	34

8 Závěr	35
Literatura	36
Příloha	I

Seznam obrázků

2.1	Příklad využití Real Time Windows target toolboxu, převzato z [2]	3
3.1	Měřicí systém	4
3.2	Porovnání single-ended vstupu s diferenciálním zapojení	5
3.3	Měřicí karta MF614, převzato z [5]	6
3.4	Označení pinů konektorů	6
3.5	Měřicí karta PCI-1711U, převzato z [6]	8
4.1	Spuštění Simulinku	10
4.2	Knihovny bloků Simulinku	10
4.3	Tvorba modelu v RTWT toolboxu	11
4.4	Parametry bloků	11
4.5	Rozsahy vstupních signálů analogového výstupu	12
4.6	Počet platných kanálů karty MF614	13
5.1	Spuštění nástroje GUIDE	15
5.2	Návrh vzhledu GUI	15
5.3	Navržené GUI k modelu Vodárny	18
6.1	Model Vodárny	20
6.2	Simulinkové schéma modelu Vodárny	21
6.3	Model Servo Amira	22
6.4	Simulinkové schéma modelu Servo Amira	23
6.5	Model Kyvadlo	23
6.6	Simulinkové schéma modelu Kyvadlo	24
6.7	Model Kuličky na tyči	25
6.8	Simulinkové schéma modelu Kulička na tyči	25
6.9	Model Vznášení	26
6.10	Simulinkové schéma modelu Vznášení	27
6.11	Model Tepelný systém	28
6.12	Simulinkové schéma modelu Tepelný systém	29
6.13	Model Servo S2	29
6.14	Simulinkové schéma modelu Servo S2	30
6.15	Model Vodní elektrárna	31
6.16	Simulinkové schéma modelu Vodní elektrárna	32

6.17 Model Rotační kyvadlo	32
6.18 Simulinkové schéma modelu Rotační kyvadlo	33
7.1 Chyba průběhu z analogových vstupů karty MF614	34

Seznam tabulek

3.1	Parametry karty MF614	7
3.2	Parametry karty PCI-1711	8
5.1	Definice událostí, převzato a upravno z [1]	18
6.1	Zapojení vstupů/výstupů - Vodárny	21
6.2	Zapojení vstupů/výstupů - Servo Amira	22
6.3	Zapojení vstupů/výstupů - Kyvadlo	24
6.4	Zapojení vstupů/výstupů - Kulička na tyči	25
6.5	Zapojení vstupů/výstupů - Vznášení	27
6.6	Zapojení vstupů/výstupů - Tepelný systém	28
6.7	Zapojení vstupů/výstupů - Servo S2	30
6.8	Zapojení vstupů/výstupů - Vodní elektrárna	31
6.9	Zapojení vstupů/výstupů - Rotační kyvadlo	33

1 Úvod

Současné laboratorní modely, které jsou umístěné v laboratořích Katedry řídicí techniky FEL ČVUT v Praze, umožňují velmi efektivně usnadnit výuku aplikací teoretických znalostí v praxi. Studenti pak mohou snáze pochopit jednotlivé souvislosti jak v teorii řízení, tak v modelování systémů. Na základě matematicko-fyzikálních vztahů, které popisují reálný systém, mohou vytvářet ovládací modely v prostředí Matlab/Simulink. Reálné systémy lze nejen ovládat, ale také jednoduše navrhovat řídicí algoritmy a získané závislosti porovnávat s teoretickými poznatky, a to vše v jediném vývojovém prostředí. Veličiny, jako jsou otáčky motoru, úhel natočení hřídele, příkon tepelného systému a další, lze nastavovat v grafickém rozhraní Simulinku a výstupy ze systému může uživatel jednoduše zobrazit na monitoru bez potřeby dalšího měřicího zařízení. Vše probíhá v reálném čase a změna parametrů v počítači se okamžitě projeví na reálném modelu. V laboratořích řídicí techniky je dlouhodobě využíváno vývojové prostředí Matlab/Simulink s nadstavbou pro Real-Time komunikaci (tzv. toolbox).

Funkce Real-Time toolboxu, který vyvinula společnost Humusoft, jsou od verze Matlabu R2012b integrovány do Real-Time Windows Target toolboxu. Při přechodu na platformu Windows 7 64bit by nebyly modely v laboratořích K23 a K26 podporovány, a proto bylo zapotřebí současné ovládací modely využívající RT toolboxu přepracovat a sestavit nové ovládací modely umožňující komunikaci prostřednictvím RTWT toolboxu. Dále bylo nutné ověřit funkčnost a případně upravit ovládání, aby byla zajištěna jednoduchost a přehlednost modelu. V práci je obsažen rovněž popis jednotlivých laboratorních modelů z laboratoře K23 a návod, jak vytvořit jednoduché grafické uživatelské rozhraní pro ovládání modelů. Popisy laboratorní modelů pro laboratoř K26 nejsou součástí práce, ale ovládací modely byly upraveny a některé vyzkoušeny.

2 Vývojové prostředí Matlab

Program Matlab, vyvíjený společností MathWorks, je široce využívaný nástroj pro technické výpočty jak v komerční, tak zejména v akademické sféře. Původně se jednalo o programovací jazyk usnadňující matematické výpočty bez potřeby znalosti jazyku Fortran¹. Tento nástroj prošel řadou změn a Matlab lze dnes považovat za interaktivní vývojové prostředí pro tvorbu algoritmů a aplikací s podporou grafického uživatelského rozhraní. Mezi základními knihovnami funkcí, nazývané také toolboxy, je možné Matlab doplnit o další knihovny, které rozšiřují jeho využití v mnoha technických odvětvích, jako je řídicí technika, zpracování signálů, modelování systémů, statistické výpočty, sběr dat a další. Jednou z nadstavbou Matlabu pro simulaci a modelování dynamických systémů je nástroj Simulink. Pomocí knihoven Simulinku lze modelovat dynamické soustavy od jednoduchých modelů až po rozsáhlé fyzikální systémy. Simulinkové modely jsou sestaveny z bloků, které jsou vzájemně propojeny. Za účelem propojení Simulinku s reálným systémem, pro měření a sběr dat, byl vyvinut Real-Time Windows Target toolbox (RTWT), jehož funkce bude popsána v následujícím odstavci.

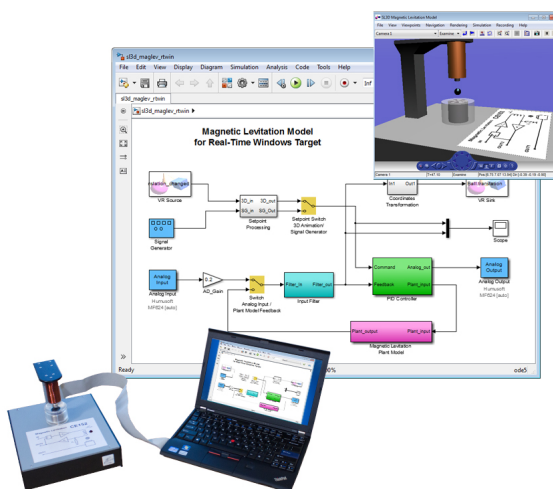
2.1 Simulink a Real Time Windows Target

Jakmile je v prostředí Simulink vytvořen model, který lze reprezentovat obyčejnými diferenciálními rovnicemi (ODEs), je potřeba určit v jakých časových okamžicích má být proveden numerický výpočet těchto rovnic. K tomu slouží tzv. solver. V knihovnách Simulinku existují dva typy solveru. Pro výpočet ODEs lze použít Fixed-Step solver nebo Variable-Step solver. Časové okamžiky, ve kterých jsou ODEs vyhodnoceny, jsou dány aktuálním simulačním časem a hodnotou kroku (step size). Rozdíl tedy mezi oběma solvery je v tom, že Fixed-Step solver během simulace nemění hodnotu kroku a rovnice jsou vypočítávány v konstantních časových úsecích, zatímco Variable-Step solver mění svou hodnotu kroku v závislosti na dynamice systému. Pokud je systém statický a v čase se příliš nemění, solver nastaví delší časový krok. V systémech, kde dochází k častým změnám, je automaticky nastaven kratší časový krok [1].

Real-Time Windows Target toolbox umožňuje komunikovat s real-time modely z prostředí Simulink, a tím získávat data ze senzorů nebo ovládat akční členy. RTWT toolbox podporuje velké množství měřicích karet a rovněž je jeho součástí knihovna bloků, jako jsou analogové a digitální vstupy/výstupy, čítače, enkodéry a další. Aby mohl Simulink komunikovat s modelem, musí být nainstalováno v počítači tzv. Real-Time Windows jádro, které přiřadí modelu nejvyšší prioritu pro obsluhu procesorem a zároveň zařídí aktualizaci modelu v daných časových intervalech [2]. V době,

¹ Programovací jazyk, vyvinutý firmou IBM v 50. letech minulého století, pro vědecké výpočty a numerické aplikace

kdy není obsluhována procesorem real-time aplikace, jádro začne obsluhovat ostatní běžící procesy ve Windows. RTWT toolbox umožňuje spouštět modely ve dvou různých režimech: normal módu a external módu.



Obrázek 2.1: Příklad využití Real Time Windows target toolboxu, převzato z [2]

External mód

Poté, co je v Simulinku vytvořen model systému, kodér (Real-Time Workshop) vygeneruje C kód, který je převeden do spustitelné podoby. RTWT toolbox následně načte ovladač zařízení a provede spojení se Simulinkem v reálném čase. Tím je zajištěna synchronizace s hardwarem se vzorkovací frekvencí až 20 kHz. Musí být však použit pouze Fixed-Step solver.

Normal mód

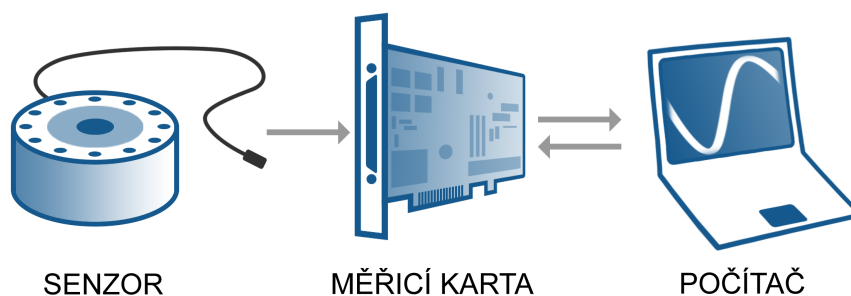
V tomto módu je spuštěn proces, kde běží simulační algoritmus, který vypočítává diferenciální rovnice. Odděleně je vytvořen druhý proces, kde je spuštěn v reálném čase ovladač zařízení. Zde je omezena vzorkovací frekvence na 500 Hz. Jelikož není simulační proces synchronizován s hardwarem, může docházet ke ztrátě vzorků. Z toho důvodu bloky obsahují parametr Maximum missed ticks, který určuje, při jakém počtu ztracených vzorků dojde k ukončení simulace. V porovnání s external módem lze použít i Variable-Step solver a není zapotřebí Simulink kodér.

Více informací o Real-Time Windows Target toolboxu lze nalézt na [2]. V kapitole 4 je popsán návod na vytvoření ovládacího modelu v RTWT toolboxu, kde je předpokládána znalost měřicích karet pro PC. Čtenáři, kteří jsou obeznámeni s tímto tématem, mohou následující kapitolu vynechat a přejít rovnou na kapitolu 4.

3 Měřicí karty pro PC

V dnešní době, kdy ke zpracování dat jsou téměř výhradně využívány počítače, je zapotřebí vhodným způsobem upravit analogové signály. Tyto signály mohou představovat nejrůznější fyzikální veličiny, jako jsou teplota uvnitř pece, tlak nebo průtok čerpadla, otáčky motoru a jiné. K tomuto účelu je možné využít měřicí kartu, označovanou jako zařízení, která provede potřebné přizpůsobení signálů pro zpracování v PC. Měřicí karta obsahuje jeden nebo více analogových vstupů, ale může obsahovat i digitální vstupy, digitální výstupy, čítače, časovače nebo může analogové signály sama generovat. Takové karty jsou potom označovány jako multifunkční.

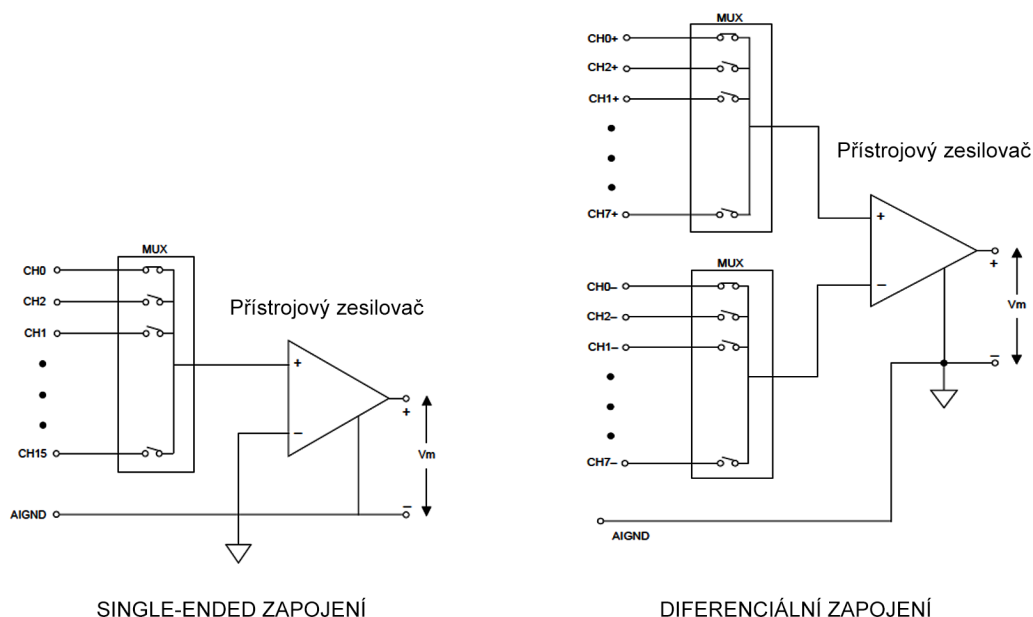
Měřicí karta tvoří rozhraní mezi senzorem dané veličiny a obslužným zařízením, ke kterému je připojena prostřednictvím sběrnice PCI, PCI Express, USB, FireWire nebo Ethernet [3]. Celý měřicí systém je zobrazen na Obr.3.1 Aby mohl být signál připojený na analogový vstup zpracován



Obrázek 3.1: Měřicí systém

v PC, musí být zdigitalizován, k čemuž slouží integrovaný čip zvaný A/D převodník. Vlastnosti A/D převodníku do značné míry určují parametry celé karty. Důležitými parametry jsou rozlišení a rychlost převodu daného převodníku. Obvykle je součástí karet jeden A/D převodník s rozlišením 8 až 24 bitů, ke kterému je připojen multiplexor a ten přepíná jednotlivé vstupy. To zapříčiní nižší vzorkovací kmitočet na jeden kanál, než udává výrobce. Existují i speciální měřicí karty obsahující pro každý kanál vlastní A/D převodník, jejich nevýhodou je ovšem vysoká cena. Součástí karet je aproximační nebo Sigma-Delta A/D převodník a nejpřesnější měřicí karty někdy používají integrační A/D převodník [4].

Nejčastěji jsou karty vybaveny napěťovými vstupy, které pracují v závislosti na způsobu uzemnění v režimu SE (Single-ended) nebo DIF (diferenciálně). Existují také modely, kde lze změnit typ vstupu přepínací propojkou nebo softwarově [4]. Single-ended vstupy mají vždy jeden kontakt uzemněn, jak je možné vidět na Obr. 3.2.



Obrázek 3.2: Porovnání single-ended vstupu s diferenciálním zapojení

Single-ended zapojení je výhodné pro signály s hodnotou řádově jednotek voltů. Pro nízkonapěťové signály se začne v signálu projevovat souhlasný šum, a proto je vhodnější použít diferenciální zapojení vstupů podle Obr. 3.2.

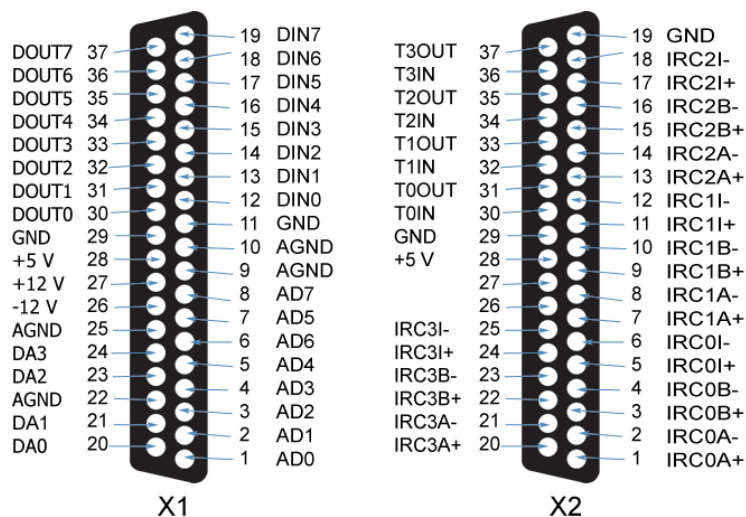
3.1 Humusoft MF614

Tato karta patří mezi multifunkční zařízení s 8 analogovými vstupy a 4 analogovými výstupy, jejichž rozlišení je 12 bitů. Zatímco vstupní rozsahy je možné softwarově nastavit jak bipolárně na $\pm 10\text{V}$, $\pm 5\text{V}$, tak unipolárně na $0 - 10\text{V}$, $0 - 5\text{V}$. Výstupní rozsah analogových výstupů je nastaven trvale na $\pm 10\text{V}$ a vzorkovací kmitočet je omezen do 100kHz . Součástí karty je i 8 digitálních vstupů a výstupů s TTL úrovněmi. Výhodou je rovněž možnost zpracovávat data až ze 4 inkrementálních snímačů v diferenciálním nebo single-ended režimu. Na kartě je také integrovaný čítač/časovač CTS9513.



Obrázek 3.3: Měřicí karta MF614, převzato z [5]

Počítač komunikuje s kartou po PCI sběrnici a samotná karta je propojena s modelem přes dvojici konektorů DB 37. Popis jednotlivých pinů je vyznačen na Obr. 3.4. Kompletní specifikace měřicí karty MF 614 byly získány z oficiálních stránek výrobce [5] a jsou shrnuty v tabulce 3.1.



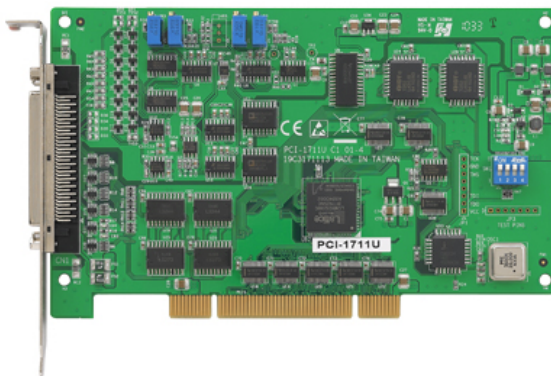
Obrázek 3.4: Označení pinů konektorů

Analogové vstupy	
Kanály	8 single-ended
A/D převodník	12 bitový, doba převodu 10 μ s
Vstupní rozsahy	± 10 V, ± 5 V, 0 – 10 V, 0 – 5 V
Spouštění	programově
Ochrana vstupů	max. ± 16 V trvale
Analogové výstupy	
Kanály	4 kanály, 12-bitů
Výstupní rozsah	± 10 V,
Výstupní proud	10 mA max
Digitální vstupy/výstupy	
Vstupní linky	8, s úrovněmi TTL
Výstupní linky	8, s úrovněmi TTL
Časovač/čítač	
Časovač	CTS9513
Rozlišení	50 ns
Vstupy inkrementálních snímačů	
Vstupní kanály	4, single-ended nebo diferencialní
Vstupy	A, B, Index
Vstupní frekvence	max 2.5 MHz
Všeobecné údaje	
Odběr proudu	100 mA @ +5 V, 50 mA @ +12 V, 50 mA @ -12 V
Pracovní teplota	0 - 50 °C (32 - 140 °F)
Konektor	2 x DB-37
Sběrnice	PCI

Tabulka 3.1: Parametry karty MF614

3.2 Advantech PCI-1711

Měřicí karta od společnosti Advantech lze rovněž zařadit do kategorie multifunkční. Její součástí je 16 analogových vstupů a 2 analogové výstupy v provedení single-ended s rozlišením 12 bitů. V případě potřeby je možné nastavit také rozsah analogových výstupů ve dvou rozsazích 0 až +5 V a 0 až +10 V. Karta obsahuje 16 digitálních vstupů a výstupů kompatibilní s TTL logikou, jeden časovač/čítač s rozlišením 16 bitů a referenční frekvencí 10 MHz.



Obrázek 3.5: Měřicí karta PCI-1711U, převzato z [6]

Propojení karty a počítače je realizováno přes PCI sběrnici, model je připojen ke kartě pomocí 68-pinového konektoru SCSI. Bližší informace a parametry karty PCI-1711 jsou uvedeny v [6] a shrnuty v tabulce 3.2.

Analogové vstupy					
Kanály	16 single-ended				
A/D převodník	12 bitový				
Vstupní rozsahy	±10 V	±5 V	±2.5 V	±1.25 V	±0.625 V
Přesnost (% z FSR ± 1LSB)	0.1	0.1	0.2	0.2	0.4
velikost FIFO	1024 vzorků				
Ochrana vstupů	30 Vp-p				
Vstupní impedance	300 MΩ/5 pF				
Analogové výstupy					
Kanály	2 kanály / 12 bitů				
Výstupní rozsah	0 až + 5 V		0 až + 10 V		
Digitální vstupy/výstupy					
Vstupní linky	16, s úrovněmi TTL				
Výstupní linky	16, s úrovněmi TTL				
Časovač/čítač					
Rozlišení	16 bitů				
Všeobecné údaje					
Odběr proudu	850 mA - 1 A				
Pracovní teplota	0 ~ 60 °C (32 ~ 140 °F)				
Konektor	68-pinový SCSI				
Sběrnice	PCI				

Tabulka 3.2: Parametry karty PCI-1711

4 Vytvoření modelu s použitím RTWT

Následující informace jsou získány z oficiálního webu společnosti MathWorks [2], kde je možnost se dozvědět více konkrétních informací pro danou úlohu. Tato kapitola obsahuje pouze základní seznámení s Real-Time Windows Target toolboxem a lze v ní nalézt nejen návod, jak postupovat při tvorbě jednodušších modelů, ale rovněž může posloužit při vývoji komplexnějších modelů. Pro vytvoření modelu je zapotřebí mít v prostředí Matlab nainstalovaný Real-Time jádro (kernel). Bez něj není možné navázat komunikaci, jak bylo uvedeno v 2.1. Pokud byl RTWT toolbox nainstalován je potřebné jádro umístěno na disku počítače a jeho samotnou instalaci lze provést zadáním příkazu

```
rtwintgt -install nebo rtwintgt -setup
```

V případě, že instalace proběhla úspěšně, objeví se následující zpráva

```
The Real-Time Windows Target kernel has been success-  
fully installed.
```

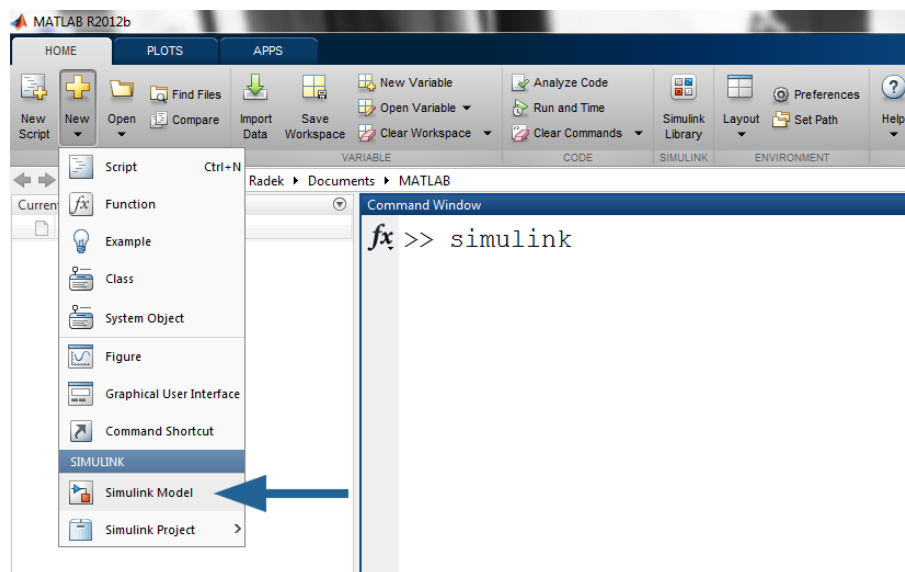
Po následném restartu počítače lze ověřit, zda bylo jádro nainstalováno příkazem

```
rtwho
```

Bližší informace lze získat v nápovědě příkazem

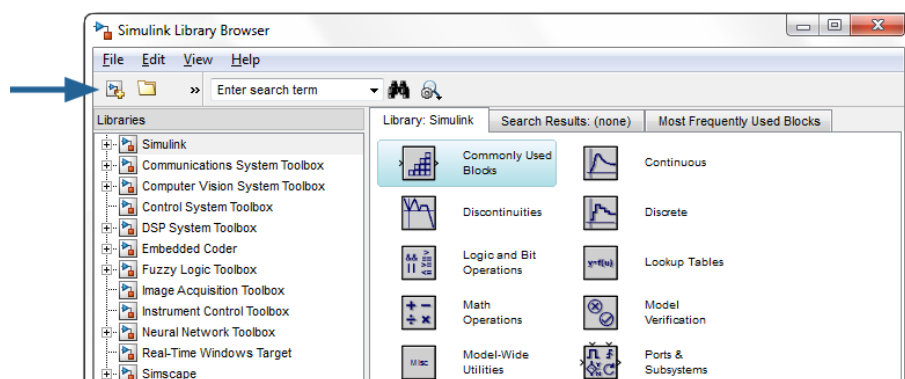
```
help rtwintgt
```

Real-time jádro zůstane nečinné, aby mohl operační systém obsluhovat běžné aplikace a je aktivován pouze v případě spuštění modelu v Simulinku. Jakmile je jádro nainstalováno, lze začít s vytvořením modelu podle Obr. 4.1 zadáním příkazu `simulink` nebo kliknutím v panelu nástrojů na `new a Simulink Model`.



Obrázek 4.1: Spuštění Simulinku

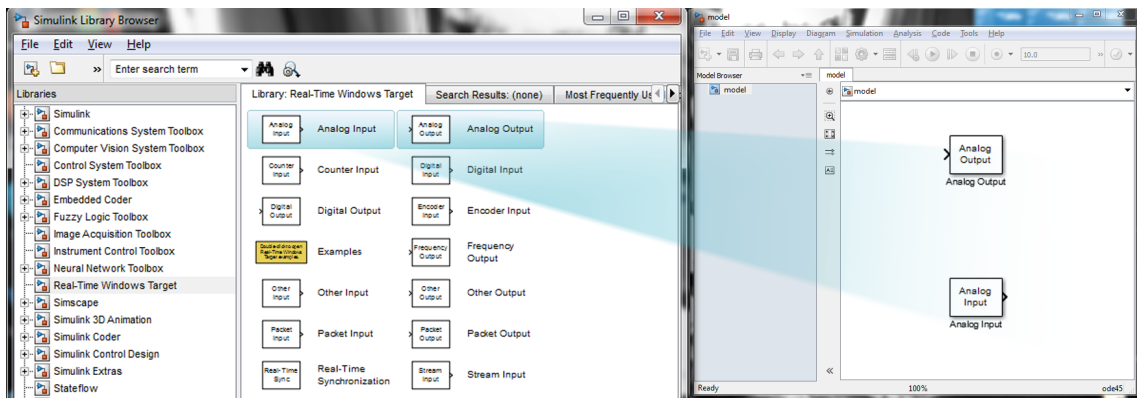
Otevře se okno, kde v levé části lze najít nainstalované toolboxy a vpravo se nacházejí jednotlivé bloky viz Obr. 4.2. Pracovní prostředí Simulinku se otevře v panelu nástrojů kliknutím na `New model`, kam lze následně jednoduše přetahovat dané bloky.



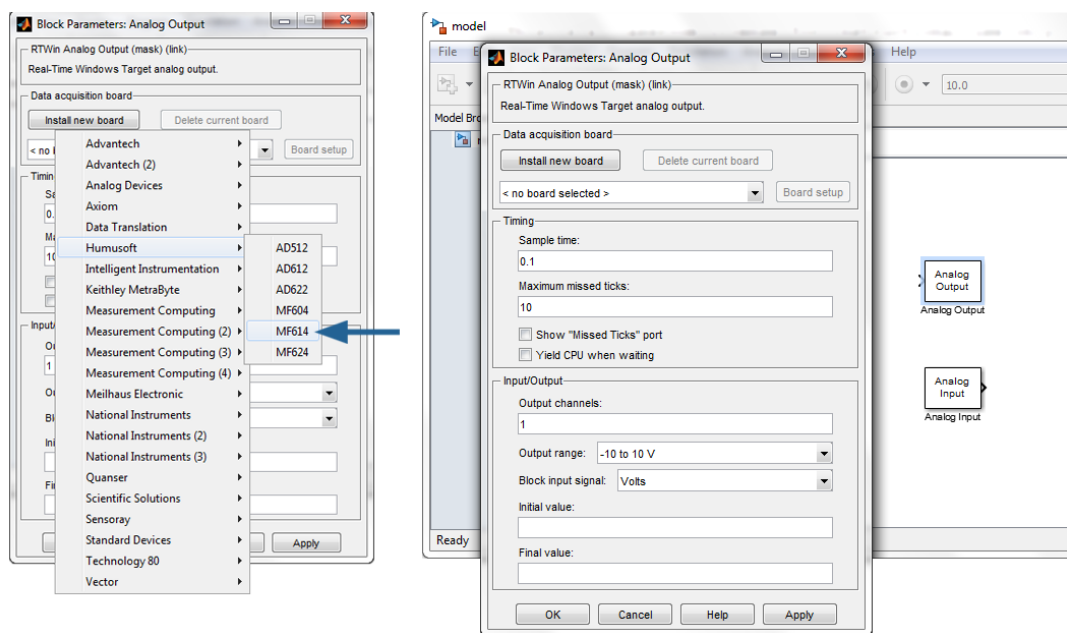
Obrázek 4.2: Knihovny bloků Simulinku

Pro praktickou ukázkou je na Obr. 4.3 zobrazen jednoduchý příklad modelu s využitím knihovny bloků RTWT toolboxu. Model může sloužit k ovládání stejnosměrného motoru a měření jeho rychlosti pomocí tachogenerátoru. Jako měřicí karta bude použita karta MF614 od českého výrobce Humusoft. Aby bylo možné ovládat motor, je zapotřebí na vstup motoru přivést stejnosměrné napětí, které bude generováno měřicí kartou ze svého analogového výstupu. Proto je v nabídce RTWT vybrán blok `analog output`. Výstupem tachogenerátoru je analogové napětí, které odpovídá otáčkám motoru. Toto napětí je následně přivedeno na analogový vstup měřicí karty. V RTWT je analogový vstup reprezentován blokem `analog input`.

Dvojklikem se provede nastavení daného bloku, kde je nutné nastavit několik parametrů podle Obr. 4.4. Nejprve musí být určeno, prostřednictvím jaké karty model komunikuje s Matlabem. V poli `Data acquisition board` je potřeba zvolit `Install new board`. V nabídce se objeví seznam výrobců s jejich podporovanými kartami. V tomto případě je zvolena karta MF614.



Obrázek 4.3: Tvorba modelu v RTWT toolboxu



Obrázek 4.4: Parametry bloků

Dále se nastaví vzorkovací perioda v poli `Sample time`, která určuje, jak často je na analogový výstup vyslána analogová hodnota (napětí motoru). V poli `Maximum missed ticks` se nastavuje maximální počet vzorků, který může být ztracen během simulace, jak bylo popsáno v odstavci 2.1. Jakmile je hodnota vyšší než uvedená v tomto poli, je simulace následně ukončena. V případě, že je potřeba zobrazovat ztracené vzorky lze zaškrtnout pole `Show "Missed Ticks" port`. U bloku se objeví port umožňující zobrazit tyto vzorky např. pomocí bloku `Scope`. Pokud jádro čeká na odpověď od hardwaru, lze v mezinápravě přidělit procesor ostatním programům, které čekají na obsluhu. Tato funkce se povoluje v poli `Yield CPU when waiting`. Všechny výše uvedené parametry se týkaly pouze časování. Pro správnou komunikaci je nejen nutné nastavit výstupní rozsahy a kanály, na které bude signál vyslán, ale také počáteční a koncové hodnoty výstupů. Zápis jednotlivých kanálů do pole `Output channels` je známý z prostředí Matlab. Např.:

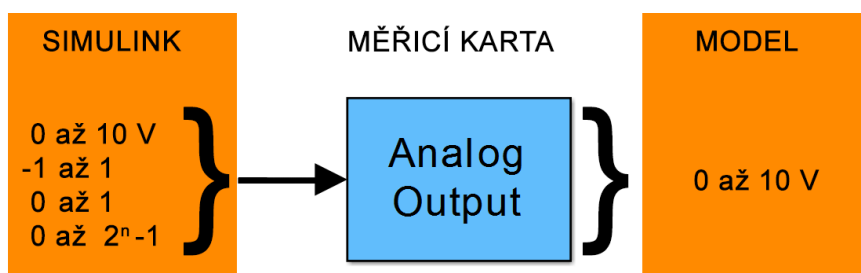
[1 2 3 4 5] ...

[1, 3, 4, 7] ...

[1 : 5] ...

[2] = 2 ...

V poli `Output range` se nastavuje výstupní rozsah analogového výstupu. Karta MF614 poskytuje pouze rozsah -10 V až 10 V . Kromě výstupního rozsahu lze nastavit i rozsah hodnot, které budou v Simulinku reprezentovat jednotlivé hodnoty na výstupu, jak je znázorněno na Obr. 4.5.



Obrázek 4.5: Rozsahy vstupních signálů analogového výstupu

- **Volts**

výstupní hodnoty jsou reprezentovány přímo hodnotami napětí (10 V na výstupy odpovídá 10 V na vstupu)

- **Normalized bipolar**

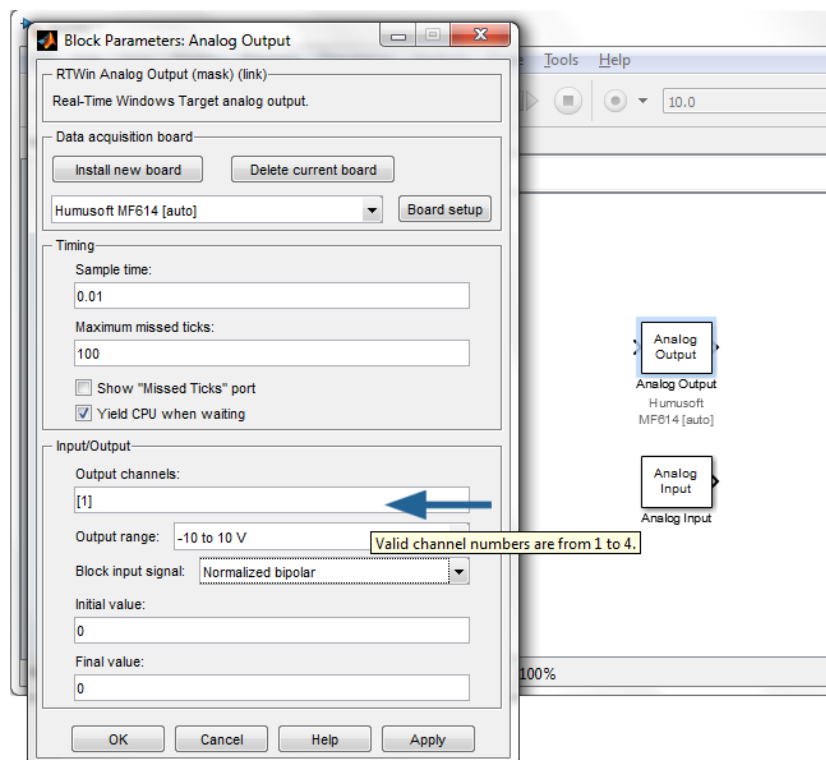
výstupní hodnoty jsou převedeny na interval -1 až 1 (10 V odpovídá 1 a -10 V odpovídá -1)

- **Normalized unipolar**

výstupní hodnoty jsou převedeny na interval 0 až 1 (10 V odpovídá 1 a -10 V odpovídá 0)

- **Raw**

výstupní hodnoty jsou reprezentovány číslem z intervalu 0 až $2^n - 1$, kde n označuje rozlišení převodníku



Obrázek 4.6: Počet platných kanálů karty MF614

Jak je možné vidět z Obr. 4.6, při najetí kurzoru do pole `Output channels` se objeví popis s počtem platných kanálů. V RTWT toolboxu jsou kanály číslovány od 1, na rozdíl od měřicí karty MF614, kde jsou číslovány od 0. Pokud při návrhu je fyzicky využít kanál 0, tak v nastavení bloku musí být označen číslem 1. Potvrzení nastavení se provede tlačítkem `Apply` nebo `Ok`. Obdobným způsobem lze nastavit i blok analogového vstupu nebo kterýkoli blok obsažený v knihovně RTWT toolboxu. Odlišnosti nastávají pouze v části `Input/Output`, kde význam jednotlivých parametrů lze intuitivně rozpoznat. V případě, že jsou všechny bloky nastaveny, lze připojit zdroj signálu např. blok `Step`. Dále bloky typu `Scope` pro zobrazení přijatých dat z modelu a spustit simulaci. V kapitole 5 bude ukázáno, jak vytvořit jednoduché grafické uživatelské rozhraní (GUI).

5 GUI pro ovládací modely

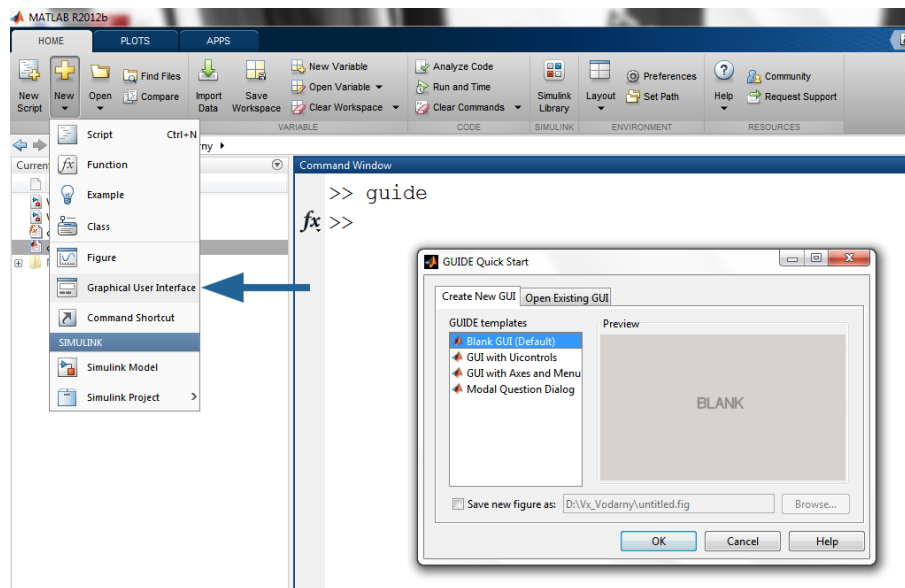
Když je ovládací model vytvořen, je možné ho ovládat klasicky z prostředí Simulink. Při změně některého z parametrů během simulace, musí být otevřen příslušný blok, kde je hodnota parametru vepsána a uložena. Někdy je zapotřebí, aby model ovládal člověk, který není seznámen s ovládacím prostředím Simulink a má pouze základní znalost práce s počítačem. Stejně jako lze ovládat model ze Simulinku, může být model ovládán pomocí grafického uživatelského rozhraní (GUI). V Matlabu existují dva způsoby, jak vytvořit GUI

- tvorba pomocí zápisu zdrojového kódu do editoru
- návrh s využitím interaktivního nástroje GUIDE

První uvedený způsob je vhodný zejména pro uživatele, kteří vyžadují maximální kontrolu nad svým návrhem. Programátor vytvoří funkci nebo skript, kde nadefinuje jednotlivé komponenty a určí, jaké příkazy budou vykonány při interakci s uživatelem. Každý způsob tvorby GUI je vhodný na různá použití podle počtu ovládacích prvků a složitosti návrhu. Zde bude popsán vývoj jednoduchého GUI pomocí nástroje GUIDE, který je součástí prostředí Matlab. Tento způsob je založený na jednoduchém přetahování komponent do připraveného okna, čímž je vytvořen základní vzhled bez psaní zdlouhavých zdrojových kódů. Podrobný postup návrhu je popsán v následujícím odstavci. Informace o návrhu GUI byly získány z [7], kde lze nalézt velké množství ukázkových příkladů.

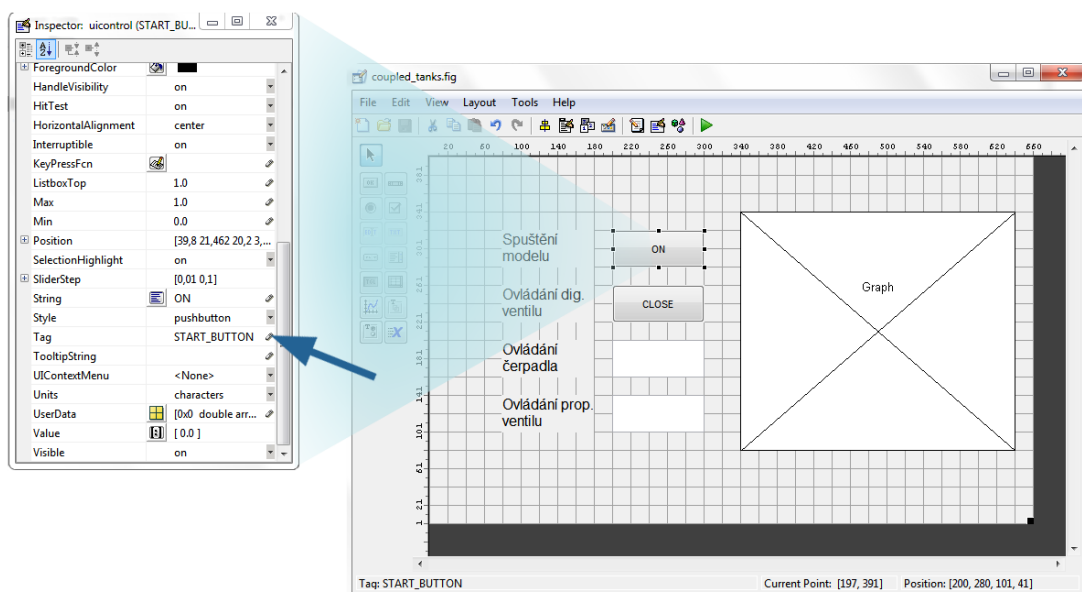
5.1 Interaktivní nástroj GUIDE

Jak již bylo řečeno, nástroj Guide je standardně součástí prostředí Matlab a lze jej spustit zadáním příkazu `guide` do příkazového řádku nebo kliknutím na `Graphical User Interface` v záložce `New`. Zde si uživatel může vybrat plochu s předem připravenými prvky nebo prázdnou plochu `Blank GUI`. Pro účely znázornění celého postupu vývoje GUI bude vytvořeno jednoduché grafické rozhraní pro ovládání simulinkového modelu Vodárny.



Obrázek 5.1: Spuštění nástroje GUIDE

Tvorba GUI probíhá tak, že nejprve je v nástroji Guide vytvořen vzhled grafického rozhraní, kde jsou umístěny ovládací a zobrazovací prvky. Z nabídky je vybrán komponent a přetáhnut do okna. Kliknutím na daný komponent se můžou měnit nejrůznější parametry, jako je velikost, barva, text na tlačítku, velikost nebo centrování písma a další. Každý blok obsahuje parametr Tag, který ho jednoznačně identifikuje a který se následně objeví ve vygenerovaném kódu. Na Obr. 5.2 je znázorněna tabulka parametrů tlačítka ON s Tagem START_BUTTON. Když je tvůrce spokojen se vzhledem GUI, může přistoupit k programování funkcí, které jsou vykonány při interakci s komponentami GUI.



Obrázek 5.2: Návrh vzhledu GUI

Je potřeba návrh uložit a tím Matlab automaticky vygeneruje zdrojový kód. V adresáři, kde

byl vytvořen a uložen návrh `coupled_tanks.fig` se objevil soubor se stejným názvem, ale příponou `.m`. Jinými slovy v souboru s příponou `.fig` je uložen vzhled GUI a v souboru s příponou `.m` je uložena výkonná část GUI, tj. jeho chování.

5.1.1 Callback funkce

Provázání GUI s akcemi uživatele je realizováno pomocí tzv. callback funkcí. Jakmile dojde k provedení akce, je zavolána příslušná callback funkce, která vykoná svůj kód. Ve vygenerovaném kódu se objeví vždy callback funkce ve tvaru

```
function tag_Callback (hObject, eventdata, handles)
```

U tlačítka ON z Obr. 5.2 byl nastaven Tag na `START_BUTTON`, a proto je jeho callback funkce ve tvaru

```
function START_BUTTON_Callback(hObject, eventdata, handles)
```

Každý komponent je charakterizován ukazatelem, nazývaný handle. Ukazatele všech prvků v GUI jsou uchovány ve struktuře `handles`, kde každému prvku odpovídá číslo. Vstupní parametr `hObject` je ukazatel na komponentu, jejíž akce vyvolala tuto funkci. Výpis struktury `handles` může vypadat např. takto

```
handles =  
  
figure1: 188.0018  
  Graph: 10.0016  
  text8: 9.0016  
  text7: 8.0016  
  text6: 7.0016  
  text5: 6.0016  
  DIG_VALVE_BUTTON: 5.0104  
  START_BUTTON: 4.0126  
  PROP_VALVE: 3.0060  
  GAIN: 189.0018  
  
output: 188.0018
```

Pro konkrétní handle je možné psát

```
handles.START_BUTTON
```

Pokud programátor zná handle na daný komponent GUI, může k jeho parametrům přistupovat pomocí příkazu `set` nebo `get`. Příkaz `get` vrátí strukturu všech parametrů, které se týkají daného komponentu. Pro konkrétní parametr je syntaxe následující

```
get(handles.START_BUTTON, 'string')
```

Tlačítko `START_BUTTON` má parametr `string`, který udává, co je napsáno na tomto tlačítku. Podobně funguje i příkaz `set`. Často je nutné sdílet informace mezi jednotlivými callback funkcemi. Vstupním parametrem callback funkcí je struktura `handles`, kam lze vkládat i proměnné, které je pak možné číst v každé funkci se vstupním parametrem `handles`. Tímto způsobem lze například vytvořit proměnnou `START_BUTTON_VALUE`, udávající, zda je tlačítko ve stavu zapnuto nebo vypnuto. Deklarace by vypadala následovně

```
handles.START_BUTTON_VALUE = 'ON'
```

5.1.2 Ovládání Simulinkového modelu

Podobně, jako se přistupuje ke komponentám GUI, se přistupuje k blokům Simulinkového modelu pomocí příkazů `set_param` a `get_param`. Syntaxe je obdobná

```
get_param('Vx_K23m_new/Pump', 'Gain')
```

```
ans =
```

```
0.45
```

První parametr udává jméno modelu, kde před lomítkem je uveden název modelu a za lomítkem název bloku. Druhý parametr představuje parametr bloku. Pokud je blok součástí subbloku, vloží se ještě název subbloku. Následující příkaz např. vypíše název ovladače k měřicí kartě

```
get_param('Vx_K23m_new/Coupled Tanks Vx/Analog Output', 'DrvName')
```

```
ans =
```

```
Advantech/PCI-1711
```

Téměř vždy je součástí simulinkového modelu nějaký zobrazovací blok pro výstupní veličinu, která je pro vývojáře zajímavá a chce, aby se během simulace zobrazovala také v GUI. Jeden ze způsobů, jak toho docílit je použít tzv. `event listener` funkce.

5.1.3 Event listener funkce

Jeden ze způsobů, jak pravidelně aktualizovat graf v GUI, který je např. svázán s výškou hladiny v nádrži, je použití funkce

```
add_exec_event_listener(blk, event, listener)
```

Funkce přiřadí bloku v Simulinku tzv. `listener` funkci, která je spuštěna vždy, když nastane předem definovaná událost. Lze také říci, že posluchač (`listener` funkce) naslouchá, která událost nastala. Pokud je událost shodná s přiřazenou událostí daného bloku, je spuštěn kód funkce.

blk Název bloku v Simulinku

event Událost, která způsobí vyvolání kódu viz tabulka 5.1

listener Představuje kód, který má být vyvolán. Může to být jednoduchý příkaz nebo handle na příslušnou funkci. Tato funkce musí mít dva vstupní parametry, první je run-time objekt simulinkového bloku a druhý je instance eventdata, která specifikuje run-time objekt a událost, která právě nastala

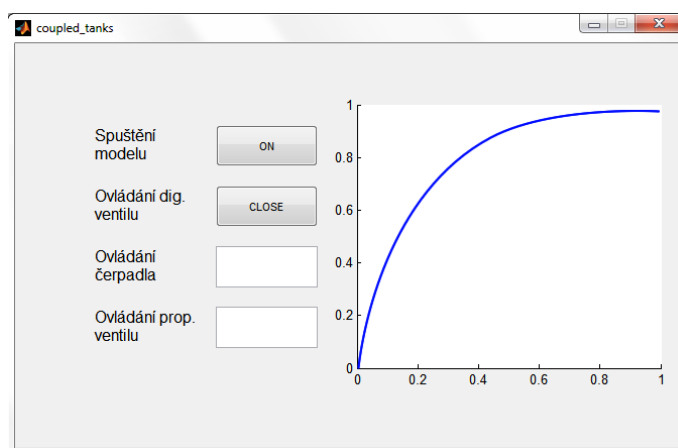
Událost	Popis
'PreDerivatives'	Nastane před výpočtem derivací bloku
'PostDerivatives'	Nastane po výpočtu derivací bloku
'PreOutputs'	Nastane před výpočtem výstupu bloku
'PostOutputs'	Nastane po výpočtu výstupu bloku
'PreUpdate'	Nastane před aktualizací stavů bloku
'PostUpdate'	Nastane po aktualizaci stavů bloku

Tabulka 5.1: Definice událostí, převzato a upraveno z [1]

Pokud chce vývojář neustále sledovat výšku hladiny v nádrži z bloku Display, musí tomuto bloku přiřadit listener funkci a určit, jaká událost ji má spouštět. Syntaxe bude následující

```
blk = 'Vx_K23m_new/Right height ' ;
event = 'PostOutputs' ;
listener = @localEventListener;
eventhandle = add_exec_event_listener(blk, event, listener)
```

Tuto funkci lze použít pouze v případě, že běží simulinkový model, proto je výhodné funkci vložit do spouštěcí funkce (StartFcn) simulinkového modelu. Jednoduché GUI může vypadat podle Obr. 5.3 Zdrojový kód je uveden v příloze na straně I.



Obrázek 5.3: Navržené GUI k modelu Vodárny

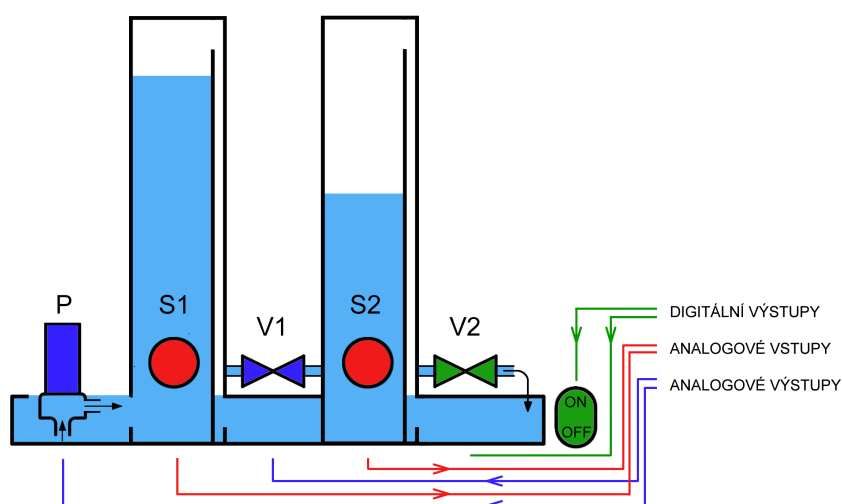
Návod má sloužit k základní orientaci ve vývoji grafického uživatelského rozhraní v prostředí Matlab. Uvedené příkazy byly vysvětleny na konkrétním příkladu modelu Vodárny, ale lze je jednoduše aplikovat na jakýkoliv jiný systém.

6 Popis laboratorních modelů

Modely jsou vytvořeny tak, aby bloky, které zajišťují komunikaci s měřicí kartou, byly součástí subbloku s názvem modelu. Do tohoto modelu vstupují jednotlivé vstupní a výstupní veličiny, které ovlivňují chování systému. Aby byla zajištěna jednotná vzorkovací frekvence u jednotlivých vstupních a výstupních bloků, je v subbloku nastaven parametr T_s . Tento parametr se nastaví poklepáním na subblok. Změna způsobí, že je parametr nastaven u všech bloků zároveň. K modelům v laboratoři K23 byly vytvořeny schémata v programu Adobe Photoshop, která jsou vyobrazena na subbloku. Tyto zdrojové obrázky musí být uloženy ve stejném adresáři jako ovládací simulinkový model. Veškeré veličiny, které reprezentují signály pro laboratorní model, jsou převedeny na hodnotu -1 až 1. Když např. uživatel chce ovládat motor v rozsahu 0 až 10 V, v Simulinku je nutné nastavit hodnotu 0 pro 0 V a 1 pro 10 V. Měřicí karta obsahuje několik vstupů/výstupů, jak je uvedeno v kapitole 3. Zatímco vstupy/výstupy (kanály) jsou v Simulinku číslovány od čísla 1, testované měřicí karty firem Humusoft a Advantech číslovají své kanály od 0. Proto je při sestavování modelu nutné si uvědomit, že pokud je ovládání motoru připojeno ke kartě na kanál 4, v Simulinku tomuto kanálu odpovídá kanál 5. Pro popis laboratorních modelů bylo čerpáno z [8, 9].

6.1 Vodárny

Výukový model vodáren představuje aplikaci pro řízení výšky hladin a průtoku, které jsou významným procesem v mnoha průmyslových odvětvích. Fyzikální systém je složen ze dvou vzájemně propojených nádrží, zásobníku vody, čerpadla a dvou elektricky ovládaných ventilů (V1 a V2). Jeden ventil je využit pro propojení nádrží a druhý je umístěn na výtoku z druhé nádrže. Model obsahuje kombinaci proporcionálního ventilu a digitálního v závislosti na použitém modelu vodárny.

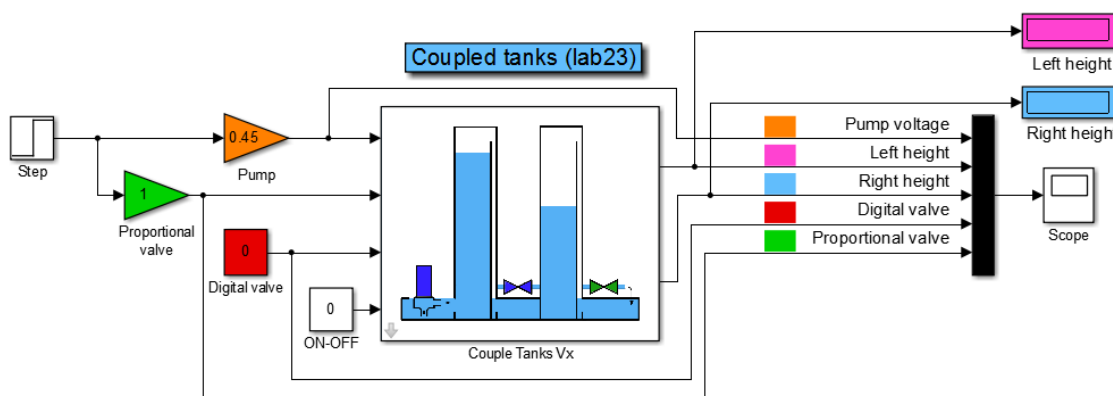


Obrázek 6.1: Model Vodárny

Zatímco digitální ventil umožňuje regulovat průtok pouze ve dvou polohách (otevřený / zavřený), proporcionálním ventilem lze nastavit průtok plynule. Výšky hladin v obou nádržích jsou měřeny tlakovými snímači (S1, S2), které převádí informaci o výšce hladiny na analogový signál, který je zobrazen na čelním panelu. V modelu je použito odstředivé čerpadlo (P), které lze ovládat vstupním napětím. Celý systém je připojen k měřicí kartě Advantech PCI-1711 a následně ovládán v programu Matlab/Simulink, kde je vytvořen model celé vodárny pomocí Real-Time Windows Target toolboxu. Z pohledu měřicí karty se v systému nacházejí dva analogové výstupy. Jeden slouží k ovládání otáček čerpadla a druhý ovládá proporcionální ventil, a tím reguluje průtok. Dále obsahuje analogové vstupy z tlakových senzorů a digitální výstupy pro zapnutí modelu a ovládání digitálního ventilu. V tabulce 6.1 je shrnuto zapojení jednotlivých vstupů/výstupů k daným analogovým a digitálním kanálům.

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Výška hladiny 1	1	-10 až 10 V	Normalized bipolar
	Výška hladiny 2	2		
Analogový výstup	Ovládání čerpadla	1	0 až 10 V	Normalized unipolar
	Ovládání prop. ventilu	2		
Digitální výstup	Ovládání modelu	1	0 nebo 1	
	Ovládání dig. ventilu	2		

Tabulka 6.1: Zapojení vstupů/výstupů - Vodárny



Obrázek 6.2: Simulinkové schéma modelu Vodárny

Na Obr. 6.2 je znázorněno simulinkové schéma pro ovládání modelu Vodárny. Význam jednotlivých bloků lze popsat následovně

ON-OFF Tímto blokem lze celý model zapnout nebo vypnout z prostředí Simulinku, kde hodnota 1 značí zapnutí a 0 vypnutí.

Pump Zeslabení jednotkového skoku Step lze nastavit v bloku Pump a určuje napětí přivedené na odstředivé čerpadlo.

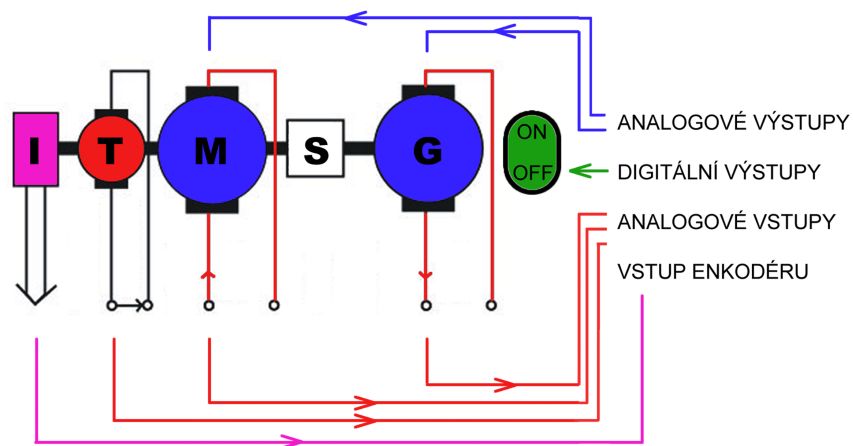
Proportional valve Tato veličina nastavuje míru otevření proporcionálního ventilu v rozsahu 0 až 1

Digital valve Určuje, zda bude během simulace digitální ventil uzavřen (0) nebo otevřen (1)

Veškeré měřitelné hodnoty lze pozorovat v bloku Scope nebo sledovat pouze výšku hladin v nádržích pomocí bloků Left/Right height. Kliknutím na blok Coupled Tanks Vx lze nastavit vzorkovací periodu T_s .

6.2 Servo Amira

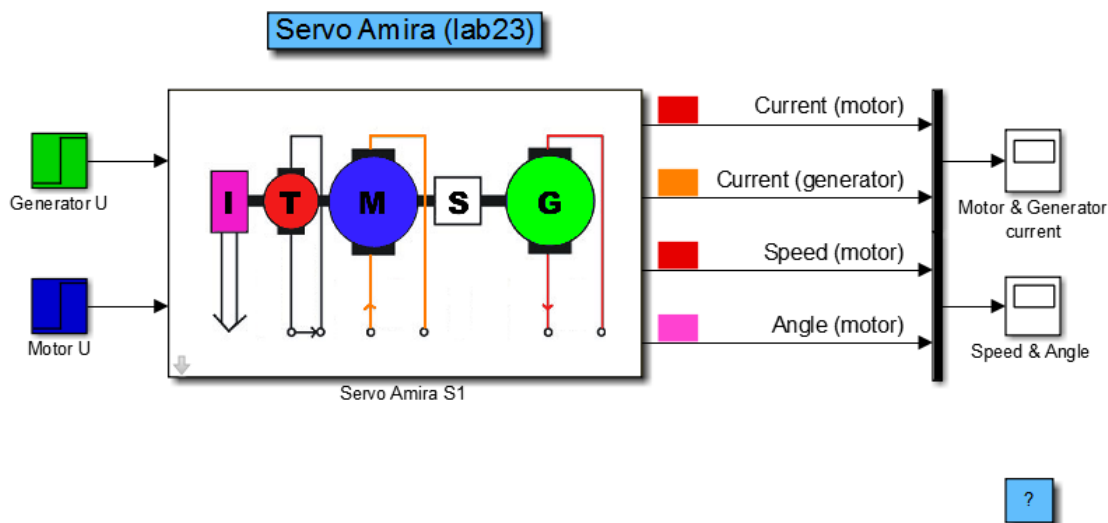
Výukový model je systém založený na spojení dvou identických stejnosměrných motorů, který umožňuje řízení rychlosti nebo úhlu natočení. S tímto problémem je možné se setkat zejména u aplikací vyžadující přesné řízení polohy, jako jsou např. robotické manipulátory. Součástí systému na Obr. 6.3 je motor (M) řízený vstupním napětím, jehož otáčky jsou měřeny tachogenerátorem (T). Pro potřeby měření přesného úhlu natočení je motor vybaven inkrementálním snímačem (I). Druhý motor (G) je vzájemně spojen s prvním motorem pomocí spojky (S) a slouží jako generátor. Model umožňuje měřit proud generátoru a proud motoru. Celý systém je připojen k multifunkční měřicí kartě MF614 firmy Humusoft, která je vhodná pro tuto aplikaci, protože její součástí je vstup pro inkrementální snímač.



Obrázek 6.3: Model Servo Amira

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Proud motoru	5	-10 až 10 V	Normalized bipolar
	Proud generátoru	6		
	Otáčky motoru	7		
Analogový výstup	Ovládání motoru	3	-10 až 10 V	Normalized bipolar
	Ovládání generátoru	4		
Digitální výstup	Ovládání modelu	6	0 nebo 1	
Vstup enkodéru	Úhel natočení	2		

Tabulka 6.2: Zapojení vstupů/výstupů - Servo Amira

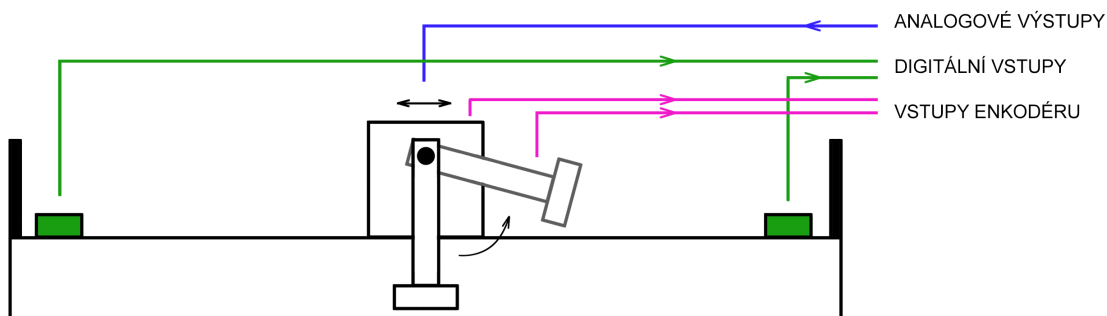


Obrázek 6.4: Simulinkové schéma modelu Servo Amira

Celý model je možné ovládat pomocí dvou napětí, napětí na motoru (Motor U) a napětí na generátoru (Generator U). Na výstupu je pak možné sledovat rychlost motoru a úhel natočení společně s průběhy proudů. Všechny veličiny jsou stejně jako u všech modelů převedeny na čísla 0 až 1.

6.3 Kyvadlo

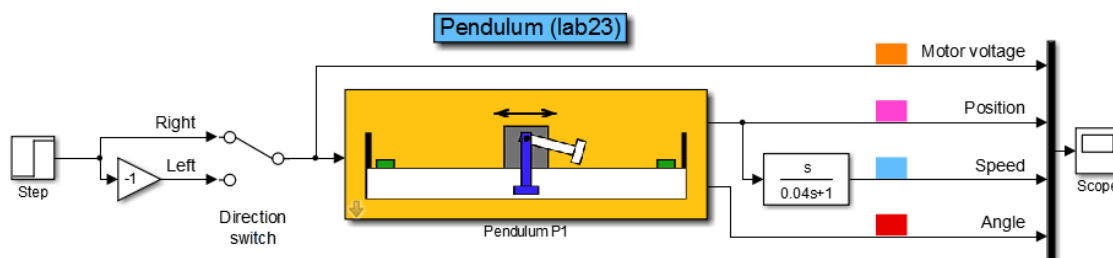
Laboratorní model představuje kyvadlo upevněné na pohyblivém vozíku, který je poháněn malým motorem. Poloha vozíku a samotného kyvadla je měřena dvěma inkrementálními snímači. Z důvodu bezpečnosti a ochrany hnacího kolečka je model opatřen softwarovými a hardwarovými dorazy. Softwarové dorazy jsou umístěny na spodní straně vozíku a při sepnutí ukončí simulaci. Podobnou funkci plní i hardwarové dorazy, které přímo odpojí motor od zdroje. Jedná se o poměrně jednoduchý systém, kde jediná veličina ovlivňující činnost modelu z pohledu obsluhy, je napětí na motoru. Zbylé veličiny udávají informaci o poloze vozíku, kyvadla a poloze vozíku na konci dráhy. Zapojení signálů k měřicí kartě MF614 je uvedeno v tabulce 6.3



Obrázek 6.5: Model Kyvadlo

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Vstup enkodéru	Inkrementální snímač kyvadla	2		
	Inkrementální snímač vozíku	1		
Analogový výstup	Ovládání motoru	1	-10 až 10 V	Normalized bipolar
Digitální vstup	Softwarové dorazy	1, 2	0 nebo 1	

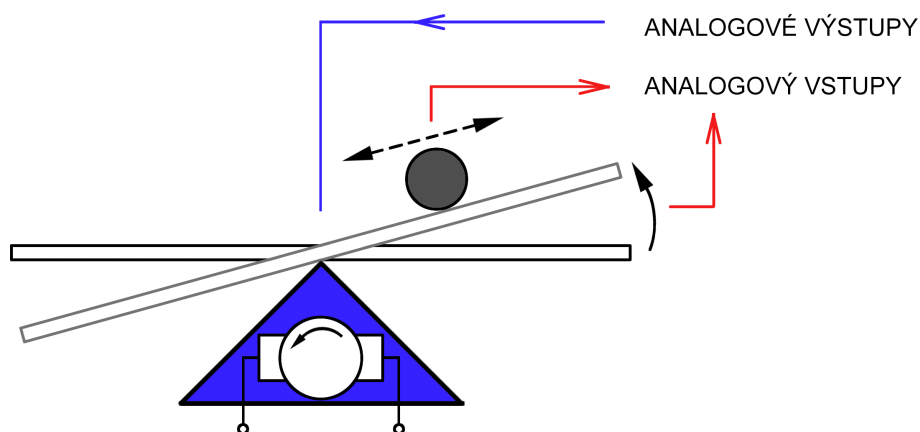
Tabulka 6.3: Zapojení vstupů/výstupů - Kyvadlo



Obrázek 6.6: Simulinkové schéma modelu Kyvadlo

6.4 Kulička na tyči

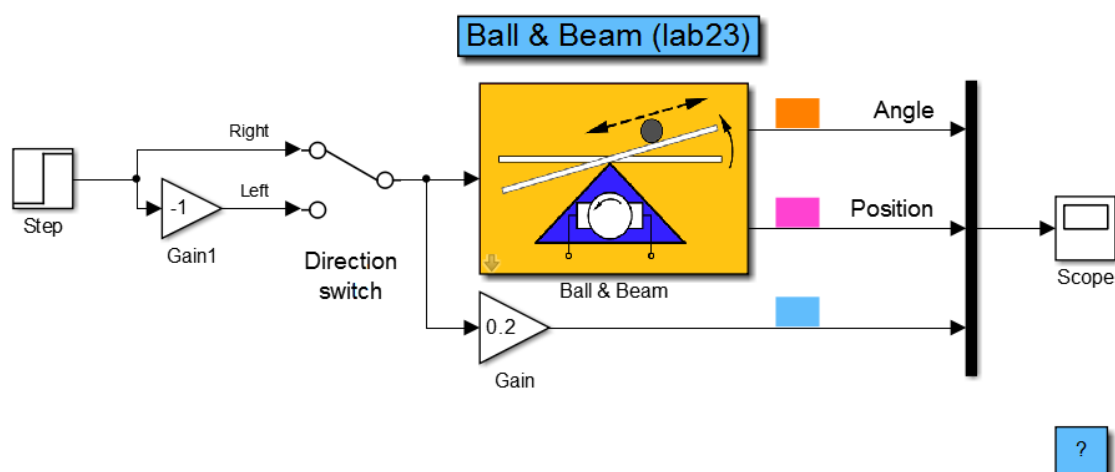
Model reprezentuje nelineární systém, jehož vlastností je nestabilita v otevřené smyčce. Uvedené chování vykazuje mnoho fyzikálních systémů jako je start raketoplánu nebo exotermní reakce v chemickém průmyslu. Součástí modelu je ocelová kulička, kolejnice umístěné na tyči. Náklon je řízen servomotorem, který je ovládán vstupním napětím. Výstupem ze systému je poloha kuličky společně s náklonem tyče. Schematicky je model znázorněn na Obr. 6.7. Model je připojen ke kartě MF614 a nachází se ve dvou verzích v laboratořích K23 a K26. Odlišnosti v zapojení kanálů jsou uvedeny v tabulce 6.4.



Obrázek 6.7: Model Kuličky na tyči

Typ	Funkce	K23	K26	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
		Kanál			
Analogový vstup	Poloha kuličky	1	1	-10 až 10 V	Normalized bipolar
	Úhel natočení tyče	2	3		
Analogový výstup	Ovládání motoru	1	1	-10 až 10 V	Normalized bipolar

Tabulka 6.4: Zapojení vstupů/výstupů - Kulička na tyči

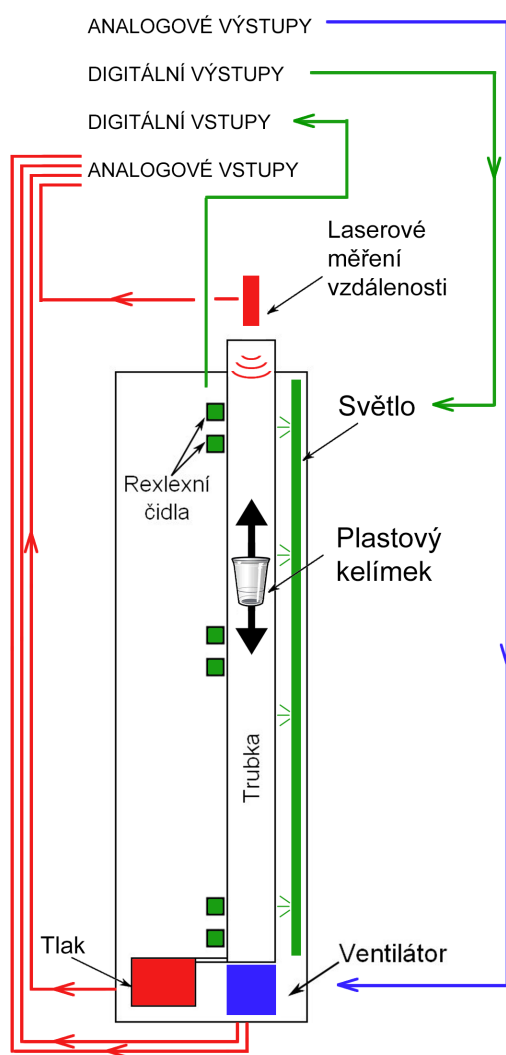


Obrázek 6.8: Simulinkové schéma modelu Kulička na tyči

Vstupní veličinou je napětí na servomotorku, kde kladná hodnota skoku způsobí, že se rameno natočí doprava. Aby se nemusela při každé změně směru měnit hodnota skoku v bloku Step, byl vložen do modelu přepínač. Změna směru tak probíhá pouze dvojklikem na blok Direction switch.

6.5 Model vznášení

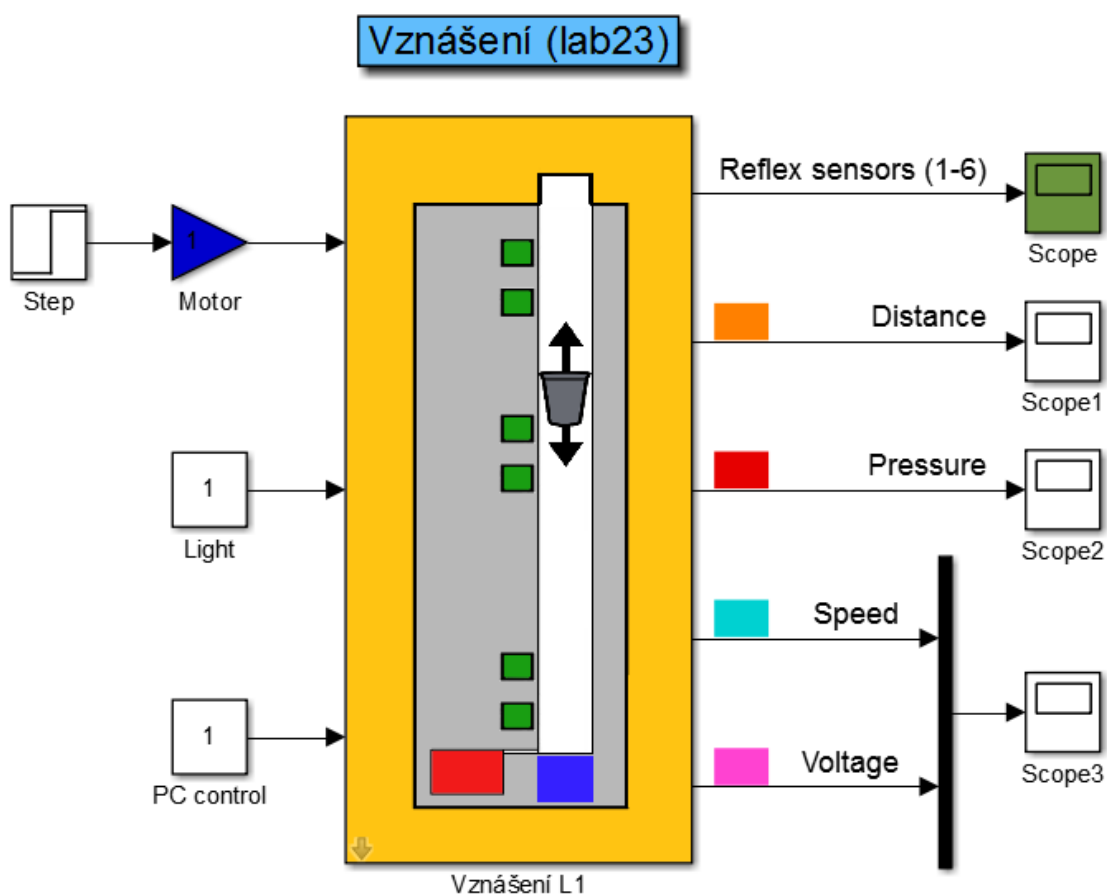
Tento jednoduchý model reprezentuje fyzikální systém sloužící pro transport materiálu, kde nosným médiem je vzduch. Využívá se dynamických účinků proudícího vzduchu např. k přepravě sypkých směsí v potravinářském průmyslu. Jedná se o model, jehož součástí je ventilátor ovládaný motorem, dále obsahuje vzduchovou trubici, ve které je uzavřen plastový kelímek. Tento kelímek je vlivem proudícího vzduchu posouván ve vertikálním směru. Pro indikaci polohy kelímku slouží diskretní čidla nebo spojitý laserový snímač. Současně jsou měřeny i otáčky ventilátoru a napětí na jeho motoru. Ve spodní části je umístěn tlakový senzor určující tlakovou diferenci mezi atmosférickým tlakem a tlakem v trubici. Schematicky je systém zobrazen na Obr. 6.9. V laboratoři je využita ke komunikaci karta Advantech PCI-1711.



Obrázek 6.9: Model Vznášení

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Měření vzdálenosti	11	-10 až 10 V	Normalized bipolar
	Rozdíl tlaků	12		
	Rychlost ventilátoru	13		
	Napětí na ventilátoru	14		
Analogový výstup	Ovládání ventilátoru	1	0 až 5 V	Normalized unipolar
Digitální výstup	Přepnutí režimu na PC	1	0 nebo 1	
	Ovládání světla	2		
Digitální vstup	Reflexní snímače	2-7	0 nebo 1	

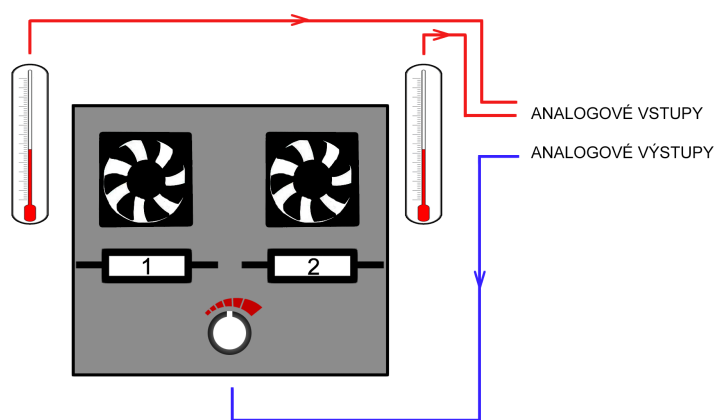
Tabulka 6.5: Zapojení vstupů/výstupů - Vznášení



Obrázek 6.10: Simulinkové schéma modelu Vznášení

6.6 Tepelný systém

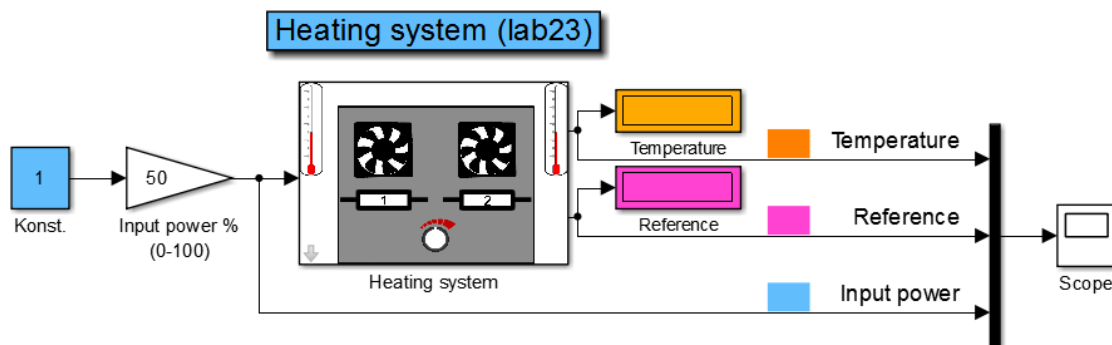
Tepelné soustavy vykazují v reálných aplikacích značnou setrvačnost při změnách teploty, proto mají charakteristiky velmi dlouho časové konstanty. Model tedy může sloužit k návrhu nebo identifikaci takových soustav jako jsou nejrůznější pece nebo topné systémy. Aby bylo možné porovnat systémy s různými časovými konstantami, je model vybaven dvojicí rezistorů. Každý je chlazen vlastním ventilátorem, který nelze ovládat z prostředí Matlab/Simulink. Rychlost je nastavována pomocí dvojice přepínačů na panelu modelu. Rovněž je na modelu umístěn přepínač pro zvolení konkrétního rezistoru. Tyto rezistory jsou napájeny ze stabilizovaného zdroje a dodávaný výkon lze nastavit pomocí PC. Výstupem ze systému jsou teploty rezistorů, které jsou měřeny polovodičovými prvky. Schematicky je model znázorněn na Obr. 6.11.



Obrázek 6.11: Model Tepelný systém

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Teplota soustavy 1/2	1	-10 až 10 V	Normalized bipolar
	Referenční teplota	2		
Analogový výstup	Ovládání výkonu	1	-10 až 10 V	Normalized bipolar

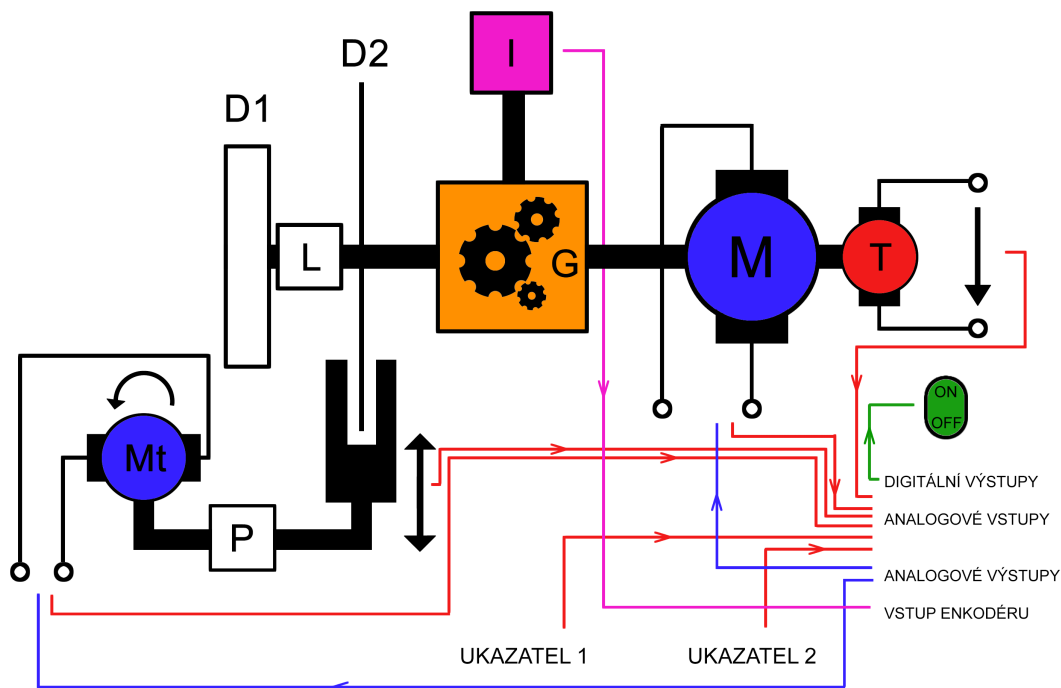
Tabulka 6.6: Zapojení vstupů/výstupů - Tepelný systém



Obrázek 6.12: Simulinkové schéma modelu Tepelný systém

6.7 Servo S2

Servomechanismus S2 podobně jako Servo Amira umožňuje regulovat rychlost a úhel natočení. Model je složen ze stejnosměrného motoru (M), jehož otáčky jsou měřeny tachogenerátorem (T).



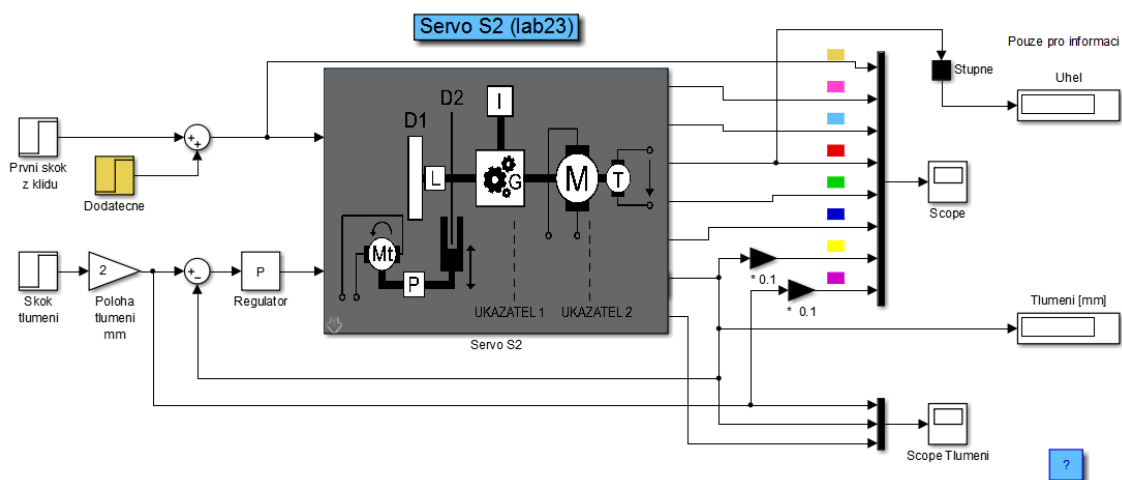
Obrázek 6.13: Model Servo S2

K motoru je připojen přes převodovku (G) výměnný kotouč (D1) a tlumící kotouč (D2). K dispozici jsou 3 výměnné kotouče s rozdílným momentem setrvačnosti. Vliv kotouče D2 se uplatňuje v závislosti na poloze pohyblivého magnetu. Tento magnet je ovládán druhým motorem (Mt) přes převod (P) z rotačního pohybu na posuvný. Ukazatel 1 je spojen s převodovkou v poměru 1:60, tj.

Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Otáčky motoru	3	-10 až 10 V	Normalized bipolar
	Úhel natočení (ukazatel 1)	1		
	Napětí servomotoru	4		
	Zadaná hodnota (ukazatel 2)	2		
	Poloha tlumení	6		
	Napětí motoru tlumení	5		
Analogový výstup	Ovládání motoru	1	-10 až 10 V	Normalized bipolar
	Ovládání motoru tlumení	2		
Digitální výstup	Ovládání modelu	1	0 nebo 1	
Vstup enkodéru	Inkrementální snímač úhlu natočení	4		

Tabulka 6.7: Zapojení vstupů/výstupů - Servo S2

jedna otáčka ukazatele odpovídá 60 otáčkám motoru. Ukazatel 2 slouží k ručnímu nastavení požadovaného úhlu natočení. Všechny veličiny, vystupující v modelu, jsou shrnuty v tabulce na této straně.



Obrázek 6.14: Simulinkové schéma modelu Servo S2

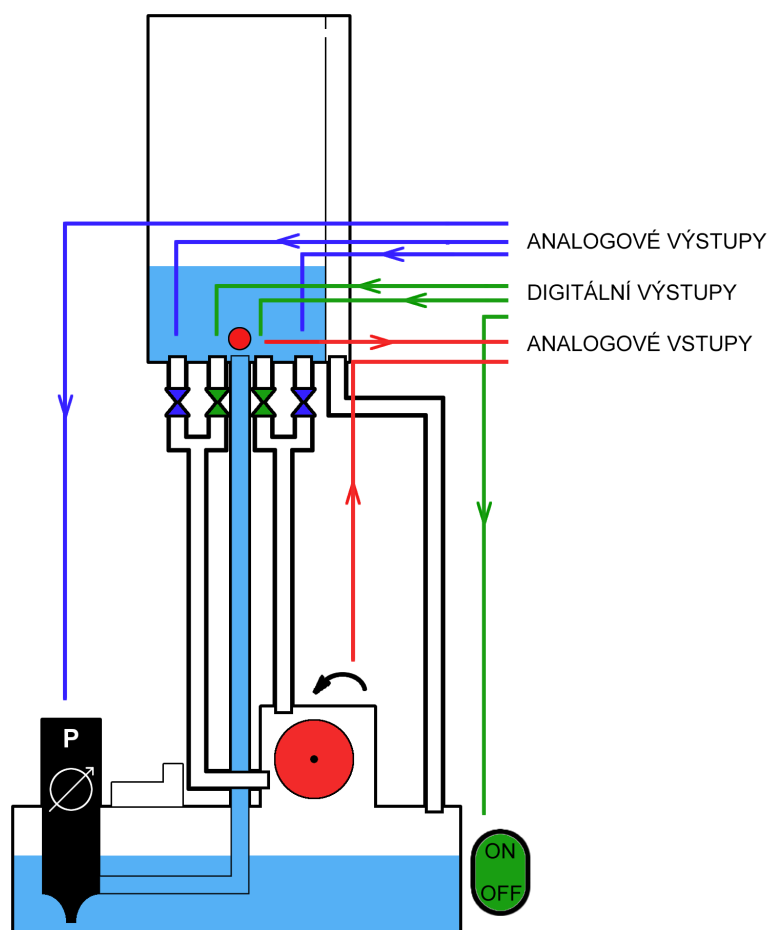
6.8 Vodní elektrárna

Model Vodní elektrárna lze považovat za systém s dopravním zpožděním, kde z rezervoáru je kapalina dopravována odstředivým čerpadlem do horní nádrže. Odtok kapaliny z nádrže je realizován pomocí dvou hadic o různé délce. Zároveň je možné regulovat průtok v každé hadici dvojicí ventilů, proporcionálního a digitálního. Výška hladiny je snímána tlakovým snímačem obdobně jako

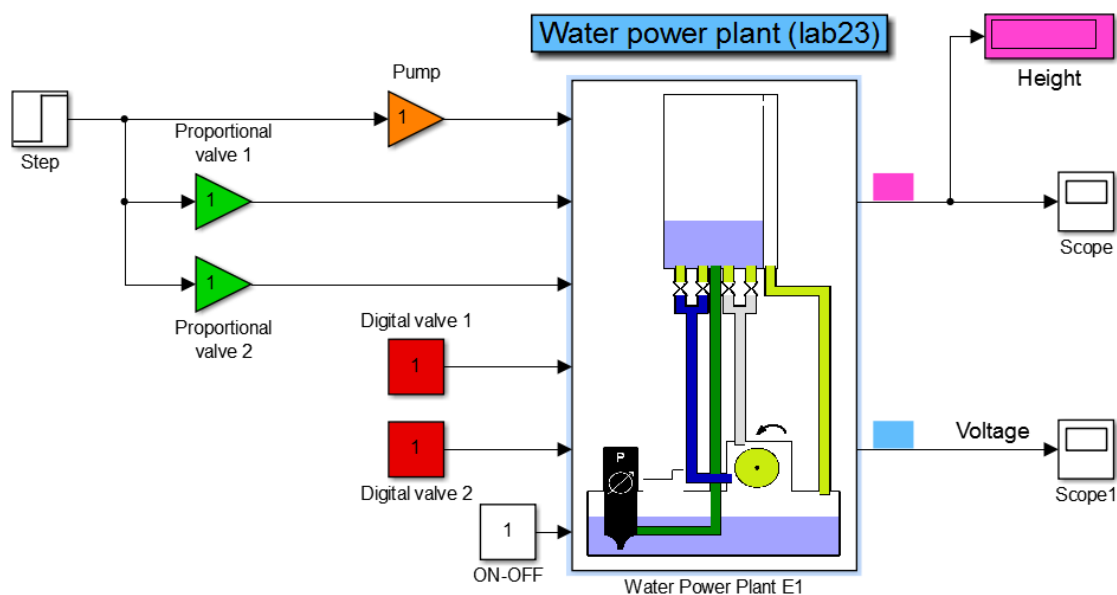
Typ	Funkce	Kanál	Vstupní/výstupní rozsah	Vstupní/výstupní signál bloku
Analogový vstup	Výška hladiny	2	-10 až 10 V	Normalized bipolar
	Napětí tachogenerátoru	3		
Analogový výstup	Ovládání čerpadla	1	-10 až 10 V	Normalized bipolar
	Proporcionální ventil 1	3		
	Proporcionální ventil 2	2		
Digitální výstup	Ovládání modelu	1	0 nebo 1	
	Digitální ventil 1	3		
	Digitální ventil 2	2		

Tabulka 6.8: Zapojení vstupů/výstupů - Vodní elektrárna

u modelu Vodárny. Vracející-se voda zpět do rezervoáru roztáčí malou turbínu a generuje napětí na tachodynamu, což představuje v reálné elektrárně soustrojí turbína-generátor.



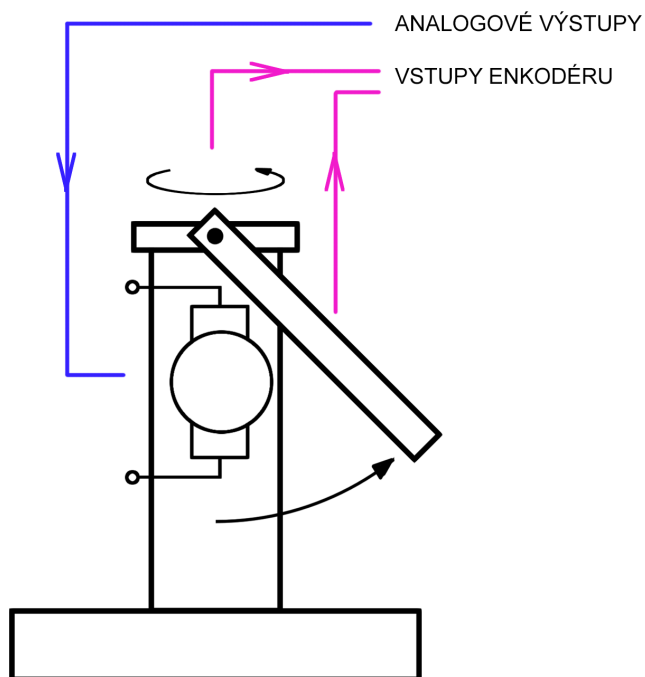
Obrázek 6.15: Model Vodní elektrárna



Obrázek 6.16: Simulinkové schéma modelu Vodní elektrárna

6.9 Rotační kyvadlo

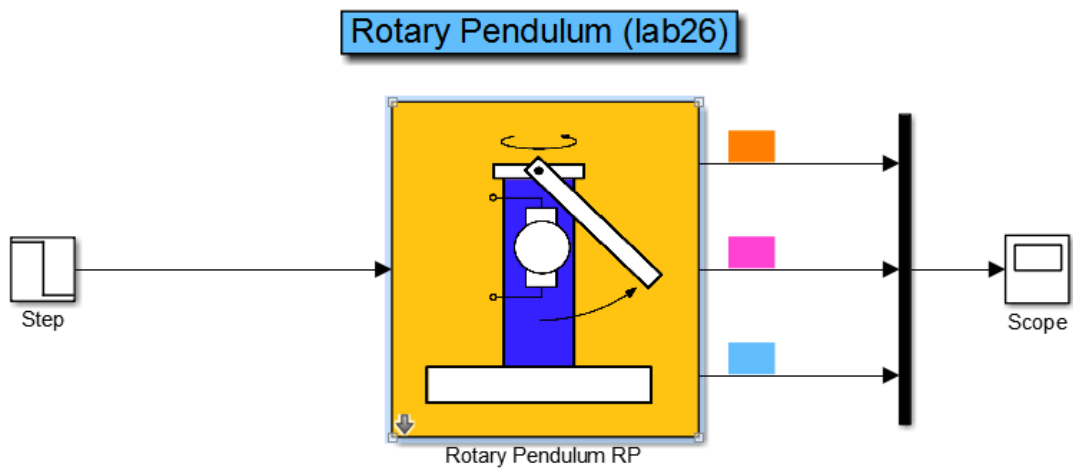
Jedná se o systém demonstrující vlastnosti přístrojů z každodenního života jako jsou např. jeřáby nebo také pojízdné vozítko Segway. Z hlediska vstupů a výstupů se jedná o jednoduchý systém, kde vstupním napětím je ovládáno otočné rameno s volně uchyceným kyvadlem. Výstupními signály jsou úhly natočení kyvadla a otočného ramene, které jsou snímány enkodéry.



Obrázek 6.17: Model Rotační kyvadlo

Typ	Funkce	Kanál
Analogový výstup	Ovládání motoru	1
Analogový vstup	Úhel natočení ramene	1
	Úhel natočení kyvadla	4

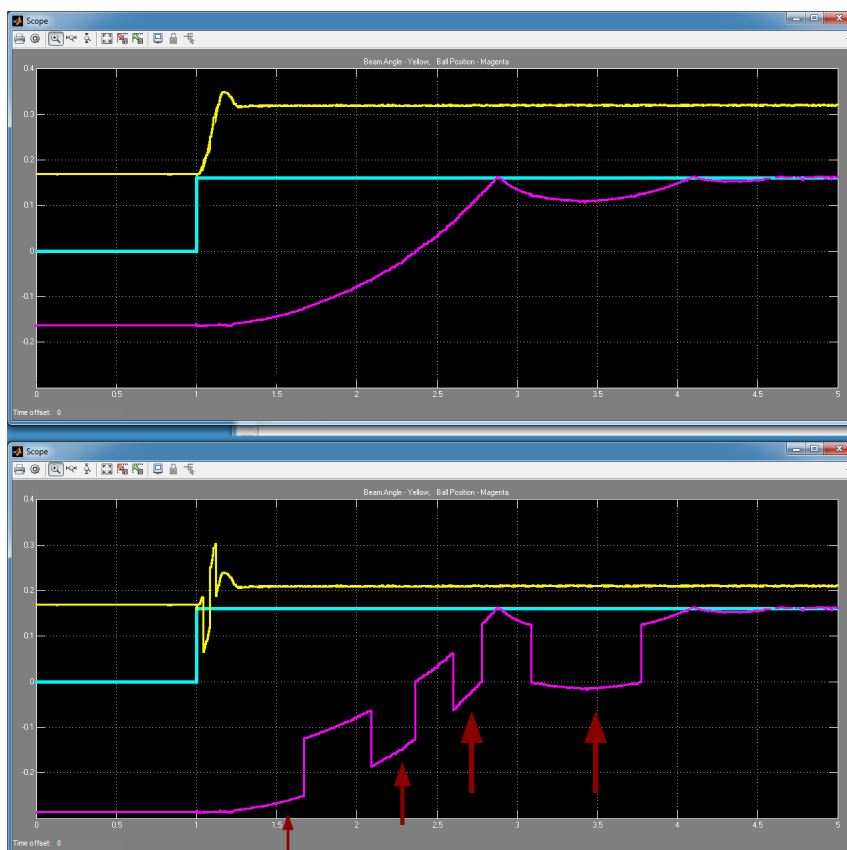
Tabulka 6.9: Zapojení vstupů/výstupů - Rotační kyvadlo



Obrázek 6.18: Simulinkové schéma modelu Rotační kyvadlo

7 Testování modelů

Nové ovládací modely byly otestovány na počítači s operačním systémem Windows 7 Ultimate 64 bitů, což je i platforma, ze které by měly být v budoucnu laboratorní modely ovládány. K samotnému zkoušení byly využívány měřicí karty společností Advantech a Humusoft, kde u měřicí karty PCI-1711 společnosti Advantech nebyly zjištěny žádné závady během testování. U druhé karty MF614 společnosti Humusoft byla zjištěna chyba při měření analogových vstupů, kde byly v signálu identifikovány neočekávané posuny v signálu. Toto nezvyklé chování vykazovaly modely: kulička na tyči, vodní elektrárna a servomechanismy. Zaznamenaný graf chybného průběhu je zobrazen na Obr. 7.1.



Obrázek 7.1: Chyba průběhu z analogových vstupů karty MF614

Místa označené šipkou jsou úseky, kde došlo k posunu signálu o více jak jednu desetinu. Fialový průběh je poloha kuličky a žlutý průběh je úhel natočení ramene. Stejný posun se objevil i u žlutého průběhu. Po konzultaci s firmou Humusoft byla zjištěna závada v ovladači. Jednalo se o softwarovou chybu 64 bitového ovladače, kde docházelo k chybnému přepočítávání přijatých dat z karty. Byl mi propůjčen nový ovladač, se kterým se chyba již neobjevila na žádném z modelů.

8 Závěr

K laboratorním modelům v laboratoři K23 byly vytvořeny nové ovládací modely, které mohou být nově využity v nové verzi prostředí Matlab/Simulink a zároveň začleněny do výuky. U modelů, které jsou připojeny k měřicím kartám MF614 se objevily při sledování analogových vstupů neočekávané skoky v signálu. Konkrétně u modelů kulička na tyči, vodní elektrárna a servomechanismu. Po konzultaci s firmou Humusoft byla závada odstraněna použitím nového ovladače. Následné testování prokázalo funkčnost modelů s použitím nového ovladače pro Matlab. K laboratorním modelům Vodní elektrárna a Vznášení musely být vytvořeny zcela nové ovládací modely, které byly rovněž vyzkoušeny. Podařilo se vytvořit jednoduché GUI, které využívá základní funkce nutné pro ovládání simulinkového modelu z grafického rozhraní. Práce může rovněž do budoucna sloužit jako zdroj informací při tvorbě nových ovládacích modelů.

U laboratorních modelů v laboratoři K26 byla zajištěna kompatibilita s novým toolboxem a vyzkoušena komunikace s reálnými modely Kulička na tyči (BB1), Kulička v obruči (BH), Spojené pohony (CD1) a (CD2), Helikoptéra SPEL (HS), Rotační kyvadlo (RP), Servo Amira (SA) a Spojená Serva ETH (SE).

Literatura

- [1] Documentation Center: Simulink Documentation. *MathWorks: MATLAB and Simulink for Technical Computing* [online]. © 1994-2013
- [2] Documentation Center: Real-Time Windows Target Documentation. *MathWorks: MATLAB and Simulink for Technical Computing* [online]. © 1994-2013 [cit. 2013-04-04]. Dostupné z: <http://www.mathworks.com/help/rtwin/index.html>
- [3] What Is Data Acquisition?. *National Instruments* [online]. © 2013 [cit. 2013-04-10]. Dostupné z: <http://sine.ni.com/np/app/main/p/ap/daq/lang/cs/pg/1/sn/n17:daq/fmid/652/>
- [4] Měřicí desky a moduly pro PC. *Automa: časopis pro automatizační techniku*. Praha: FCC Public, 2009, roč. 2009, č. 11. ISSN 1210-9592. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/39934.pdf>
- [5] Měřicí karty: MF614. *Humusoft: Technické výpočty, řídicí technika, simulace* [online]. © 1991 - 2013 [cit. 2013-04-04]. Dostupné z: <http://www.humusoft.cz/produkty/datacq/old/mf614/>
- [6] Industrial Automation: Data Acquisition - PCI-1711U. *Advantech* [online]. © 1983-2013 [cit. 2013-04-04]. Dostupné z: http://www.advantech.com/products/PCI-1711U/mod_B8EF5337-44F0-4C36-9343-AD87D01792D1.aspx
- [7] Matlab GUI. *MathWorks: MATLAB and Simulink for Technical Computing* [online]. © 1994-2013 Dostupné z: <http://www.mathworks.com/discovery/matlab-gui.html>
- [8] Laboratorní modely. *DCE Wiki* [online]. © 2013 [cit. 2013-05-19]. Dostupné z: http://support.dce.felk.cvut.cz/mediawiki/index.php/Laboratorn%C3%AD_modely
- [9] Laboratorní modely. *Laboratoř teorie automatického řízení K26* [online]. © 2013 [cit. 2013-05-19]. Dostupné z: <https://support.dce.felk.cvut.cz/lab26/index.php?page=models>

Příloha

Zdrojový kód GUI

```
function varargout = coupled_tanks(varargin)
% COUPLED_TANKS MATLAB code for coupled_tanks.fig
%   COUPLED_TANKS, by itself, creates a new COUPLED_TANKS or raises the existing
%   singleton*.
%
%   H = COUPLED_TANKS returns the handle to a new COUPLED_TANKS or the handle to
%   the existing singleton*.
%
%   COUPLED_TANKS('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in COUPLED_TANKS.M with the given input arguments.
%
%   COUPLED_TANKS('Property','Value',...) creates a new COUPLED_TANKS or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before coupled_tanks_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to coupled_tanks_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help coupled_tanks

% Last Modified by GUIDE v2.5 14-May-2013 11:00:28

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @coupled_tanks_OpeningFcn, ...
                  'gui_OutputFcn',  @coupled_tanks_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before coupled_tanks is made visible.
function coupled_tanks_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn. hObject handle to
% figure eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA) varargin
% command line arguments to coupled_tanks (see VARARGIN)

% Choose default command line output for coupled_tanks
handles.output = hObject;
```

```

% Uložení názvu simulinkového modelu do struktury handles
handles.modelName = 'Vx_K23m_new';

% Stav tlačítka ON-OFF
handles.START_BUTTON_VALUE = 'OFF';

% Stav tlačítka pro ovládání digitálního ventilu
handles.DIG_VALVE_BUTTON_VALUE = 'OPEN';

% Testování, zda je model otevřen
openModels = find_system();
if isempty(strncmp(handles.modelName,openModels,20))
    open_system(handles.modelName);
end

% Nastaví modelu Vx_K23m_new funkci localAddEventListener, která je
% spuštěna před startem simulace. Funkce line vytvoří osy v grafu
%
set_param(handles.modelName,'startFcn','localAddEventListener');
line('XData',[],'YData',[]);

% Aktualizace struktury handles
guidata(hObject, handles);

% UIWAIT makes coupled_tanks wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = coupled_tanks_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function GAIN_Callback(hObject, eventdata, handles)
% hObject handle to GAIN (see GCBO) eventdata reserved - to be defined
% in a future version of MATLAB handles structure with handles and user
% data (see GUIDATA)

% Vrací hodnotu prvku GAIN, jako string hodnotu a kontroluje typ zadanych
% dat
value = get(handles.GAIN,'string');

value_num = str2double(value);

if value_num > 1 || value_num < 0
    errordlg('Hodnota musí být mezi 0 a 1','Error');
return
end
if isnan(value_num)
    errordlg('Hodnota není číslo','Error');
return
end

```



```

set_param('Vx_K23m_new/Pump','Gain',value);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GAIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GAIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function PROP_VALVE_Callback(hObject, eventdata, handles)
% hObject    handle to PROP_VALVE (see GCBO) eventdata reserved - to be
% defined in a future version of MATLAB handles    structure with handles
% and user data (see GUIDATA)

% Vrací hodnotu prvku PROP_VALVE, jako string hodnotu a kontroluje typ
% zadanych dat
value = get(handles.PROP_VALVE,'String');
value_num = str2double(value);
if value_num > 1 || value_num < 0
    errordlg('Hodnota musí být mezi 0 (maximálně uzavřený) a 1 (maximálně otevřený)','Error');
return
end
if isnan(value_num)
    errordlg('Hodnota není číslo','Error');
return
end
set_param('Vx_K23m_new/Proportional valve','Gain',value);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PROP_VALVE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PROP_VALVE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in START_BUTTON.
function START_BUTTON_Callback(hObject, eventdata, handles)
% hObject    handle to START_BUTTON (see GCBO) eventdata reserved - to be
% defined in a future version of MATLAB handles    structure with handles
% and user data (see GUIDATA)

```

```

% podle aktualniho stavu tlacitka (START_BUTTON_VALUE) meni pozadi a
% hodnotu bloku ON-OFF
    if strcmp(handles.START_BUTTON_VALUE, 'OFF')
        set(handles.START_BUTTON, 'String', 'OFF');
        handles.START_BUTTON_VALUE = 'ON';
        set_param('Vx_K23m_new/ON-OFF', 'Value', '1');
        set_param('Vx_K23m_new/ON-OFF', 'BackgroundColor', 'green');
        set_param('Vx_K23m_new', 'SimulationCommand', 'start');

    else
        set(handles.START_BUTTON, 'String', 'ON');
        handles.START_BUTTON_VALUE = 'OFF';
        set_param('Vx_K23m_new/ON-OFF', 'Value', '0');
        set_param('Vx_K23m_new/ON-OFF', 'BackgroundColor', 'red');
        set_param('Vx_K23m_new', 'SimulationCommand', 'stop');
    end
guidata(hObject, handles);

function eventhandle = localAddEventListener

% registrace bloku Right height a prirazeni udalosti, která spusti kod ve
% funkci localEventListener

blk = 'Vx_K23m_new/Right height';
event = 'PostOutputs';
listener = @localEventListener;

eventhandle = add_exec_event_listener(blk, event, listener);

global ad;
% get the application data
ad = guidata(gcbo);

function localEventListener(block, eventdata)

% get the application data
global ad;

% získání vystupnich dat z bloku
time = block.CurrentTime;
data = block.InputPort(1).Data;

% nacteni jiz zobrazenych dat v grafu
old_x = get(get(ad.Graph, 'children'), 'XData');
old_y = get(get(ad.Graph, 'children'), 'YData');

% doplneni vektoru dat o nove vzorky z promennych time a data
new_x = [old_x time];
new_y = [old_y data];

% nastavuje pouze poslednich 500 vzorku pro vykresleni
if length(xdata) < 501

```

```

        new_x = [old_x time];
        new_y = [old_y data];
    else
        new_x = [old_x(2:end) time];
        new_y = [old_y(2:end) data];
    end

    % vykreslení vzorku
    set(get(ad.Graph,'children'),'XData',new_x);
    set(get(ad.Graph,'children'),'YData',new_y);

    % --- Executes on button press in DIG_VALVE_BUTTON.
    function DIG_VALVE_BUTTON_Callback(hObject, eventdata, handles)
    % hObject    handle to DIG_VALVE_BUTTON (see GCBO) eventdata reserved - to
    % be defined in a future version of MATLAB handles    structure with
    % handles and user data (see GUIDATA)

    % podle aktualního stavu tlačítka (DIG_VALVE_BUTTON_VALUE) mení pozadí a
    % hodnotu bloku Digital valve
    if strcmp(handles.DIG_VALVE_BUTTON_VALUE,'CLOSE')
        set(handles.DIG_VALVE_BUTTON,'String','CLOSE');
        handles.DIG_VALVE_BUTTON_VALUE = 'OPEN';
        set_param('Vx_K23m_new/Digital valve','Value','0');
        set_param('Vx_K23m_new/Digital valve','BackgroundColor','green');
    else
        set(handles.DIG_VALVE_BUTTON,'String','OPEN');
        handles.DIG_VALVE_BUTTON_VALUE = 'CLOSE';
        set_param('Vx_K23m_new/Digital valve','Value','1');
        set_param('Vx_K23m_new/Digital valve','BackgroundColor','red');
    end
    guidata(hObject, handles);

    % --- Executes when user attempts to close figure1.
    function figure1_CloseRequestFcn(hObject, eventdata, handles)
    % hObject    handle to figure1 (see GCBO) eventdata reserved - to be
    % defined in a future version of MATLAB handles    structure with handles
    % and user data (see GUIDATA)

    % zobrazení dotazovacího okna: ANO - uzavření GUI a zastavení simulace
    volba = questdlg('Opravdu chcete ukončit simulaci?','Ukončení simulace','Ano','Ne','Ano');
    switch volba
        case 'Ano'
            delete(hObject);
            if isempty(find_system('type','block_diagram','Name',handles.modelName)) == 0
                set_param(handles.modelName,'SimulationCommand','stop');
                bdclose(handles.modelName);
            end
        case 'Ne'
    end
end

```