



**FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI**

Bakalářská práce

Vývoj metod pro pokročilou integraci simulačních nástrojů na bázi standardu "FMI 2.0 for Co-Simulation" a její validace na modelu vybraného energetického zařízení.

Vedoucí : Ing. Martin Čech, Ph.D.

Vypracoval : Karel Kubíček - A14B0518P

Titulní list

Zadání

Vývoj metod pro pokročilou integraci simulačních nástrojů na bázi standardu "FMI 2.0 for Co-Simulation" a její validace na modelu vybraného energetického zařízení.

- **1.** Analyzujte aktuální stav vývoje a možné přínosy integrace simulačních nástrojů na bázi standardu "FMI 2.0 for Co-Simulation" z pohledu návrhu a vývoje řídicích systémů.
- **2.** Vytvořte zjednodušený model vybraného energetického zařízení (např. turbína, reaktor) pomocí komponentového přístupu na bázi jazyka Modelica; použijte vhodnou dostupnou simulační platformu (OpenModelica, Dymola, SimulationX).
- **3.** Vyvíňte postup pro integraci modelu jako FMU (Function-Mockup-Unit) do HIL (Hardware-in-the-Loop) simulátoru obsahujícím řídicí subsystém vytvořený v systému REX (alternativně Matlab/Simulink).
- **4.** Ověřte postup a výsledky simulace na dostupných datech.

Assignment

Development of methods for advanced integration of simulation tools based on "FMI 2.0 for Co-Simulation" standard and its validation on selected model of energetic device.

- **1.** Analyze the current state of development and potential benefits of the integration simulation tools based on standard "FMI 2.0 for Co-Simulation" in point of view of design and development of control systems.
- **2.** Create a simplified model of the selected power equipment (for example turbine, reactor) using a component-based modeling language Modelica; Use appropriate available simulation platform (OpenModelica, Dymola, simulationx).
- **3.** Develop a procedure for the integration model as FMU (Function-Mockup-Unit) in HIL (Hardware-in-the-Loop) simulator including a control subsystem created in REX (alternatively Matlab / Simulink).
- **4.** Verify the procedure and simulation results on available data.

Čestné prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

.....

Anotace

Cílem této práce je objasnění fungování FMI 2.0. Budou ukázány základní principy tohoto rozhraní. Všechny kroky budou nejprve demonstrovány na modelu kyvadla. Přejdeme od matematicko - fyzikálního popisu přes model v Simulinku, přes komponentový přístup v Open Modelice až po vyexportovaný model ve tvaru FMU. Následně získané modely porovnáme. V druhé části bude celý proces proveden znovu na značně složitějším modelu energetického zařízení. Tím bude jednoduchý model vodní elektrárny, který se bude skládat z modelu vodní turbíny, generátoru a regulátoru.

Klíčová slova: FMI 2.0, FMU, Co-Simulation, Matlab, Simulink, OpenModelica, kyvadlo, vodní turbína, vodní elektrárna, RexLib, REX

Objectives

The aim of this thesis is to clarify the functioning of FMI 2.0. The basic principles of this interface will be shown. All steps will be first demonstrated on the pendulum model. We move from the mathematical-physical modeling through the model in Simulink, to the component access in the Open Model to the exported model in the form of FMU. We will compare all the created models. In the second part, the whole process will be carried out again in more complex model of the energy equipment. That will be a simple model of a hydropower plant that will consist of a model of a water turbine, generator and regulator.

Key words: FMI 2.0, FMU, Co-Simulation, Matlab, Simulink, OpenModelica, pendulum, water turbine, water powerplant, RexLib, REX

Poděkování

V první řadě bych chtěl poděkovat svému vedoucímu panu Ing. Martinu Čechovi, Ph.D. za skvělý přístup, odborné rady a obzvlášť za věnovaný čas, kterého bylo opravdu hodně. Vždy když jsem potřeboval, tak si na mě udělal čas a vždy mi vyšel vstříc a poradil. Dále bych chtěl určitě poděkovat panu Ing. Janu Reitingerovi za pomoc a věnovaný čas při sestavování modelů v OpenModelice a také za poskytnutí cenných rad a informací o celé problematice. Také paní Mgr. Janě Königsmarkové bych rád poděkoval za pomoc při odvozování a sestavování analytického řešení kyvadla a sestavení modelu pomocí RexLibu a v REXu. V neposlední řadě bych chtěl poděkovat svojí rodině, přítelkyni a přátelům za podporu během studia.

Obsah

0.1	Obecný úvod	1
0.2	Motivace	1
1	Vysvětlení základních principů modelování	3
1.1	Vysvětlení pojmů	3
1.1.1	Systém a jeho okolí	3
1.1.2	Vstupní a výstupní proměnné	4
1.1.3	Stavové proměnné	4
1.1.4	Stav systému	4
1.1.5	Model	5
1.1.6	Modelování	5
1.1.7	Simulace	5
1.2	Matematicko - fyzikální modelování vs identifikace systému	5
1.3	The Functional Mock-up Interface	6
2	Model kyvadla	7
2.1	Odvození diferenciální rovnice	8
2.2	Analytické řešení diferenciální rovnice	9
2.3	Simulink	13
2.4	SimScape v Simulinku	14
2.5	OpenModelica	15
2.6	FMU	16
2.6.1	Jak vyexportovat FMU	17
2.6.2	Jak pracovat s FMU v Matlabu	18
2.7	Porovnání	20
2.7.1	V klidu	21
2.7.2	S náhodnou funkcí na vstupu	22
2.8	Porovnání s generátory náhodných signálů	25
2.8.1	Mackey-Glassův generátor	25
2.8.2	Rösslerův generátor	28
2.9	Problémy při řešení	32
2.10	Zhodnocení	34
2.10.1	Ohlédnutí za projektem 5	34
2.10.2	Model kyvadla	34
3	Model vodní turbíny	35
3.1	Sestavení modelu v Simulinku	36
3.1.1	Parametry a rovnice	36
3.1.2	Nelineární model turbíny	36
3.2	Sestavení modelu v Open Modelice	38
3.3	Porovnání modelů	39
4	Model synchronního generátoru připojeného k silné síti	42
4.1	Sestavení modelu v Simulinku	43
4.2	Sestavení modelu v Open Modelice	45
4.3	Porovnání modelů	46

5 Model servomotorů	49
6 Model elektricko-hydraulického regulátoru	50
6.1 Paralelní realizace	50
6.2 Sériová realizace	51
7 Hodnoty parametrů	52
7.1 Parametry pro turbínu	52
7.2 Parametry pro generátor	52
7.3 Parametry pro servomotory	52
7.4 Parametry pro regulátor	52
8 Model jednoduché vodní elektrárny	53
8.1 V Simulinku	53
8.2 Pomocí FMU - Model Exchange	53
8.3 Pomocí FMU - Co-Simulation	54
8.4 Porovnání modelů	54
8.4.1 Porovnání mechanického výkonu	55
8.4.2 Porovnání elektrického výkonu	57
9 Sestavení modelu v REXu	59
9.1 Pomocí knihovny RexLib v Simulinku	59
9.1.1 Schéma modelu	59
9.1.2 Porovnání modelů	60
9.2 V Rexu	63
9.2.1 Regulátor	64
9.2.2 Turbína s generátorem pomocí FMUs	65
10 Závěr	66
Seznam obrázků	69
Seznam tabulek	71
Seznam literatury	72

Úvod

0.1 Obecný úvod

V této práci je rozebrána práce se standardem FMI 2.0 (The Functional Mock-up Interface). Je zde ukázáno, jak získat FMU (Functional Mock-up Unit, všechny použité zkratky jsou uvedeny na konci této práce) a jak s ním pracovat. Projdeme si proces vytváření modelu v Simulinku, přes Open Modelicu až po vyexportování FMU a jejich následné porovnání.

Celý tento proces bude nejprve demonstrován na jednoduchém modelu kyvadla, ve kterém se seznámíme s celým FMI 2.0. Poté přejdeme ke složitějšímu a komplexnějšímu modelu jednoduché vodní elektrárny. Ta se skládá z modelu vodní turbíny, generátoru a regulátoru.

0.2 Motivace

FMI vzniklo jako odpověď na inženýrský problém modelovat a simulovat složité systémy. Každý si jistě dovede představit, že modelování a simulace je velice komplexní problém. Existuje mnoho přístupů, jak modelovat a simulovat, stejně tak existuje nepřeberné množství simulačních a modelačních nástrojů. Každý se hodí na něco jiného. Máme speciální programy, které slouží například pro modelování fyzikálních jevů, pružnost a pevnost materiálů, návrhy řídicích systémů atd [3]... Jedná se vlastně o jakýsi komponentový přístup. Snahu nahradit složité a provázané struktury systémů něčím jednodušším a přehlednějším. V případě potřeby změny nějakého bloku potom stačí dané FMU nahradit novější verzí. V FMU se také dají nadefinovat parametry, které po rozkliknutí bloku FMU můžeme jednoduše měnit. Oproti simulaci daného modelu v původním softwaru odpadá nutnost vyřešit import/export jednotlivých výsledků. Někdy je totiž poměrně obtížné tato data interpretovat a zajistit správnou interakci mezi nimi. Z tohoto důvodu vzniklo FMI.

Realita je samozřejmě mnohem složitější, protože komplexnější systémy jsou velmi provázané a ne vždy je prostá výměna původního modelu za FMU možná. Musí se vždy vyřešit začlenění a komunikace FMU s ostatními částmi systému, ale je z toho jasně patrné, jakým směrem tato technologie směřuje [8].

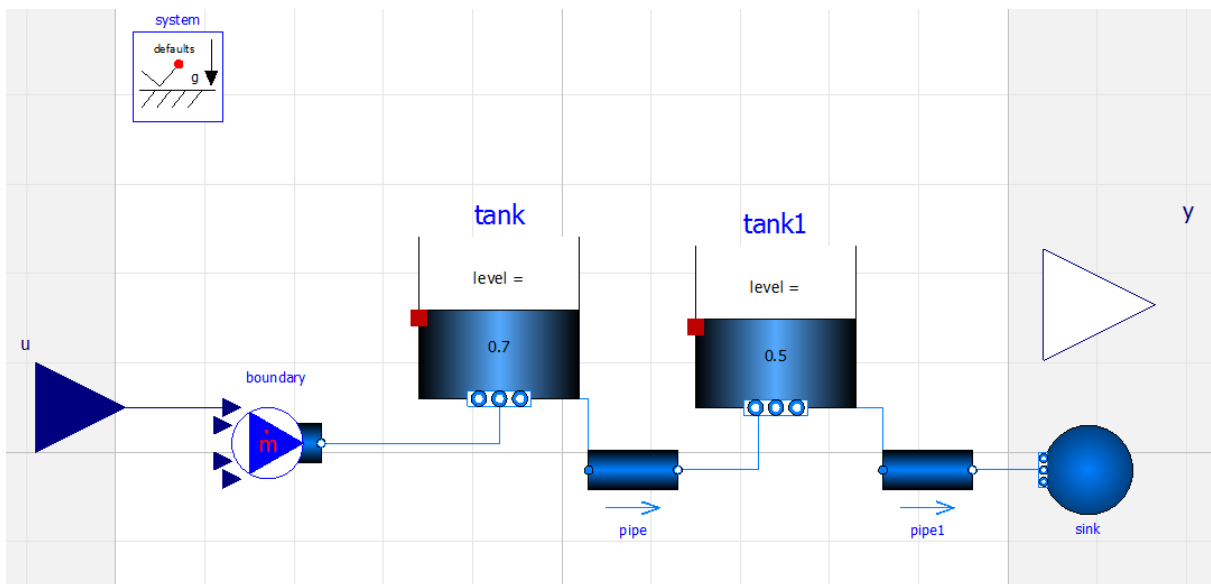
Díky tomuto standardu se může každý specialista soustředit na svoji práci. Jeden expert se může věnovat například mechanickému návrhu, druhý může řešit elektroinstalaci a další třeba řídicí systém daného objektu [5]. Každý potom svůj výsledek exportuje jako FMU a s ním může pracovat další odborník. Jiný expert zase vezme všechny tyto FMUs a poskládá je do výsledného modelu. Platí, že není nutné, aby přesně znal všechny detaily, co se skrývají za daným FMU. Je však nutné, aby chápal vztahy a interakce mezi jednotlivými FMU. Co je přivedeno na vstup a co naopak z daného FMU vystupuje.

Představme si to na modelu auta. Někdo musí sestavit model karoserie, elektroinstalace, motoru, převodovky, řídicích systémů atd... Toto všechno poté můžeme dát dohromady a simulovat jako "celé auto" naráz. To přispívá ke zvýšené efektivitě celého procesu návrhu, ale také kvalitě. Každý odborník se soustředí hlavně na svoji úlohu. Nejdříve se provede

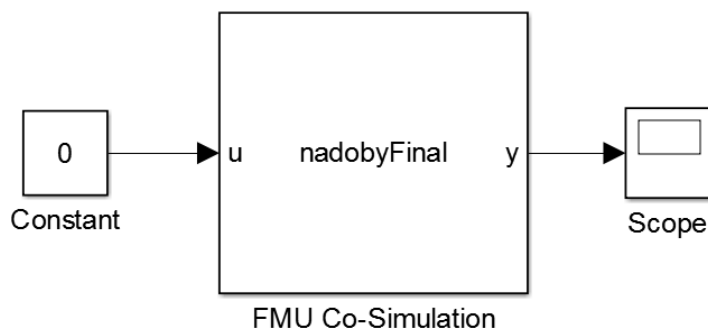
množství simulací a na základě jejich průběhů jsou upraveny jednotlivé části tak, aby bylo dosaženo požadovaných výsledků. Teprve až budeme spokojeni se simulacemi, tak přistoupíme k vytvoření fyzického prototypu. Dochází tak také ke značným časovým a finančním úsporám [8], [31-33].

Tento standard vznikl za účelem snadné výměny jednotlivých částí komplexnějšího modelu mezi různými softwary. Díky FMI můžeme libovolný model exportovat jako FMU [5]. To si lze představit jako určité zapouzdření daného modelu do jednoho jediného bloku (komponenty), který má jasně definované vstupy a výstupy. Tím dochází k zřehlednění celého systému. A snadnějšímu propojení všech součástí. Hlavní věc, kterou přináší FMI je snadná přenositelnost daného modelu mezi různými zařízeními a hlavně mezi různými softwary [3], [8], [31], [33].

Pro názornost si to uvedeme na příkladu modelu spojených nádob. Tento model je znázorněn na obrázku č.1 a v Modelice se skládá z několika bloků: dvě nádoby, dvě potrubí, zdroj přitékající vody, odtok vody a vstupní a výstupní port. Tj. celkem osm bloků.



Obrázek 1: Model spojených nádob v Open Modelice



Obrázek 2: Model spojených nádob importovaný do Matlabu jako FMU

Jak je patrné z obrázku č.2 vstupní a výstupní porty zůstaly zachovány. Všechny komponenty, které tvoří daný model, však byly "zabaleny" do FMU. FMI poskytuje dvě možnosti exportů - Model Exchange a Co-Simulation. Ty tu nyní však rozepisovat nebudeme a dostaneme se k nim v teoretické a praktické části této práce.

1 Vysvětlení základních principů modelování

Ještě než se začneme zabývat interpretací a možnostmi FMI ukázaných na jednoduchém modelu, je důležité si připomenout některé základní pojmy, které se používají v problematice modelování a simulací [10].

1.1 Vysvětlení pojmů

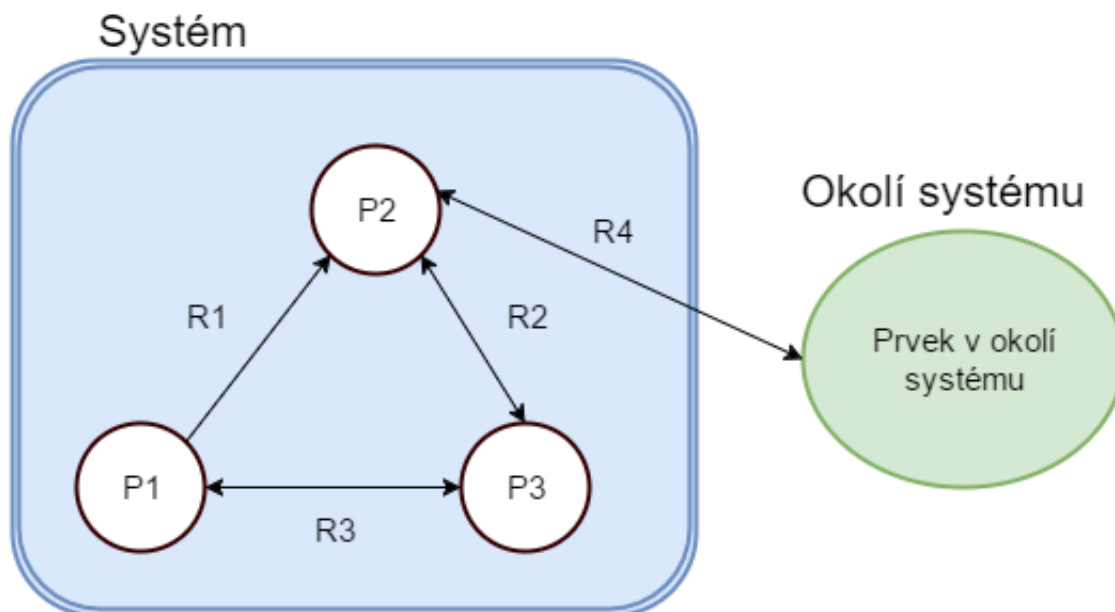
Existuje mnoho různých definic pro následující pojmy. Vybral jsem proto ty, které jsou mně osobně nejbližší.

1.1.1 Systém a jeho okolí

Systém se dá pochopit jako množina určitých prvků (příznaků) a vazeb (relací) mezi nimi. Relace mohou být jednosměrné nebo obousměrné. Tato množina má určité vlastnosti. Vazby se dají ještě dělit na vnitřní (vazby mezi jednotlivými prvky systému) a vnější (vazby mezi prvky systému a jeho okolí).

Vše co je vně daného systému se označuje jako okolí systému. Systém s okolím komunikuje pomocí vstupních a výstupních proměnných.

Na Obrázku č.3 je znázorněn systém se svým okolím. P1 - P3 jsou prvky systému. Mezi nimi jsou relace R1 - R3, ty jsou až na R1 obousměrné. Dále vidíme, že jediný prvek P2 komunikuje s okolím systému pomocí vnější relace R4. Můžeme ho tedy označit za vstupně/výstupní prvek (proměnnou).



Obrázek 3: Systém a jeho okolí

1.1.2 Vstupní a výstupní proměnné

Vstupní proměnné se zpravidla označují u_1, u_2, \dots, u_n . Vstupní proměnné mohou mít různou interpretaci a představují informace, na jejichž základě daný systém provede sérii činností, které povedou na nějaký výstup. Výstup systému je popsán výstupními proměnnými. Ty se označují y_1, y_2, \dots, y_n .

1.1.3 Stavové proměnné

Jsou to takové proměnné, které se používají k matematickému popisu stavu dynamického systému, popisují vnitřní parametry modelu. Označují se jako x_1, x_2, \dots, x_n . Všechny proměnné (vstupní, výstupní a stavové) se často zapisují jako vektory.

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_n \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{bmatrix}$$

1.1.4 Stav systému

Stav systému je reprezentován hodnotami stavových proměnných $x(t)$. Typ proměnných můžeme rozdělit na diskrétní v čase a spojitě. Stavové proměnné společně se vstupem systému definují jeho výstup. Stav systému k určitému časovému okamžiku se určuje pomocí vstupu a jeho výstupu v předcházejícím časovém okamžiku. Stav systému nám dává dostatečné informace o systému k tomu, abychom byli schopní určit jeho budoucí chování. Rozlišujeme rovnovážný stav a nerovnovážný stav systému.

1.1.5 Model

Model se dá chápat jako abstrakce reálného objektu.

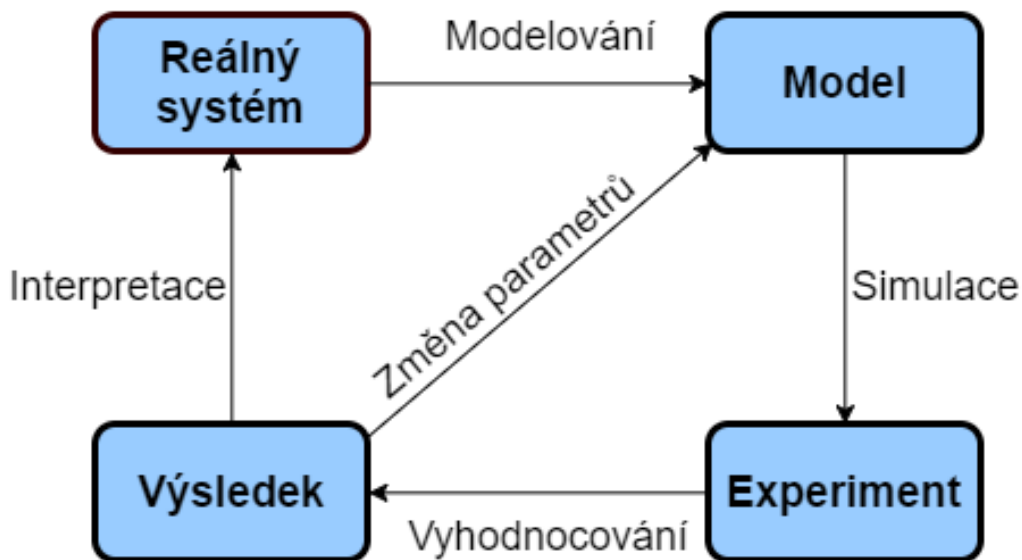
1.1.6 Modelování

Smyslem modelování je nahrazení skutečného systému jeho abstraktním modelem. Na modelu provedeme řadu experimentů, s cílem získat co nejvíce informací o původním zkoumaném systému. Výsledkem modelování je tedy sestavený model.

1.1.7 Simulace

Provádí se s daným modelem, na který je přiveden určitý vstup a my na základě výsledků na výstupu buď změním některé parametry modelu a nebo, pokud jsme s výsledky spokojeni, aplikujeme je na reálný systém.

Proces přechodu od reálného systému k modelu, jeho simulace a interpretace výsledků je znázorněna na následujícím diagramu:



Obrázek 4: Princip postupu při modelování, simulacích a vyhodnocení výsledků

1.2 Matematicko - fyzikální modelování vs identifikace systému

Získat model systému můžeme jedním z těchto způsobů. Matematicko - fyzikální modelování využívá přesné matematické znalosti a fyzikální zákonitosti. Identifikace systémů se používá tehdy, kdy nejsme schopní pomocí matematicko - fyzikálního modelování získat funkční model. Systém se nám jeví jako "black box".

Matematicko - fyzikální modelování

Vytváření modelu probíhá v matematické rovině na základě fyzikálních zákonů. To poskytuje numerickou analýzu problému s cílem proniknout do jeho podstaty - získat informace

o zkoumaném problému. K odvození diferenciálních rovnic modelů se využívají fyzikální zákony. Například v oblasti elektřiny se jedná o Kirchhoffovy zákony, v mechanice o Newtonovy zákony, atd... Dalším z možných přístupů je použití Lagrangeovy funkce nazývané též Lagrangian. Tuto funkci zavedl francouzský matematik Joseph-Louis Lagrange ve svém popisu klasické mechaniky. Je to funkce, která v sobě zahrnuje popis dynamického chování systému.

Komponentové modelování je speciální typ matematicko - fyzikálního modelování. Jedná se o přístup, kdy je inženýr odstíněn od nutné znalosti diferenciálních rovnic. V daném prostředí poskládá model z funkčních bloků a překladač daného softwaru se o přesný tvar diferenciálních rovnic postará sám, na základě zapojení jednotlivých částí (komponent) modelu. S tímto přístupem pracuje například Open Modelica.

Identifikace systému

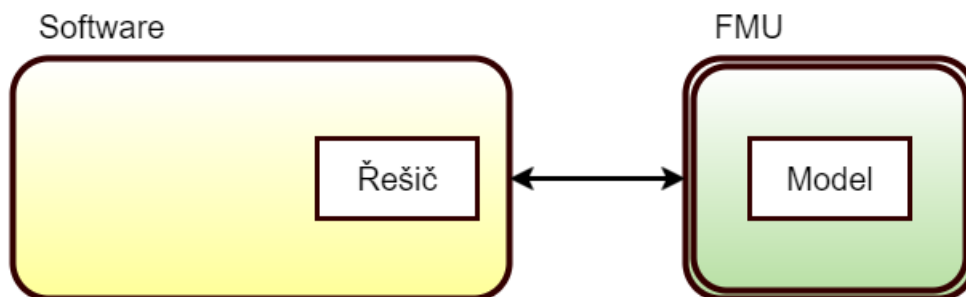
Identifikace se zabývá nalezením modelu systému z pozorovaných dat. Využívá se zejména tam, kde náš sledovaný systém je příliš složitý pro použití matematicko - fyzikálního modelování a není ani možná jeho dekompozice na subsystémy. Mezi metody identifikace systémů patří například metody filtrace, aproximace přechodových charakteristik a mnoho dalších. Ty jsou používány v technických disciplínách zabývajících se automatickým řízením a zpracováním signálů, v ekonomii, ekologii, biologii, lékařství atd... Umožňují nám získat vhodné modely pro návrh regulátorů, předvídání vývoje sledovaných veličin, pro simulaci a ke zvyšování znalostí o identifikovaném systému.

1.3 The Functional Mock-up Interface

The Functional Mock-up Interface je standard pro výměnu kompilovaných dynamických modelů. Je určen pro podporu znovu používání daného modelu. Ten exportujeme jako FMU (popřípadně FMUs) - Functional Mock-up Units s definovanými vstupy a výstupy [3]. Daný model bude potom reprezentován pouze jedním blokem. Odpadají tak starosti se složitým začleněním celého modelu do naší práce. Mnoho nástrojů umožňuje exportování/importování do FMU [6]. Všechny modely ve formě FMU mohou být použity za pomoci FMI toolboxu pro Matlab (Přesný název balíčku je v sekci - Problémy při řešení). FMI poskytuje dva odlišné formáty pro výměnu modelů - Model Exchange a Co-Simulation [7], [9].

FMI pro Model Exchange (FMI-ME)

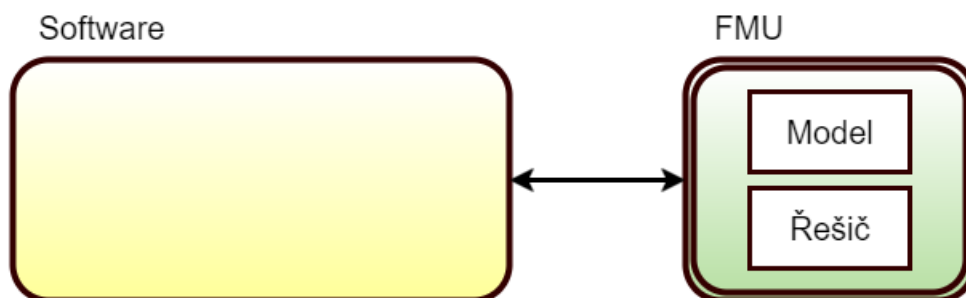
FMI-ME je specifikace založená na reprezentaci obyčejných diferenciálních rovnicí (ODE). Lze také reprezentovat diferenciální algebraické rovnice (DAE) [32]. FMI-ME poskytuje vstupy, výstupy a funkce pro nastavení jednotlivých parametrů a samotné počítání derivací ODE (popřípadě DAE). Pracovní prostředí, do kterého importujeme FMU-MEs, musí poskytovat ODE (popřípadě DAE) řešič (solver), který numericky interpretuje dynamiku daného modelu. Na obrázku č.5 je znázorněna komunikace mezi softwarem (Simulink) a FMU pro Model Exchange [4], [6], [9].



Obrázek 5: FMI pro Model Exchange

FMI pro Co-Simulation (FMI-CS)

FMI-CS specifikace poskytuje reprezentaci modelu, ve které je obsažen jak daný model, tak ODE(DAE) řešič [32]. Obě tyto části jsou potom obsaženy uvnitř FMU-CS. Stejně jako FMI-ME, FMU-CS poskytuje vstupy a výstupy, které nastavují parametry modelu. Poskytuje také funkci "posunutí" modelu v čase o určitý časový interval. Pracovní prostředí, do kterého importujeme FMU-CS proto nemusí obsahovat vlastní řešič. Na obrázku č.6 je znázorněna komunikace mezi softwarem (Simulink) a FMU pro Co-Simulation. U FMU-CS je důležité nastavení kroku vzorkování. My jsme při srovnávání modelu nejčastěji používali hodnotu 0.001 a 0.00001 [4], [6], [9][34].



Obrázek 6: FMI pro Co-Simulation

2 Model kyvadla

Na modelu kyvadla si ukážeme, jak pracovat s FMI 2.0. Nejdříve si odvodíme matematicko - fyzikální model, který následně sestavíme jako diferenciální rovnici v Simulinku, dále pomocí knihovny Simscape v Simulinku, komponentově v OpenModelice a poté si ukážeme, jak z OpenModelicy vyexportovat FMU jako model Exchange a Co-Simulation. Na závěr budou všechny získané modely porovnány. Pro zajímavost si také ukážeme analytické řešení diferenciální rovnice kyvadla.

Nejdříve je však důležité uvést všechny verze programů, které zde jsou použity. A to proto, že zatím pouze některé verze podporují práci s FMI 2.0, do jiných jsme zase museli stáhnout speciální softwarové balíčky, které tuto podporu dodatečně přidávají.

Pro sestavení modelu jsou použity tyto programy: Matlab R2016a a Modelica nightly build. Oba programy jsou 64bitové. Do Matlabu je nutné nainstalovat - "Pilot Support Package", který přidává podporu FMI 2.0. Tento softwarový balíček nám byl poskytnut přímo společností Mathworks. Veškeré problémy, na které jsem při řešení tohoto úkolu narazil, jsou sepsány v sekci - Problémy při řešení.

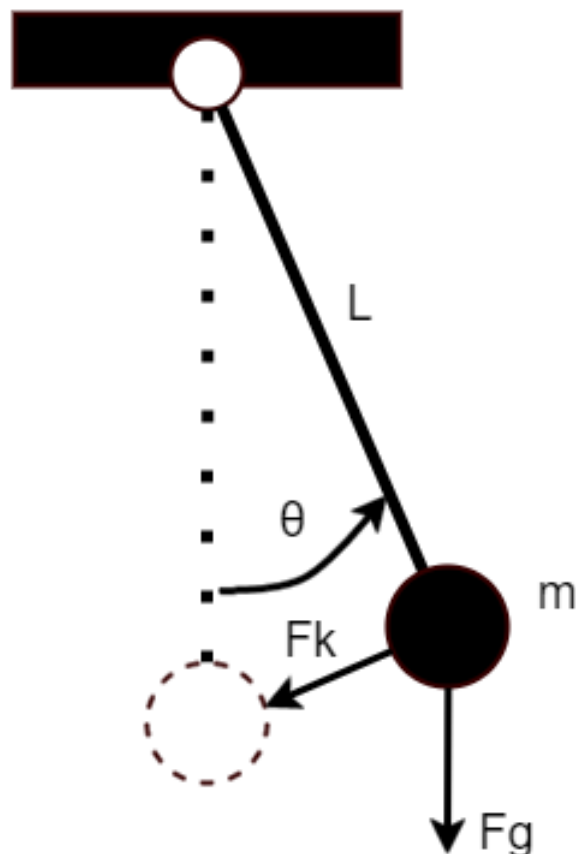
2.1 Odvození diferenciální rovnice

Při odvozování diferenciálních rovnic jsme vycházeli ze znalosti vzorce pro moment síly a moment hybnosti. Tato metoda pro odvození se nazývá Newton - Eulerova metoda.

$$M_c = L \times F = l \times mg$$

$$H_c = L \times P = l \times mv_c$$

Nejdříve si popíšeme jednotlivé proměnné. L představuje délku kyvadla, m je hmotnost, v_c je úhlová rychlost, g je gravitační konstanta, b je parametr tlumení, F_g je gravitační síla, θ je úhel vychýlení kyvadla a F_k je síla, která se snaží kyvadlo dotáhnout do rovnovážné polohy. Kyvadlo na obrázku č.7, které je vykresleno čárkovaně/tečkovaně znázorňuje rovnovážnou polohu kyvadla.



Obrázek 7: Schéma kyvadla

Úhlovou rychlost můžeme vyjádřit následovně:

$$v_c = \omega \cdot l = \dot{\theta}l$$

Získaný tvar dosadíme do rovnice pro moment hybnosti:

$$H_c = l \times m\dot{\theta}l = ml^2\dot{\theta}$$

Nyní sestavíme moment síly pro sílu F_k , kterou si můžeme označit jako F :

$$\begin{aligned} F_K &= \sin(-\theta) \cdot F_g = \sin(-\theta) \cdot mg = F \\ M_c &= l \times \sin(-\theta) \cdot mg = l \cdot \sin(-\theta) \cdot mg \end{aligned}$$

Na základě vztahu mezi momentem síly a momentem hybnosti sestavíme diferenciální rovnici kyvadla:

$$\begin{aligned} M_c &= \dot{H}_c \\ l \cdot \sin(-\theta) \cdot mg &= ml^2\ddot{\theta} \\ \sin(-\theta) \cdot g &= l\ddot{\theta} \\ \frac{g}{l} \cdot \sin(-\theta) &= \ddot{\theta} \end{aligned}$$

Výsledný tvar diferenciální rovnice tedy je:

$$\ddot{\theta} + \frac{g}{l} \cdot \sin(\theta) = 0 \quad (1)$$

2.2 Analytické řešení diferenciální rovnice

Pro zajímavost zde uvedu analytické řešení diferenciální rovnice, které poté porovnám s modelem v Simulinku a v SimScapu. V tomto případě není kyvadlo modelováno jako reálná koule, ale jako hmotný bod. Budeme vycházet z rovnice (1). Nejdříve si určíme počáteční podmínky. θ_0 označuje vychýlení kyvadla. Potom tedy θ_0 odpovídá amplitudě $[-\theta_0; +\theta_0]$ [11-12].

$$\begin{aligned} \theta(0) &= \theta_0 \\ \dot{\theta} &= 0 \end{aligned} \quad (2)$$

Dále si určíme tvar pro ω_0 a T , kde T je perioda:

$$\begin{aligned} \omega_0 &= \sqrt{\frac{g}{l}} \\ T &= \frac{2\pi}{\omega} \end{aligned}$$

Rovnici (1) rozšíříme $\dot{\theta}$:

$$\ddot{\theta} \cdot \dot{\theta} + \omega_0^2 \cdot \sin(\theta) \cdot \dot{\theta} = 0$$

Vytkneme derivaci:

$$\begin{aligned} \frac{d}{dt} \cdot \left[\frac{1}{2} \cdot \ddot{\theta} - \omega_0^2 \cdot \cos(\theta) \right] &= 0 \\ \ddot{\theta} &= 2 \cdot \omega_0^2 \cdot \cos(\theta) \end{aligned}$$

Upravíme a přidáme počáteční podmínky:

$$\ddot{\theta} = 2 \cdot \omega_0^2 \cdot \left[\cos(\theta) - \cos(\theta_0) \right]$$

Pomocí trigonometrického vzorce pro $\cos(\theta)$:

$$\cos(\theta) = 1 - 2 \cdot \sin^2 \left(\frac{\theta}{2} \right)$$

Dostaneme rovnici do následující tvaru:

$$\ddot{\theta} = 4 \cdot \omega_0^2 \cdot \left[\sin^2 \left(\frac{\theta_0}{2} \right) - \sin^2 \left(\frac{\theta}{2} \right) \right] \quad (3)$$

Zavedeme si nové proměnné y a k :

$$y = \sin^2 \left(\frac{\theta}{2} \right) \quad (4)$$

$$k = \sin^2 \left(\frac{\theta_0}{2} \right) \quad (5)$$

Z rovnic (2), (4) a (5) potom můžeme vyjádřit, že $y(0) = \sqrt{k}$. Úpravou rovnice (5). Dále zavedeme hodnotu θ jako funkci podle $\frac{dy}{dt}$. Z rovnice (5) dostáváme:

$$\frac{dy}{dt} = \frac{dy}{d\theta} \cdot \frac{d\theta}{dt} = \frac{1}{2} \cdot \frac{d\theta}{dt} \cdot \cos \left(\frac{\theta}{2} \right)$$

Umocněním rovnice (6) na druhou získáme vyjádření pro $\ddot{\theta}$:

$$\ddot{\theta} = \frac{4}{1 - y^2} \cdot \ddot{y} \quad (6)$$

Zavedením rovnice (4),(5) a (6) do rovnice (3) získáme tvar:

$$\frac{4}{1 - y^2} \cdot \ddot{y} = 4 \cdot \omega_0^2 \cdot (k - y^2)$$

Vyjádříme si \ddot{y} :

$$\ddot{y} = \omega_0^2 \cdot k \cdot (1 - y^2) \cdot \left(1 - \frac{y^2}{k} \right) \quad (7)$$

Zavedeme proměnné τ a t :

$$\begin{aligned}\tau &= \omega_0 \cdot t \\ t &= \frac{y}{\sqrt{k}}\end{aligned}\tag{8}$$

Potom můžeme rovnici (7) zapsat jako:

$$\left(\frac{dz}{d\tau}\right)^2 = (1 - z^2) \cdot (1 - k \cdot z^2)\tag{9}$$

Kde pro k platí $0 < k < 1$ a kde také platí následující:

$$\begin{aligned}z(0) &= 1 \\ \left(\frac{dz}{d\tau}\right)\Big|_{\tau=0} &= 0\end{aligned}\tag{10}$$

Vyřešením rovnice (9) pro $d\tau$ dostaneme:

$$d\tau = \pm \frac{dz}{\sqrt{(1 - z^2) \cdot (1 - k \cdot z^2)}}\tag{11}$$

Rovnici (11) zintegrujeme, abychom vyjádřili τ . Zintegrujeme to jako funkci z , proto za z zavedeme ξ :

$$\tau = - \int_1^z \frac{d\xi}{\sqrt{(1 - \xi^2) \cdot (1 - k \cdot \xi^2)}}\tag{12}$$

Výraz v (12) můžeme rozepsat do tvaru:

$$\tau = \int_0^1 \frac{d\xi}{\sqrt{(1 - \xi^2) \cdot (1 - k \cdot \xi^2)}} - \int_0^z \frac{d\xi}{\sqrt{(1 - \xi^2) \cdot (1 - k \cdot \xi^2)}}\tag{13}$$

To nám umožňuje vyjádřit τ jako funkci z a k :

$$\tau(z) = K(k) - F(\arcsin(z), k)\tag{14}$$

Kde funkce $K(m)$ reprezentuje kompletní eliptický integrál a funkce $F(\phi, m)$ je nekompletní eliptický integrál prvního druhu [12]. Tvary funkcí jsou potom následující:

$$K(m) = \int_0^1 \frac{dz}{\sqrt{(1 - z^2) \cdot (1 - m \cdot z^2)}}\tag{15}$$

$$F(\phi, m) = \int_0^\phi \frac{dz}{\sqrt{(1 - z^2) \cdot (1 - m \cdot z^2)}}\tag{16}$$

Víme, že $z = \sin(\phi)$. Potom je perioda oscilací čtyřnásobek času, který kyvadlo potřebuje, aby se z hodnoty $\theta = 0$ ($z = 0$) do $\theta = \theta_0$ ($z = 1$). Periodu tedy můžeme zapsat jako:

$$T = 4 \cdot t(0) = \frac{4 \cdot \tau(0)}{\omega_0} = \frac{4}{\omega_0} \cdot K(k) = \frac{2}{\pi} \cdot T_0 \cdot K(k) \quad (17)$$

Kde T_0 definujeme jako:

$$T_0 = \frac{2 \cdot \pi}{\omega_0} = 2\pi \cdot \sqrt{\frac{l}{g}} \quad (18)$$

T_0 je perioda kyvadla pro malé oscilace. Potom rovnici (14) můžeme zapsat jako:

$$F(\arcsin(z), k) = K(k) - \tau \quad (19)$$

To lze zapsat pomocí Jacobiovy eliptické funkce $sn(n, m)$. Eliptické Jacobiovy funkce jsou zobecněním funkcí sinus a kosinus. Funkci sinusamplituda lze definovat jako inverzní funkci k eliptickému integrálu prvního druhu, který jsme uváděli výše [11].

$$z = sn(K(k) - \tau, k) \quad (20)$$

Na základě rovnic (4),(5),(8) a (17) můžeme zapsat rovnici (20) jako:

$$\sin\left(\frac{\theta}{2}\right) = \sin\left(\frac{\theta_0}{2}\right) \cdot sn\left[K \cdot \left(\sin^2\left(\frac{\theta_0}{2}\right)\right) - \omega_0 \cdot t; \sin^2\left(\frac{\theta}{2}\right)\right] \quad (21)$$

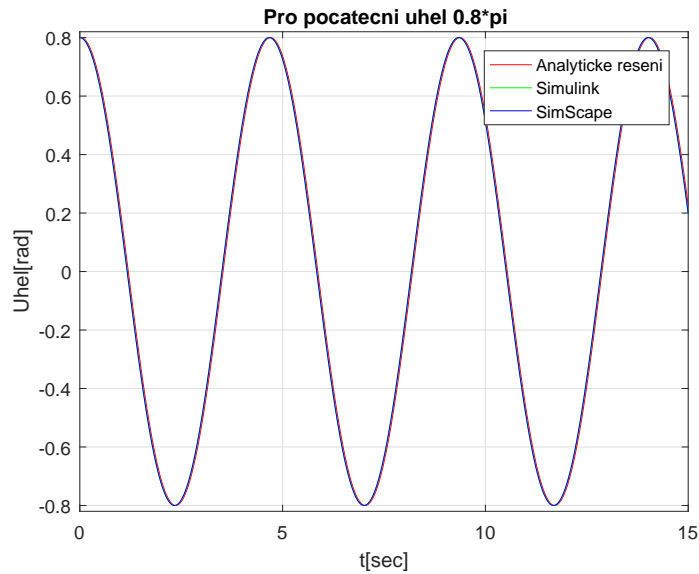
Nyní můžeme vyjádřit θ jako funkci času:

$$\theta(t) = 2 \cdot \arcsin\left\{\sin\left(\frac{\theta_0}{2}\right) \cdot sn\left[K \cdot \left(\sin^2\left(\frac{\theta_0}{2}\right)\right) - \omega_0 \cdot t; \sin^2\left(\frac{\theta}{2}\right)\right]\right\} \quad (22)$$

Což je přesně to, co jsme chtěli získat. Rovnice (22) popisuje pohyb kyvadla. Tato rovnice odpovídá pro amplitudu $\theta_0 < 0.75 \cdot \pi$ (135). Pokud budeme chtít větší amplitudu musíme za ω_0 dosadit tento výraz [11]:

$$\omega(\theta_0) = \frac{\pi \cdot \omega_0}{2 \cdot K\left[\sin^2\left(\frac{\theta_0}{2}\right)\right]} \quad (23)$$

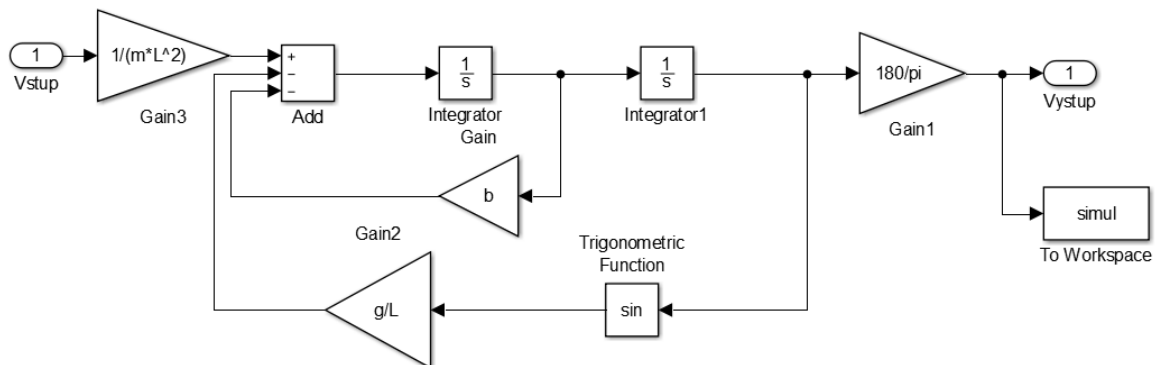
Pro ilustraci si zvolíme počáteční úhel jako 0.8π . S tímto nastavením spolu porovnáme naše analytické řešení s modelem v Simulinku a v SimScapu. Oba modely jsou rozebrány v následující kapitole 2.3. Výsledný graf je zobrazen na obrázku č.8.



Obrázek 8: Porovnání analytického řešení kyvadla

2.3 Simulink

Jedná se o nelineární model. Sestavení modelu pro diferenciální rovnici v Simulinku.



Obrázek 9: Model kyvadla v Simulinku

Nastavení jednotlivých parametrů je:

$$\begin{aligned}
 b &= 0 \\
 g &= 9.81 \\
 L &= 5
 \end{aligned}$$

Na vstup systému bude přivedena náhodná spojitá funkce. Gain1 slouží k přepočtu radiánů na stupně.

Protože modelujeme kyvadlo jako reálnou kouli, tak je nutné upravit model v Simulinku. Tuto úpravu děláme proto, abychom zohlednili matice setrvačnosti a abychom mohli na všechny modely použít stejný vstup. Za vstupní funkci je nutné dát gain3, který obsahuje

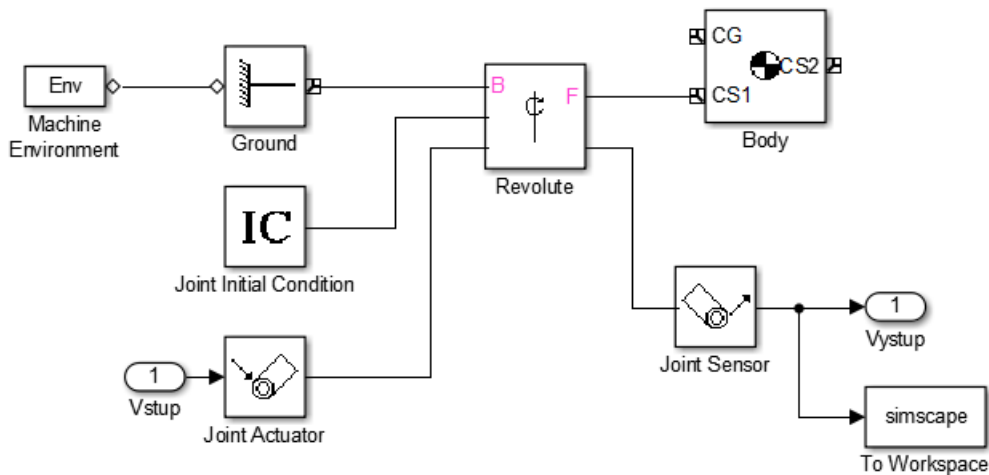
tento vzorec:

$$gain = \frac{1}{m \cdot L^2}$$

S touto úpravou dává model v Simulinku na ten samý vstup jako ostatní modely téměř stejné výsledky.

2.4 SimScape v Simulinku

Schéma modelu kyvadla pomocí knihovny SimScape v Simulinku.



Obrázek 10: Model Kyvadla pomocí SimScape v Simulinku

Je důležité si ukázat, jak je nastavený blok Body. V něm totiž musíme nadefinovat hmotnost, matici setrvačnosti a hlavně polohu těžiště. Délku kyvadla jsme definovali jako 5, proto polohu těžiště musíme posunout v ose y o 5, tj na -5 . V bodě $[0, 0, 0]$ je kloub kyvadla.

Mass properties

Mass: kg

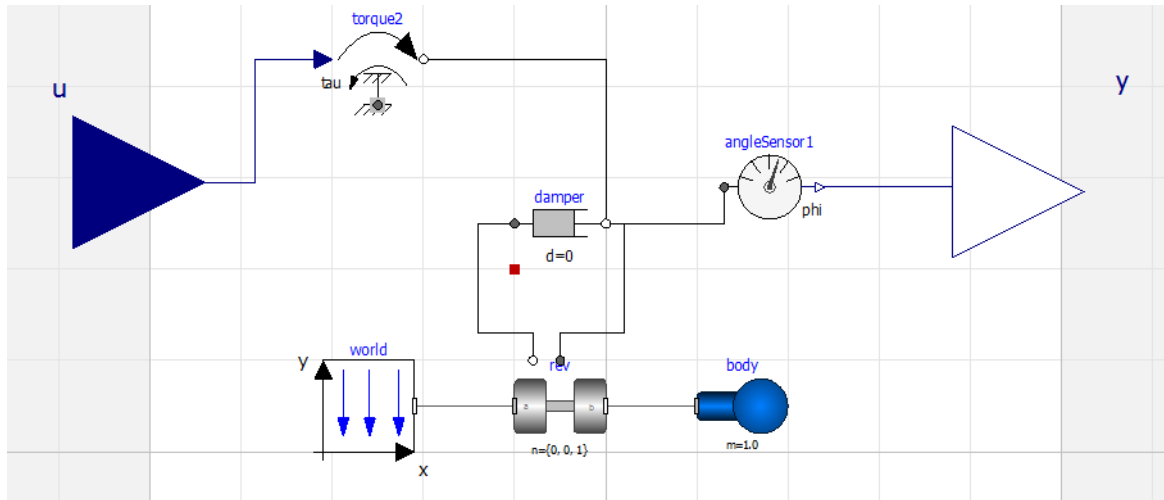
Inertia: g*cm²

Position	Orientation	Visualization				
Show Port	Port Side	Name	Origin Position Vector [x y z]	Units	Translated from Origin of	Components in Axes of
<input checked="" type="checkbox"/>	Left	CG	[0 -5 0]	m	World	World
<input checked="" type="checkbox"/>	Left	CS1	[0 0 0]	m	Adjoining	Adjoining
<input checked="" type="checkbox"/>	Right	CS2	[0 0 0]	m	CG	CG

Obrázek 11: Nastavení bloku Body

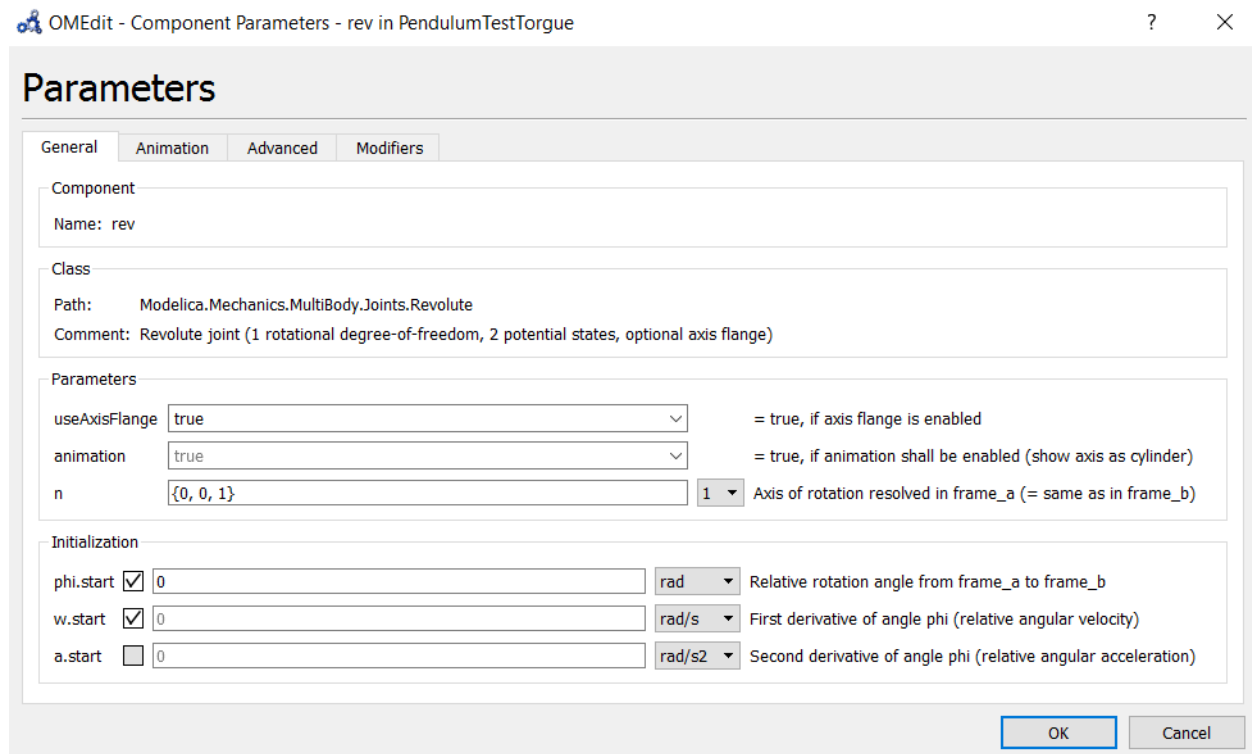
2.5 OpenModelica

Schéma modelu kyvadla v OpenModelice.



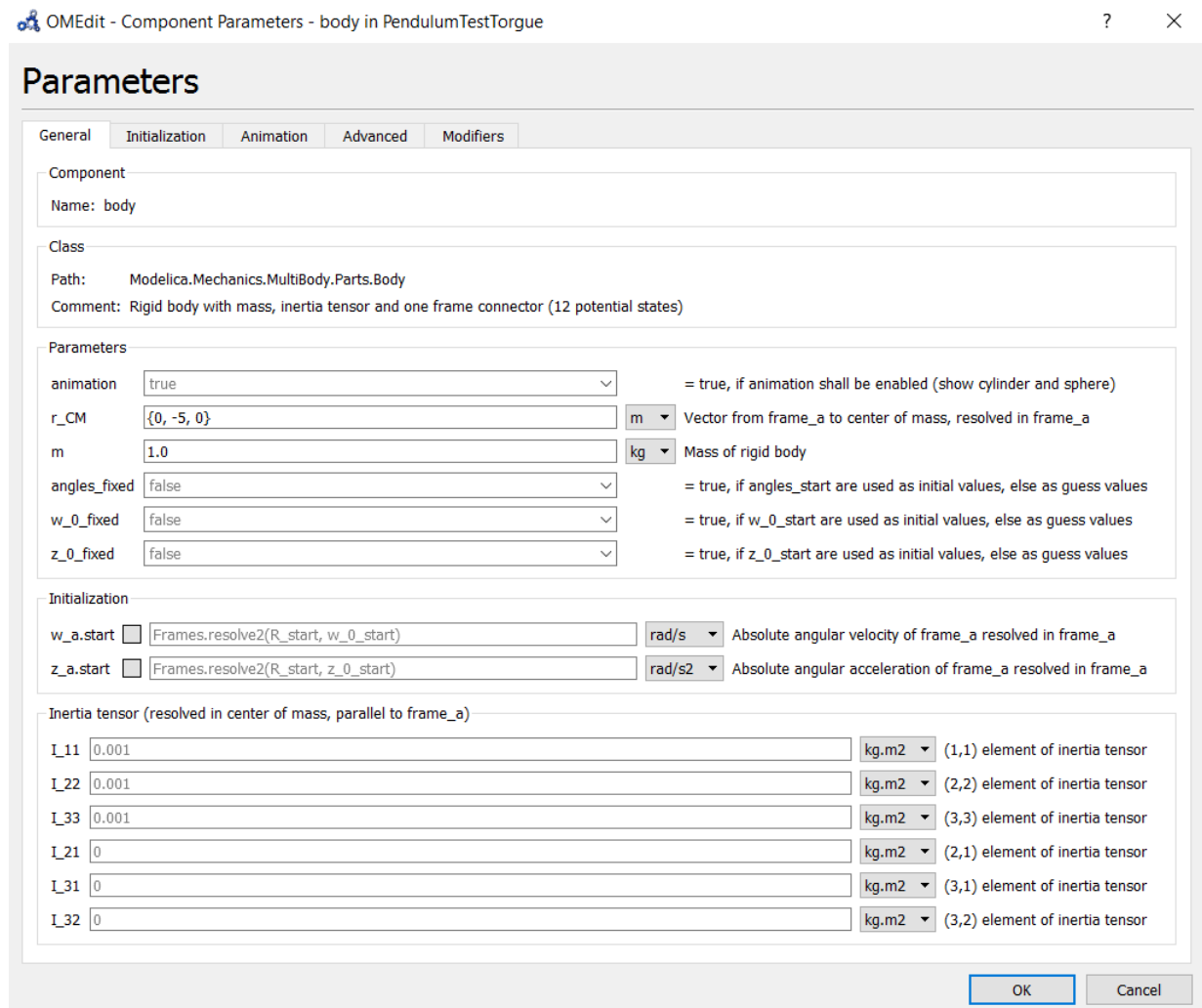
Obrázek 12: Model kyvadla v OpenModelice

Na obrázku č.13 je zobrazeno nastavení bloku rev, tedy kloubu, který umožňuje otáčení kyvadla.



Obrázek 13: Nastavení bloku rev v OpenModelice

Blok body nastavíme tak, že jeho těžiště, stejně jako v případě SimScapu, posuneme na ose y o 5 níž, tj. na -5 . Matici setrvačnosti nastavíme v osách I_{11} , I_{22} a I_{33} na 0.001. Hmotnost m nastavíme na 1.



Obrázek 14: Nastavení bloku body v Open Modelice

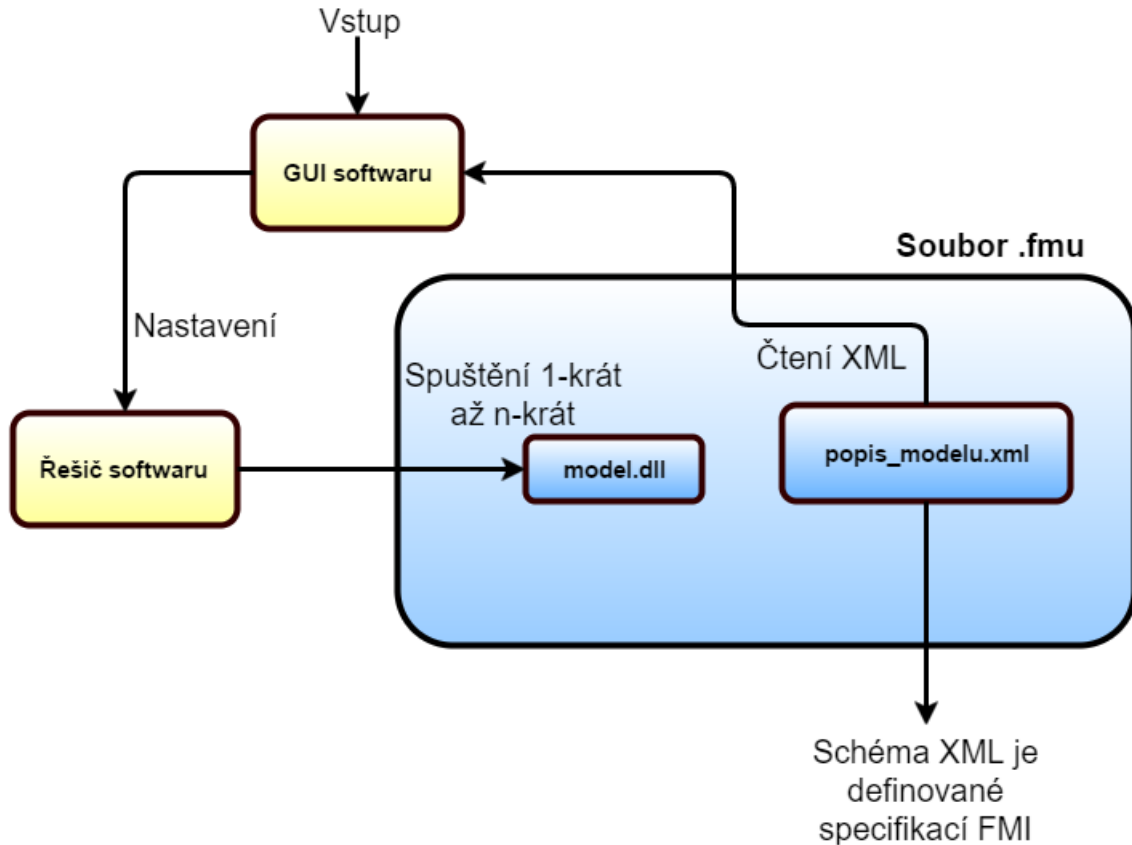
2.6 FMU

Před tím, než provedeme samotnou simulaci, tak si musíme říct, co to vlastně FMU je. Jedná se v podstatě o zip archiv, který v sobě obsahuje popis modelu v XML, rovnice popisující model a zdrojové soubory [4-9].

Popis modelu v XML - Obsahuje informace o modelu, které nejsou potřebné během simulace modelu. Jedná se například o jména vstupních signálů a atributů [3], [7].

Rovnice popisující model - Jsou definovány v jazyce C jako funkce. V FMU jsou obsaženy jako zdrojové kódy v jazyce C a jako binární kód v podobě .dll knihoven. DLL knihovny jsou definovány pro jednu nebo více platform (Windows, Linux...) [3], [7].

Zdrojové soubory - Obsahují dokumentaci v podobě html souborů. Ikony použité v modelu, tedy grafické bitmapové soubory. Nastavení map a tabulek, se kterými model pracuje během své inicializace[3], [7].



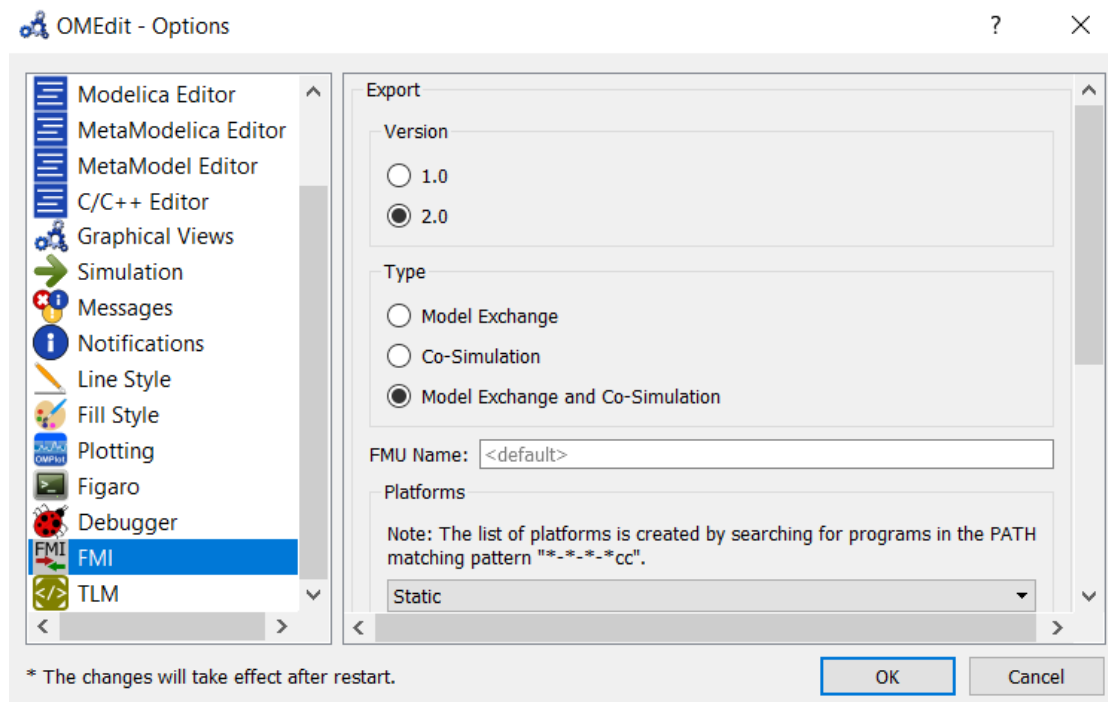
Obrázek 15: Schéma práce s .fmu

Na obrázku č.15 je znázorněna práce s .fmu souborem. Vstup je naše nastavení .fmu, například vychýlení kyvadla, vstup na daný systém atd... GUI softwaru představuje prostředí, ve kterém pracujeme s .fmu. V našem případě se jedná o Simulink. Řešič softwaru reprezentuje nastavený solver (např.: euler, ode...). Uvnitř .fmu model.dll znázorňuje DLL knihovny modelu a popis_modelu.xml obsahuje informace o modelu [8-9].

2.6.1 Jak vyexportovat FMU

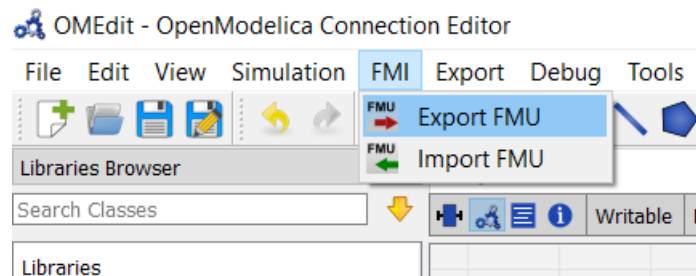
Nyní si ukážeme, jak vyexportovat model ve tvaru .fmu z Open Modelicy. Je důležité mít správně nakonfigurované FMI v nastavení Open Modelicy. To znamená, mít vybrané zda chceme jen Model Exchange nebo Co-Simulation a nebo obě naráz. Jakou chceme verzi FMI, zda 1.0 nebo 2.0. Verze 2.0 je aktuální.[1]

Nejprve si otevřeme nastavení v Open Modelice. Poté v nastavení vybereme sekci FMI, viz obrázek č.16.



Obrázek 16: Sekce FMI v nastavení

Celý model vyexportujeme tak, že vybereme na liště v Open Modelice FMI a klikneme na export FMU, viz obrázek č.17.



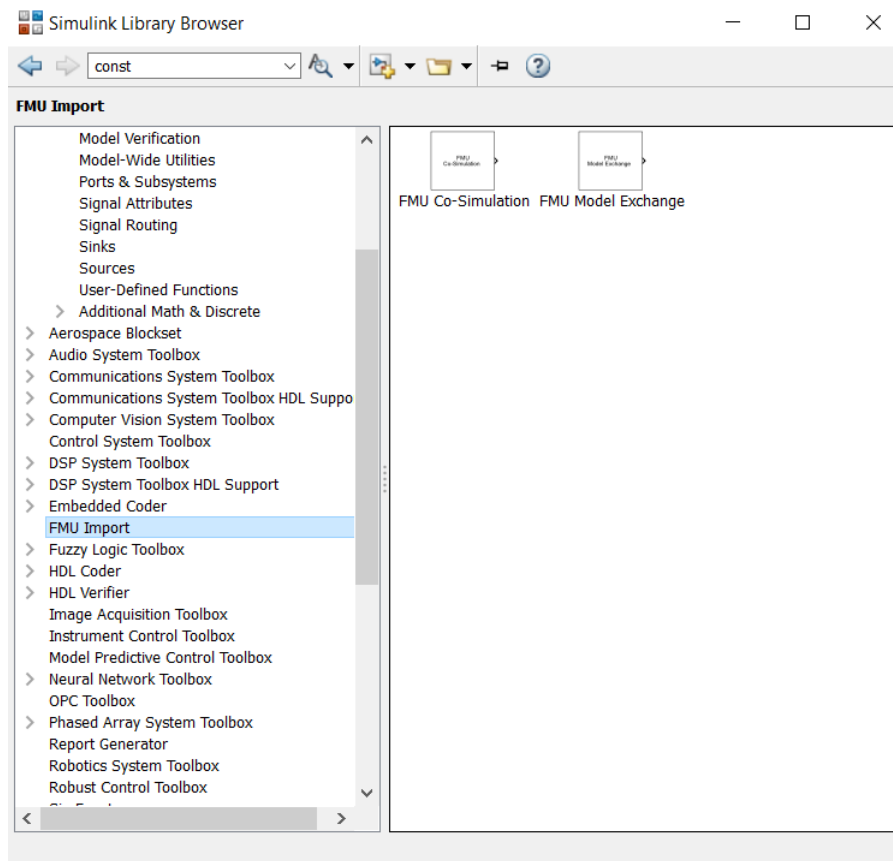
Obrázek 17: Vyexportování modelu jako FMU

FMU se ukládá na systémový disk do složky daného uživatele. Celá cesta k FMU na Win10 vypadá následovně :

C:\Users\NazevUzivatele\AppData\Local\ Temp\OpenModelica\OMEdit

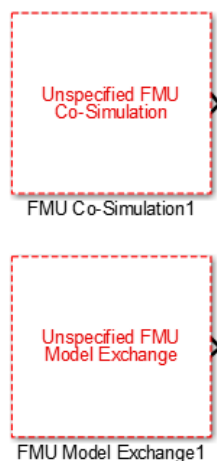
2.6.2 Jak pracovat s FMU v Matlabu

Pokud máme nainstalovaný balíček pro FMI, najdeme v knihovně bloků sekci FMU import. Zde jsou dva bloky. Jeden pro Co-Simulation a druhý pro Model Exchange [6].



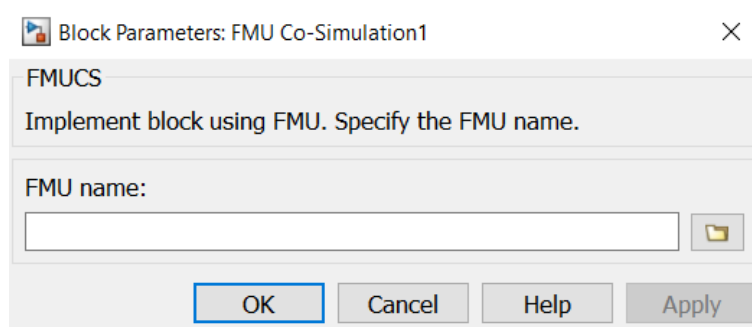
Obrázek 18: Bloky z FMU import v Simulinku

Po přetažení bloků do prostředí Simulinku se nám zobrazí takto:



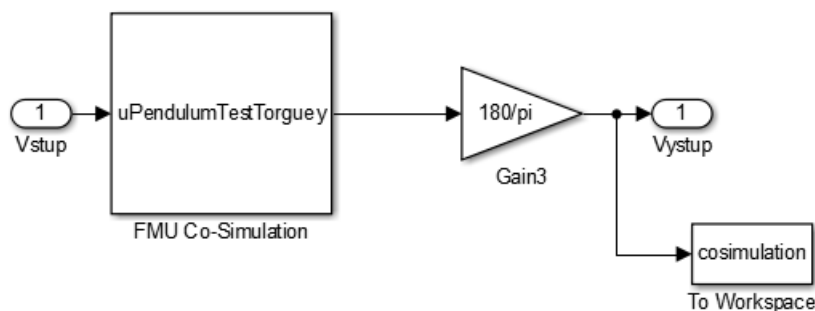
Obrázek 19: Nastavení bloků pro FMU

Červeně jsou zobrazeny proto, že nemají definovanou cestu k modelu ve tvaru FMU. Po rozkliknutí se nám zobrazí okno, které je stejné pro oba bloky a tak ho uvedu jen pro jeden. V něm vybereme cestu k požadovanému FMU.



Obrázek 20: Sekce FMU import v Matlabu

Náš model ve tvaru FMU vypadá následovně. Pro Model Exchange je schéma stejné. Jen místo FMU Co-Simulation jsme použili blok FMU Model Exchange.



Obrázek 21: Implementace bloku pro FMU Co-Simulation

2.7 Porovnání

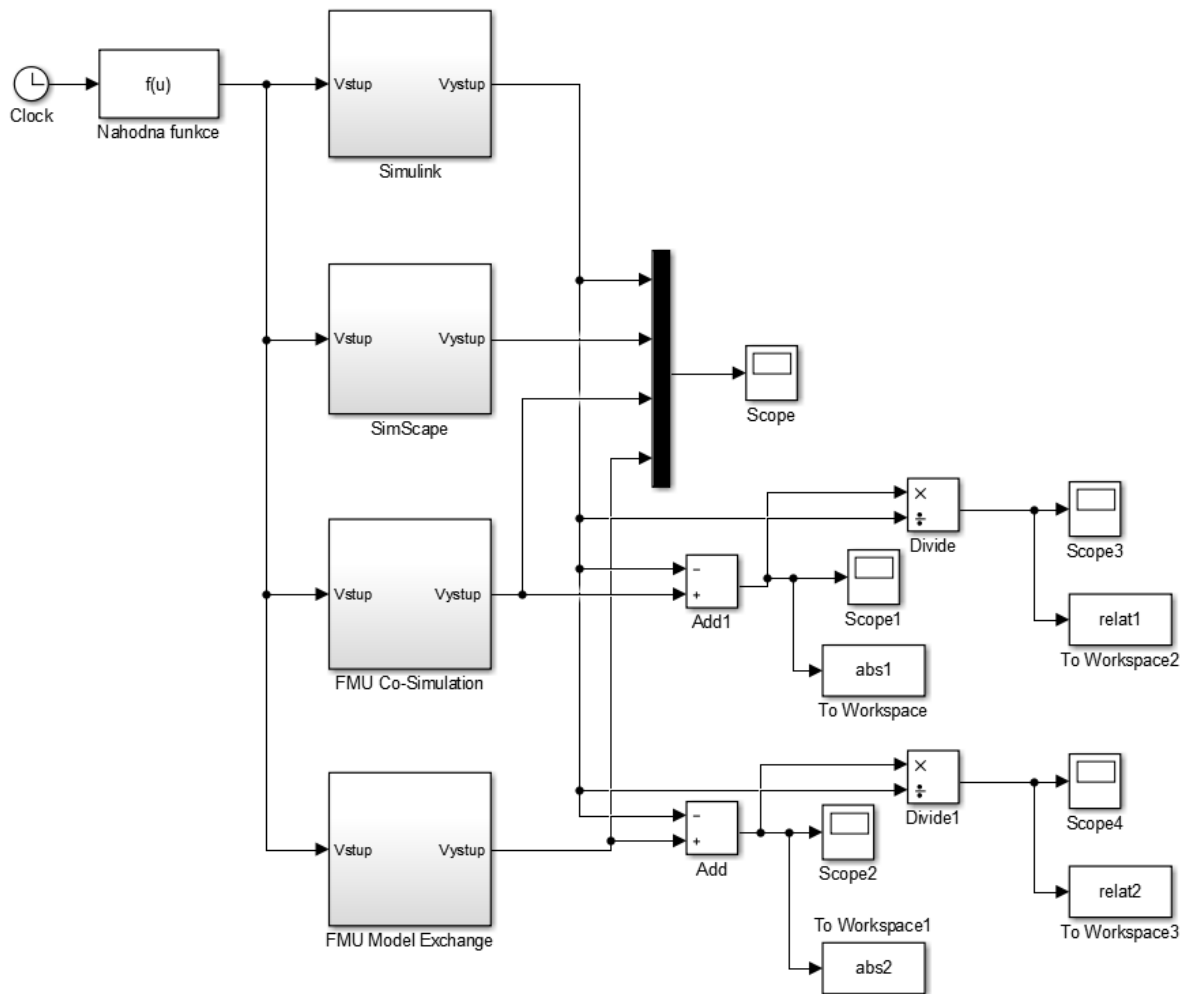
Pro porovnání všech modelů si vytvoříme schéma, které je na obrázku číslo 22. Porovnááme tedy model ze Simulinku, ze Simulinku pomocí knihovny SimScape a Open Modelicy vyexportované jako FMU a to jak Model Exchange, tak i Co-Simulation. Při simulaci jsme nastavili maximální krok solveru na $1e - 5$, čímž jsme získali simulaci s velmi jemným krokem.

U všech porovnání (tj. jak u kyvadla, tak u modelů energetického zařízení) budeme vykreslovat průběh absolutní a relativní chyby. Absolutní chyba je algebraický rozdíl mezi ukazovanou hodnotou (výstup ze Simulinku) a porovnávanou hodnotou (výstup z FMUs).

$$absolutní_chyba = výstup_FMU - výstup_Simulink$$

Relativní chyba je poměr absolutní chyby k porovnávané hodnotě.

$$relativní_chyba = \frac{absolutní_chyba}{výstup_Simulink}$$



Obrázek 22: Porovnání všech modelů kyvadla

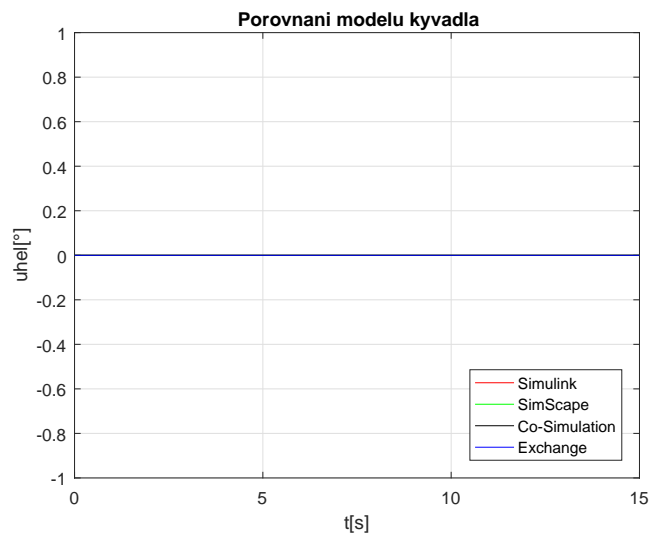
Vstupní náhodná funkce je v následujícím tvaru. Více informací je o ni v sekci - Problémy při řešení.

$$f(u) = (s(2) \cdot \sin(s(1) \cdot u) + s(2)) \cdot \sin\left(\frac{s(5) \cdot u \cdot [s(3) \cdot \cos(u) + s(3)]}{s(4)}\right)$$

U FMU Co-Simulation máme nastavený step time na 0.001 a poté na 0.00001. Čím menší ho nastavíme, tím dostaneme přesnější výsledky. To má však za následek výrazné prodloužení doby simulace.

2.7.1 V klidu

Jako vstup jsme zvolili místo náhodné funkce $f(u)$ konstantu, jejíž hodnota je nastavena na 0. Dostáváme tak, že na modely nepůsobí žádné vnější síly.

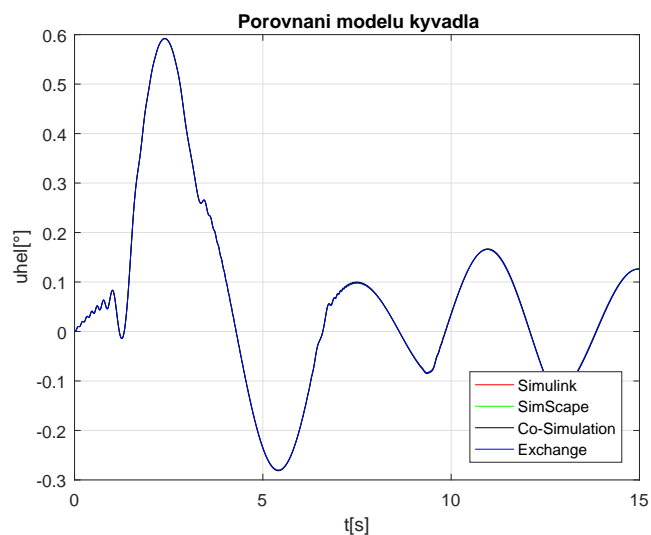


Obrázek 23: Odezva všech modelů na nulový vstup

Z obrázku č.23. vidíme, že všechny modely setrvaly v klidu.

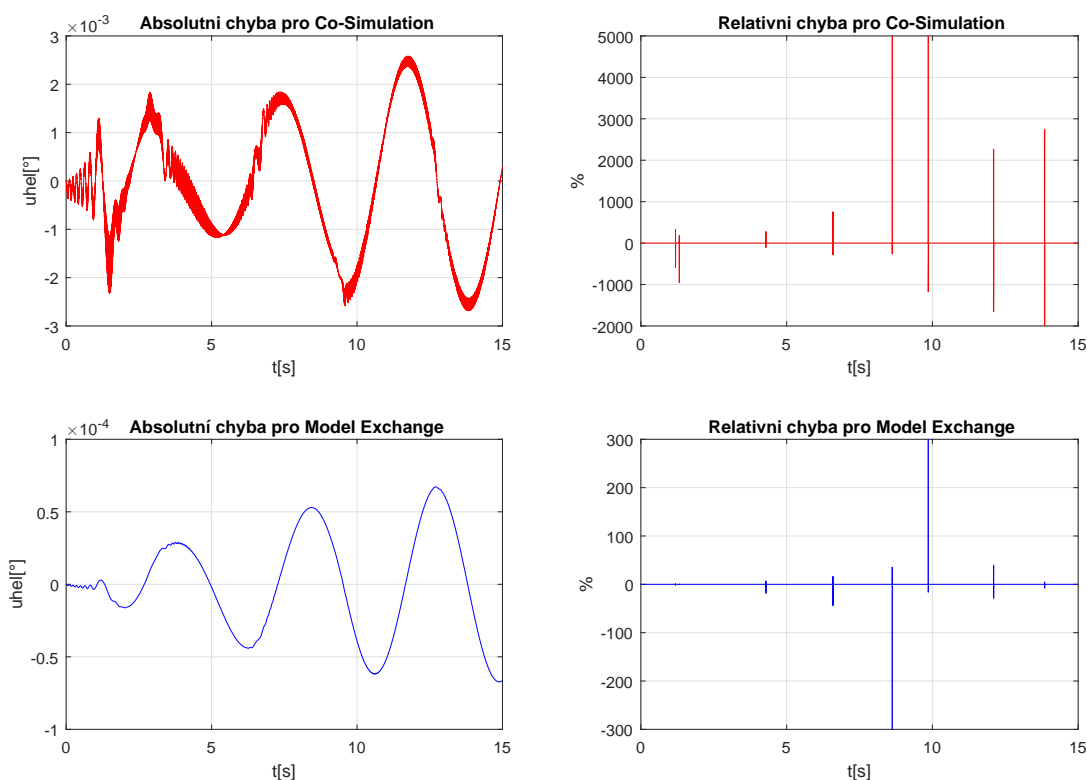
2.7.2 S náhodnou funkcí na vstupu

Vlivem nastavení náhodné funkce dostáváme pro každou simulaci různé výsledky. Někdy se parametry náhodné funkce vygenerují tak, že kyvadlo vychýlí jen o desetiny stupně a jindy zase o desítky stupňů. Tyto grafy jsme získali při nastaveném náhodném vektoru s na $[1, 9419; 2, 0782; 6, 5514; 2, 9567; 12, 355]$.



Obrázek 24: Odezva modelů na naší spojitou funkci

Průběh absolutní a relativní chyby pro Simulink a FMU Co-Simulation se step timem 0.001 je na obrázku č.25.



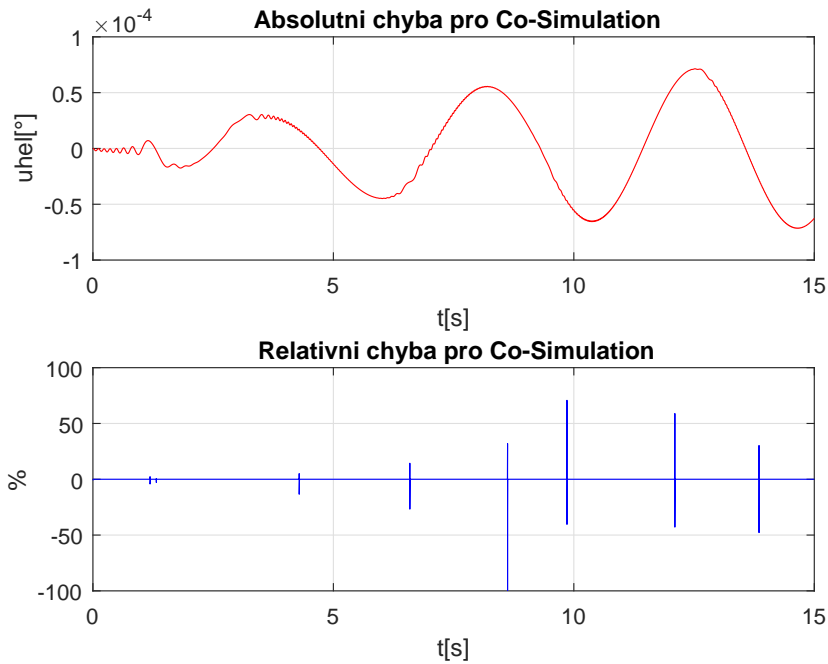
Obrázek 25: Absolutní a relativní chyba mezi Simulinkem a FMU

Na obrázku č.25 je porovnání FMU jako Co-Simulation (horní dva grafy) a Model Exchange (spodní dva grafy) s modelem v Simulinku. Vidíme, že s takto nastaveným step timem dává Model Exchange přesnější výsledky. Absolutní chyba pro nás nemá příliš velkou váhu a je tu uvedena spíše pro názornost. Větší vypovídající hodnotu má relativní chyba.

Pokud zjenníme nastavený step time v případě Co-Simulation, tak získáme přesnější výsledky. Pro nastavený step time na 0.00001 dostaneme grafy chyb, které jsou zobrazeny na obrázku č.26.

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	$2.58 \cdot 10^{-3} [^\circ]$	2742 %
CS - 0.00001	$7.1385 \cdot 10^{-5} [^\circ]$	70.75 %
ME	$6.722 \cdot 10^{-5} [^\circ]$	38.7 %

Tabulka 1: Maximální hodnoty chyb pro model kyvadla



Obrázek 26: Chyby mezi Simulinkem a Co-Simulation s krokem 0.00001

Maximální hodnoty pro chyby jsou obsaženy v tabulce č.1. Můžeme si všimnout, že oproti kroku 0.001 se absolutní chyba zmenšila řádově na 10^{-4} . S tím ale také vzrostla doba potřebná na simulaci. V kombinaci s malým maximálním krokem simulace byla doba potřebná na proběhnutí simulace několikanásobně vyšší než v předchozím případě. Proto je důležité si ujasnit, v případě složitějších simulací, jak moc přesný model potřebujeme.

U relativní chyby často dochází k tomu, že většinu času je na hodnotách velice blízkých nule, ale pokud model dosáhne nějakých extrémních hodnot (maximální/minimální hodnota, průchod nulou), tak její hodnota na krátký časový (většinou jeden krok simulace) okamžik mnohonásobně vzroste a poté opět klesá na hodnotu blízkou nule. Vždy jsem se snažil zvolit limity pro osu y tak, aby nezanikly ostatní výkyvy a spíše než jeden velký výkyv jsme viděli větší počet menších. Pokud totiž maximální chyba byla například řádu 10^5 , tak spolehlivě zakryla všechny ostatní menší výkyvy relativní chyby. Zmínka o tomto problému je i v podkapitole Problémy při řešení.

Ten samý postup s nastavováním limit pro osu y platí u všech následujících porovnáních a tak to již nebudeme znovu zmiňovat. Vždy jsme se snažili o to, aby byl graf co nejvíce vypovídající.

2.8 Porovnání s generátory náhodných signálů

Pro ověření výsledků a přesnosti mezi modely je odsimulujeme za použití dvou generátorů náhodných signálů, které přivedeme na vstup systému. Zdrojové soubory k oběma generátorům jsou v přílohách.

2.8.1 Mackey-Glassův generátor

Generátor náhodného signálu, který je založen na nelineární časově zpožděné diferenciální rovnici [13-14]. Ta má následující tvar:

$$\frac{dx(t)}{dt} = \frac{ax \cdot (t - \tau)}{1 + x \cdot (t - \tau)^{10}} - bx(t) \quad (24)$$

Ta může být numericky spočtena například metodou Runge-Kutta čtvrtého řádu v diskrétním čase s daným krokem [14]. Tvar pro výpočet následujícího stavu potom vypadá takto:

$$x(t + \Delta t) = \text{mackeyglass4k}(x(t), x(t - \tau), \Delta t, a, b)$$

Mackeyglass4k je funkce, která numericky vypočte Mackey-Glassovu diferenciální rovnici [13]. Pro matematický popis použijeme *m_q* místo *mackeyglass_eq*. Funkce v sobě ukrývá tyto výpočty:

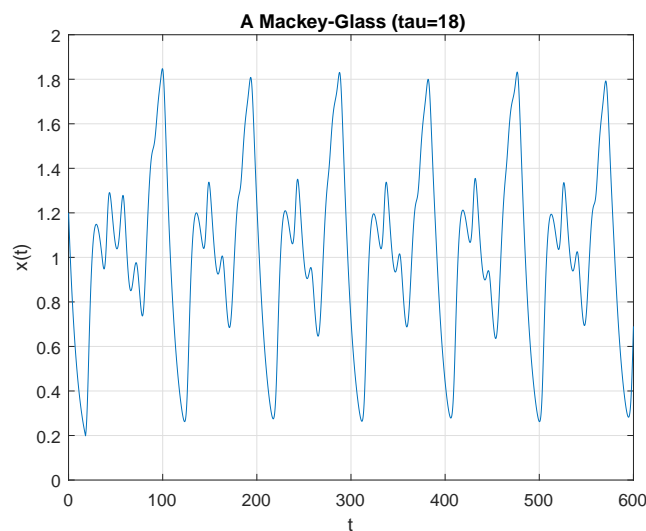
$$\begin{aligned} k_1 &= \Delta t \cdot m_q(x(t), x(t - \tau), a, b) \\ k_2 &= \Delta t \cdot m_q(x(t + 0.5 \cdot k_1), x(t - \tau), a, b) \\ k_3 &= \Delta t \cdot m_q(x(t + 0.5 \cdot k_2), x(t - \tau), a, b) \\ k_4 &= \Delta t \cdot m_q(x(t + k_3), x(t - \tau), a, b) \\ x(t + \Delta t) &= x(t) + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{6} + \frac{k_4}{6} \end{aligned}$$

Mackeyglass_eq je funkce, která se zadanými parametry *a* a *b* provede výpočet rovnice (24). Nyní si nadefinujeme naše zvolené parametry [14].

Parametr	Hodnota
a	0.3
b	0.1
τ	18
x_0	1.2
Δt	0.05
počet vzorků	12000

Tabulka 2: Hodnoty parametrů použitých v Mackeyglassovu generátoru

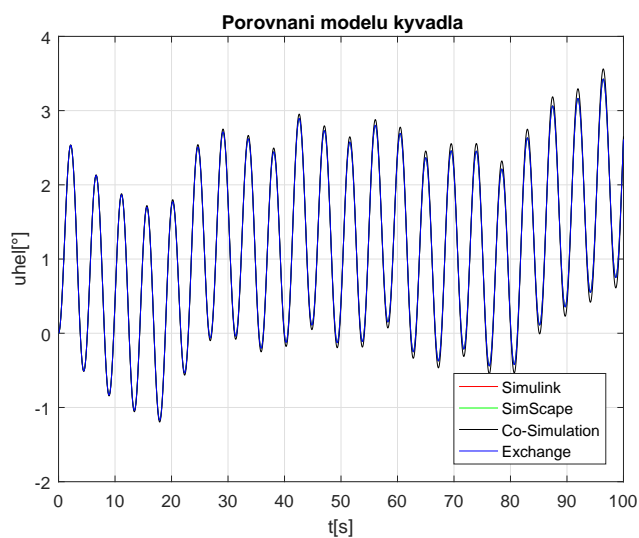
Tvar signálu takto nastaveného generátoru je vidět na obrázku č.27.



Obrázek 27: Výstup z Mackey Glassova generátoru

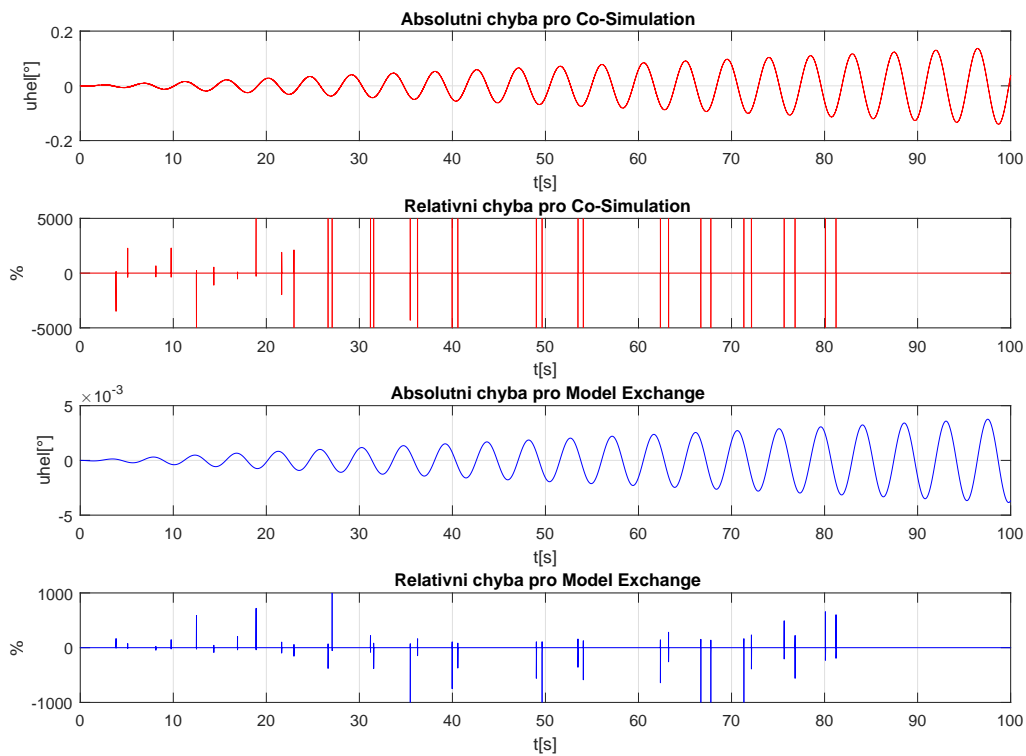
Takovýto signál použijeme jako vstup na náš systém kyvadel. Podoba modelu se od zobrazení na obrázku č. 22 nezměnila, pouze jsme změнили vstupní signál. Dobu simulace jsme pro názornost nastavili na 100 sekund.

Odezva jednotlivých modelů na takovýto signál je vidět na obrázku č.28.



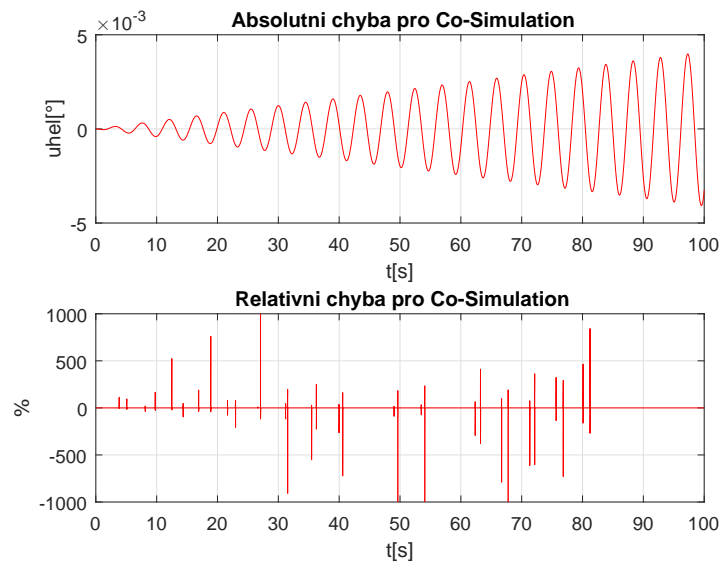
Obrázek 28: Odezva modelů na signál z Mackey Glassova generátoru

Absolutní a relativní chyba jednotlivých FMU je znázorněna na obrázku č.29.



Obrázek 29: Absolutní a relativní chyba modelů za použití Mackey Glassova generátoru

Z obrázku č. 29 je patrné, že chyba FMU se s délkou simulace neustále zvyšuje. Model Exchange si zachová chybu řádově kolem 10^{-3} , ale Co-Simulation se na konci simulace blíží k hodnotě 0.2. Vyzkoušíme zmenšit nastavený krok v bloku pro Co-Simulation na hodnotu 0.00001 za účelem zvýšit přesnost.



Obrázek 30: Absolutní a relativní chyba pro jemnější krok v Co-Simulation

Z obrázku č. 30 vidíme, že jsme dostali přesnější výsledky. Klesla také relativní chyba. V tabulce č.3 jsou maximální hodnoty získaných chyb.

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	0.14 [°]	4304.75 %
CS - 0.00001	$4.074 \cdot 10^{-3}$ [°]	907.467 %
ME	$3.85 \cdot 10^{-3}$ [°]	718.7 %

Tabulka 3: Maximální hodnoty chyb pro Mackey Glassův generátor

Zjemněním kroku se hodnoty maximálních chyb pro Co-Simulation blíží pro hodnoty Model Exchange. Při porovnání hodnot v tabulkách č.3 a č.4 je vidět, že pro rychleji se měnící vstupní signál generují modely větší chybu. Tento jev se projeví ještě mnohem více u Rösslerova generátoru. Pro nastavení limit grafů u relativní chyby platí to samé, jako v předcházejícím případě.

2.8.2 Rösslerův generátor

Generátor, který je založen na třech nelineárních obyčejných diferenciálních rovnicích (rovnice (25), (26) a (27)). Tyto rovnice původně studoval Otto Rössler [15]. Těmito rovnicemi je definován dynamický systém spojitého času, který ale vykazuje chaotickou dynamiku spojenou s fraktálami (členitý geometrický tvar, který lze rozdělit na části, které jsou každá zmenšenou kopií celku) generátoru.

$$\dot{x} = -z - y \quad (25)$$

$$\dot{y} = x + ay \quad (26)$$

$$\dot{z} = b + z \cdot (x - c) \quad (27)$$

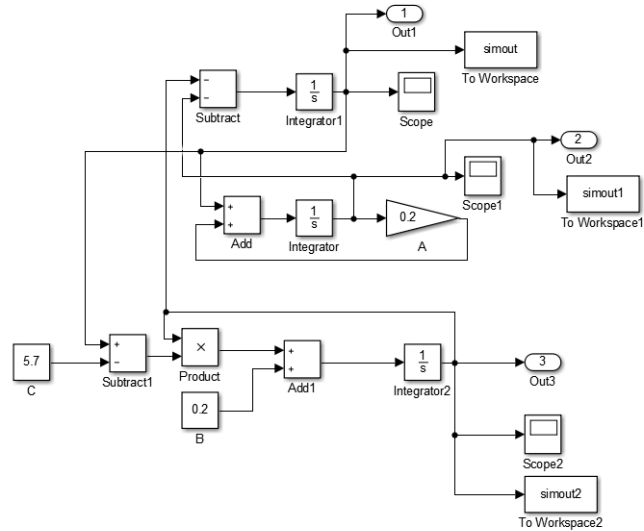
Takovýto systém se často označuje jako Rösslerův systém. Kde (x,y,z) jsou proměnné a náleží \mathbb{R}^3 . Tyto proměnné definují fázový prostor. (a,b,c) jsou nastavitelné parametry a také náleží \mathbb{R}^3 [15].

Rössler při studování tohoto chaotického generátoru stanovil jeho parametry na hodnoty ve sloupci Hodnota 1. Obecně se však častěji používají parametry ve sloupci Hodnota 2.

Parametr	Hodnota 1	Hodnota 2
a	0.2	0.1
b	0.2	0.1
c	5.7	14

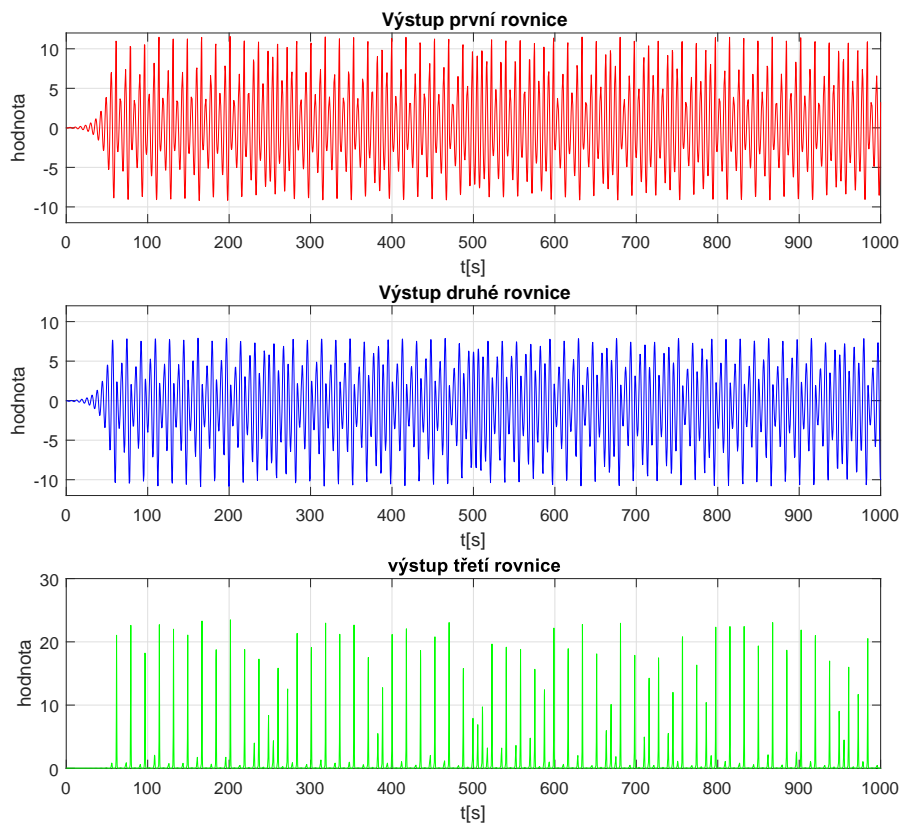
Tabulka 4: Hodnoty používaných parametrů pro Rösslerův generátor

Tento generátor jsme poskládali v Simulinku a jeho podoba je na obrázku č. 31.



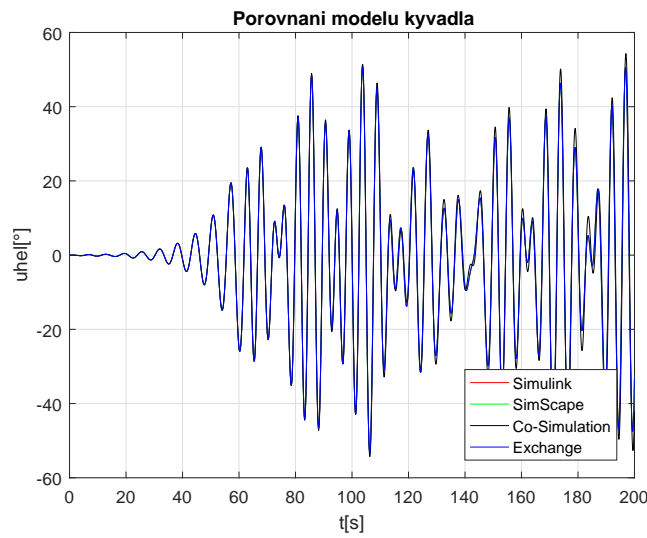
Obrázek 31: Model Rösslerova generátoru

My použijeme pro simulaci nastavení parametrů odpovídající sloupci Hodnota 1. Máme tři diferenciální rovnice, tzn. že budeme mít také tři výstupy. Jejich tvar pro naše nastavení je vidět na obrázku číslo 32.



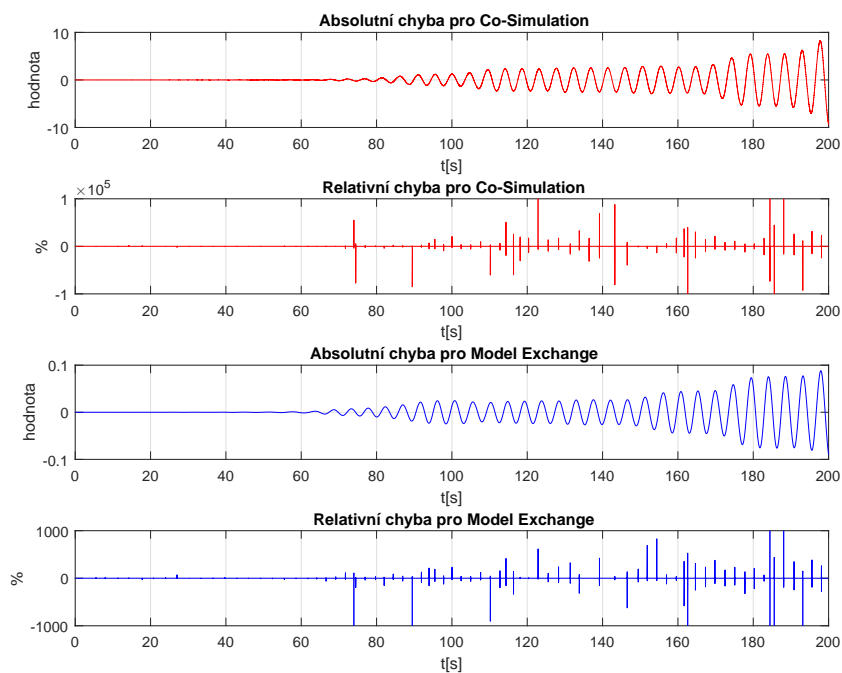
Obrázek 32: Výstupy z Rösslerova generátoru

Jako vstupní signál jsme si zvolili výstup druhé rovnice (tj. modrý graf na obrázku č. 32). Odezva systému na náš vybraný vstupní signál je na obrázku č.33.



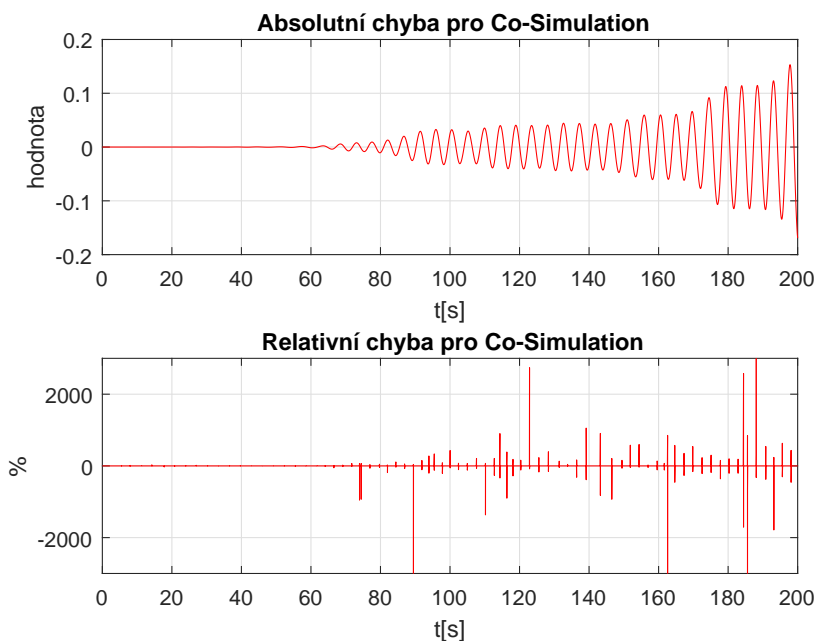
Obrázek 33: Odezva modelů na signál z Rösslerova generátoru

Absolutní a relativní chyba pro Model Exchange a Co-Simulation s krokem 0.001 je na obrázku č.34.



Obrázek 34: Absolutní a relativní chyba modelů za použití Rösslerova generátoru

Při zjemnění kroku u Co-Simulation dostaneme výsledky, které jsou zobrazeny na obrázku č.35.



Obrázek 35: Absolutní a relativní chyba s jemnějším krokem pro Rösslerův generátor

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	8.245 [°]	$8.796 \cdot 10^4$ %
CS - 0.00001	0.15321 [°]	2738.25 %
ME	0.08822 [°]	904.25 %

Tabulka 5: Maximální hodnoty chyb pro Rösslerův generátor

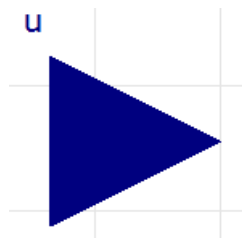
Maximální naměřené hodnoty chyb jsou uvedeny v tabulce č.5. Je vidět, že při rychle se měnícím periodickém signálu generují modely mnohem větší chybu než-li u naší testovací funkce. Tato chyba se s rostoucím časem zvětšuje. Z grafu č.34 a č.35 je patrné, že zjemněním kroku pro Co-Simulation dostaneme přesnější výsledky, ale ani s krokem 0.00001 se nedostaneme na úroveň chyby pro Model Exchange.

V případě Rösslerova generátoru jsou chyby ještě větší než u předcházejících dvou porovnání. To je dáno frekvencí generovaného signálu. V případě Co-Simulation můžeme zvýšením kroku dosáhnout přesnějších výsledků, ale jak bylo již zmíněno, je to vykoupeno mnohonásobně většími nároky na výpočetní výkon a hlavně reálný čas.

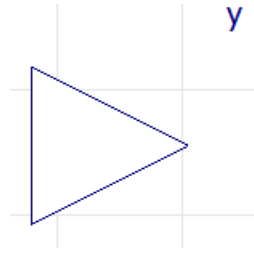
Simulace s Rösslerovým generátorem byla časově nejnáročnější ze všech tří porovnání. Pro představu na počítači s i7 (CPU - Intel Core i7 - 5700HQ 2.7Ghz) běžela simulace téměř dvě hodiny.

2.9 Problémy při řešení

1. **Instalace FMI modulu do Matlabu** - Tento problém byl vyřešen díky komunikaci s Mathworks, který nám poskytl dodatečný balíček do Matlabu - "Pilot Support Package", který umožňoval importovat FMI jako Co-Simulation a Model Exchange.
2. **Export z Modelicy** - Je důležité mít v Modelice správně nastavené možnosti pro FMI. Pokud chceme používat Co-Simulation i Exchange, je nutné mít vybrané obě metody současně. Důležitá je také verze FMI. My jsme používali verzi FMI 2.0.
3. **Import do Matlabu** - Pro správný import je nutné dodržet parametry viz. bod 2. Jinak dochází k chybovým hláškám, které hlásí problém se správným interpretováním dll knihovny daného modelu.
4. **Model Exchange z Modelicy do Matlabu** - Na 32 bitových verzích nefungoval správně. Tento problém byl vyřešen použitím 64 bitových verzí programů. Aktuální stav je takový, že novější verze Matlabu jsou pouze 64 bitové, ale Modelica je v základu stále 32 bitová. Poskytuje ale experimentální verzi - "Nightly Builds", která je 64 bitová.
5. **Modelica nightly build** - Jedná se o 64 bitovou verzi Modelicy, která je schopná exportovat funkční FMU. Pro otevření 64 bitových FMU je nutné, aby byl Matlab také 64 bitový.
6. **Export FMU z Matlabu** - Tato funkce zatím není podporovaná. Jedno z řešeních je FMI toolbox od Modelon, které jsme ale neměli možnost vyzkoušet. Odkaz - <http://www.modelon.com/products/fmi-tools/fmi-toolbox-for-matlab-simulink/>
7. **FMU se vstupy a výstupy** - Je nutné k modelu v Modelice přidat z knihovny Modelica -> Blocks -> Interface -> Realinput a Realoutput. Input připojit na akční člen a output na výstup z čidla. Je zde také možnost zapsat rovnici pro výstup přímo do kódu modelu. Tím pádem nám odpadá nutnost spojit výstup s jinou částí, toto spojení zajišťuje již zmíněná rovnice.



Obrázek 36: Blok pro reálný vstup



Obrázek 37: Blok pro reálný výstup

8. **Porovnávání solverů** - Zjistili jsme, že Matlab a Modelica obsahují pár stejných solverů. Jedná se například o solver euler a trapezoid. Jejich rozdíly jsou v řádu desetitisícin. Defaultně nastavené solvery jsou v Modelice dassl a v Simulinku je to Automatic solver selection a jeho typ je Variable-step. I tyto dva solvery se liší v řádu desetitisícin. Nicméně jejich interpretace se liší. V našem případě byly rozdíly tak malé, že nám příliš neovlivnily výsledky. Můžeme tedy říct, že defaultně nastavené solvery v obou programech poskytují velice podobné výsledky. Samozřejmě záleží na modelech, které simulujeme. V případě kyvadla dostáváme rozdíly řádově kolem tisícín, které pro nás nehrají větší roli a můžeme tak tento rozdíl zanedbat.
9. **Vstup se spojitou derivací** - Vytvořili jsme náhodnou spojitou funkci závislou na čase(u) a na náhodně vygenerovaném vektoru o pěti složkách($s(1) - s(5)$), kterou jsme použili jako vstup u našeho modelu kyvadla. Náhodný vektor se generuje pomocí exponenciálního rozdělení v Matlabu.

$$f(u) = (s(2) \cdot \sin(s(1) \cdot u) + s(2)) \cdot \sin\left(\frac{s(5) \cdot u \cdot [s(3) \cdot \cos(u) + s(3)]}{s(4)}\right)$$

10. **Nastavení limit os u grafů** Všechny grafy jsme vytvářeli v Matlabu. Snažili jsme se nastavit limity os tak, abychom maximalizovali vypovídající hodnotu grafu. To se týká hlavně relativní chyby, kde často docházelo k takovému skoku v průběhu, že vlivem jeho velikosti byly ztraceny nižší hodnoty.

2.10 Zhodnocení

2.10.1 Ohlédnutí za projektem 5

V rámci projektu 5 jsme s kolegou řešili základní možnosti FMI 2.0. Vyzkoušeli jsme si také vyzkoušeli, jaké jsou rozdíly mezi různými řešiči (solvery) v jednotlivých programech. Všechny tyto informace jsme demonstrovali na jednoduché diferenciální rovnici prvního řádu. Na základě naší analýzy můžeme říct, že defaultní nastavení řešice (solveru) v Matlabu a v OpenModelice poskytuje velmi podobné výsledky a to i když se jedná o dva odlišné solvery (V Matlabu Automatic solver selection typ - Variable-step a v OpenModelice dassl) [31-32].

2.10.2 Model kyvadla

Model kyvadla je značně složitější než obyčejná diferenciální rovnice prvního řádu. Největším problémem je odlišný přístup v modelování modelu. Diferenciální rovnice (model v Simulinku) nezohledňuje hmotnost a matice setrvačnosti kyvadla (reálné kyvadlo). SimScape a Open Modelica však ano, proto se výsledky mírně rozcházejí. Proto jsme museli model v Simulinku upravit.

Při použití naší náhodné funkce se tedy Model Exchange liší řádově o 10^{-5} a Cosimulation s nastaveným step timem na 0.001 o 10^{-3} . Pokud zjemníme step time na 0.00001 dostaneme se řádově na 10^{-5} . Toto zjemňování je však vykoupeno delší dobou potřebnou na simulaci. Relativní chyba má občasné velké výkyvy, ale jinak se po většinu času drží na hodnotě velice blízké nule. Situace je stejná i u Mackey-Glassova a Rösslerova generátoru. Zde jsou ovšem chyby výrazně větší, což je dáno tvarem vstupního signálu.

Někoho může zarazit průběh absolutní a relativní chyby. Tím je myšleno, že může nabývat i záporných hodnot. Problém je, že jako vstup používáme periodické funkce, které se dostanou i do záporných hodnot a poté se vrátí do kladných. Toto se přenáší na náš systém a ten tedy kmitá na jednu a poté zase na druhou stranu. Tzn. že to způsobuje oscilace mezi kladnými a zápornými hodnotami. U relativní chyby je důležité si uvést, že u naměřených veličin nabývá hodnot mezi 0-100%. To proto, že máme jasně definovanou hodnotu pro 100%, kterými hodnotu podělíme. Jenže v našem případě jsou za 100% brány hodnoty ze Simulinku, který se ale mění v čase. Z toho plyne, že se našich 100% neustále mění. Když se hodnoty ze Simulinku blíží k 0 nebo extrémním hodnotám, tak relativní chyba vystřelí velmi vysoko.

FMI 2.0 značně urychluje práci s exportovaným modelem. Dříve, když jsme chtěli data ze simulace z Open Modelicy interpretovat v Matlabu, museli jsme data importovat ze souboru, který byl ve složce daného modelu. Problémem bylo, že tento soubor v sobě v podstatě obsahoval jednu velkou matici, ve které však nebyly pouze hodnoty, ale také názvy proměnných a vše to bylo zapsané po sloupcích ze shora dolů. Museli jsme tuto matici transponovat, abychom se v tom mohli orientovat. Po identifikaci dat v matici jsme je zavolali jako daný řádek matice.

Díky FMI 2.0 si celý model exportujeme jako FMU a s ním dále pracujeme. Celý proces exportu a importu je značně jednodušší, rychlejší a uživatelsky mnohem příjemnější.

3 Model vodní turbíny

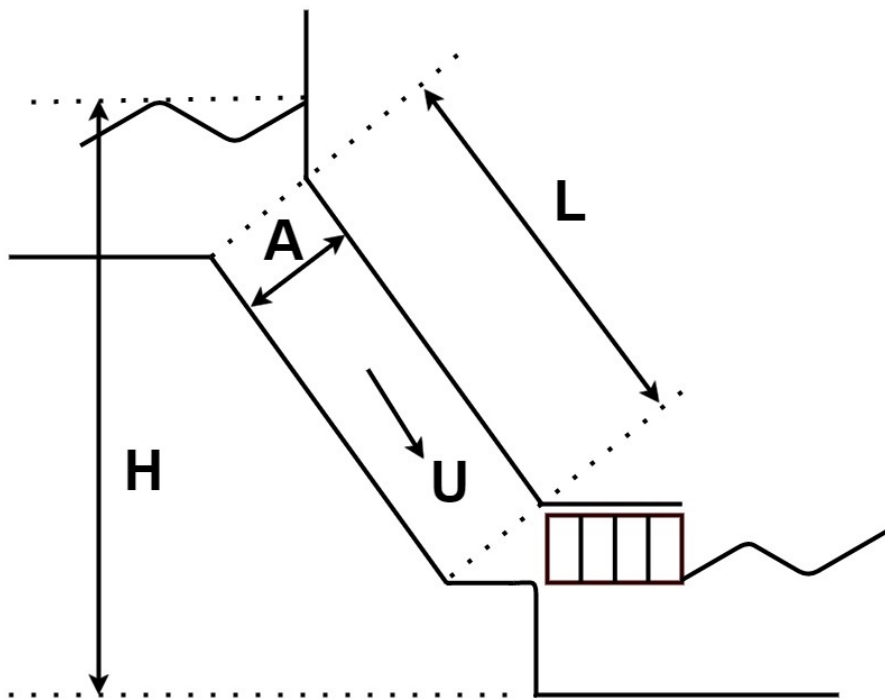
Vodní turbína je točivý mechanický stroj, který přeměňuje kinetickou či tlakovou energii vody na mechanickou energii. Vodní turbíny se rozdělují do dvou skupin.

- **Rovnotlaké-impulsní** - Jsou turbíny, kde se tlaková energie vody v rozváděcím zařízení přeměňuje na kinetickou energii vody a ta se poté přeměňuje na kinetickou energii oběžného kola turbíny. Tlak vody je všude stejný a za oběžným kolem se nemění. Do této skupiny patří například Walterova, Peltonova, turbína atd...
- **Přetlakové-reakční** - Takové turbíny, kde se tlaková energie vody přeměňuje na kinetickou energii oběžného kola turbíny. Tlak vody poté za oběžným kolem klesá. Sem patří například Francisova, Kaplanova, Tysonova turbína atd...

Při modelování vodní turbíny nás nezajímá o jaký přesně typ turbíny jde. Je jedno jaký model turbíny přesně jde. Základní princip fungování mají všechny stejný a to přeměnu vodního proudění (energie) na mechanický výkon, který potom roztáčí generátor [16-20], [22-23], [28-30].

Pro zjednodušení modelu uvažujeme tyto tři předpoklady:

- Je zanedbaný hydraulický odpor
- Přívodní potrubí je nepružné
- Voda je nestlačitelná



Obrázek 38: Schéma vodní turbíny

3.1 Sestavení modelu v Simulinku

Na základě diferenciálních rovnic sestavíme nelineární model vodní turbíny v Simulinku. Je důležité uvést, že nemám k dispozici žádná reálná data, na kterých bych mohl model postavit. Proto jsou všechny rovnice v normalizovaném tvaru. Tzn., že jejich jednotky jsou pu (per unit) [19], [22-23], [25], [28-29].

3.1.1 Parametry a rovnice

V následující tabulce jsou popsány všechny proměnné a parametry, které se v modelu vyskytují.

Značka	Význam
G	pozice ventilu
H	vzdálenost hladin(hydraulic haed)
A	průměr potrubí
ρ	hustota vody
g	gravitační konstanta
K_u, K_p, K_d	proporční koeficienty
L	délka potrubí
μ	efektivita turbíny
A_t	konstanta
qnl	ztráta proudění v potrubí

Tabulka 6: Názvy parametrů použitých v modelu turbíny

Vycházíme ze třech základních rovnic. Jedná se o rovnice pro velikost proudění(28), mechanický výkon(29) a změnu proudění(30).

$$U = K_u \cdot G \cdot \sqrt{H} \quad (28)$$

$$P_m = K_p \cdot H \cdot U \quad (29)$$

$$\frac{d\Delta U}{dt} = \frac{-A \cdot \rho \cdot g \cdot \Delta H}{\rho \cdot L \cdot A} \quad (30)$$

3.1.2 Nelineární model turbíny

Začneme definicí výstupní veličiny modelu turbíny. Jedná se o mechanický výkon, který lze také zapsat jako [23], [25-26]:

$$P_m = A_t \cdot H \cdot (U - qnl) \cdot \mu - K_d \cdot \Delta\omega \cdot G$$

Hodnotu parametru A_t lze získat z toho vztahu. *MVA* se rozumí Mega Volt Ampere, což je jednotka pro transformátory a generátory [23].

$$A_t = \frac{1}{H \cdot (U - qnl)} \cdot \frac{VykonTurbiny(MW)}{MVAhodnotaGeneratoru}$$

Protože nemám k dispozici žádné reálné informace o turbíně, tak zanedbáme její účinnost (tj. za μ dosadíme 1).

$$P_m = A_t \cdot H \cdot (U - gnl) - K_d \cdot \Delta\omega \cdot G$$

Ještě je důležité uvést si výpočet pro T_w . To je "startovní čas vody", tzn. čas potřebný k dosažení velikosti průtoku vody U_0 ze startovní pozice H_0 .

$$T_w \cdot \frac{d\Delta U}{dt} = -\Delta H \Rightarrow T_w = \frac{L \cdot U_0}{g \cdot H_0}$$

Nyní můžeme zapsat rovnice, které jsou uvažovány při modelování vodní turbíny [23]:

$$U = G \cdot \sqrt{H} \quad (31)$$

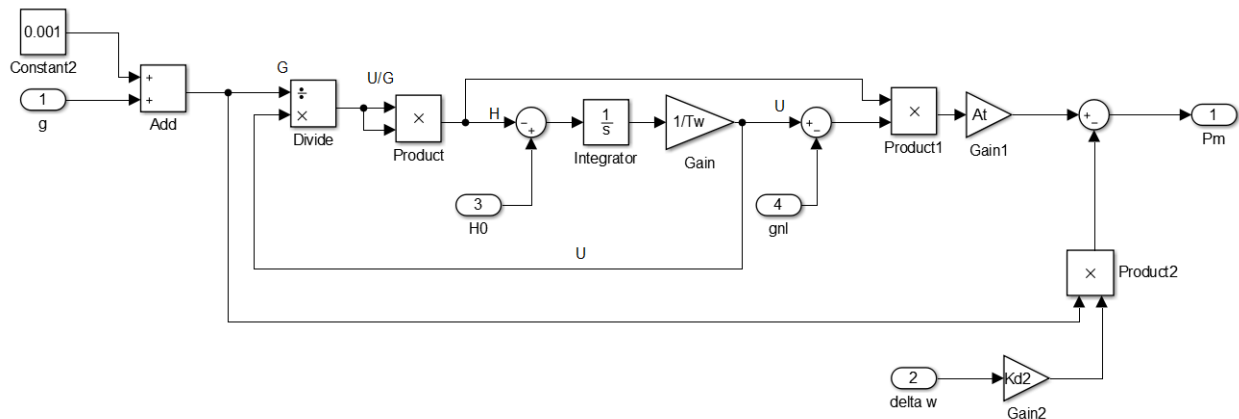
$$P_m = A_t \cdot H \cdot (U - gnl) - K_d \cdot \Delta\omega \cdot G \quad (32)$$

$$\frac{d\Delta U}{dt} = \frac{H - H_0}{T_w} \quad (33)$$

Rovnici (33) lze pomocí Laplaceovy transformace zapsat takto:

$$\frac{U}{H - H_0} = \frac{-1}{T_w \cdot S} \quad (34)$$

Z rovnic (31),(32) a (34) lze sestavit následující model vodní turbíny.



Obrázek 39: Schéma vodní turbíny v Simulinku

Tento model nejde odsimulovat, pokud je G rovno 0. Proto k hodnotě G přičítám malou hodnotu, abychom zamezili dělení nulou.

Do modelu tedy vstupuje nastavení ventilu G a změna rychlosti $\Delta\omega$. Vystupuje z něj mechanická energie P_m .

3.2 Sestavení modelu v Open Modelice

V Open Modelice nebudeme turbínu modelovat komponentově, ale zapíšeme do zdrojového kódu modelu přímo diferenciální rovnice. K modelu pouze přidáme vstupy a výstupy, na které nadefinujeme požadované proměnné. Nejde nám totiž o komponentový přístup, ale o to získat funkční FMU, které následně porovnáme s modelem v Simulinku [30-31].

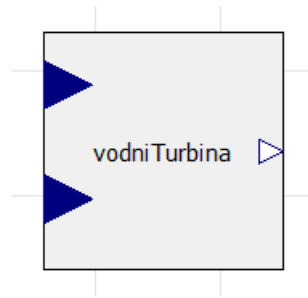
Zdrojový kód turbíny vypadá následovně:

```
model VodniTurbina
// konstanty pro turbinu
parameter Real Tw = 10;
parameter Real At=10;
parameter Real H0=10;
parameter Real gnl=0.01;
parameter Real Kd=0.01;
//promene pro turbinu
Real H;
Real U;
// G je vstup
// dw je vstup
// Pm je vystup
// bloky pro vstup a vystup
Modelica.Blocks.Interfaces.RealOutput Pm
annotation(Placement (...));
Modelica.Blocks.Interfaces.RealInput G
annotation(Placement (...));
Modelica.Blocks.Interfaces.RealInput dw
annotation(Placement (...));
// rovnice
initial equation
H=0;
equation
U = G * sqrt(H);
Pm = At * H * (U - gnl) - Kd * dw * G;
der(U) = (H0 - H) / Tw;
annotation(uses(Modelica(version = "3.2.2")));
end VodniTurbina;
```

Obrázek 40: Zdrojový kód vodní turbíny v Open Modelice

Jak je vidět z obrázku č.40, tak jsme pro přehlednost skryli automaticky generovaný kód po přidání bloku pro vstup a výstup. Důležitý je pouze jejich název.

Pokud takto popsaný model, budeme chtít použít v Open Modelice, bude jeho blok vypadat jako na obrázku č.41.

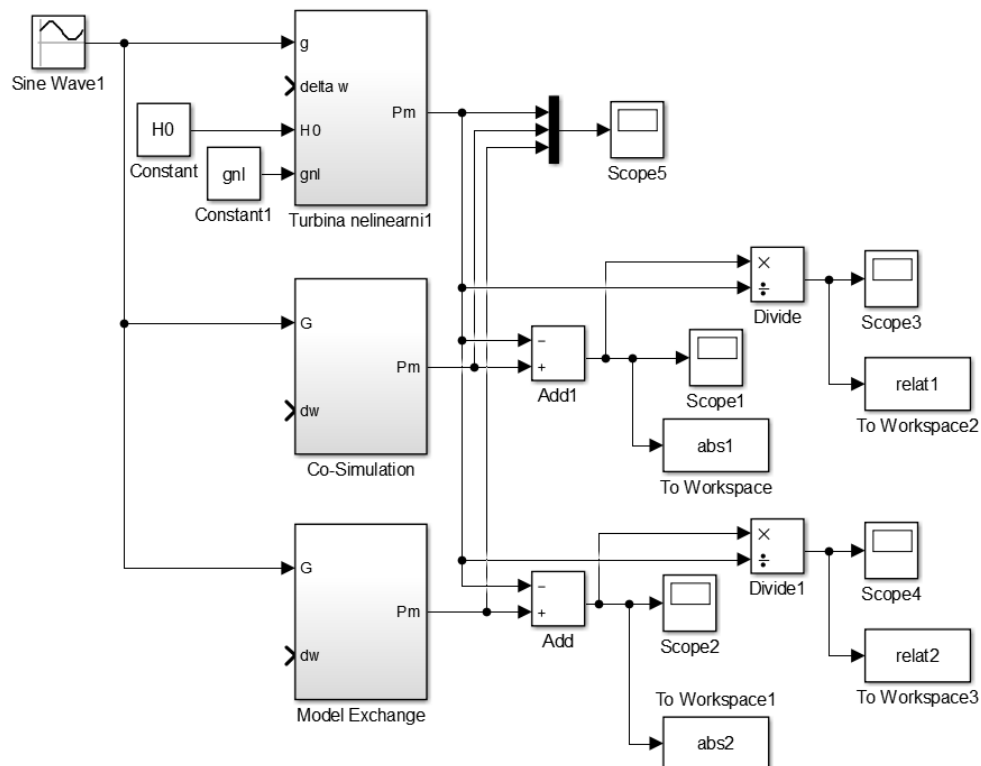


Obrázek 41: Blok pro vodní turbínu v Open Modelice

3.3 Porovnání modelů

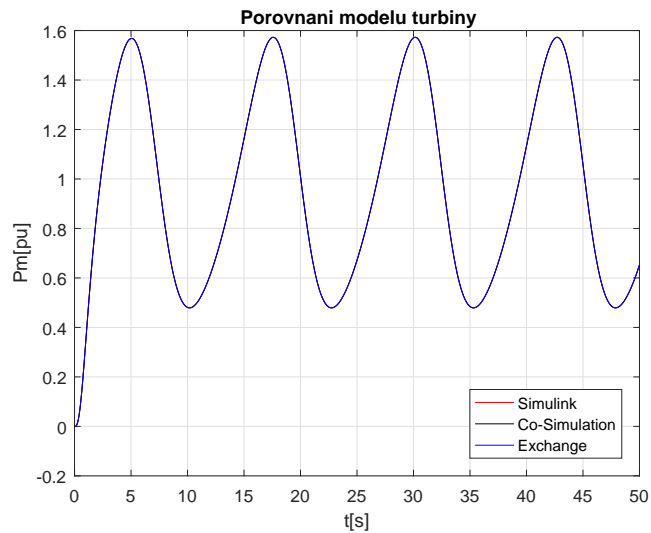
Při porovnávání jednotlivých modelů mezi sebou si jako vstup zvolíme generátor sinusové vlny. V Simulinku blok "Sin Wave". Amplitudu nastavíme na 0.5, bias na hodnotu 1 a frekvenci na 0.5. Toto nastavení použijeme z důvodu, aby byl vstup kladný. Je jasné, že za G nemůžeme dostat záporné číslo, protože G reprezentuje pozici brány.

Tento vstup použijeme místo například funkce Step, protože vlivem kmitů se zde více projeví rozdíly mezi jednotlivými modely. Budeme tedy porovnávat model turbíny v Simulinku s importovaným FMU z Open Modelicity a to jak Co-Simulation, tak Model Exchange.

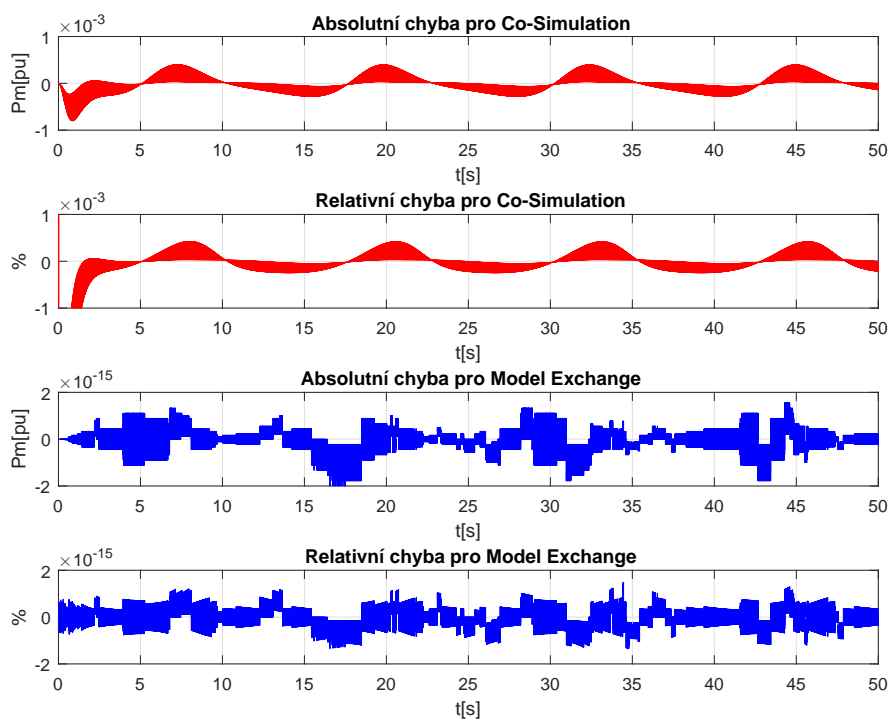


Obrázek 42: Schéma porovnání modelů vodní turbíny

Graf porovnání výstupu jednotlivých modelů je na obrázku č. 43. Maximální krok simulace je opět nastaven na $1e^{-5}$ a step v případě Co-Simulation je 0.001. Výsledky pro toto nastavení jsou na obrázku č.44. Maximální hodnoty chyb pro jednotlivé modely jsou potom uvedeny v tabulce č.7.

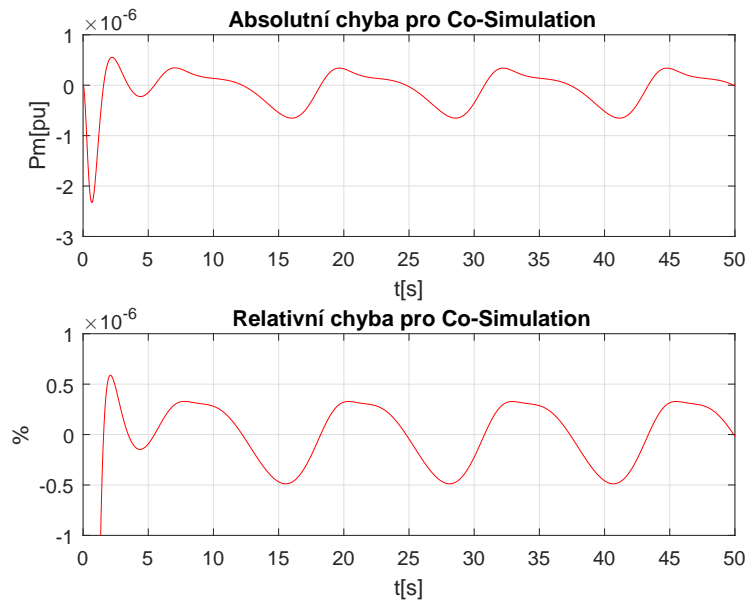


Obrázek 43: Odezva modelů vodní turbíny na sinusový vstup



Obrázek 44: Absolutní a relativní chyba FMU pro vodní turbínu

Graf chyb při snížení kroku v Co-Simulation na 0.00001 je na obrázku č.45.



Obrázek 45: Absolutní a relativní chyba FMU-CS vodní turbíny s jemnějším krokem

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	$8 \cdot 10^{-4}$ pu	$4.2 \cdot 10^{-4}$ %
CS - 0.00001	$2.372 \cdot 10^{-6}$ pu	$5.8784 \cdot 10^{-7}$ %
ME	$1.77 \cdot 10^{-15}$ pu	$1.474 \cdot 10^{-15}$ %

Tabulka 7: Maximální hodnoty chyb pro model vodní turbíny

Z hodnot v tabulce č.7 je patrné, že modely si jsou podobné a jejich chyby nabývají velice malých hodnot. Zjemněním kroku v Co-Simulation dostaneme o něco přesnější výsledek, stále se ale nepřiblížíme hodnotám pro Model Exchange, u kterého chyba dosahuje řádově na hodnoty 10^{-15} .

4 Model synchronního generátoru připojeného k silné síti

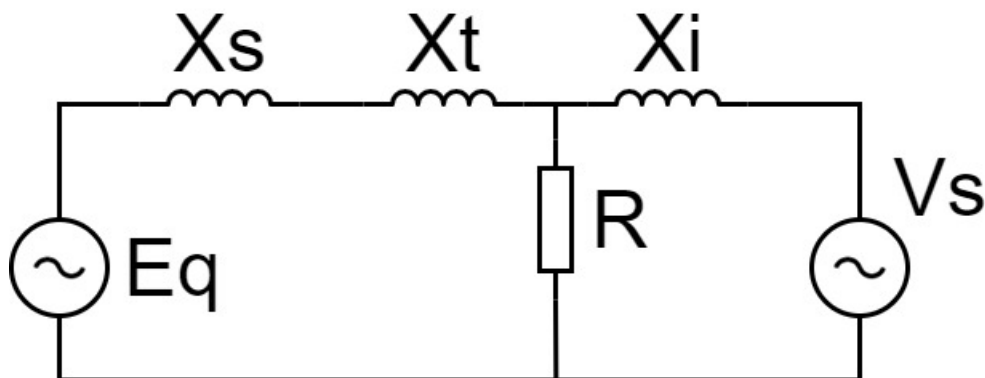
Pro naše potřeby bude stačit velice jednoduchý model generátoru. Některé proměnné jsou zcela zanedbány. Potřebujeme pouze, aby model byl schopný přeměnit mechanickou energii z turbíny na rychlost rotoru [21], [24]. Tu potřebujeme pro pozdější návrh regulační smyčky, protože servomotory ovládající polohu ventilu mají na vstupu změnu rychlosti. Z toho plyne, že na vstupu celého systému bude naše požadovaná rychlost nebo otáčky a od této nastavené hodnoty se bude odečítat hodnota vycházející z generátoru [23], [28-29].

Generátor modelujeme jako jednu linku, která je připojena do nekonečné sběrnice. Parametry a proměnné, které se v modelu generátoru vyskytují, jsou popsány v následující tabulce [21], [23]:

Značka	Význam
V_s	zdroj voltu na nekonečné sběrnici
R	zátěž
V_g	volty na generátoru
X_t	impedance měniče
X_s	impedance zdroje
$R_i + jX_i$	impedance přenosu
E_q	elektromotorická síla
H, D	konstanty
δ	úhel natočení rotoru generátoru
ω_0	zesílení generátoru

Tabulka 8: Názvy parametrů použitých v modelu generátoru

Elektrické schéma našeho generátoru je na obrázku č.46.

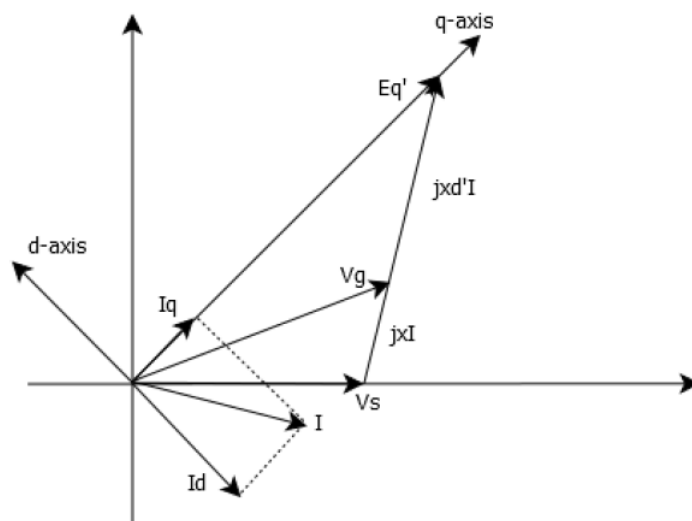


Obrázek 46: Elektrické schéma generátoru

4.1 Sestavení modelu v Simulinku

Při sestavení modelu v Simulinku vyjdeme z podoby fázorového diagramu. Z něj můžeme určit vztahy mezi jednotlivými veličinami [23], [25].

Veškeré hodnoty použitých parametrů jsou uvedeny v kapitole 7 - Hodnoty parametrů.



Obrázek 47: Fázorový diagram generátoru

Závislost mezi V_g a V_s lze zapsat jako:

$$V_g = V_s + j \cdot x_d \cdot I$$

Vztah mezi E'_g a V_s je:

$$E'_g = V_s + j \cdot x'_d \cdot I$$

Rovnice popisující točení generátoru má tedy tento tvar:

$$M \cdot \frac{d\Delta\omega}{dt} = P_m - P_e - P_d$$

Kde P_e představuje elektrický výkon (někdy se označuje také jako "air-gap"), P_d je tlumení, M je točivý moment.

Dále si vyjádříme M , P_d a P_e :

$$\begin{aligned} M &= 2 \cdot H \\ P_d &= D \cdot \Delta\omega \\ P_e &= \delta \cdot K_e \end{aligned}$$

Vztah pro úhel rotoru a jeho rychlosti je popsán takto:

$$\frac{d\delta}{dt} = \Delta\omega$$

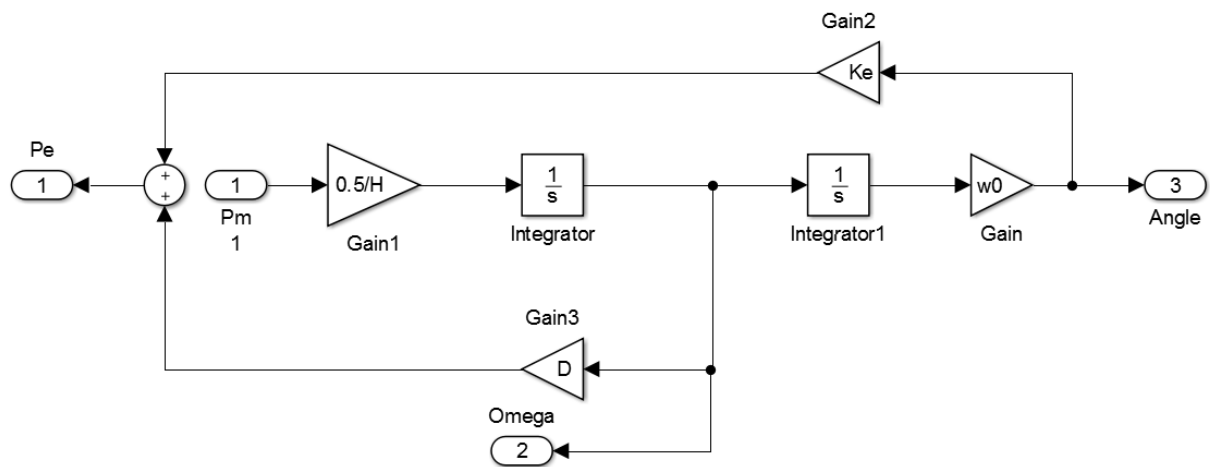
Celý model si zjednodušíme tím, že zanedbáme E_f (vnitřní elektromotorická síla) a I_d (proud kotvy). To si můžeme dovolit, protože obě hodnoty jsou velmi malé. Tím pádem můžeme zapsat E'_g jako konstantu. Efekt tlumení vinutí je také zanedbán. To má za následek, že můžeme zapsat E'_d jako konstantu [21], [23].

S tímto zjednodušením lze při modelování generátoru uvažovat jen následující rovnice:

$$M \cdot \frac{d\Delta\omega}{dt} = P_m - P_e - P_d \quad (35)$$

$$\frac{d\delta}{dt} = \Delta\omega \quad (36)$$

Na základě rovnic (35) a (36) sestavíme model generátoru.



Obrázek 48: Model generátoru v Simulinku

Do generátoru tedy vstupuje mechanická energie P_m a vystupuje elektrická energie P_e , rychlost rotoru ω a úhel natočení rotoru $\text{Angle}(\delta)$.

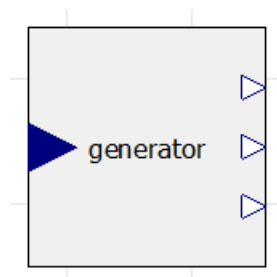
4.2 Sestavení modelu v Open Modelice

K sestavení modelu v Open Modelice jsme přistoupili stejně jako v případě vodní turbíny. Přímo do zdrojového kódu modelu jsme sepsali rovnice popisující model generátoru a tento celek opět vyexportovali jako FMU a to následně porovnali.

```
model GeneratorV3
  parameter Real H = 3.1;
  parameter Real w0 = 314;
  // 100 * pi
  parameter Real Ke = 1.27;
  parameter Real D = 2;
  //vstup Pm
  Real Angle1;
  Modelica.Blocks.Interfaces.RealInput Pm
  annotation (...);
  // vystup omega
  Modelica.Blocks.Interfaces.RealOutput Omega
  annotation (...);
  // vystup Pe
  Modelica.Blocks.Interfaces.RealOutput Pe
  annotation (...);
  // vystup Angle
  Modelica.Blocks.Interfaces.RealOutput Angle
  annotation (...);
  //Upraveny Pm
  Modelica.Blocks.Interfaces.RealOutput UpravenyPm
  annotation (...);
equation
  UpravenyPm = Pm - Pe;
  der(Omega) = UpravenyPm * 0.5 / H;
  der(Angle1) = Omega;
  Angle = Angle1 * w0;
  Pe = Angle * Ke + Omega * D;
  annotation(uses(Modelica(version = "3.2.2")));
end GeneratorV3;
```

Obrázek 49: Zdrojový kód generátoru v Open Modelice

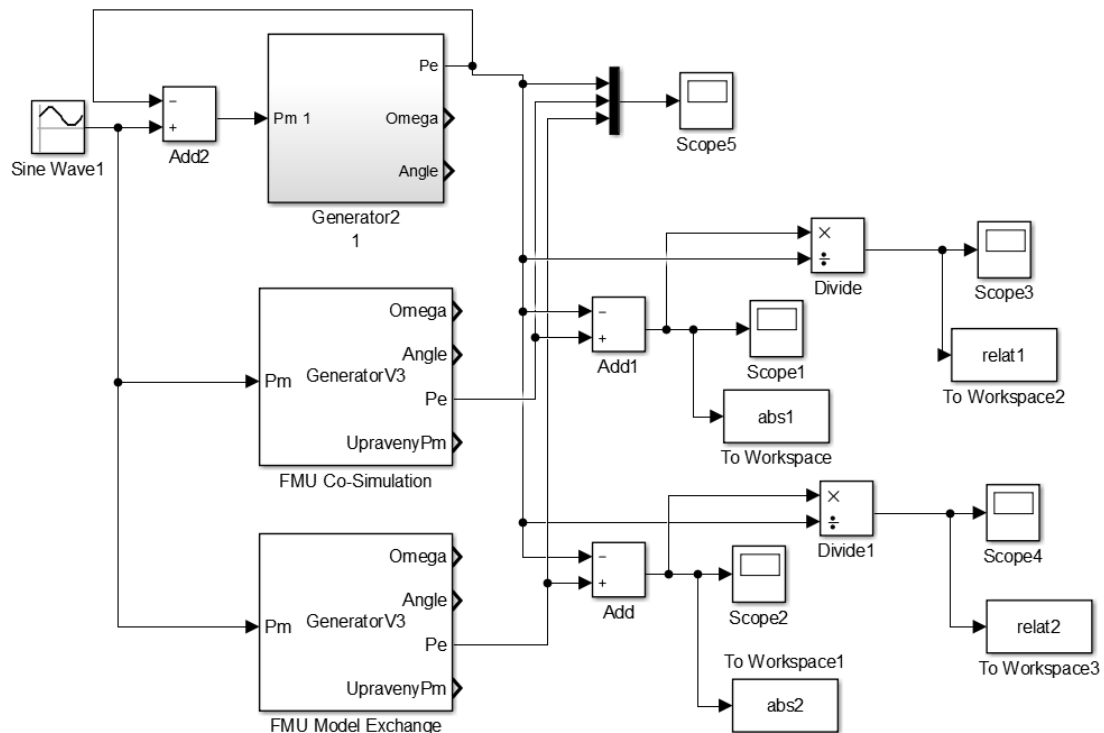
Opět jsem v ukázce kódu na obrázku č.49 zkrátil automaticky generované části kódu pro vstupy a výstupy. Důležité jsou pouze jejich názvy. Na obrázku č.50 je znázorněn blok generátoru.



Obrázek 50: Blok pro generátor v Open Modelice

4.3 Porovnání modelů

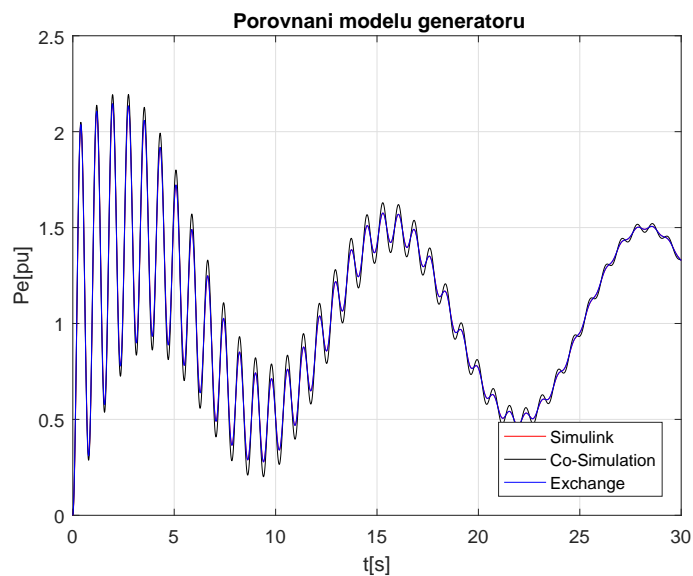
Pro porovnání jednotlivých modelů použijeme stejný vstup jako při porovnávání modelů vodní turbíny. Opět zde budeme porovnávat modely vytvořené v Simulinku a v Open Modelice jako FMU Co-Simulation a FMU Model Ex-Change. Schéma porovnání je znázorněno na obrázku č.51.



Obrázek 51: Porovnání modelů generátoru

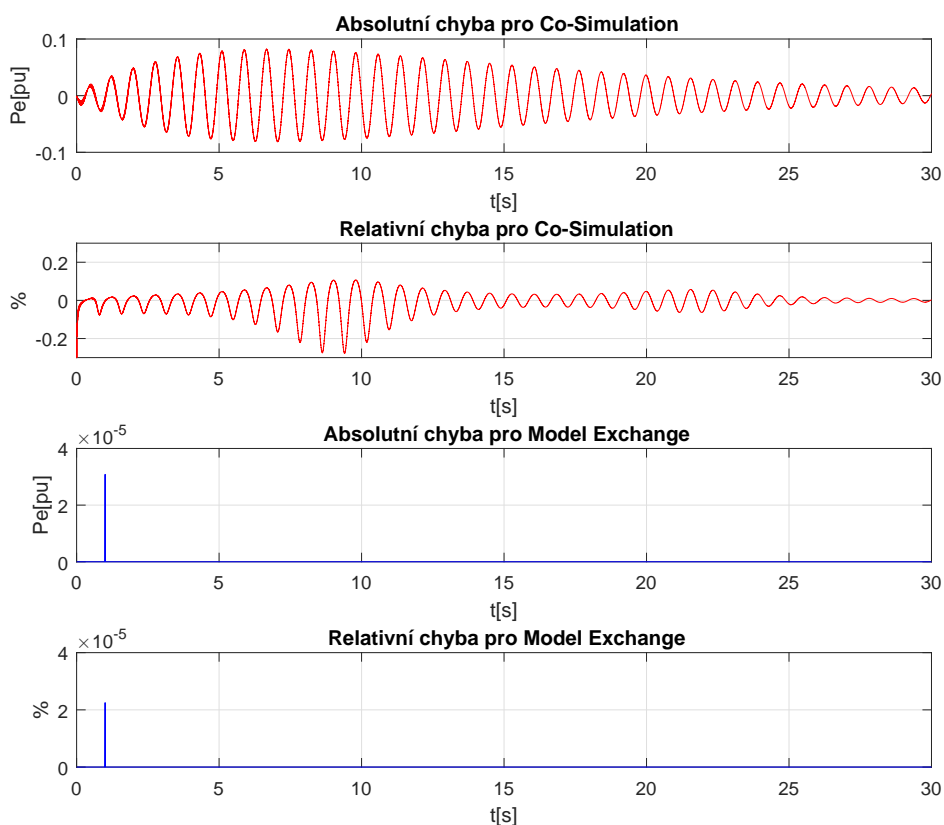
Postup při simulaci je stejný jako v případě vodní turbíny. Nejdříve provedeme simulaci pro krok v Co-Simulation s hodnotou 0.001 a poté ho zjermíme na 0.00001.

Odezva všech modelů na sinusový vstup je na obrázku č.52.



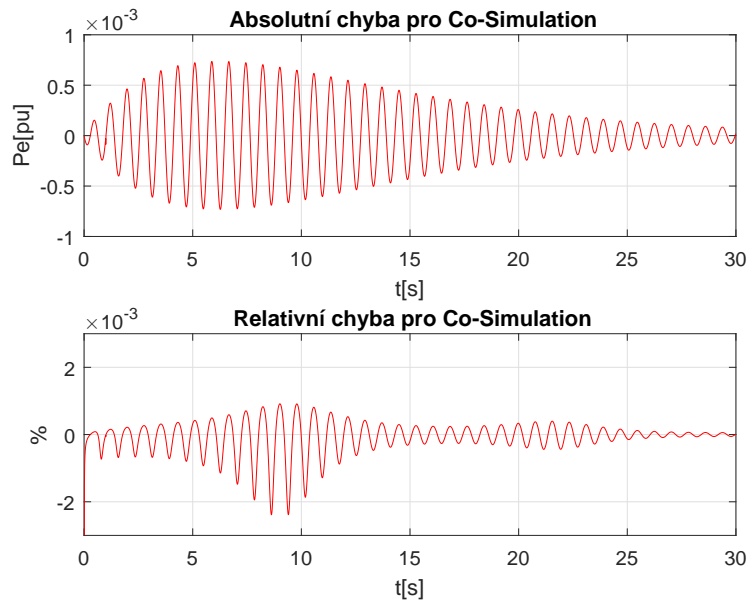
Obrázek 52: Odezva modelů generátoru na sinusový vstup

Průběh absolutní a relativní chyby pro Model Exchange a Co-Simulation s krokem 0.001 je znázorněn na obrázku č.53.



Obrázek 53: Absolutní a relativní chyba generátoru jako FMU

Graf chyb při snížení kroku v Co-Simulation na 0.00001 je na obrázku č.54.



Obrázek 54: Absolutní a relativní chyba generátoru jako FMU-CS s jemnějším krokem

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	0.0822 pu	0.278 %
CS - 0.00001	$7.3572 \cdot 10^{-4}$ pu	$2.3872 \cdot 10^{-3}$ %
ME	$3.0855 \cdot 10^{-5}$ pu	$2.2313 \cdot 10^{-5}$ %

Tabulka 9: Maximální hodnoty chyb pro model generátoru

Maximální hodnoty chyb jsou uvedeny v tabulce č.9. Modely generátoru nedávají již tak stejné výsledky jako modely vodní turbíny. To je pravděpodobně způsobeno návrhem modelu generátoru, který je značně zjednodušen a je modelován jako, že je připojen k nekonečně silné síti. Z výsledků je patrné, že Co-Simulation s krokem 0.001 dává oproti předešlým případům již značně nepřesné výsledky. S jemnějším krokem, ale dosáhneme lepších výsledků a přiblížili jsme se hodnotám pro Model Exchange.

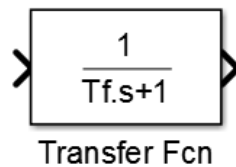
V případě použití složitějšího generátoru by mohlo být dosaženo lepších výsledků při porovnání a později i při simulování jednoduché vodní elektrárny. Avšak tento jednoduchý model je pro naše účely dostačující.

5 Model servomotorů

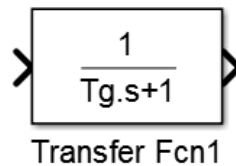
Používání servomotorů u vodní turbíny je dáno již z historie. První vodní turbíny se začaly objevovat počátkem 19. století. V té době řídily systém vodní turbíny pomocí mechanicko-hydraulického regulátoru. Je potřeba velká síla, aby ovládací brána (gate) mohla překonat vysoký tlak vody a vysoké třecí síly [16], [23], [27]. Právě z tohoto důvodu jsou použity v řídicím systému dva servomotory: pilot servomotor a gate servomotor. Jejich názvy by se daly přeložit jako řídicí servomotor a servomotor brány.

Servomotory se používají pro ovládání brány (gate), kterou přitéká voda do potrubí turbíny. Tzn., čím bude hodnota výstupu servomotorů větší, tím bude do systému proudit více vody.

Oba servomotory budeme reprezentovat pomocí přenosových funkcí. Parametry T_f a T_g jsou časové konstanty. Tvar funkcí je následující:



Obrázek 55: Tvar přenosové funkce řídicího servomotoru



Obrázek 56: Tvar přenosové funkce servomotoru brány

Při přechodu na elektricko-hydraulický řídicí systém se tvary servomotorů nemění. Protože rychlost turbíny je měřena elektronicky, má elektricko-hydraulický řídicí systém vyšší přesnost a téměř žádné mrtvé pásmo (pásmo necitlivosti) na rozdíl od mechanicko-hydraulického řídicího systému.

Začlenění servomotorů do modelu regulátoru bude ukázáno v následující kapitole.

6 Model elektricko-hydraulického regulátoru

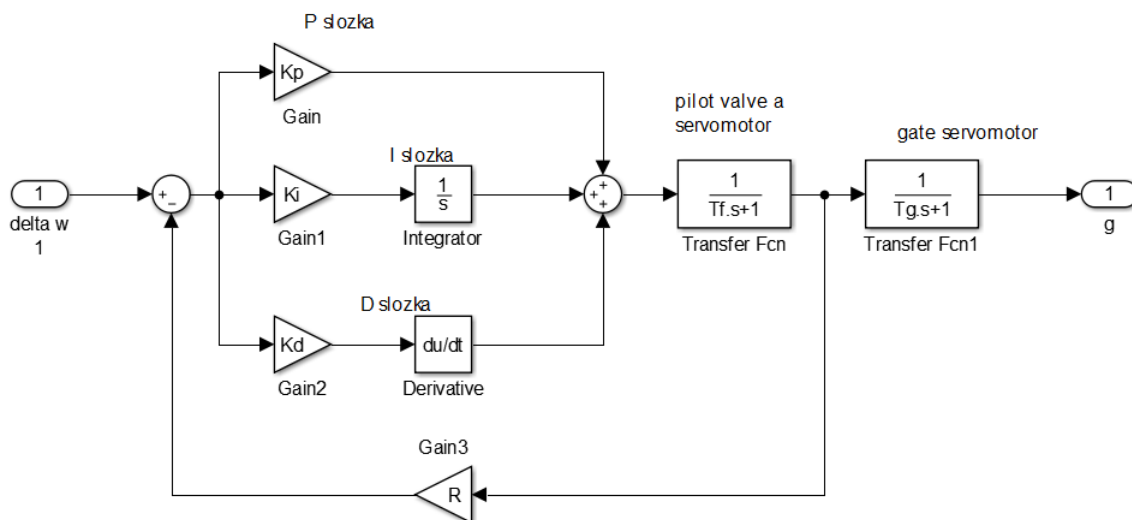
V našem modelu chceme regulovat otevírání nebo zavírání brány podle naší zadané rychlosti tj. otáček [16]. Proto sestavíme PI regulátor, který na základě rozdílu zadané a aktuální rychlosti bude regulovat otevírání brány [17], [19], [26-27], [29]. Při navrhování regulátoru pro bránu se často vychází z PID regulátoru a PI získáme tak, že za K_d v derivační větvi dáme 0 [23-25].

6.1 Paralelní realizace

Předpis pro paralelní verzi regulátoru je následující:

$$C_{paralelni}(s) = K_p + \frac{K_i}{s} + K_d s$$

Kde K_p je proporční člen, K_i je integrační člen a K_d je derivační složka. Paralelní realizace regulátoru včetně servomotorů je zobrazena na obrázku č.57.



Obrázek 57: Paralelní verze elektricko-hydraulického regulátoru

Za parametr K_d dosadíme 0 a zbylé parametry určíme podle vztahu:

$$K_p = \frac{1}{r}$$

$$K_i = \frac{1}{r \cdot T_r}$$

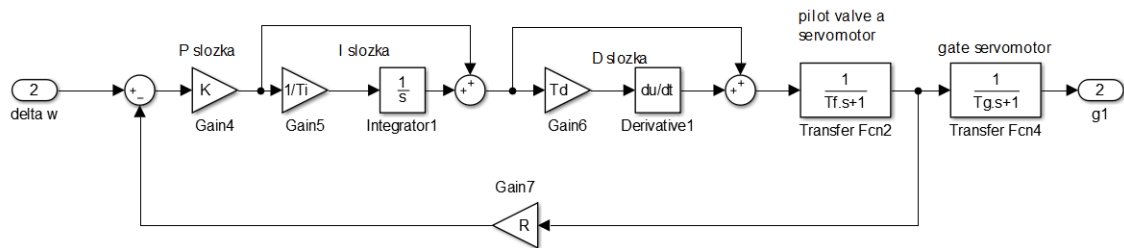
Kde r je přechodný pokles, T_r je časová konstanta, R je trvalý pokles. Hodnoty všech parametrů jsou uvedeny v 7. kapitole - Hodnoty parametrů.

6.2 Sériová realizace

Předpis pro sériovou verzi regulátoru je následující:

$$C_{seriovy}(s) = K \cdot \left(1 + \frac{1}{T_i s}\right) \cdot (1 + sT_d)$$

Kde K je zesílení, T_i je integrační časová konstanta a T_d je derivační časová konstanta. Sériová realizace regulátoru včetně servomotorů je potom zobrazena na obrázku č.58.



Obrázek 58: Sériová verze elektricko-hydraulického regulátoru

Přepočítání parametrů mezi verzemi regulátorů se provede podle následujících rovnic [2]:

$$K_p = K \cdot \left(1 + \frac{T_d}{T_i}\right)$$

$$K_i = \frac{K}{T_i}$$

$$K_d = KT_d$$

Po dosazení získáme, že $K = 2.6316$, $T_i = 9.99848$ a $T_d = 0$, protože chceme realizovat PI regulátor. Všechny hodnoty parametrů jsou znovu uvedeny v 7. kapitole - Hodnoty parametrů.

Je jedno, jaký typ použijeme. Obě realizace regulátoru dávají stejné výsledky.

7 Hodnoty parametrů

Zde jsou uvedeny veškeré hodnoty parametrů použité v modelech [22-23], [25-26].

7.1 Parametry pro turbínu

Parametr	Popis	Hodnota
H_0	startovní výška hladiny	1
T_w	časová konstanta přenosové funkce	1
A_t	konstanta pro MVA	1
gnl	ztráta proudění v potrubí	0.01
K_{d2}	proporční koeficient	0.01

Tabulka 10: Hodnota parametrů pro model turbíny

7.2 Parametry pro generátor

Parametr	Popis	Hodnota
H	konstanta	3.1
D	konstanta	2
K_e	elektrické zesílení	1.27
ω_0	zesílení generátoru	100*pi

Tabulka 11: Hodnota parametrů pro model generátoru

7.3 Parametry pro servomotory

Parametr	Popis	Hodnota
T_f	časová konstanta pro pilot s.	0.1
T_g	časová konstanta pro gate s.	0.5

Tabulka 12: Hodnota parametrů pro model servomotorů

7.4 Parametry pro regulátor

Parametr	Popis	Hodnota
r	přechodový pokles	0.38
R	trvalý pokles	2
T_r	časová konstanta	10
K_p	proporční člen	2.6316
K_i	zesílení integrační složky	0.2632
K_d	zesílení derivační složky	0
T_d	časová derivační konstanta	0
T_i	časová integrační konstanta	9.99848
K	zesílení	2.6316

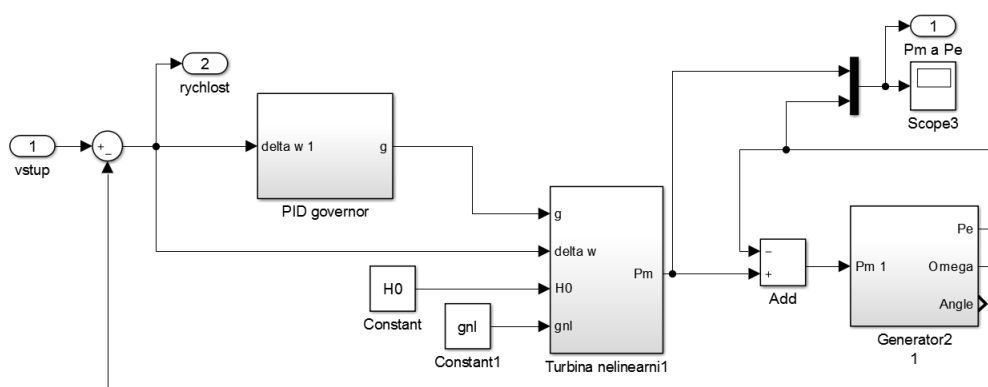
Tabulka 13: Hodnota parametrů pro model regulátor

8 Model jednoduché vodní elektrárny

V této kapitole spojíme všechny předešlé modely, které jsme si ukázali. Náš vytvořený systém se tedy bude skládat z vodní turbíny, generátoru a regulátoru se servomotory [18], [20-21], [23]. Sestavíme takto celkem tři modely. První bude sestaven kompletně v Simulinku. Druhý bude mít turbínu a generátor z Open Modelicy ve formě FMU typu Model Exchange a třetí bude také z Open Modelicy pouze s tím rozdílem, že FMU bude ve formě Co-Simulation [30].

8.1 V Simulinku

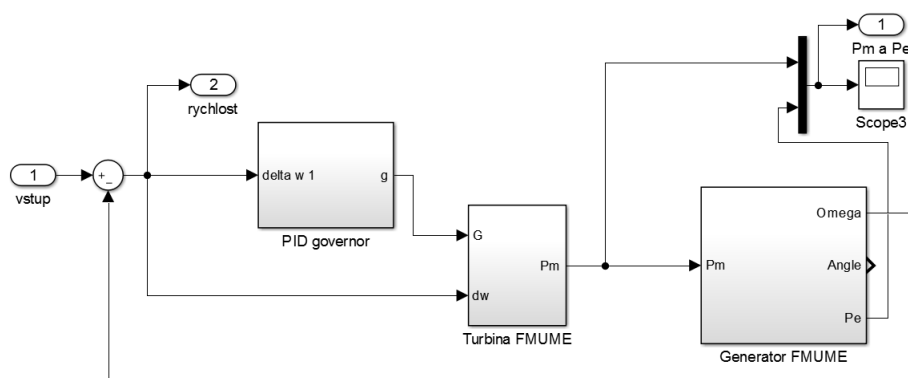
Na obrázku č.59 je schéma modelu jednoduché vodní elektrárny poskládané pomocí modelů uvedených v předcházejících kapitolách a vytvořených v Simulinku [25].



Obrázek 59: Schéma modelu jednoduché vodní elektrárny v Simulinku

8.2 Pomocí FMU - Model Exchange

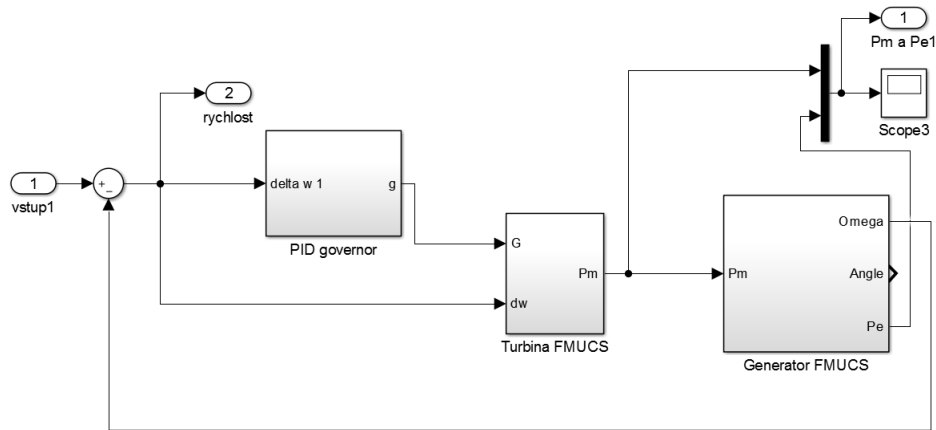
Na obrázku č.60 je schéma modelu jednoduché vodní elektrárny, která má turbínu a generátor vytvořené v Open Modelice a nainportované jako FMU - Model Exchange. Regulátor je vytvořený v Simulinku.



Obrázek 60: Schéma modelu jednoduché vodní elektrárny pomocí FMU - Model Exchange

8.3 Pomocí FMU - Co-Simulation

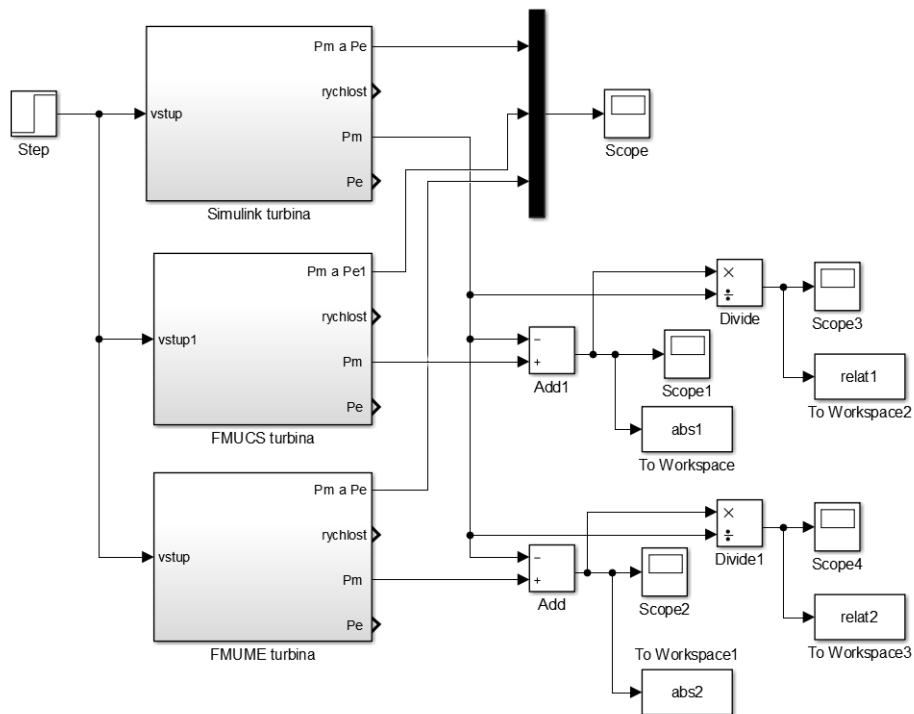
Na obrázku č.61 je schéma modelu jednoduché vodní elektrárny, která má turbínu a generátor vytvořené v Open Modelice a nainportované jako FMU - Co-Simulation. Regulátor je vytvořený v Simulinku.



Obrázek 61: Schéma modelu jednoduché vodní elektrárny pomocí FMU - Co-Simulation

8.4 Porovnání modelů

Schéma porovnání všech modelů je na obrázku č. 62.

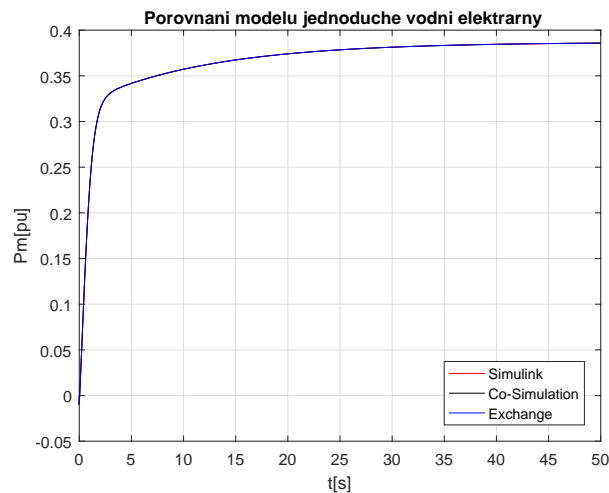


Obrázek 62: Schéma porovnání všech modelů pro vodní elektrárnu

Nejprve provedeme porovnání a výpočet absolutní a relativní chyby pro mechanický výkon (P_m). V případě porovnání elektrického výkonu (P_e) je model totožný, pouze se liší v tom, že místo P_m je tam přivedeno P_e . Jako vstup na naše modely použijeme step se začátkem v nulovém čase, počáteční hodnota bude 0.2 a konečná hodnota bude 0.8.

8.4.1 Porovnání mechanického výkonu

Na obrázku č.63 je znázorněn průběh mechanického výkonu. Vidíme, že systémy najedou zhruba na 90% výkonu, pro danou velikost vstupu, za sedm a půl vteřiny. V 47.5 vteřině dosáhnou systémy rovnovážného bodu a pokud se nezmění vstup, tak v něm setrvají i nadále.

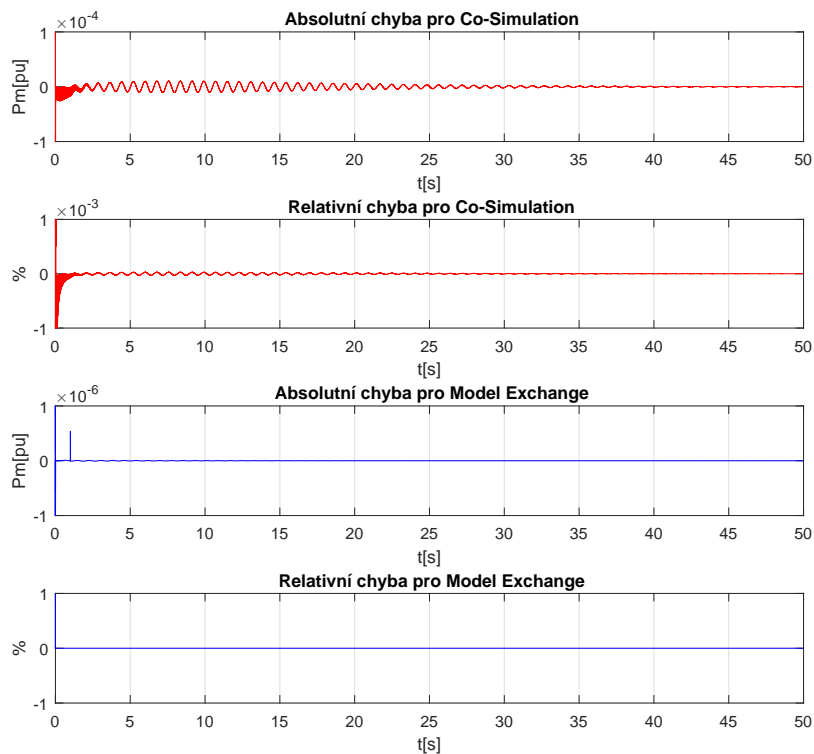


Obrázek 63: Mechanický výkon vodní elektrárny

Jak je patrné již z grafů chyb a tabulky pro Co-Simulation, tak se maximální hodnoty chyb drží velice blízko nule a s rostoucím časem se snižují. Počáteční velké hodnoty chyb neuvažují, protože tyto hodnoty jsou výrazně větší, ale pouze v jednom časovém kroku simulace a poté ihned jejich hodnota klesá. Z toho plyne, že čím více se modely blíží rovnovážnému bodu, tím klesá chyba. Maximální hodnoty chyb jsou dosaženy v prvních patnácti vteřinách simulace.

Model Exchange dává téměř po celou dobu simulace nulovou chybu a to jak absolutní, tak relativní. Pouze u absolutní chyby v první vteřině je výraznější výkyv, ale v dalším časovém kroku se její hodnota opět blíží nule.

Průběh absolutní a relativní chyby je znázorněn na obrázku č.64. Nejdříve jsme provedli takové porovnání, že parametry kroku pro Cosimulation byly 0.0001 pro turbínu a 0.001 pro generátor.



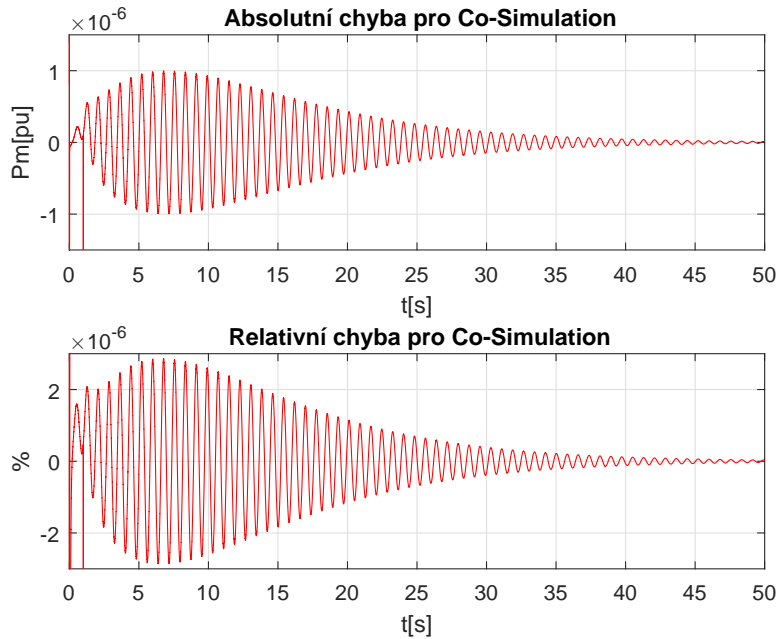
Obrázek 64: Absolutní a relativní chyba pro mechanický výkon

Absolutní a relativní chyba pro nastavený krok v Cosimulation na 0.00001 pro turbínu a 0.0001 pro generátor je na obrázku č.65.

Maximální hodnoty chyb jsou uvedeny v tabulce č.14.

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.0001 a 0.001	$90.2 \cdot 10^{-6}$ pu	$3.025 \cdot 10^{-5}$ %
CS - 0.00001 a 0.0001	$9.965 \cdot 10^{-7}$ pu	$2.864 \cdot 10^{-6}$ %
ME	$2.72 \cdot 10^{-6}$ pu	$1.2 \cdot 10^{-5}$ %

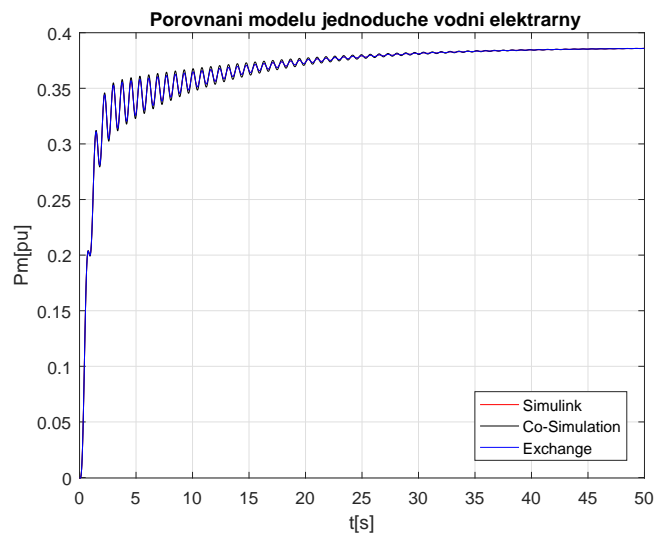
Tabulka 14: Maximální hodnoty chyb pro mechanický výkon



Obrázek 65: Absolutní a relativní chyba pro mechanický výkon s jemnějším krokem

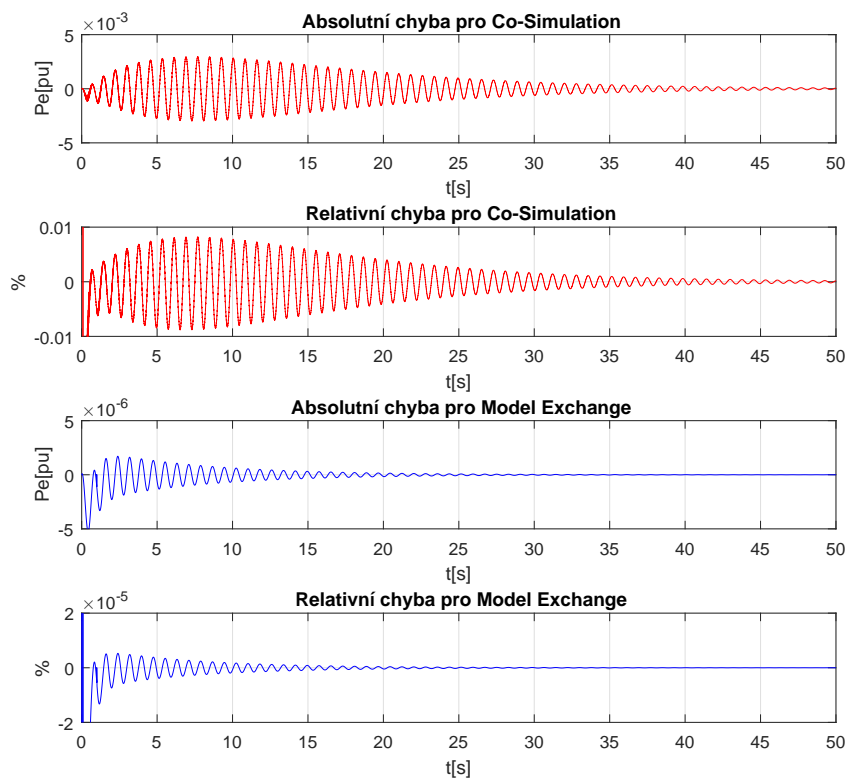
8.4.2 Porovnání elektrického výkonu

Na obrázku č.66 je znázorněn průběh elektrického výkonu. Z grafu je patrné, že generovaný elektrický výkon z generátoru má kmitavý průběh, ale s rostoucím časem simulace se jeho kmity postupně ustálí. V 47.5 vteřině dosáhnou systémy rovnovážného bodu.



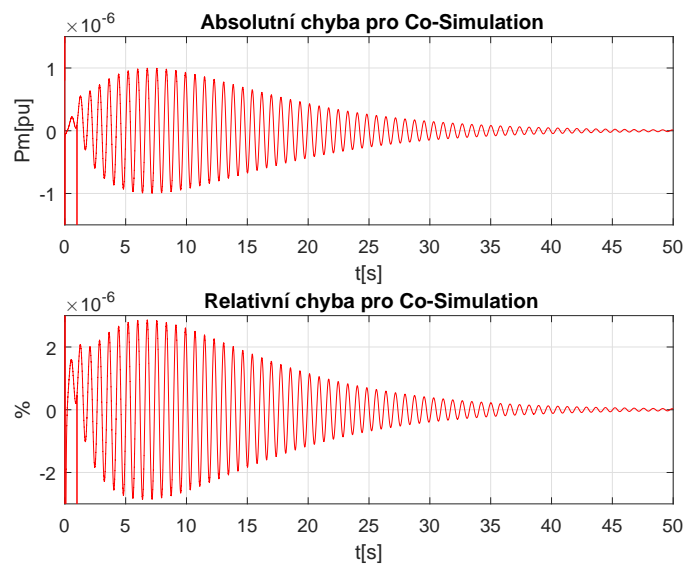
Obrázek 66: Elektrický výkon vodní elektrárny

Absolutní a relativní chyba pro elektrický výkon je znázorněna na obrázku č.67. Kroky pro Co-Simulation jsou nastavené stejně jako v případě porovnání mechanického výkonu.



Obrázek 67: Absolutní a relativní chyba pro elektrický výkon

Na obrázku č.68 je průběh absolutní a relativní chyby pro Co-Simulation když zjenníme krok u turbíny na 0.00001 a u generátoru na 0.0001.



Obrázek 68: Absolutní a relativní chyba pro elektrický výkon s jemnějším krokem

Maximální dosažené hodnoty chyb jsou uvedeny v tabulce č.15.

Typ modelu	Absolutní chyba	Relativní chyba
CS - 0.001	$3.032 \cdot 10^{-3}$ pu	$8.2 \cdot 10^{-3}$ %
CS - 0.00001	$2.8312 \cdot 10^{-4}$ pu	$7.856 \cdot 10^{-4}$ %
ME	$2.523 \cdot 10^{-5}$ pu	$1.34 \cdot 10^{-5}$ %

Tabulka 15: Maximální hodnoty chyb pro elektrický výkon

9 Sestavení modelu v REXu

V této kapitole se budeme věnovat programu REX. V první podkapitole porovnáme model jednoduché vodní elektrárny v Simulinku s modelem vytvořeným pomocí knihovny RexLib. Což znamená, že nahradíme bloky a regulátor vytvořený v Simulinku za bloky z REXu [31], [33-34]. V druhé kapitole se již přesuneme přímo do prostředí programu REX, kde vytvoříme model řízeného systému a regulátor, který následně pomocí exekutivy propojíme a budeme simulovat v reálném čase [2]. Protože pracujeme s 64 bitovými FMU musí být i REX 64 bitový, jinak je nebudeme moci správně spustit [33-34].

9.1 Pomocí knihovny RexLib v Simulinku

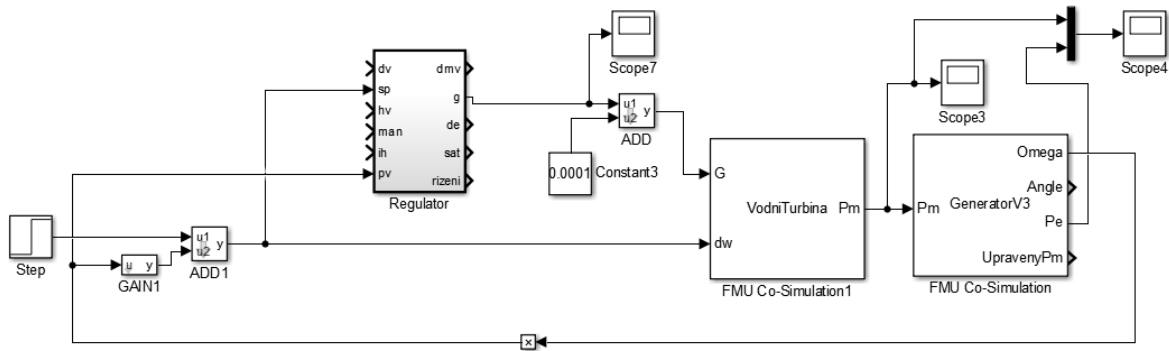
RexLib je knihovna funkčních bloků, která obsahuje více než 130 bloků použitelných v Simulinku i v REXu [2]. Tyto bloky jsou určeny pro průmyslové řízení technologických procesů, strojů, manipulátorů a robotů. Podle oblasti použití jsou bloky rozděleny do několika skupin. Jedná se například o GenLib, RegLib, LogicLib a další... Všechny bloky vyžadují diskrétní solver. Pouze bloky jako add, gain a lpbrk jsou schopné fungovat i ve spojitém solveru. Proto musíme provést úpravu, která je popsána v následujícím odstavci. Bez této úpravy není model vůbec možné spustit [2].

Je velmi důležité, aby veškeré složitější Rexovské bloky, které pracují v diskrétním čase, byly zabaleny do subsystému. V parametrech tohoto subsystému nastavíme, že se jedná o atomickou jednotku a požadovaný krok. My jsme použili nastavení kroku jako 0.00001.

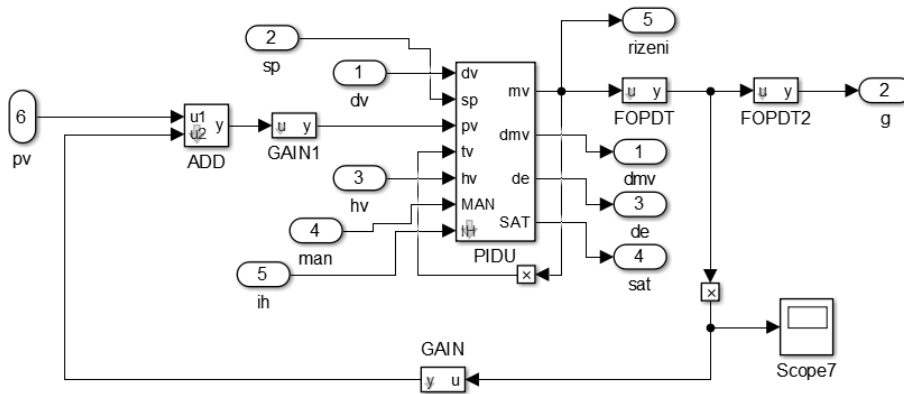
K tomuto nastavení se dostaneme tak, že pomocí pravého tlačítka klikneme na subsystém. Rozbalí se nám nabídka a v ní vybereme "Block Parameters (Subsystem)".

9.1.1 Schéma modelu

Schéma modelu je na obrázku č.69. Subsystém "Regulator" obsahuje elektricko-hydraulický regulátor, který je kompletně vytvořen pomocí bloků RexLibu. Na obrázku č.70 je znázorněna struktura regulátoru. V bloku PIDU vybereme možnost PI regulátoru. Parametry použijeme stejné jako jsou v případě sériové realizace regulátoru v Simulinku.



Obrázek 69: Schéma modelu vodní elektrárny pomocí RexLibu



Obrázek 70: Schéma regulátoru vytvořeného pomocí RexLibu

Bloky *FOPDT* a *FOPDT2* jsou přenosové funkce, které reprezentují oba servomotory. Nastavujeme do nich tedy hodnoty parametrů T_f a T_g jako v případě servomotorů vytvořených pomocí Simulinkovských bloků.

Pro bezrázovou regulaci určíme ještě výsledovací časovou konstantu tt . Níže si uvedeme výpočet této časové konstanty pro PID a PI regulátor [2]:

$$tt_{PID} = \sqrt{T_i \cdot T_d}$$

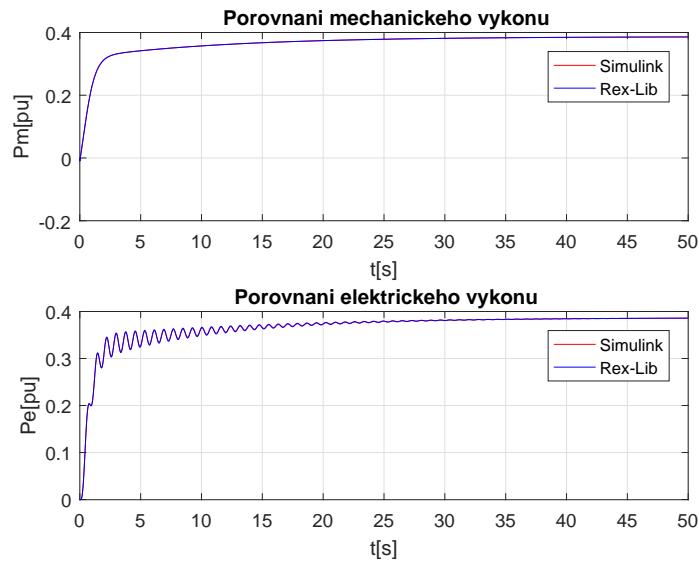
$$tt_{PI} = 2 \cdot \sqrt{T_i}$$

Pro naše hodnoty a nastavení získáváme, že $tt_{PI} = 6.324$.

9.1.2 Porovnání modelů

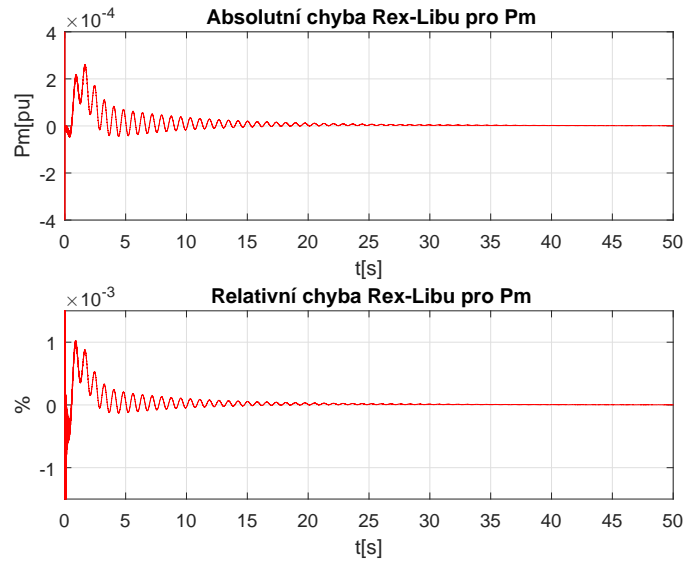
Porovnávat budeme model s FMU Co-Simulation a regulátorem v RexLibu s modelem realizovaným čistě v Simulinku. Jako vstupní signál použijeme opět step se stejnými parametry jako v předchozí kapitole, tj. od 0.2 do 0.8. Schéma porovnání obou modelů vychází ze stejné koncepce jako porovnání jednoduché vodní elektrárny na obrázku č.61. Pouze s tím rozdílem, že místo subsystémů pro Co-Simulation a Model Exchange je pouze subsystém pro RexLib. Model v Simulinku zůstal zachován.

Krok v FMU Co-Simulation nastavíme v případě turbíny na 0.0001 a pro generátor na 0.001. Nejdříve provedeme porovnání a výpočet chyb pro mechanický výkon a poté pro elektrický výkon. Toto porovnání je zobrazeno na obrázku č.71.



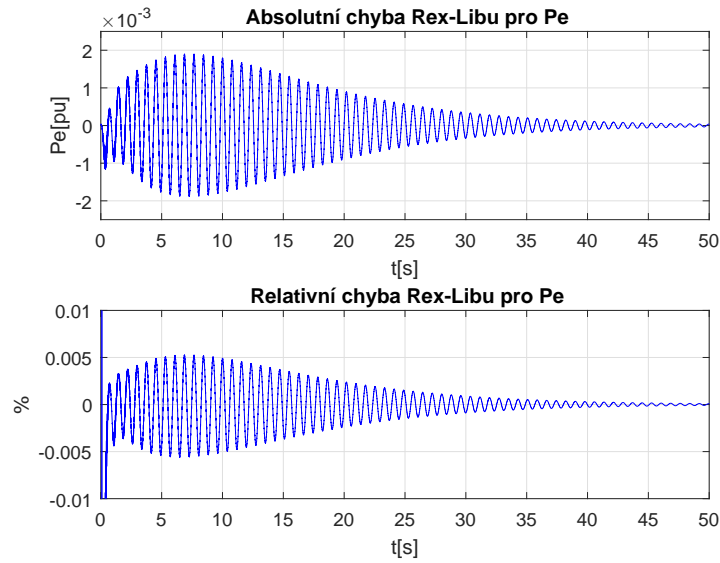
Obrázek 71: Porovnání P_m a P_e ze Simulinku a pomocí RexLibu

Obrázek č.72 znázorňuje průběh absolutní a relativní chyby pro mechanický výkon.



Obrázek 72: Absolutní a relativní chyba mezi Simulinkem a RexLibem

Na obrázku č.73 je znázorněn průběh absolutní a relativní chyby pro elektrický výkon.



Obrázek 73: Absolutní a relativní chyba mezi Simulinkem a RexLibem s jemnějším krokem

Tabulka č.16 obsahuje maximální dosažené hodnoty pro absolutní a relativní chybu pro P_m a P_e .

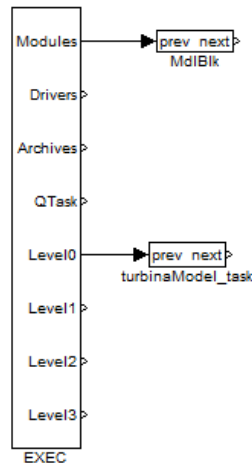
Porovnávaná veličina	Absolutní chyba	Relativní chyba
P_m	$2.614 \cdot 10^{-4}$ pu	$1.0285 \cdot 10^{-3}$ %
P_e	$1.894 \cdot 10^{-3}$ pu	$5.277 \cdot 10^{-3}$ %

Tabulka 16: Maximální hodnoty chyb pro P_m a P_e pro RexLib

9.2 V Rexu

Řídicí systém REX je soubor softwarových nástrojů pro řízení strojů, technologií a procesů. Lze ho použít ve všech oblastech automatizace, robotiky, měření a regulace [2]. Zde se již dostáváme do vývojové části SIL, což znamená, že modely jsou simulovány v reálném čase [33-34]. Všechny předešlé modely neběžely v reálném čase. Jednalo se o tzv. MIL. V případě spojení modelu se skutečným řídicím systémem nebo model řídicího systému se skutečným systémem, tak se jedná o HIL [31], [33-34].

Nejprve tedy sestavíme v Rexu regulátor a poté model složený z FMUs pro turbínu a generátor. Ověříme, že každá tato část odpovídá jejich realizaci v Simulinku. Sestavený model v Rexu spustíme a poté si výsledky z bloku TRND vyexportujeme ve formátu csv. Toto csv naimportujeme do Matlabu a vybereme si pouze hodnoty, které budeme chtít porovnávat. Model v Rexu potřebuje spouštěcí sekvenci tzv. *exec*, ve kterém se definují všechny časové kroky včetně priorit operací. Podoba *execu* je zobrazena na obrázku č.74. Na pozici *Level0* připojíme nejdříve náš vytvořený generátor a poté turbínu s generátorem. Kvůli blokům *FMUCS* musíme naimportovat modul *MdlBlk* [34].



Obrázek 74: Exec v Rexu

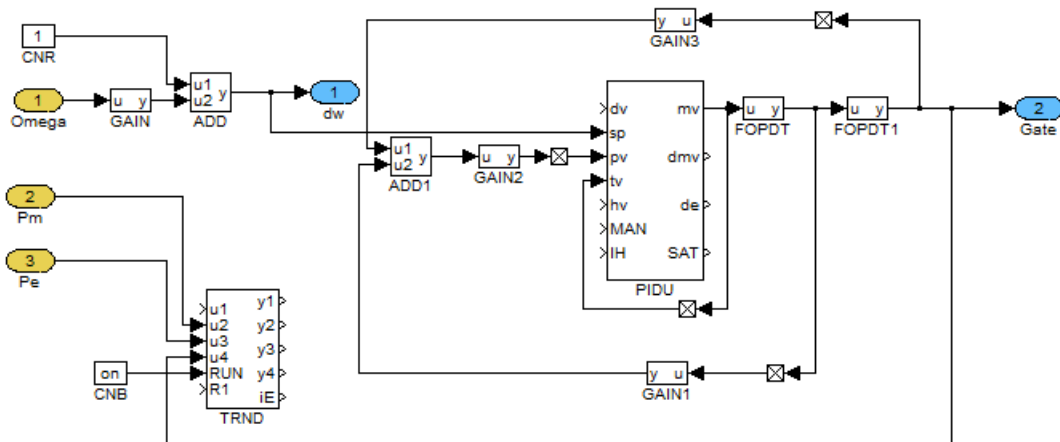
Tabulka č.17 obsahuje hodnoty pro jednotlivé parametry v *exec*. Bez správného nastavení hodnot nepůjde model vůbec spustit. Nebo nebude fungovat správně.

Parametr	Hodnota
tick	0.005
ntick0	1
ntick1	50
ntick2	100
ntick3	1200
pri0	5
pri1	9
pri2	13
pri3	18

Tabulka 17: Nastavení hodnot v *exec*

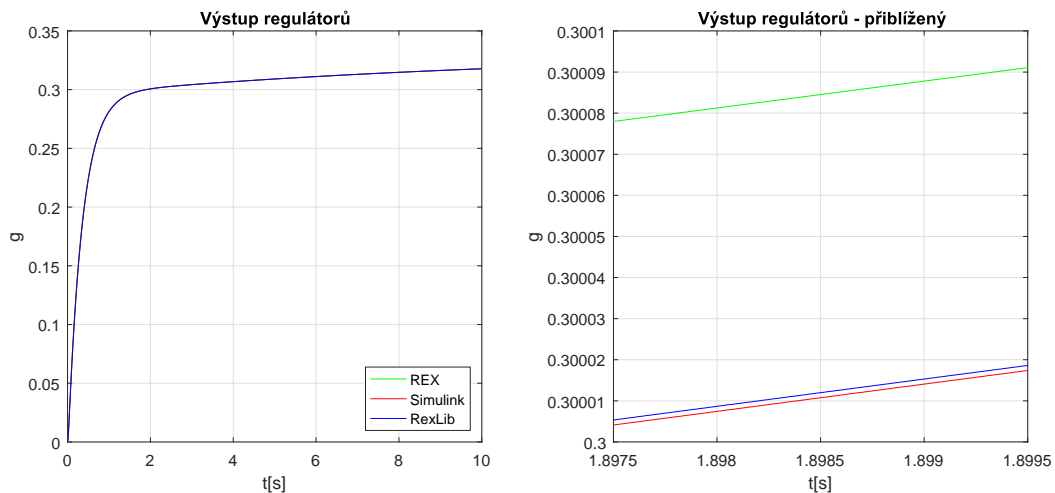
9.2.1 Regulátor

Schéma sestaveného regulátoru je znázorněno na obrázku č.75. Jako vstup použijeme konstantu rovnou jedné (to odpovídá stepu v čase 0 na hodnotu 1 v Simulinku).



Obrázek 75: Schéma regulátoru v Rexu

Takto sestavený regulátor jsme porovnali s modely sestavenými v Simulinku a pomocí RexLibu. Hodnoty jsme použili stejné jako v předchozím případě.

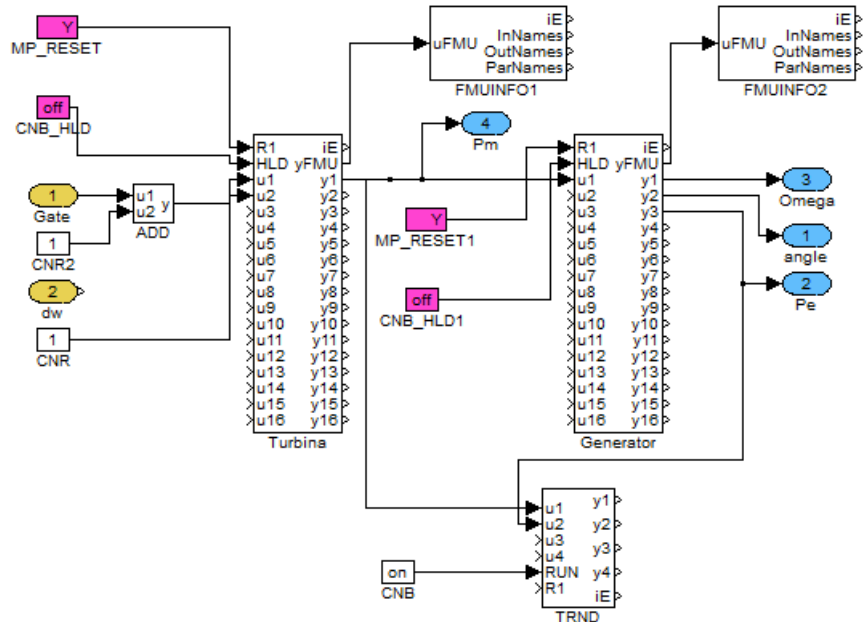


Obrázek 76: Porovnání regulátor z REXu se Simulinkem a RexLibem

Z obrázku č.76 je patrné, že si regulátory téměř odpovídají, rozdíl v případě Rexu je daný pouze krokem a periodou nastavenými v execu. S vyšší periodou bychom dosáhli větší shody.

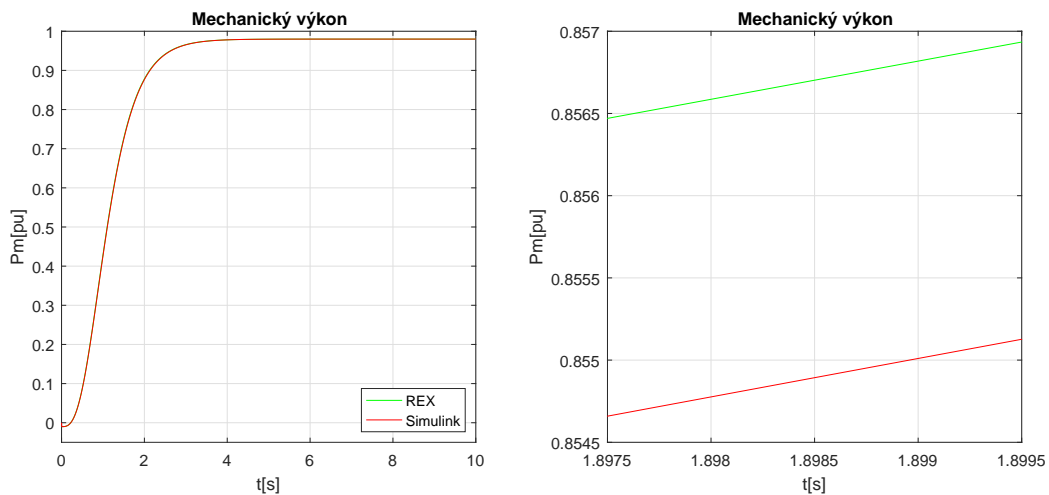
9.2.2 Turbína s generátorem pomocí FMUs

Schéma sestaveného modelu, tj. turbíny a generátoru, pomocí FMUs v Rexu je znázorněno na obrázku č.77. Jako vstup použijeme konstanty rovny jedné (to odpovídá stepu v čase 0 na hodnotu 1 v Simulinku).



Obrázek 77: Model turbíny a generátoru v Rexu pomocí bloků FMUCS

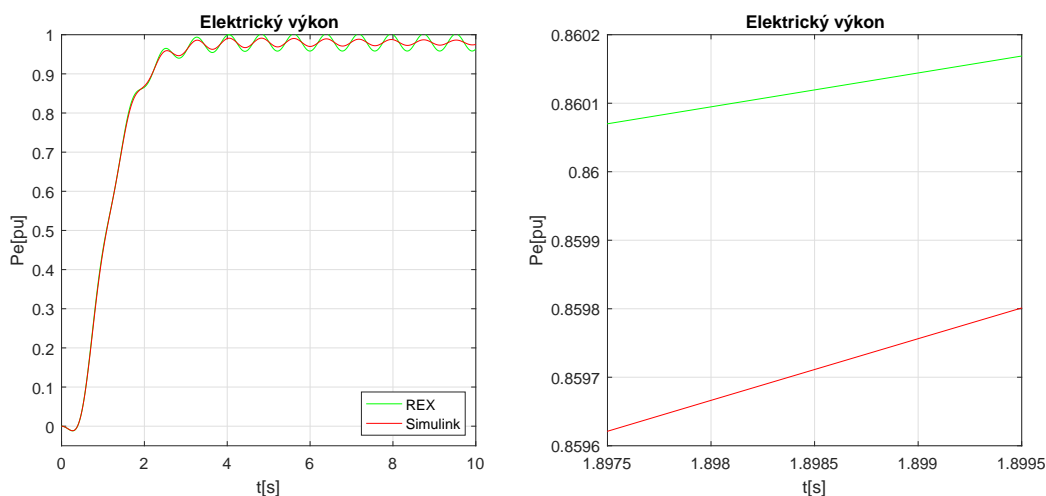
Takto sestavený model budeme opět porovnávat s modelem sestaveným v Simulinku pomocí bloků FMUs jako Co-Simulation. Porovnávat budeme mechanický a elektrický výkon.



Obrázek 78: Porovnání mechanického výkonu pro Rex

Z obrázku č.78 je patrné, že v případě mechanického výkonu dávají modely téměř totožné výsledky. Rozdíl REXu je opět způsoben nastaveným krokem a periodou v execu.

Na obrázku č.79 je vidět, že zatímco model v Simulinku se utlumí, tak model v Rexu nikoliv. To je způsobeno nastavením execu, pokud zvolíme vyšší periodu (tj. ntick0) dojde ke zpřesnění a utlumení modelu stejně jako v případě Simulinku.



Obrázek 79: Porovnání elektrického výkonu pro Rex

10 Závěr

Cílem této práce bylo demonstrování možností rozvíjejícího se standardu FMI 2.0. V první části této práce jsme se věnovali modelu kyvadla. Odvodili jsme diferenciální rovnici pro tento model a poté jsme si ukázali i analytické řešení této rovnice. To pracuje s kyvadlem jako s hmotným bodem. Abychom ověřili jeho správnost porovnali jsme ho s modelem v Simulinku a v Simulinku pomocí knihovny SimScape. Následně jsme sestavili model kyvadla, které jsme modelovali jako skutečnou kouli, v Simulinku, v SimScapu a v Open Modelice. Z Open Modelicy jsme poté vyexportovali FMU a to jak verzi Model Exchange, tak Co-Simulation. Všechny tyto modely jsme mezi sebou porovnali a věnovali jsme se určení absolutní a relativní chyby pro FMU vůči Simulinku. Jako vstupní funkci jsme použili námi sestavenou spojitou funkci a poté dva generátory náhodného signálu. Jednalo se o Mackey-Glassův a Rösslerův generátor. Zjistili jsme, že čím jemnější nastavíme krok v FMU Co-Simulation, tím dostaneme přesnější výsledky. To je ovšem vykoupeno výraznou výpočetní náročností simulace. V některých případech jsou výsledné chyby zanedbatelné. To samé platí i pro modely energetického zařízení.

Ve druhé části jsme se věnovali sestavení modelu energetického zařízení. Jednalo se o model vodní turbíny a generátoru. Při vytváření modelu vodní turbíny jsme narazili na to, že takovýto stroj se dá modelovat mnoha způsoby. Jedním z nich je například model, který je založen na změně tlaku v turbíně. My jsme však zvolili jednodušší model, který je postaven na rovnicích popisujících proudění vody a mechanický výkon. Model generátoru jsme zvolili také jednoduchý, protože jsme potřebovali pouze převést mechanickou energii turbíny na rychlost rotoru tj. otáčky. Veškeré hodnoty parametrů jsou však čistě teoretické, protože jsme neměli k dispozici žádná reálná data. Oba modely jsme sestavili nejprve v Simulinku a poté jsme je pomocí rovnic sestavili v Open Modelice. Z Open

Modelicy jsme opět vyexportovali FMUs jak verzi Model Exchange tak Co-Simulation. Všechny tyto modely jsme porovnali a opět určili absolutní a relativní chyby získaných FMUs.

V další části jsme se rozhodli sestavit jednoduchý model vodní elektrárny. Modely vodní turbíny a generátoru jsme měli již sestavené a zbývalo nám sestavit regulátor. Ten jsme zvolili jako PI a spřáhli jsme ho se dvěma servomotory. Ty se používají k ovládání brány, kterou přitéká voda do potrubí vodní turbíny. U PI regulátoru jsme použili paralelní realizaci, ale ukázali jsme si také, jak od paralelní realizace přejít na sériovou. Takto sestavený model jsme vytvořili v Simulinku a to jak pomocí reprezentace modelů v Simulinku, tak pomocí FMUs (pro turbinu a generator) jako Model Exchange a Co-Simulation. Tyto modely jsme mezi sebou porovnali a také určili absolutní a relativní chybu celého systému.

V závěru práce jsme se přenesli do programu REX, což je program pro návrh řídicích systémů reálného času. Zde jsme nejprve pomocí knihovny RexLib sestavili model v Simulinku a ověřili jeho správnou funkčnost, tzn. že jsme ho porovnali s modelem sestaveným v Simulinku za pomoci bloků FMUs pro Co-Simulation a určili chyby generované modelem pomocí RexLibu. Poté jsme sestavili model regulátoru a model turbíny a generátoru přímo v Rexu. Jejich odsimulováním jsme získali data ve formátu csv, která jsme naimportovali do Matlabu a porovnali s modely v Simulinku. V případě regulátoru jsme porovnávali model v Rexu se Simulinkem a RexLibem. Vodní turbínu s generátorem jsme porovnávali pouze s modelem v Simulinku pomocí bloků FMU Co-Simulation. Z toho jsme zjistili, že velmi záleží na nastaveném kroku a periodě v Rexu.

Možné zlepšení v modelu energetického zařízení je určitě ve složitosti modelu generátoru. My jsme zvolili pro naše potřeby co nejjednodušší model synchronního generátoru. S tím také souvisí validace modelu na reálných datech. Porovnání našich modelů na dostupných teoretických hodnotách dávalo však přijatelné výsledky. Maximální hodnoty chyb se pohybovaly řádově kolem 10^{-3} a s jemnějším krokem v Co-Simulation ještě klesaly.

Seznam Všech použitých zkratek

- **FMI** - The Functional Mock-up Interface
- **FMU** - The Functional Mock-up Unit
- **FMI-ME** - The Functional Mock-up Interface - Model Exchange
- **ODE** - Ordinary Differential Equations
- **DAE** - Differential Algebraic Equations
- **FMI-CS** - The Functional Mock-up Interface - Co-Simulation
- **XML** - eXtensible Markup Language
- **DLL** - Dynamic Link Library
- **GUI** - Graphical User Interface
- **CPU** - Central Processing Unit
- **pu** - per unit (za jednotku)
- **MVA** - Mega Volt Amper (jednotka používaná u transformátorů)
- **PI** - Proporcionálně, Integrační regulátor
- **PID** - Proporcionálně, Integračně, Derivační regulátor
- **REX** - Rapid development, EXcellent performance
- **RexLib** - knihovna obsahující bloky REXu pro Simulink
- **SIL** - Software In the Loop
- **MIL** - Model In the Loop
- **HIL** - Hardware In the Loop

Seznam obrázků

1	Model spojených nádob v Open Modelice	2
2	Model spojených nádob importovaný do Matlabu jako FMU	3
3	Systém a jeho okolí	4
4	Princip postupu při modelování, simulacích a vyhodnocení výsledků	5
5	FMI pro Model Exchange	7
6	FMI pro Co-Simulation	7
7	Schéma kyvadla	8
8	Porovnání analytického řešení kyvadla	13
9	Model kyvadla v Simulinku	13
10	Model Kyvadla pomocí SimScape v Simulinku	14
11	Nastavení bloku Body	14
12	Model kyvadla v OpenModelice	15
13	Nastavení bloku rev v OpenModelice	15
14	Nastavení bloku body v Open Modelice	16
15	Schéma práce s .fmu	17
16	Sekce FMI v nastavení	18
17	Vyexportování modelu jako FMU	18
18	Bloky z FMU import v Simulinku	19
19	Nastavení bloků pro FMU	19
20	Sekce FMU import v Matlabu	20
21	Implementace bloku pro FMU Co-Simulation	20
22	Porovnání všech modelů kyvadla	21
23	Odezva všech modelů na nulový vstup	22
24	Odezva modelů na naší spojitou funkci	22
25	Absolutní a relativní chyba mezi Simulinkem a FMU	23
26	Chyby mezi Simulinkem a Co-Simulation s krokem 0.00001	24
27	Výstup z Mackey Glassova generátoru	26
28	Odezva modelů na signál z Mackey Glassova generátoru	26
29	Absolutní a relativní chyba modelů za použití Mackey Glassova generátoru	27
30	Absolutní a relativní chyba pro jemnější krok v Co-Simulation	27
31	Model Rösslerova generátoru	29
32	Výstupy z Rösslerova generátoru	29
33	Odezva modelů na signál z Rösslerova generátoru	30
34	Absolutní a relativní chyba modelů za použití Rösslerova generátoru	30
35	Absolutní a relativní chyba s jemnějším krokem pro Rösslerův generátor	31
36	Blok pro reálný vstup	32
37	Blok pro reálný výstup	33
38	Schéma vodní turbíny	35
39	Schéma vodní turbíny v Simulinku	37
40	Zdrojový kód vodní turbíny v Open Modelice	38
41	Blok pro vodní turbínu v Open Modelice	39
42	Schéma porovnání modelů vodní turbíny	39
43	Odezva modelů vodní turbíny na sinusový vstup	40
44	Absolutní a relativní chyba FMU pro vodní turbínu	40
45	Absolutní a relativní chyba FMU-CS vodní turbíny s jemnějším krokem	41

46	Elektrické schéma generátoru	42
47	Fázorový diagram generátoru	43
48	Model generátoru v Simulinku	44
49	Zdrojový kód generátoru v Open Modelice	45
50	Blok pro generátor v Open Modelice	45
51	Porovnání modelů generátoru	46
52	Odezva modelů generátoru na sinusový vstup	47
53	Absolutní a relativní chyba generátoru jako FMU	47
54	Absolutní a relativní chyba generátoru jako FMU-CS s jemnějším krokem	48
55	Tvar přenosové funkce řídicího servomotoru	49
56	Tvar přenosové funkce servomotoru brány	49
57	Paralelní verze elektricko-hydraulického regulátoru	50
58	Sériová verze elektricko-hydraulického regulátoru	51
59	Schéma modelu jednoduché vodní elektrárny v Simulinku	53
60	Schéma modelu jednoduché vodní elektrárny pomocí FMU - Model Exchange	53
61	Schéma modelu jednoduché vodní elektrárny pomocí FMU - Co-Simulation	54
62	Schéma porovnání všech modelů pro vodní elektrárnu	54
63	Mechanický výkon vodní elektrárny	55
64	Absolutní a relativní chyba pro mechanický výkon	56
65	Absolutní a relativní chyba pro mechanický výkon s jemnějším krokem	57
66	Elektrický výkon vodní elektrárny	57
67	Absolutní a relativní chyba pro elektrický výkon	58
68	Absolutní a relativní chyba pro elektrický výkon s jemnějším krokem	58
69	Schéma modelu vodní elektrárny pomocí RexLibu	60
70	Schéma regulátoru vytvořeného pomocí RexLibu	60
71	Porovnání P_m a P_e ze Simulinku a pomocí RexLibu	61
72	Absolutní a relativní chyba mezi Simulinkem a RexLibem	61
73	Absolutní a relativní chyba mezi Simulinkem a RexLibem s jemnějším krokem	62
74	Exec v Rexu	63
75	Schéma regulátoru v Rexu	64
76	Porovnání regulátor z REXu se Simulinkem a RexLibem	64
77	Model turbíny a generátoru v Rexu pomocí bloků FMUCS	65
78	Porovnání mechanického výkonu pro Rex	65
79	Porovnání elektrického výkonu pro Rex	66

Seznam tabulek

1	Maximální hodnoty chyb pro model kyvadla	23
2	Hodnoty parametrů použitých v Mackeyglassovu generátoru	25
3	Maximální hodnoty chyb pro Mackey Glassův generátor	28
4	Hodnoty používaných parametrů pro Rösslerův generátor	28
5	Maximální hodnoty chyb pro Rösslerův generátor	31
6	Názvy parametrů použitých v modelu turbíny	36
7	Maximální hodnoty chyb pro model vodní turbíny	41
8	Názvy parametrů použitých v modelu generátoru	42
9	Maximální hodnoty chyb pro model generátoru	48
10	Hodnota parametrů pro model turbíny	52
11	Hodnota parametrů pro model generátoru	52
12	Hodnota parametrů pro model servomotorů	52
13	Hodnota parametrů pro model regulátor	52
14	Maximální hodnoty chyb pro mechanický výkon	56
15	Maximální hodnoty chyb pro elektrický výkon	59
16	Maximální hodnoty chyb pro P_m a P_e pro RexLib	62
17	Nastavení hodnot v exec	63

Seznam literatury

Reference

- [1] FMI 2.0 for Co-Simulation specification (<https://www.fmi-standard.org/downloads>)
- [2] Začínáme se systémem REX na platformě Raspberry Pi (<https://www.rexcontrols.cz/dokumentace-a-podpora>)
- [3] T. Blochwitz, M. Otter. The Functional Mockup Interface 2.0: The standard for Tool independent Exchange of Simulation Models
- [4] C. Noll, T. Blochwitz, et. all. Implementation of Modelisar Functional Mock-up Interfaces in SimulationX
- [5] Bernhard Bachmann, Willi Braun. Functional Mockup Interface for Tool and Model Interoperability (towards FMI 2.0)
- [6] FMI Toolbox User's Guide 2.3. Publication date 2015-03-03 Copyright © 2016 Modelon AB
- [7] Christian Schubert, Thomas Neidhold, Günter Kunze. Experiences with the new FMI Standard Selected Applications at Dresden University. January 31, 2011.
- [8] Peter Fritzson. Functional Mockup Interface (FMI) A general standard for Model Exchange and Simulator Coupling. Linköping University 2011-02-10.
- [9] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauss, H. Elmqvist, A. Jungmanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.V. Peetz, S. Wolf. The Functional Mockup Interface for Tool independent Exchange of Simulation Models
- [10] Michal Dorda. Úvod do modelování a simulace systémů
- [11] Augusto Beléndez, Carolina Pascual Villalobos, David Méndez, Cristian Neipp. Exact solution for the nonlinear pendulum. January 2007
- [12] D.G. Simpson. The Nonlinear Pendulum. 2010
- [13] Scholarpedia Mackey-Glassův generátor. http://www.scholarpedia.org/article/Mackey-Glass_equation
- [14] MathWorks Mackey-Glassův generátor. https://www.mathworks.com/matlabcentral/fileexchange/24390-mackey-glass-time-series-generators_equation
- [15] Scholarpedia, Rösslerův attractor. https://en.wikipedia.org/wiki/R%C3%B6ssler_attractor
- [16] Thierry Van Cutsem. Power system dynamics, control and stability. Turbines and speed governors. November 2015

- [17] Weijia Yang, Jiandong Yang, Wencheng Guo, Wei Zeng, Chao Wang, Linn Saarinen, Per Norrlund. A Mathematical Model and Its Application for Hydro Power Units under Different Operating Conditions. 2015
- [18] Jai Dev Sharma, Avneesh Kumar. Development and Implementation of Non-Linear Hydro Turbine Model with Elastic Effect of Water Column and Surge Tank. December 2014.
- [19] Yin Chin Choo, Kashem M. Muttaqi, M. Negnevitsky. Modeling of hydraulic governor-turbine for control stabilisation.
- [20] Nanaware R. A., Sawant S. R., Jadhav B. T.. Modeling of Hydraulic Turbine and Governor for Dynamic Studies of HPP. 2012
- [21] Dietmar Winkler, Hege Marie Thoresen, Ingvar Andreassen, Magamage Anushka Sampath Perera, Behzad Rahimi Sharefi. Modelling and Optimisation of Deviation in Hydro Power Production. 2011
- [22] Tzuu Bin Ng, G.J. Walker and J.E. Sargison. Modelling of Transient Behaviour in a Francis Turbine Power Plant. 2004
- [23] Peining Wu. Stability of Hydro Power Plants in Island Operation. 2016
- [24] Dr. R. Thapar. Hydro-Turbine Governing System.
- [25] NEPLAN AG. TURBINE-GOVERNOR MODELS Standard Dynamic Turbine-Governor Systems in NEPLAN Power System Analysis Tool.
- [26] R. A. Naghizadeh, S. Jazebi, B. Vahidi. Modeling Hydro Power Plants and Tuning Hydro Governors as an Educational Guideline. 2012
- [27] J. Culberg, M. Negnevitsky and M. A. Kashem. Hydro-Turbine Governor Control: Theory, Techniques and Limitations. 2015
- [28] Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies. HYDRAULIC TURBINE AND TURBINE CONTROL MODELS FOR SYSTEM DYNAMIC STUDIES. February 1992
- [29] E. De Jaeger, N. Janssens, B. Malfiet, F. Van De Meulebroeke. HYDRO TURBINE MODEL FOR SYSTEM DYNAMIC STUDIES. November 1994
- [30] J. Reitinger, M. Schlegel, P. Zídek. Matematické modely parní turbíny
- [31] V. Kučera, P. Zídek, P. Balda, M. Čech, J. Königsmarková, J. Reitinger. Zpráva o testování RT simulátoru pro komponentové modely.
- [32] Jana Königsmarková. Analýza numerických metod pro diferenciální algebraické soustavy
- [33] Martin Čech, Jana Königsmarková, Jan Reitinger, Pavel Balda. Novel tools for model-based control system design based on FMI/FMU standard with application in energetics.
- [34] P. Balda, M. Schlegel, J. Reitinger. RT Simulátor pro komponentové modely. 2017

Přílohy

Na přiloženém CD se nacházejí skripty a modely uvedené v této práci. Jsou rozděleny do několika složek, které si zde popíšeme.

- **FMU** - Obsahuje model kyvadla, vodní turbíny a generátoru ve formě vyexportovaného FMU (CS i ME).
- **Matlab-Simulink** - Modely sestavené v Simulinku, tzn. modely kyvadla, generátorů náhodného signálu, vodní turbíny, generátoru a regulátoru. Nachází se zde také modely určené pro porovnávání. Přiložen je i skript pro generování grafů.
- **OpenModelica** - Realizace modelů kyvadla, vodní turbíny a generátoru v programu OpenModelica.
- **Reference** - Veškeré materiály použité v této práci v elektronické podobě.
- **REX** - Modely pro regulátor a systém dvou FMUs spolu s exekutivou v programu RexDraw.
- **ElektronickaVerze** - Elektronická verze této práce.