

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2017

Marek Bukovan

VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra kybernetiky a biomedicínského inženýrství

**Návrh a realizace PIL a SIL simulátorů pro výukový fyzikální model
inverzního kyvadla**

**Design and Implementation of PIL and SIL Simulators for Educational
Physical Model of Inverted Pendulum**

2017

Ing. Marek Bukovan

Zadání bakalářské práce

Student: **Ing. Marek Bukovan**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: **Návrh a realizace PIL a SIL simulátorů pro výukový fyzikální model
inverzního kyvadla
Design and Implementation of PIL and SIL Simulators for Educational
Physical Model of Inverted Pendulum**

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s řídicím systémem REX se zaměřením na platformu Raspberry Pi.
2. Seznámení se s technikami RT modelování. Popis aplikace těchto technik na regulované soustavě, motivace a ukázky použití, vlastnosti.
3. Návrh a implementace SIL a PIL simulátorů pro zvolenou regulovanou soustavu, vizualizace činnosti simulátorů.
4. Ověření činnosti PIL a SIL simulátorů srovnáním s reálnými daty naměřenými na fyzikálním modelu regulované soustavy.
5. Zhodnocení dosažených výsledků závěrečné práce, závěr.

Seznam doporučené odborné literatury:

- [1] XUE, Dingyü, Yangquan CHEN a Derek P. ATHERTON. *Linear feedback control: analysis and design with MATLAB*. 4th rev. and enlarged ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, c2007. xii, 354 p. ISBN 08-987-1638-1/978-0-898716-38-2.
- [2] NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. 1. vyd. Ostrava: MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.
- [3] VÍTEČKOVÁ, Miluše a Antonín VÍTEČEK. *Základy automatické regulace*. 2. vyd. Ostrava: VŠB-TU Ostrava, 2008. 243 s. ISBN 978-80-248-1924-2.
- [4] VAVŘÍN, Petr. *Teorie automatického řízení I (Lineární spojité a diskrétní systémy)*. 2. přepracované vyd. Brno: VUT Brno, 1991. 158 s. ISBN 80-214-0244-X.
- [5] VAVŘÍN, Petr. *Teorie dynamických systémů*. 1. vyd. Brno: VUT Brno, 1989. 177 s.
- [6] ZÍTEK, P., M. HOFREITER a J. HLAVA. *Automatické řízení*. Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 1999. 148 s. ISBN 80-010-2044-4.
- [7] ŠULC, Bohumil. *Teorie automatického řízení s počítačovou podporou*. Vyd. 1. Praha: ČVUT, Strojní fakulta, 1999. 154 s. ISBN 80-010-1974-8.
- [8] OŽANA, Štěpán. *Navrhování a realizace regulátorů*. vyd. 1. Ostrava: VŠB - TU Ostrava, 2012. Studijní materiály. 136 s. ISBN 978-80-248-2605-9.
- [9] ZEZULKA, František, Petr FIEDLER a Zdeněk BRADÁČ. *Prostředky průmyslové automatizace*. Učební texty. Brno: VUT v Brně, 2002.
- [10] Firemní dokumentace Matlab and Simulink (MathWorks).
- [11] Firemní dokumentace Rex Controls.

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

Vedoucí bakalářské práce: **doc. Ing. Štěpán Ožana, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum odevzdání bakalářské práce: 28. 4. 2017



.....
Marek Bukovan

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce panu doc. Ing. Štěpánu Ožanovi, Ph.D. za ochotu, vedení, konzultace a připomínky, díky kterým se podařilo tuto práci dotáhnout do konce. Dále bych také rád poděkoval panu Ing. Tomáši Dočkalovi za poskytnuté konzultace a v neposlední řadě také své manželce, která mě podporovala a stála při mně po celou dobu studia.

Abstrakt

Tato práce se zabývá problematikou SIL a PIL simulace. Cílem práce bylo navržení SIL a PIL simulátoru pro výukový fyzikální model inverzního kyvadla. V úvodní části je popsán řídicí systém REX, který byl využit pro realizaci SIL a PIL simulátoru implementovaného na platformu Raspberry Pi. Dále je zde popsána problematika modelování a simulace na kterou navazuje kapitola ve které je odvozen matematický model inverzního kyvadla. Na základě tohoto modelu byly pak sestaveny modely pro jednotlivé simulace. V závěru práce se pak nachází porovnání výstupů jednotlivých modelů s daty získanými z reálného kyvadla.

Klíčová slova

modelování, simulace, inverzní kyvadlo, REX, PIL, SIL, Raspberry Pi, Simulink

Abstract

This work deals with issue of SIL and PIL simulation. The aim of the work is the design of SIL and PIL simulator for educational physical model of inverted pendulum. In the first part of thesis there is description of control system REX, which has been used for SIL and PIL simulator, which is working on Raspberry Pi. There is also described the issue of modeling and simulation which is followed by the chapter with description of mathematic model of the inverted pendulum. Models for simulators have been designed on basis of mathematic model. In the conclusion there is a comparison of the outputs from the simulator with the data from real pendulum.

Key words

modeling, simulation, inverted pendulum, REX, PIL, SIL, Raspberry Pi, Simulink

Obsah

Seznam použitých symbolů a zkratk	1
Seznam obrázků	2
Úvod.....	3
1 Popis řídicího systému REX a platformy Raspberry Pi	4
1.1 Řídicí systém REX.....	4
1.1.1 RexDraw.....	5
1.1.2 RexComp.....	5
1.1.3 RexCore.....	5
1.1.4 RexVeiw.....	5
1.1.5 RexHMI Designer	6
1.2 Raspberry Pi.....	6
1.2.1 Popis Raspberry Pi	6
1.2.2 Raspberry Pi s řídicím systémem REX	7
2 Modelování a simulace.....	8
2.1 Modelování	8
2.1.1 Motivační příklad	9
2.2 Simulace.....	10
2.2.1 Model in the loop	10
2.2.2 Software in the loop	11
2.2.3 Processor in the loop	12
2.2.4 Hardware in the loop	13
2.2.5 Srovnání odezvy jednotlivých typů simulací	14
3 Model inverzního kyvadla na vozíku řízeného DC motorem	15
3.1 Fyzikální model inverzního kyvadla	15
3.2 Matematický model inverzního kyvadla na vozíku	16
3.3 Stavový popis inverzního kyvadla na vozíku.....	18
4 Koncepce řízení modelu inverzního kyvadla	21
5 Návrh a realizace SIL a PIL simulátoru v prostředí REX.....	23
5.1 Model regulátoru v prostředí REX.....	23
5.2 Model soustavy v prostředí REX	23
5.2.1 SIL simulace pro inverzní kyvadlo	25
5.2.2 PIL simulace pro inverzní kyvadlo	26
5.3 Vizualizace.....	27
5.4 Realizace testovacího stojanu.....	29
6 Ověření činnosti SIL a PIL simulátoru.....	30
6.1 Určení koeficientu tření b.....	30
6.2 Ověření výšvihu a regulace v horní poloze	31
7 Závěr	36
Literatura.....	38
Přílohy.....	40

Seznam použitých symbolů a zkratk

zkratka	význam anglicky	význam česky
MIL	model in the loop	model ve smyčce
PIL	processor in the loop	procesor ve smyčce
SIL	software in the loop	software ve smyčce
HIL	hardware in the loop	hardware ve smyčce
FBD	function block diagram	jazyk blokových schémat
GPIO	general puprose input/output	programovatelné vstupně-výstupní piny
SoC	system on chip	system na čipu
RISC	reduced instruction set	redukováná instrukční sada
CISC	complex instruction set	komplexní instrukční sada
ARM	Acorn RISC machine	označení architektury procesoru vycházejí z názvu výrobce
CSI	camera interface	rozhraní pro připojení kamery
DSI	display interface	rozhraní pro připojení displeje
PLC	programmable logic controller	programovatelný logický automat
PWM	pulse width modulation	pulzně šířková modulace
RT modeling	real-time modeling	modelování v reálném čase
PID regulator	proportional – integral – derivate regulator	proporcionálně integračně derivační regulátor
DAQ	data acquisition	sběr dat
LQR	linear quadratic regulator	lineárně-kvadratický regulátor
IP	iverted pendulum	inverzní kyvadlo

Seznam obrázků

Obrázek 1: Schéma systému REX.	4
Obrázek 2: Ukázka práce překladače.	5
Obrázek 3: Raspberry Pi 3 Model B v1.2.	6
Obrázek 4: Komunikační rozhraní Raspberry Pi 3 Model B v1.2.	7
Obrázek 5: Systém druhého řádu.	9
Obrázek 6: Model soustavy vyjádřený ve Frobeniově kanonickém tvaru.	10
Obrázek 7: MIL simulace řídicího systému v prostředí Simulink.	11
Obrázek 8: Ukázka SIL simulace v řídicím systému REX.	12
Obrázek 9: PIL simulace – regulátor běžící na cílovém zařízení.	12
Obrázek 10: PIL simulace – matematický model běžící v reálném čase na simulačním HW.	13
Obrázek 11: Model HIL simulace.	13
Obrázek 12: Srovnání odezvy MIL, SIL, PIL simulace.	14
Obrázek 13: Fyzikální model kyvadla.	15
Obrázek 14: Matematický model inverzního kyvadla na vozíku.	16
Obrázek 15: Vnitřní stavové schéma inverzního kyvadla.	18
Obrázek 16: Vnitřní stavové schéma rychlostní smyčky.	19
Obrázek 17: Stavové schéma modelu inverzního kyvadla řízeného stejnosměrným motorem.	20
Obrázek 18: Schéma soustavy tvořené stavovým regulátorem a matematickým modelem.	20
Obrázek 19: Struktura se dvěma stupni volnosti.	21
Obrázek 20: Schéma pro návrh LQR regulátoru R.	21
Obrázek 21: Zapojení stavového regulátoru.	22
Obrázek 22: Celkové zapojení regulačního obvodu pro řízení reálného modelu.	23
Obrázek 23: Model soustavy s rychlostní smyčkou aproximovanou přenosem 1. řádu.	24
Obrázek 24: Model soustavy s PID regulátorem.	25
Obrázek 25: Blokové schéma SIL simulace pro inverzní kyvadlo.	26
Obrázek 26: Blokové schéma PIL simulace inverzního kyvadla pro master a slave.	26
Obrázek 27: Nastavení komunikace prostřednictvím Modbus.	27
Obrázek 28: Ukázka přiřazení proměnných.	28
Obrázek 29: Ukázka vizualizace inverzního kyvadla.	28
Obrázek 30: Ukázka praktického řešení PIL simulátoru.	29
Obrázek 35: Neřízené spuštění kyvadla pro $b = 0.0225$	30
Obrázek 36: Neřízené spuštění kyvadla pro $b = 0.0210$	31
Obrázek 31: Srovnání x_1 (poloha ramene).	32
Obrázek 32: Srovnání x_2 (rychlost ramene).	33
Obrázek 33: Srovnání x_3 (poloha vozíku).	33
Obrázek 34: Srovnání x_4 (rychlost vozíku).	34

Úvod

V praxi se lze velice často setkat s případy, kdy je nezbytné navržený algoritmus řídicího systému otestovat pomocí simulace. Simulace umožňuje odhalit chyby již ve fázi vývoje a dokáže tak předejít značným finančním ztrátám, zraněním nebo dokonce ztrátám na lidských životech. Pro tyto účely existuje několik druhů simulací např. simulace typu MIL, SIL, PIL a HIL, přičemž tato práce se zaměřuje zejména na simulaci typu PIL a SIL.

Cílem této práce je návrh a realizace PIL a SIL simulátoru pro výukový fyzikální model inverzního kyvadla na vozíku. U simulace typu PIL je matematický model simulován v reálném čase a řídicí systém je provozován na tzv. „target device“ – čili na cílovém zařízení, pro které je regulátor navržen. Na rozdíl od simulace typu SIL, kdy je matematický model spolu s řídicím systémem provozován na jednom počítači.

Pro účely simulace byl vybrán klasický problém teorie řízení – inverzní kyvadlo, na jehož příkladu budou demonstrovány oba typy simulací. Jako vývojové prostředí, ve kterém bude simulace realizována byl zvolen řídicí systém REX běžící na platformě Raspberry Pi.

Úvodní část této práce je věnována seznámení se s řídicím systémem REX a mikrokontrolérem Raspberry Pi. Další část se pak věnuje problematice modelování a simulace kde je uveden motivační příklad, který demonstruje tuto problematiku na příkladu sériové zapojení dvou RC článků.

V následující části práce je popsán model inverzního kyvadla na vozíku, na jehož základě byl vytvořen model pro simulaci typu SIL a PIL. Výpočty stavového regulátoru a optimálních trajektorií, pro jejichž ověření slouží tyto simulace především, nebyly předmětem této práce. Nicméně z důvodu komplexního pojetí této práce zde byla zařazena i kapitola koncepce řízení modelu inverzního kyvadla, která popisuje způsob řízení.

Závěr práce je pak věnován porovnání jednotlivých průběhů stavových veličin a jejich srovnání s daty získanými reálným měření.

1 Popis řídicího systému REX a platformy Raspberry Pi

Jak již bylo v úvodu předznamenáno, pro účely simulace byl zvolen řídicí systém REX, který poběží na jednodeskovém počítači Raspberry Pi 3, přičemž obě tyto platformy budou popsány v následující kapitole.

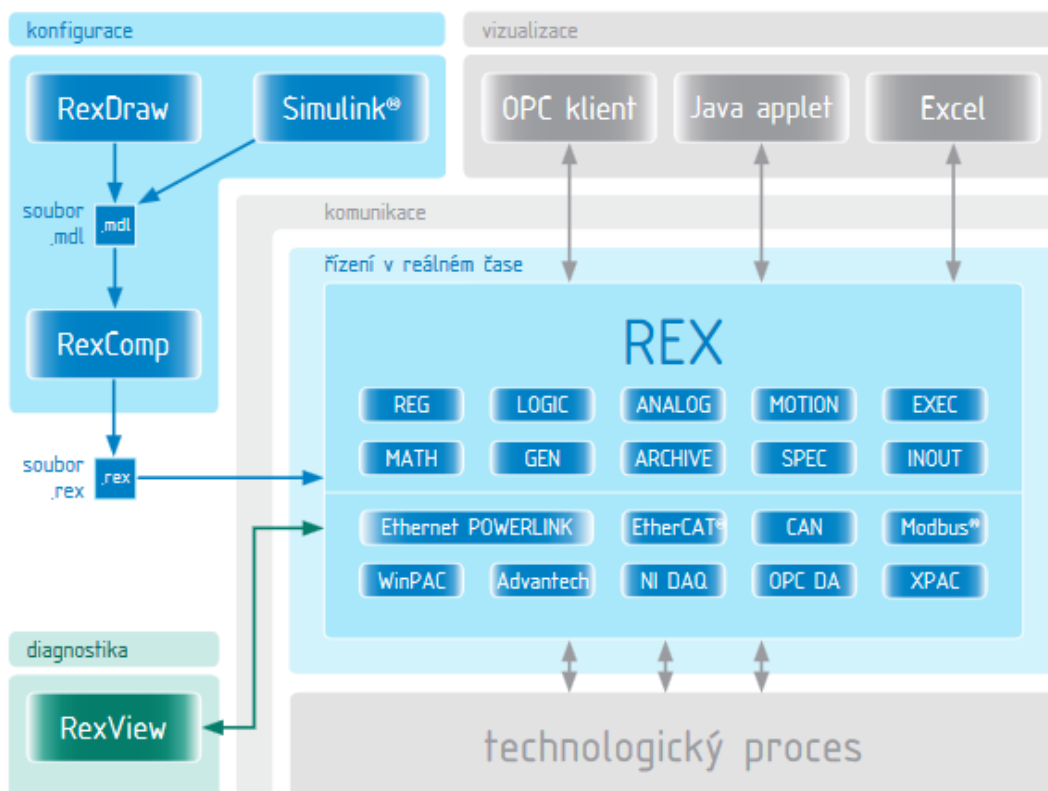
1.1 Řídicí systém REX

Jedná se o vyspělý nástroj pro návrh a realizaci komplexních algoritmů automatického řízení, který byl vyvinu plzeňskou společností REX Controls. Součástí řídicího systému REX je knihovna funkčních bloků, která umožňuje sestavit výsledný řídicí algoritmus, ať už pro oblast automatizace a regulace, nebo i pro celou řadu pokročilých algoritmů řízení. Umožňuje však také programování v programovatelném bloku pro speciální aplikace. [12]

Další přednost systému REX spočívá i v jeho kompatibilitě se systémem Matlab-Simulink a umožňuje tak využít všechny výhody pro simulaci a testování. Pro tuto možnost je ovšem potřeba doinstalovat knihovnu funkčních bloků RexLib. [12]

Přehledné schéma systému REX zobrazuje obrázek 1. Základní součástí řídicího systému pak tvoří:

- RexDraw – vývojové prostředí
- RexComp – překladač
- RexCore – runtime jádro
- RexView – diagnostický nástroj
- RexHMI designer – nástroj pro tvorbu vizualizací



Obrázek 1: Schéma systému REX. [12]

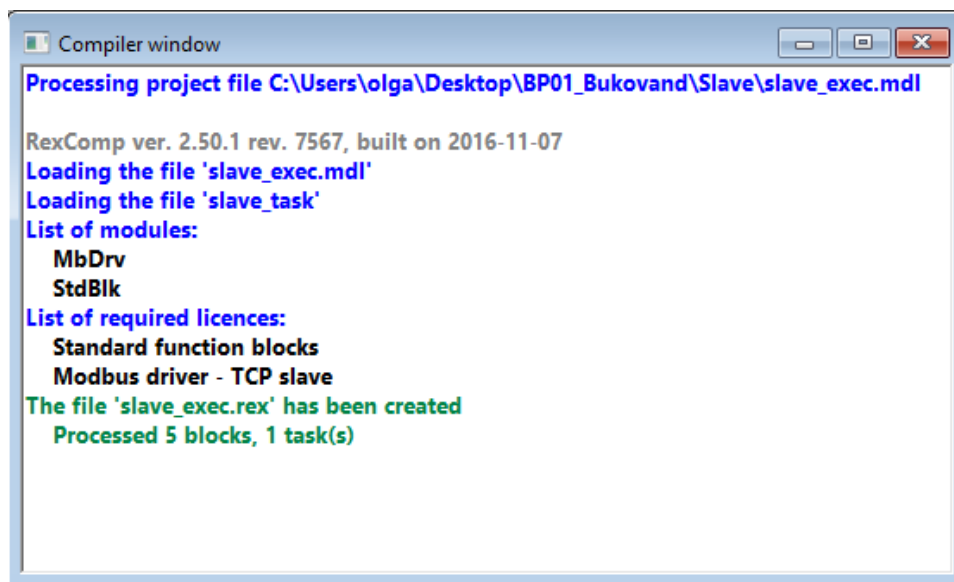
1.1.1 RexDraw

Jednou ze základních komponent řídicího systému Rex je grafické vývojové prostředí RexDraw, které slouží k vytváření algoritmů na základě rozsáhlé knihovny funkčních bloků a umožňuje tak programování FBD, tak jak jej definuje norma IEC 61131-3.

Součástí knihovny bloků jsou například jednoduché komparátory, časovače, pokročilé bloky pro zpracování analogových signálů a regulaci, PID regulátory s automatickým laděním parametrů a mnoho dalších. Algoritmus lze ihned zkompileovat a spustit na cílovém zařízení (Linux IPC, WinPAC, Raspberry Pi atd.). Při běhu algoritmu umožňuje RexDraw sledovat v reálném čase všechny signály a funkční bloku. [13]

1.1.2 RexComp

Pro kompilaci algoritmu v řídicím systému REX slouží překladač RexComp. Ten převádí algoritmy vytvořené v RexDraw, případně v Simulinku do binárního formátu systému REX. Tento formát lze pak přímo spustit v RexDraw. Při kompilaci se vypisují informace o překládaných souborech či případných chybách, tak jak lze vidět na obrázku 2. [13]



Obrázek 2: Ukázka práce překladače.

1.1.3 RexCore

Jádro RexCore běží přímo na cílovém zařízení na rozdíl od komponent uvedených výše. Zajišťuje spouštění a časování programů a současně provádění dalších činností, které jsou obvyklé v řídicích systémech. Jednotlivé úlohy jsou vykonávány na základě priorit v režimu preemptivního multitaskingu. [13]

1.1.4 RexVeiw

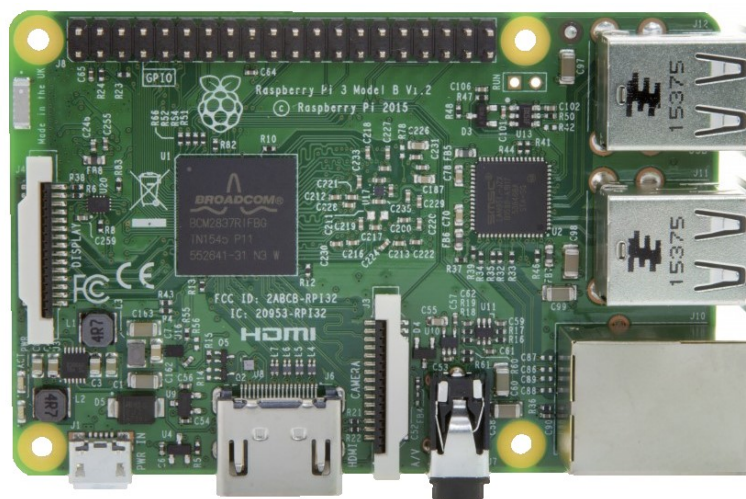
Jedná se o diagnostický nástroj, jenž umožňuje sledovat, jak jádro řídicího systému REX vykonává program. Významnou roli hraje zejména při uvádění do provozu řídicího systému a při diagnostice problémů během rutinního provozu. [13]

1.1.5 RexHMI Designer

Pro vytváření vizualizací lze použít nástroj RexHMI designer, který funguje na bázi Inkscape – jedná se v podstatě o jeho rozšíření. S novou verzí řídicího systému REX 2.50.1. došlo k implementaci RexHMI Designer přímo do RexDraw a uživatel jej tak může snadno spustit při programování. Vizualizaci lze pak sledovat na tabletu nebo chytrém telefonu.

1.2 Raspberry Pi

Raspberry Pi je jednodeskový počítač vyvinutý společností Raspberry Pi Foundation, který vznikl původně jako myšlenka levného počítače s cílem zvýšit zájem o programování u malých dětí. Postupem času se však ukázalo, že jeho využití je daleko širší a předčilo očekávání tvůrců. Lze jej použít jak k řízení hardware, k čemuž můžeme využít GPIO piny, tak také jako multimediální centrum. Na obrázku 3 lze vidět nejnovější model uvedený na trh v únoru roku 2016. Ten oproti svému předchůdci nabízí výkonnější procesor a širší možnosti konektivity jako je wi-fi nebo Bluetooth 4.1. [9], [19]



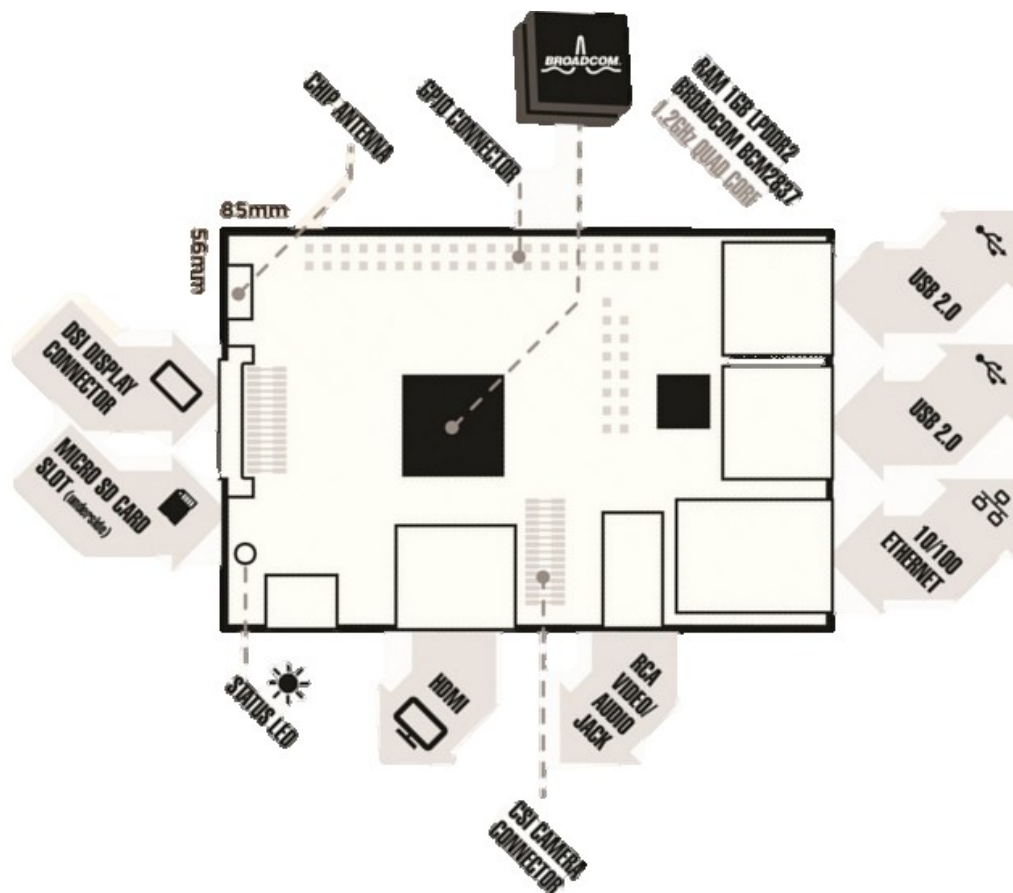
Obrázek 3: Raspberry Pi 3 Model B v1.2. [8]

1.2.1 Popis Raspberry Pi

Základem Raspberry Pi 3 Model B v1.2 je 64 bitový čtyřjádrový procesor ARMv8 Broadcom BCM 837 s frekvencí 1.2 GHz typu SoC. To znamená, že většina systémových komponent, včetně hlavního a grafického procesoru spolu se zvukovým a komunikačním hardwarem je integrovaná do jediného čipu s kapacitou paměti 1 GB RAM. Tento typ procesoru s architekturou ARM je velice rozšířen v mobilních telefonech díky vysokému výkonu a nízké spotřebě. [19]

ARM architektura využívá redukované instrukční sady RISC, což se významně projevuje na nízké spotřebě energie a tyto procesory tak nepotřebují dodatečné chlazení na rozdíl od procesoru stolních počítačů, které pracují s komplexní instrukční sadou CISC. Z těchto důvodů lze Raspberry Pi napájet přes integrovaný port mikro USB zdrojem napájení s napětím 5 V a proudem 1 A. Nevýhodou této architektury je nekompatibilita většiny softwaru určených pro stolní počítače, jelikož ten využívá architekturu s instrukční sadou x86. [19]

Pro komunikaci s periferiemi je Raspberry Pi vybaveno celou řadou komunikačních rozhraní, jako jsou GPIO piny, Wi-fi standard 802.11n, USB, HDMI port, Ethernet, kombinovaný 3.5 mm jack pro audio a video, CSI, DSI a slot pro micro SD kartu. Přehledný popis všech komunikačních rozhraní zobrazuje obrázek 4. [9]



Obrázek 4: Komunikační rozhraní Raspberry Pi 3 Model B v1.2. [8]

1.2.2 Raspberry Pi s řídicím systémem REX

V úvodní části této podkapitoly bylo zmiňováno, že lze vytvářet aplikace pro řízení hardware například pomocí kódu jazyka Python, který je součástí většiny distribucí Linux. Pro Raspberry Pi byl speciálně vyvinut a optimalizován operační systém Raspbian jenž vychází z operačního systému Debian (jedna z distribucí Linuxu). Operační systém obsahuje sadu základních programů a nástrojů, které zajišťují samotný běh Raspberry Pi. [19], [10].

S řídicím systémem REX lze z Raspberry Pi vytvořit programovatelné PLC, které lze programovat a konfigurovat z běžného stolního počítače v nástroji RexDraw. Pomocí funkčních bloků můžeme definovat libovolnou zavislost mezi senzory a akčními členy. Pro interakci s okolím slouží GPIO piny, jenž se dají konfigurovat jako vstupní nebo výstupní. Raspberry Pi však neobsahuje žádný analogový vstup. Pin č. 18 je konfigurovatelný jako PWM. [11], [20]

2 Modelování a simulace

V této kapitole je uveden popis nejdůležitějších pojmů z oblasti modelování a simulace se zaměřením na simulace typu SIL a PIL, jež jsou předmětem této práce. Informace obsažené v této kapitole vycházejí zejména z [6] a [3].

2.1 Modelování

Základem každého modelování je systém, přičemž pod pojmem *system* se rozumí izolovaná část světa, kterou pozorujeme ve vymezeném čase a z určitého důvodu. Izolovaná část, která je předmětem našeho pozorování se považuje za systém a vše ostatní se považuje za okolí systému. Jiná, častější definice, definuje systém jako množinu určitých prvků a vazeb mezi nimi, které jako celek mají určité vlastnosti. [6], [3]

Například pokud astrofyzici chtějí studovat gravitační síly mezi Sluncem a planetami, budou pozorovat Solární systém. Zbytek galaxie nebo dokonce malé nevýznamné události odehrávající se na Zemi nebo ostatních planetách budou pro ně představovat okolí systému. Stejně tak jako pro zákazníka, který si chce koupit auto, bude auto představovat izolovanou část reálného světa, ačkoliv auto je tvořeno celou řadou systémů jako je spalovací, brzdový, palivový nebo elektrický systém. V danou chvíli zajímá zákazníka vše, co se týče barvy vozu, ceny na trhu, spotřeby paliva atd. Celkově se snaží najít takové auto, který by odpovídal jeho potřebám. Stejně auto, avšak z jiného pohledu, posuzují inženýři, kteří zkoumají různé subsystémy. Ti auto považují za soubor subsystémů. Subsystém je celkový systém, který posuzujeme a ostatní subsystémy stejného systému jsou pro něj okolním systémem. [6]

Přístup systémového inženýrství umožňuje jasně rozlišit mezi posuzovanou soustavou a tím co není v daný okamžik důležité. Pokud hranice systému jsou jasně vymezené, lze provést posouzení systému uvnitř této oblasti velice snadno. Přístup systémového inženýrství vede ke kvantifikaci objektů a jejich řešení zavedenými analytickými metodami. Obecně platí, že přístup systémového inženýrství kvantifikuje dynamiku systému prostřednictvím matematického aparátu v příslušném časovém rámci. Dynamický systém je takový systém, který mění s časem svůj stav, přičemž v reálném světě neexistuje jediný systém, který by svůj stav neměnil. Systém, který se jeví jako statický, buď není posuzován v odpovídajícím časovém rámci nebo jeho změna je mimo rozsah pozorování. Například hvězda Polárka pomáhala po staletí s navigací a orientací na noční obloze, ale jednou za 5000 let v důsledku precese dojde ke změně zdánlivé polohy Polárky a její funkce bude nahrazena jinou hvězdou. Tento příklad dokládá, že i zdánlivě statický systém vykazuje jistou dynamiku. [6]

Pro definici jakéhokoliv dynamického systému existují dva kritické aspekty. První aspekt je zvolit vhodný časový rámec ve kterém je změna pozorovatelná a druhý aspekt je zvolit stav který se bude měnit. Systém, který mění své stavy v dostatečném rozsahu, se považuje za dynamický systém. Jiné měřítko pro pozorování systému bude mít astrofyzik, který uvažuje změny v řádech tisíců let a jiné měřítko bude mít například komunikační inženýr, který uvažuje změny v řádech milisekund. Definování izolované části reálného světa je důležité pro zvolení vhodného časového měřítka a zvolení vhodné stavové proměnné. [6]

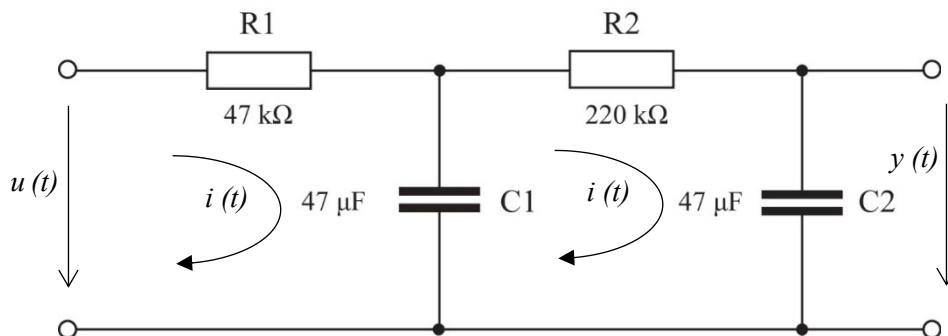
Termín stavová proměnná je definovaná jako soubor proměnných, které jsou nezbytné k jednoznačnému popisu systému. Tyto proměnné představují minimální soubor těch proměnných, které ho zcela popisují. Kombinace těchto proměnných se nazývá stavový vektor. [6]

Každý systém interaguje s okolním systémem prostřednictvím vstupu nebo výstupu. Vstup popisuje změny systému představované okolím, kdežto výstup představuje příspěvek systému do okolí a obvykle bývá vstupem dalšího systému. Například vozidlo, které jede po silnici nasává do spalovacího systému vzduch, ten se považuje za vstup systému, a naopak výstupem jsou spaliny a rychlost vozidla. [6]

Pro práci se systémem je často vyžadován jeho model, jenž představuje zjednodušenou reprezentaci části reality se souvisejícími prvky. Proces tvorby vizuální, koncepční nebo matematické reprezentace se nazývá modelování. Může se jednat o graf, vývojový diagram, matematické rovnice nebo málo-rozměrovou repliku systému či procesu. Pro proces modelování a simulace lze využít celou řadu softwarových nástrojů jako je RexDraw nebo nejznámější Simulink. [6]

2.1.1 Motivační příklad

Následující příklad demonstruje vytvoření matematického modelu soustavy druhého řádu, jenž ve skutečnosti představuje sériové zapojení dvou RC článků dle obrázku 5. Kompletní odvození matematického modelu a jeho převedení na Frobeniův tvar je uveden v příloze I. Obrázek 6 zobrazuje model soustavy ve Frobeniově tvaru v prostředí Simulinku, který se používá na vnitřní popis soustavy.



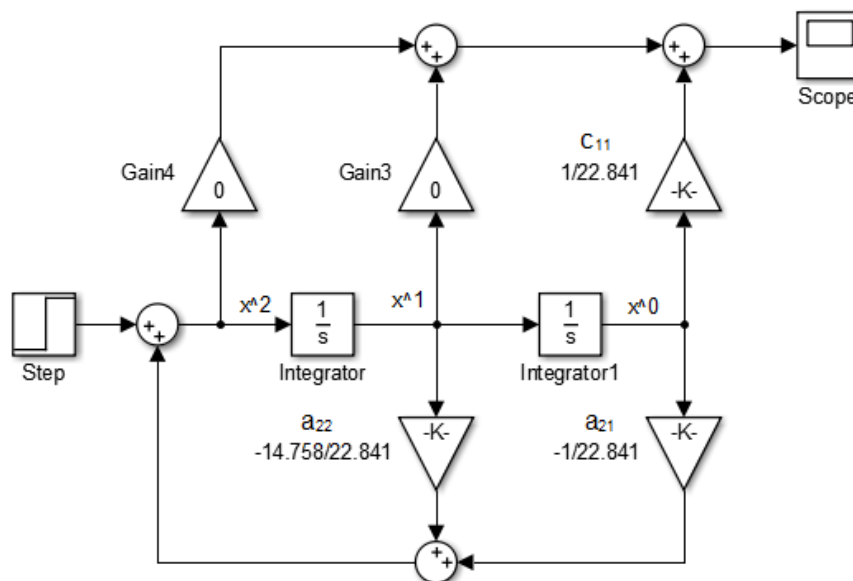
Obrázek 5: Systém druhého řádu.

Přenos takového systému je pak následující:

$$G = \frac{Y}{U} = \frac{1}{R_1 C_1 R_2 C_2 s^2 + R_1 C_1 s + R_2 C_2 s + R_1 C_2 s + 1} \quad (2.1)$$

Po vyčíslení a upravení rovnice dostaneme následující tvar:

$$G = \frac{1}{22,841s^2 + 14,758s + 1} \quad (2.2)$$



Obrázek 6: Model soustavy vyjádřený ve Frobeniově kanonickém tvaru.

2.2 Simulace

Simulace je jeden z nejmocnějších nástrojů, který pomáhá v rozhodování lidem, jež jsou zodpovědní za návrh a provoz komplexních systémů a procesů. Umožňuje jim studovat, analyzovat a posuzovat situace, které by jinak nebyly možné.

V odborné literatuře lze najít různé vysvětlení pojmu *simulace*, ale často se lze setkat s definicí podle Shannona, který definuje simulaci jako proces tvorby modelu reálného systému a provádění experimentů s tímto modelem za účelem dosažení lepšího pochopení chování studovaného systému či za účelem posouzení různých variant činnosti tohoto systému. [1]

V ranné fázi vývoje řídicího systému je prakticky nemožné jeho testování přímo na reálné soustavě. Z těchto důvodů se používá několik typů simulací, které umožňují testovat řídicí systém v různých fázích vývoje, ve kterých lze odhalit celou řadu chyb. Rozlišují se následující typy simulací:

- Model in the Loop
- Software in the Loop
- Processor in the Loop
- Hardware in the Loop

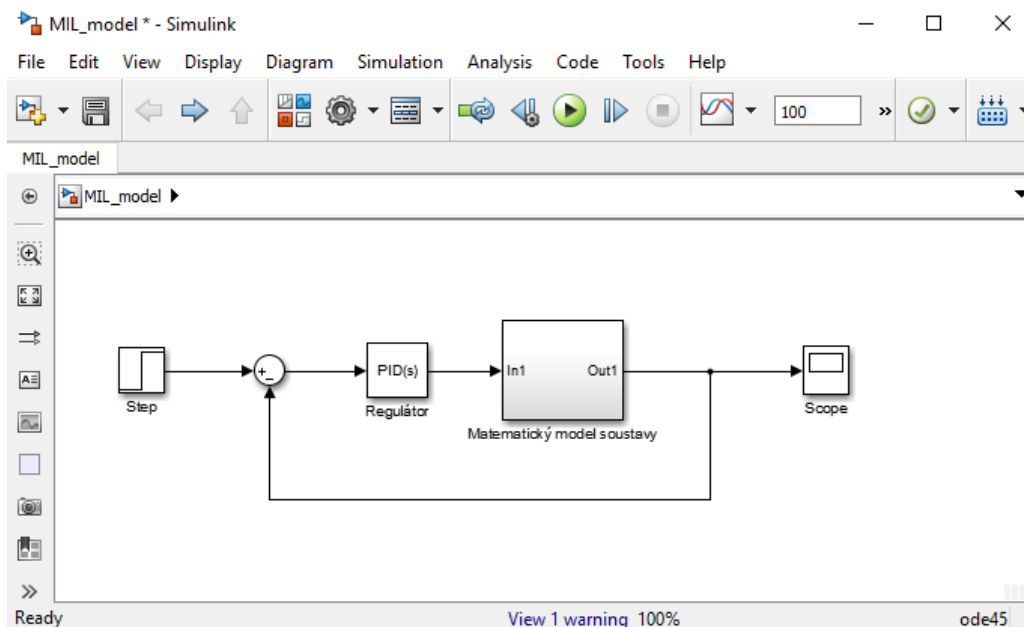
Na závěr, když je vše odladěno se provádí finální systémový test na prototypu.

V následující kapitole budou uvedeny základní informace o simulacích MIL, SIL, PIL a HIL a budou zde uvedeny ukázky realizace technik těchto simulací na motivačním příkladu z podkapitoly 2.1.1.

2.2.1 Model in the loop

O simulaci typu MIL se hovoří tehdy, je-li matematický model i řídicí systém simulován v simulačním prostředí na jednom počítači. V tomto případě se nejedná o real-time simulaci. Model soustavy i řídicí algoritmus je ve formě tzv. „source – code“, což je nezkompilovaná forma kódu.

MIL simulace slouží k předběžnému ověření požadavků a algoritmů předpokládaného řešení. Využívá se k optimalizaci konstrukce stroje nebo technologie a definování nutných vstupních a výstupních signálů. Příklad takovéto simulace lze vidět na obrázku 7. Důležitým aspektem je, že čas simulace je předem vymezen. [18], [14]

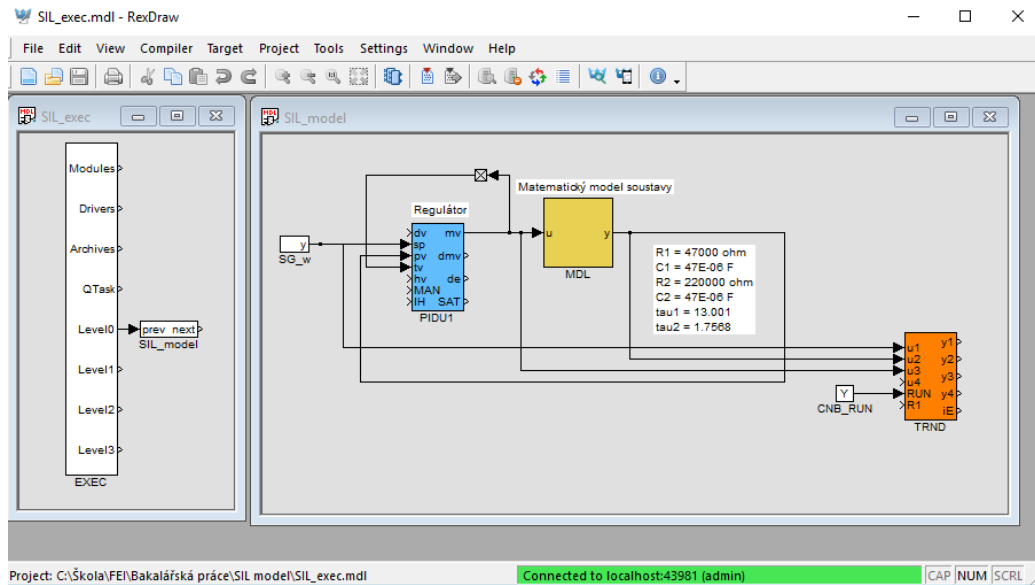


Obrázek 7: MIL simulace řídicího systému v prostředí Simulink.

2.2.2 Software in the loop

U simulace typu SIL je matematický model spolu s řídicím systémem simulován na jednom počítači. V tomto případě se pracuje s tzv. „target – code“, což je zkompilevaná forma kódu. Tato simulace se odehrává v reálném čase, nicméně kompilátor a procesor počítače může jinak pracovat na cílovém zařízení, než na kterém je simulace prováděna. Z tohoto důvodu se v některé literatuře uvádí, že se nejedná o simulaci v reálném čase. Definice hovoří o synchronizaci mezi simulací řízeného systému a simulací řídicího systému [2]

Cílem simulace je ověření výpočetní náročnosti řízení se zvolenou periodou vzorkování a vybrání správného výkonu řídicího hardware. Většina chyb může být odhalena již v této úrovni simulace. Obrázek 8 zobrazuje simulaci typu SIL v řídicím systému REX. [17]

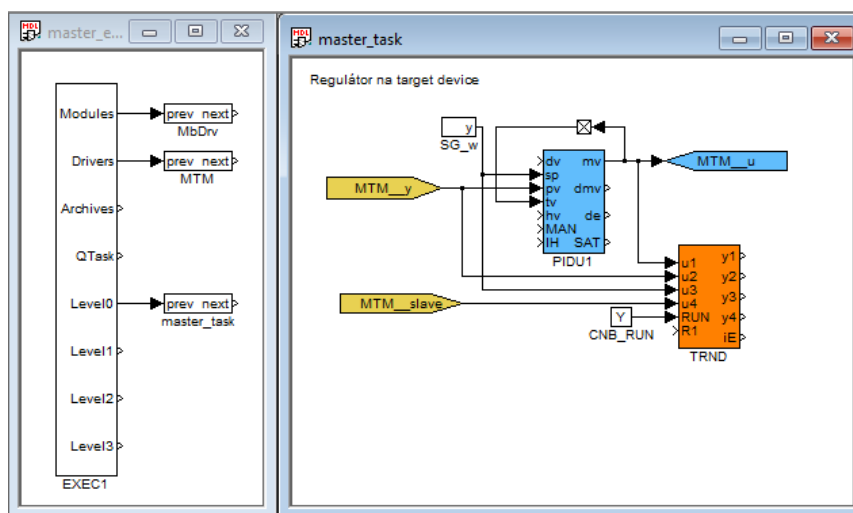


Obrázek 8: Ukázka SIL simulace v řídicím systému REX.

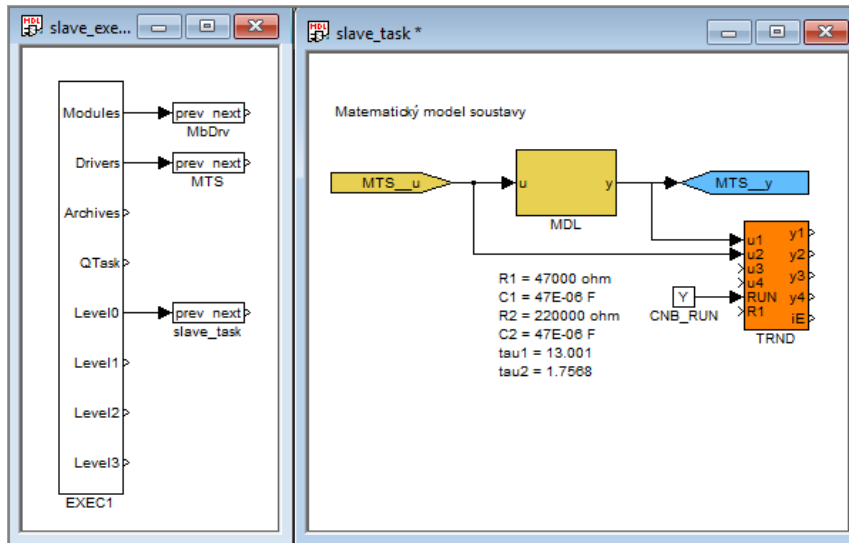
2.2.3 Processor in the loop

Simulace typu PIL je charakteristická tím, že matematický model je simulován v reálném čase, ale řídicí systém je již provozován na cílové HW platformě – tzv. „embedded target“. Při komunikaci nejsou použity žádné I/O karty, čidla či aktuátory, ale data mezi modelem a řídicím systémem se vyměňují prostřednictvím průmyslové sběrnice. Typ této simulace se využívá zejména k ověřování výpočetního výkonu řídicího HW a testování kritických situací. [16]

Podobně jako v předchozím případě i zde je uvedena ukázka simulace v prostředí řídicího systému REX. Na obrázku 9 lze vidět FBD regulátoru soustavy, který je provozován na cílovém zařízení. Ten komunikuje s řízenou soustavou, jenž lze vidět na obrázku 10 (v tomto případě stále ještě matematickým modelem soustavy), prostřednictvím průmyslové sběrnice, například ethernet využívající protokolu MODBUS. Základní myšlenkou je, aby regulátor poskytoval stejné řídicí signály soustavě a bylo tak prakticky jedno, zda reguluje matematický model soustavy nebo skutečnou soustavu.



Obrázek 9: PIL simulace – regulátor běžící na cílovém zařízení.



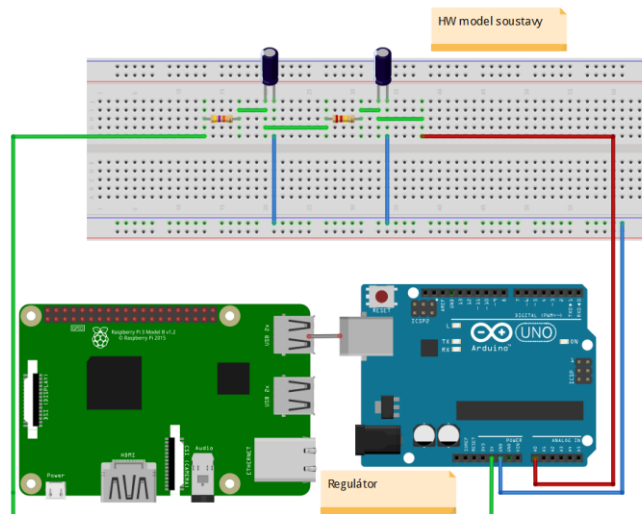
Obrázek 10: PIL simulace – matematický model běžící v reálném čase na simulačním HW.

2.2.4 Hardware in the loop

Simulace typu HIL představuje poslední stádium před vývojem prototypu. Tak jako u simulace typu PIL je řídicí systém provozován na cílovém zařízení, a navíc jsou použity i vstupně výstupní karty, případně emulace čidel a akčních členů. V této fázi umožňuje otestovat kompletní řídicí systém včetně DAQ karet a jejich ovladačů, navíc lze najít velice nepatrné chyby, které jsou způsobeny například provozním prostředím.[15]

Ve srovnání se simulacemi typu MIL nebo SIL představuje simulace HIL nejdražší variantu testování, nicméně i přes to představuje významnou úsporu nákladů oproti stavbě prototypu. Je výrazně levnější najít chybu v brzké fázi vývoje než při testování reálného prototypu. [18]

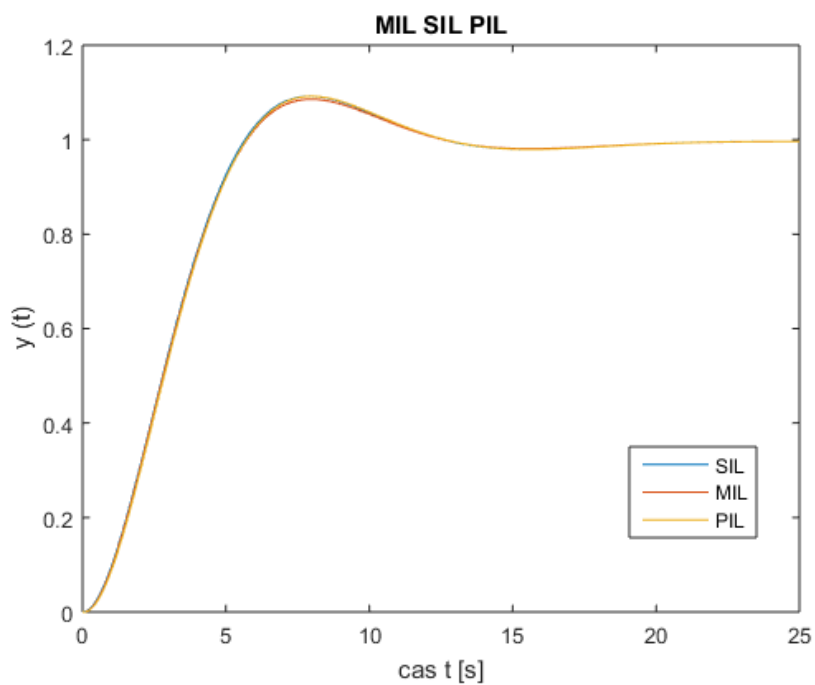
V programu Fritzing byl vytvořen model pro HIL simulaci, jež lze vidět na obrázku 11. Protože u Raspberry nejsou k dispozici analogové vstupy, bylo potřeba využít mikrokontroléru Arduino. Obě tyto desky spolu komunikují přes USB rozhraní a využívají software REXduino.



Obrázek 11: Model HIL simulace

2.2.5 Srovnání odezvy jednotlivých typů simulací

Následující obrázek 12 srovnává odezvy RC článku ze simulace typu MIL, SIL a PIL. Lze vidět, že rozdíly mezi průběhy jsou velice nepatrné.



Obrázek 12: Srovnání odezvy MIL, SIL, PIL simulace

3 Model inverzního kyvadla na vozíku řízeného DC motorem

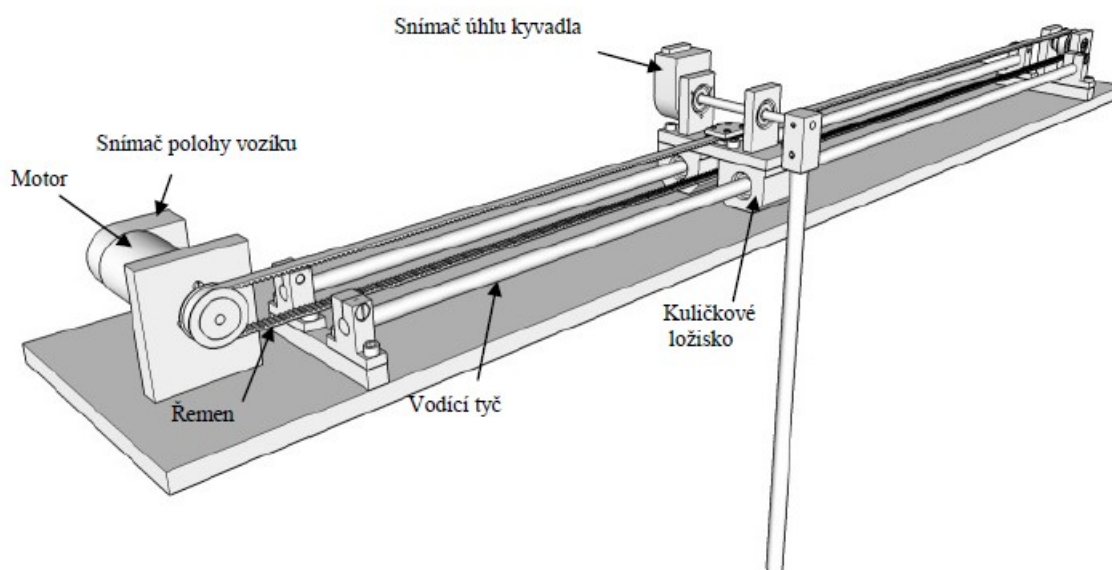
Inverzní kyvadlo je nelineární astatický systém složený z lineárně se pohybujícího pohonu a na něm volně uchyceného kyvadla. Níže bude popsán fyzikální model inverzního kyvadla na vozíku řízeného DC motorem (dále jen model inverzního kyvadla) včetně jeho matematického popisu.

3.1 Fyzikální model inverzního kyvadla

Fyzikální model inverzního kyvadla (obrázek 13) je tvořen stejnosměrným motorem se snímačem polohy vozíku, řemenem pro přenos momentu na vozík, na kterém je upevněné kyvadlo se snímačem úhlu natočení. Vozík se pohybuje po vodících tyčích. Jako hlavní konstrukční materiál pro kyvadlo, s výjimkou ložisek, byl zvolen dural, jenž se vyznačuje vysokou pevností a nízkou vahou.

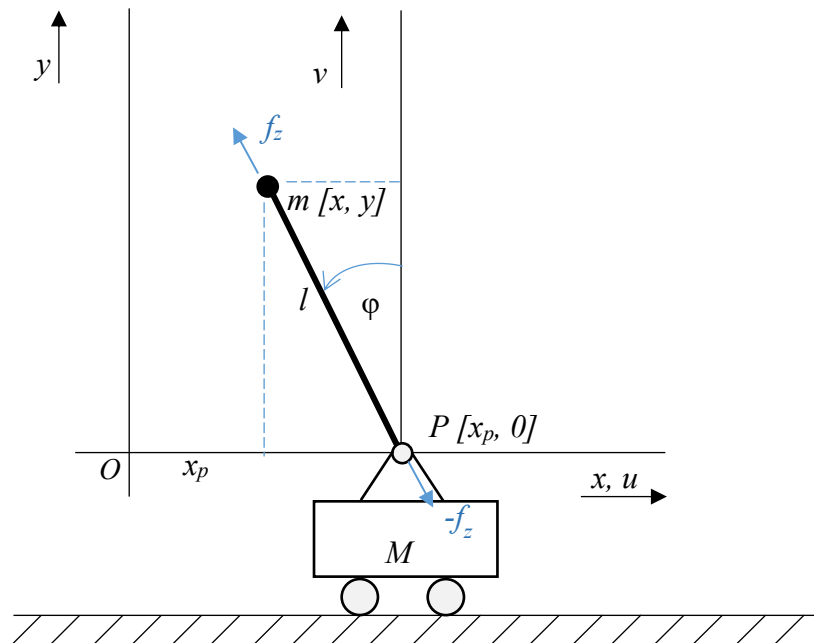
Pro pohon vozíku byl konkrétně zvolen motor Maxon A-max, který má výborné dynamické vlastnosti a vysokou účinnost. Předností tohoto motoru je okamžitá reakce na změnu napájecího napětí, takže se hodí právě pro tento typ úlohy, kdy se vyžaduje velice krátká odezva.

K měření polohy vozíku a úhlu natočení kyvadla byly použity optoelektrické inkrementální snímače HEDM-5500J12. Tyto snímače používají jako zdroj světla LED diodu. Snímač je tvořen integrovaným obvodem s detektorem, výstupním obvodem a kódovacím kotoučem, který se otáčí mezi zdrojem světla a detektorem a převádí tak pohyb na elektrické impulsy. Počet impulsů odpovídá velikosti změny polohy a frekvence odpovídá rychlosti pohybu kyvadla. [5], [7]



Obrázek 13: Fyzikální model kyvadla [5]

3.2 Matematický model inverzního kyvadla na vozíku



Obrázek 14: Matematický model inverzního kyvadla na vozíku

Pro účely matematického popisu soustavy se uvažuje jako pevná soustava O_{xy} a pohybující se soustava P_{uv} (tj. soustava pevně spojenou s pohybujícím se vozíkem) – viz. obrázek 14.

$$\begin{aligned} u &= -l \sin \varphi \\ v &= l \cos \varphi \end{aligned} \tag{3.1}$$

Označení:

f_z síla kterou působí vozík (přes nehmotné rameno) na hmotný bod m

$[x, y]$ souřadnice kyvadla (hmotného bodu m) v soustavě O_{xy} .

$[u, v]$ souřadnice kyvadla (hmotného bodu m) v soustavě P_{uv} .

F_x, F_y síly působící na hmotný bod m po řadě ve směru x a y .

Vztahy mezi souřadnicemi x, y a u, v :

$$\begin{aligned} x &= x_p + u = x_p - l \sin \varphi \\ y &= v = l \cos \varphi \end{aligned} \tag{3.2}$$

Newtonův pohybový zákon pro kyvadlo (hmotný bod m):

$$F_x = -F_z \cdot \sin \varphi = \frac{d^2}{dt^2}(x_p + u) \cdot m \quad (3.3)$$

$$F_y = -mg + F_z \cdot \cos \varphi = \frac{d^2}{dt^2} v \cdot m$$

$$F_x = -F_z \cdot \sin \varphi = \frac{d^2}{dt^2}(x_p + u) \cdot m = \frac{d^2}{dt^2}(x_p \cdot m - l \cdot \sin \varphi \cdot m) \quad (3.4)$$

$$F_y = -mg + F_z \cdot \cos \varphi = \frac{d^2}{dt^2} v \cdot m = \frac{d^2}{dt^2} l \cdot \cos \varphi \cdot m$$

Odvození derivací:

$$\frac{d^2}{dt^2}(x_p \cdot m - l \cdot \sin \varphi \cdot m) = \frac{d}{dt}(\dot{x}_p \cdot m - l \cdot \cos \varphi \cdot m \cdot \dot{\varphi}) = \ddot{x}_p \cdot m + l \cdot m \cdot \sin \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \cos \varphi \cdot \ddot{\varphi}$$

$$\frac{d^2}{dt^2} l \cdot \cos \varphi \cdot m = \frac{d}{dt}(-l \cdot m \cdot \sin \varphi \cdot \dot{\varphi}) = -l \cdot m \cdot \cos \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \sin \varphi \cdot \ddot{\varphi} \quad (3.5)$$

Odtud plyne:

$$F_x = -F_z \cdot \sin \varphi = \ddot{x}_p \cdot m + l \cdot m \cdot \sin \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \cos \varphi \cdot \ddot{\varphi} \quad (3.6)$$

$$F_y = -mg + F_z \cdot \cos \varphi = -l \cdot m \cdot \cos \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \sin \varphi \cdot \ddot{\varphi}$$

Po úpravě (vynásobení první rovnice $\cos \varphi$ a druhé $\sin \varphi$):

$$-F_z \cdot \sin \varphi \cdot \cos \varphi = \ddot{x}_p \cdot m \cdot \cos \varphi + l \cdot m \cdot \sin \varphi \cdot \cos \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \cos^2 \varphi \cdot \ddot{\varphi} \quad (3.7)$$

$$F_z \cdot \sin \varphi \cdot \cos \varphi = mg \cdot \sin \varphi - l \cdot m \cdot \sin \varphi \cdot \cos \varphi \cdot \dot{\varphi}^2 - l \cdot m \cdot \sin^2 \varphi \cdot \ddot{\varphi}$$

Po sečtení obou rovnic:

$$0 = mg \cdot \sin \varphi + \ddot{x}_p \cdot m \cdot \cos \varphi - l \cdot m \cdot \ddot{\varphi} \quad (3.8)$$

Po úpravě:

$$l\ddot{\varphi} - g \sin \varphi - \ddot{x}_p \cos \varphi = 0 \quad (3.9)$$

Bude-li pohyb vozíku řízen jeho zrychlením, lze \ddot{x}_p považovat za vstup soustavy. V tomto případě je tedy systém tvořený vozíkem s kyvadlem popsán rovnicí:

$$l\ddot{\varphi} - g \sin \varphi - u \cos \varphi = 0 \quad (3.10)$$

Kde u označuje vstup systému. Tento model však předpokládá, že tření je zanedbatelné. Pokud tření z nějakého důvodu nelze zanedbat, je nutno jej vzít v úvahu připočtením členu $b\dot{\varphi}$, tak jak lze vidět v následující rovnici.

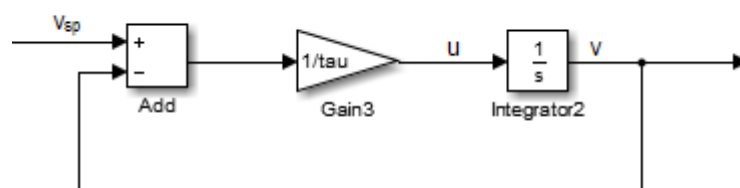
$$l\ddot{\varphi} - g \sin \varphi + b\dot{\varphi} - u \cos \varphi = 0 \quad (3.11)$$

Protože akční člen pro řízení inverzního kyvadla je stejnosměrný motor, řízený pomocí rychlostní smyčky, lze s dostatečnou přesností namodelovat vztah mezi požadovanou rychlostí vozíku a jeho skutečnou rychlostí.

Pro vyjádření tohoto vztahu lze využít přenos 1. řádu dle rovnice 3.15, jenž za předpokladu kvalitního a rychlého regulátoru bude aproximovat chování stejnosměrného motoru pomocí experimentálně stanovené časové konstanty τ .

$$G_v = \frac{1}{\tau s + 1} \quad (3.15)$$

Odpovídající vnitřní stavové schéma rychlostní smyčky zobrazuje obrázek 16.

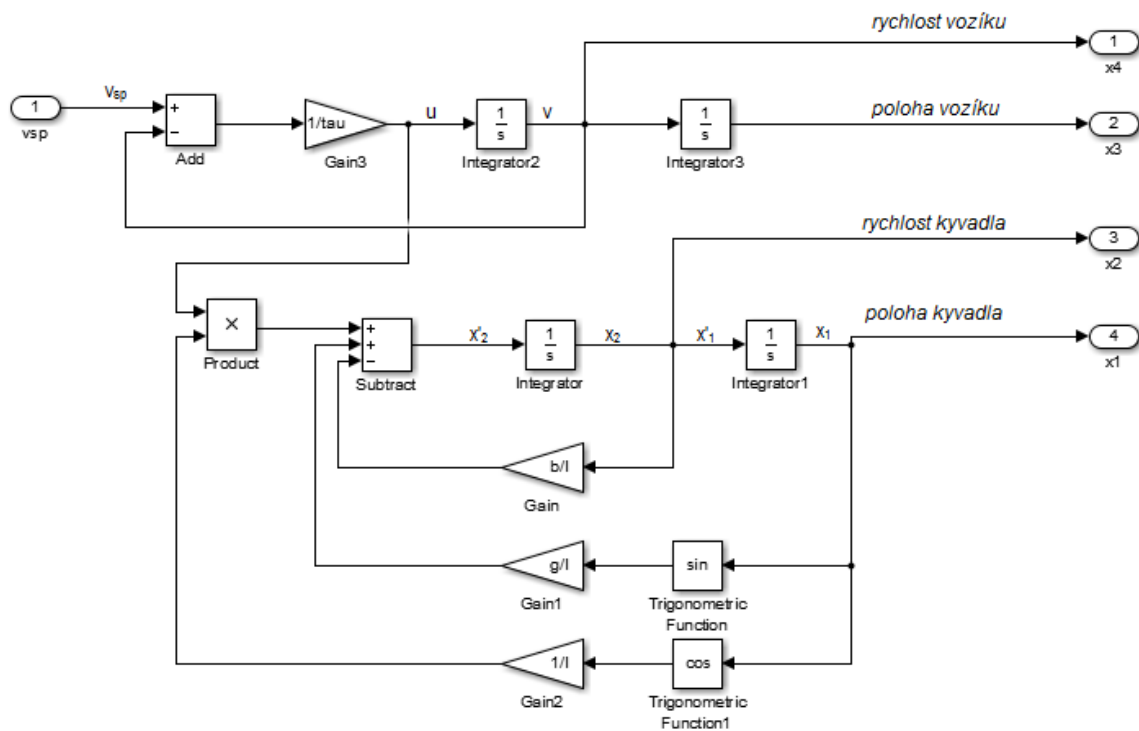


Obrázek 16: Vnitřní stavové schéma rychlostní smyčky

Vysvětlivky k obrázku 14:

v_{sp}	požadovaná rychlost
u	skutečné zrychlení
v	skutečná rychlost

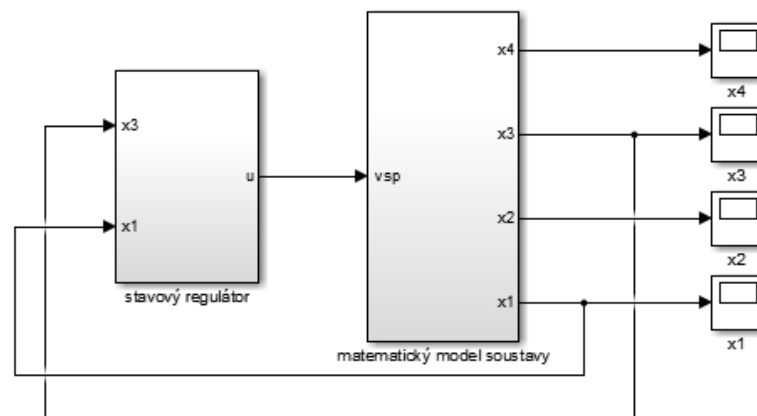
Akční člen lze prohlásit za součást celé soustavy a rozšířit stavový popis o rychlost a polohu vozíku podle obrázku 17.



Obrázek 17: Stavové schéma modelu inverzního kyvadla řízeného stejnosměrným motorem

Pro potřeby pozdějšího využití techniky PIL se provede úprava modelu s ohledem na to, že ve skutečnosti bude měřena poloha vozíku a ramene kyvadla, nikoliv však jejich rychlosti. Naměřené veličiny budou pak aproximovány pomocí numerických derivací podle rovnice 3.16.

$$\begin{aligned}
 x_2 &= \frac{dx_1}{dt} \approx \frac{\Delta x_1}{\Delta t} \\
 x_4 &= \frac{dx_3}{dt} \approx \frac{\Delta x_3}{\Delta t}
 \end{aligned}
 \tag{3.16}$$



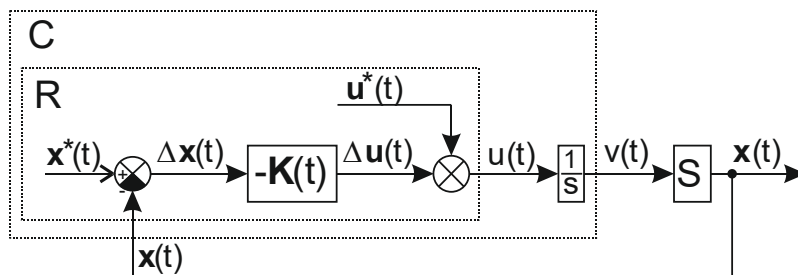
Obrázek 18: Schéma soustavy tvořené stavovým regulátorem a matematickým modelem

V reálných podmínkách bude stavový regulátor zpracovávat dvě naměřené polohy a dvě aproximované rychlosti.

4 Konceptce řízení modelu inverzního kyvadla

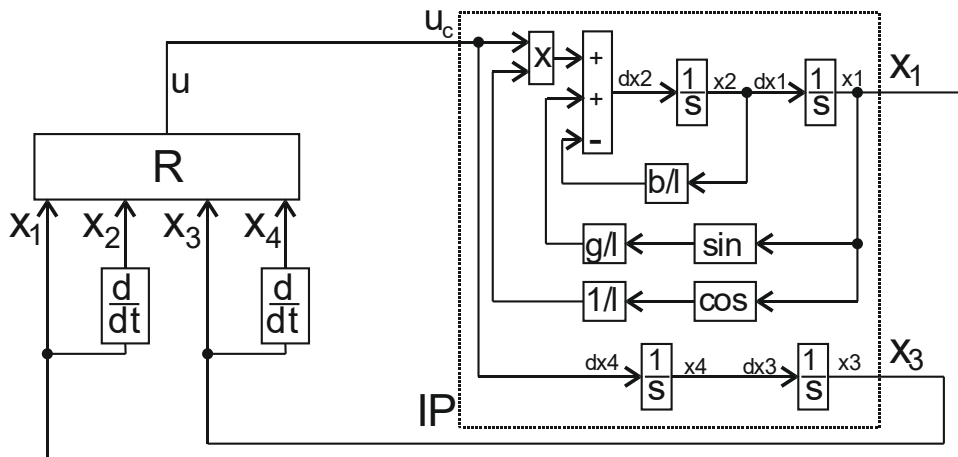
Následující kapitola se zabývá popisem konceptce řízení modelu inverzního kyvadla a byla vypracována na základě podkladů dodaných vedoucím bakalářské práce.

Z hlediska struktury a konceptce řízení byla použita struktura se dvěma stupni volnosti, tzv. „Two-degree of freedom design“ podle obrázku 19. Řízení je realizováno stavovým regulátorem $K(t)$ s časově proměnnými parametry vypočtenými jako řešení úlohy LQR na konečném horizontu. Podrobnější popis struktury řízení, založený na tomto návrhu je uveden na [4].



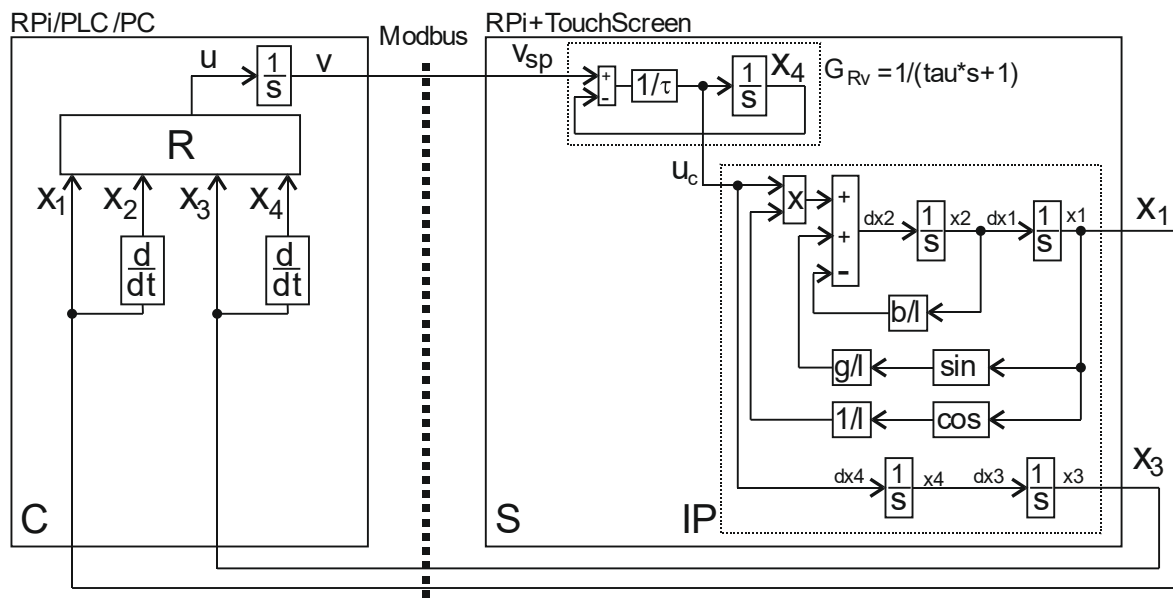
Obrázek 19: Struktura se dvěma stupni volnosti

Pro návrh LQR regulátoru R je použito následující schéma, které předpokládá, že se soustava IP řídí pomocí zrychlení. V reálných podmínkách je fyzikální veličinou působící na soustavu rychlost, nikoliv zrychlení, přičemž je použit DC motor. Dále je pro návrh stavového regulátoru uvažován předpoklad ideálního regulátoru rychlosti motoru. To znamená, že se přepokládá, že vozík je řízen takovou rychlostí, která způsobí předepsané zrychlení – viz obrázek 20.



Obrázek 20: Schéma pro návrh LQR regulátoru R

Navržený stavový regulátor se použije v následujícím zapojení (obrázek 21), které ukazuje řízení celé soustavy reprezentované blokem S , který lze dále rozložit na blok reprezentující model samotného inverzního kyvadla (IP) a model aktuátoru (reprezentovaný DC motorem a mechanickým příslušenstvím) včetně řízení rychlosti, popsany modelem 1. řádu s přenosem dle rovnice 3.15, kde hodnota τ byla stanovena experimentálně.



Obrázek 21: Zapojení stavového regulátoru

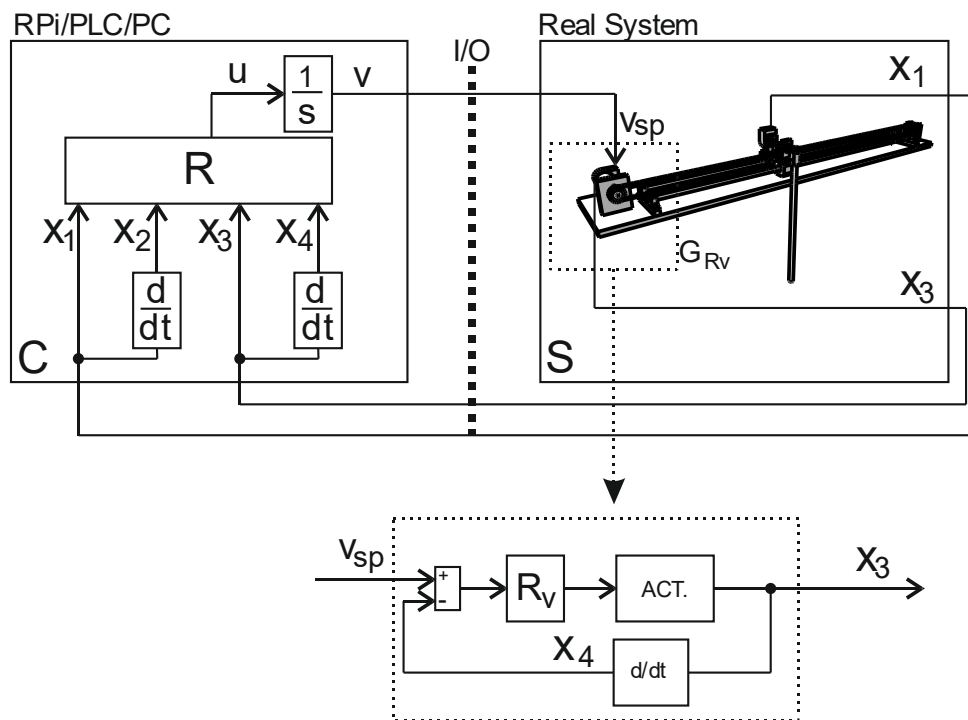
5 Návrh a realizace SIL a PIL simulátoru v prostředí REX

V kapitole dvě byly zmíněny některé druhy simulací, které se používají k návrhu regulátoru pro řízení soustavy. V této části bude matematický model soustavy aplikovaný na model pro SIL a PIL simulaci v prostředí REX.

5.1 Model regulátoru v prostředí REX

Stavový regulátor řízení použitý v této práci, jehož blokové schéma lze vidět v příloze II a III, byl dodán vedoucím bakalářské práce. Úkolem této práce bylo navázat na poskytnutý stavový regulátor řízení a vytvořit modely, které budou základem pro simulace typu SIL a PIL.

Za účelem přiblížení se reálným podmínkám, byl přenos 1. řádu nahrazen modelem rychlostní smyčky, ve které figuruje regulátor R navržený pro aktuátor ACT, viz obrázek 22. Aproximovaný model aktuátoru byl experimentálně identifikován tak, že do aktuátoru byl zaveden puls reprezentující střihu PWM signálu, byly odměřeny pulsy enkodéru a následně převedeny na polohu vozíku. Tato veličina byla numericky zderivována a tato data byla předložena identifikační metodě. Celkové blokové zapojení regulačního obvodu odpovídá obrázku 23.

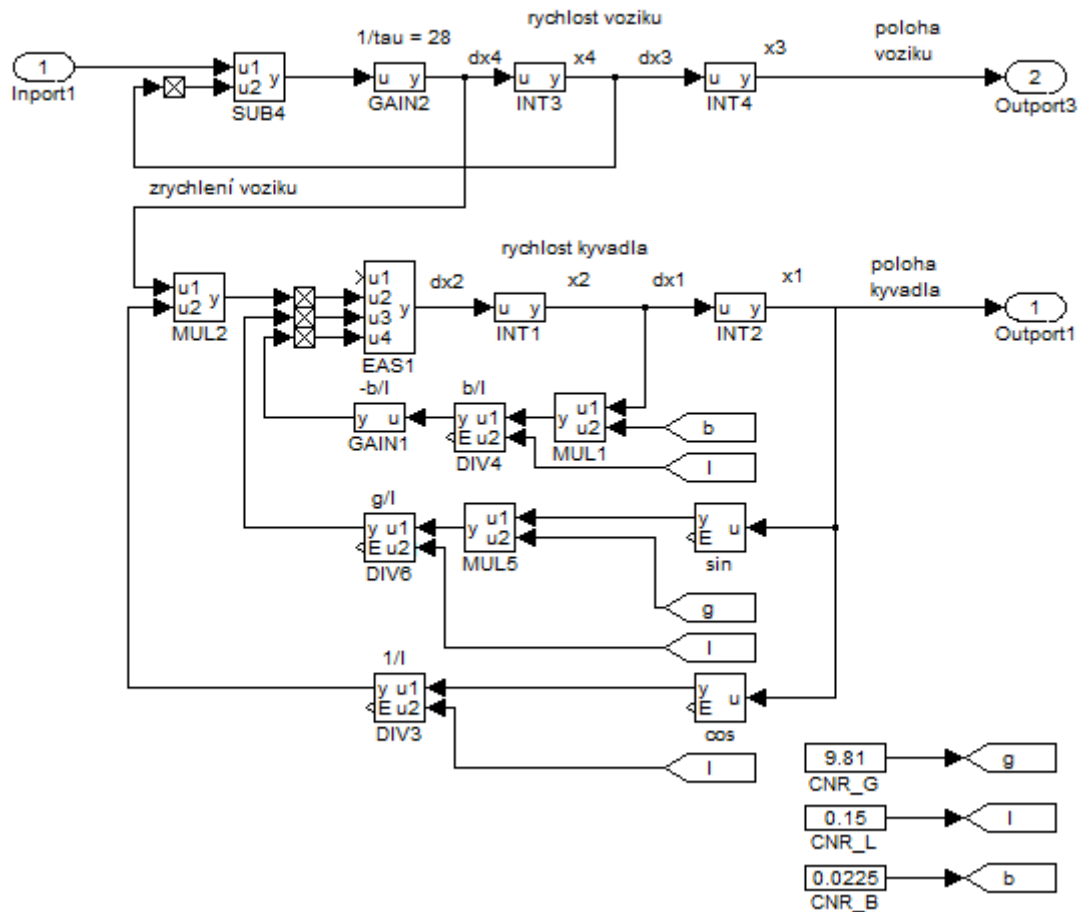


Obrázek 22: Celkové zapojení regulačního obvodu pro řízení reálného modelu

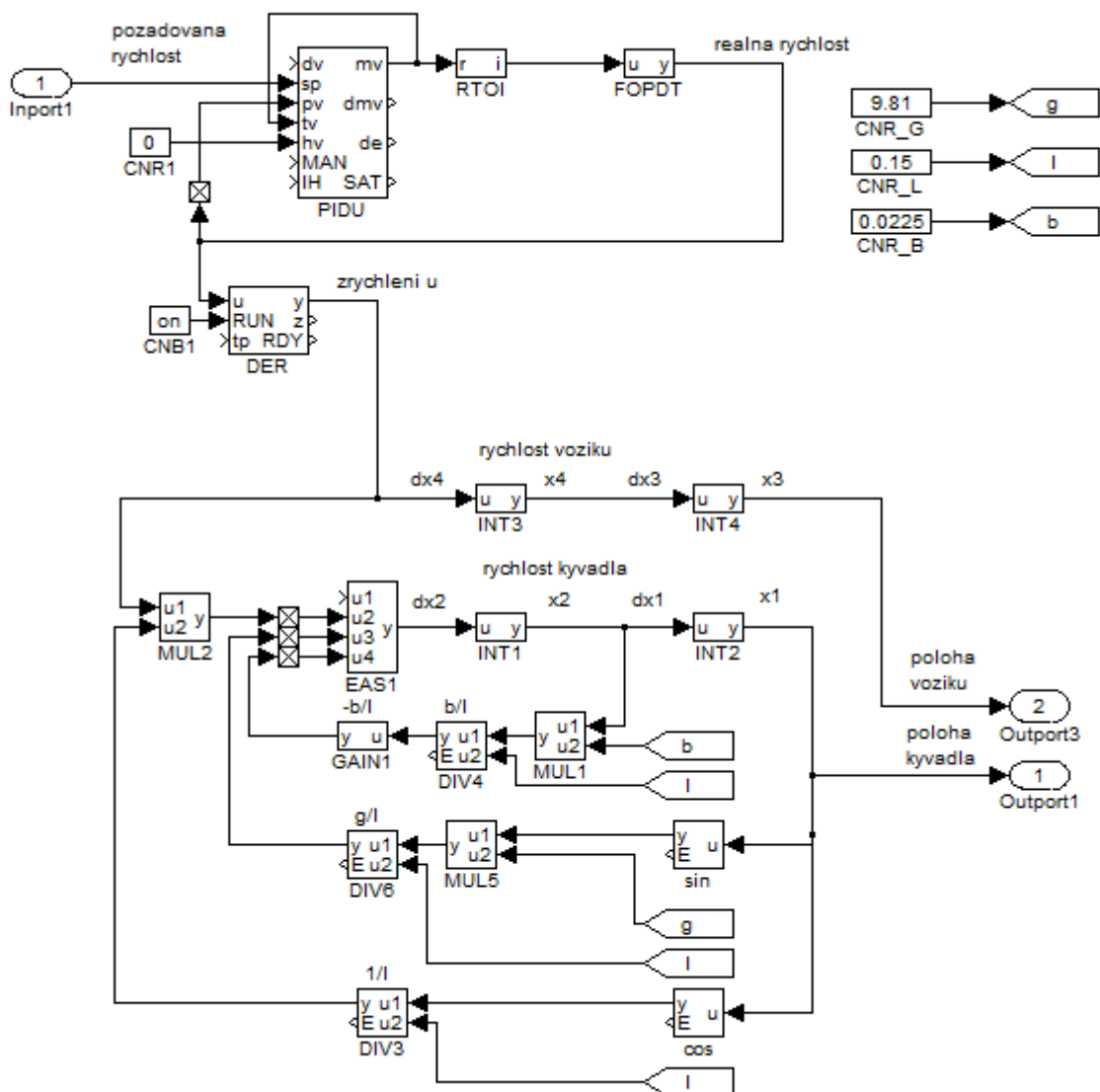
5.2 Model soustavy v prostředí REX

Model soustavy v prostředí REX (obrázek 22) je velice podobný modelu soustavy v prostředí Simulink z obrázku 16. Vstupem soustavy je požadovaná rychlost a výstupem jsou poloha vozíku a poloha kyvadla. Tyto dva měřené signály jsou společně s aproximovanými signály x_2 (rychlost vozíku) a x_4 (rychlost kyvadla) vstupem stavového regulátoru, který upravuje řídicí veličinu, kterou je opět požadovaná rychlost.

Jak již bylo uvedeno v předcházející kapitole, hodnota tau pro rychlostní smyčku byla stanovena experimentálně. Pro ještě větší přiblížení realitě byla rychlostní smyčka (původně aproximována přenosem 1. řádu, čemuž odpovídá schéma z obrázku 23) nahrazena PID regulátorem navrženým pro DC motor s přenosem 2. řádu dle obrázku 24.



Obrázek 23: Model soustavy s rychlostní smyčkou aproximovanou přenosem 1. řádu

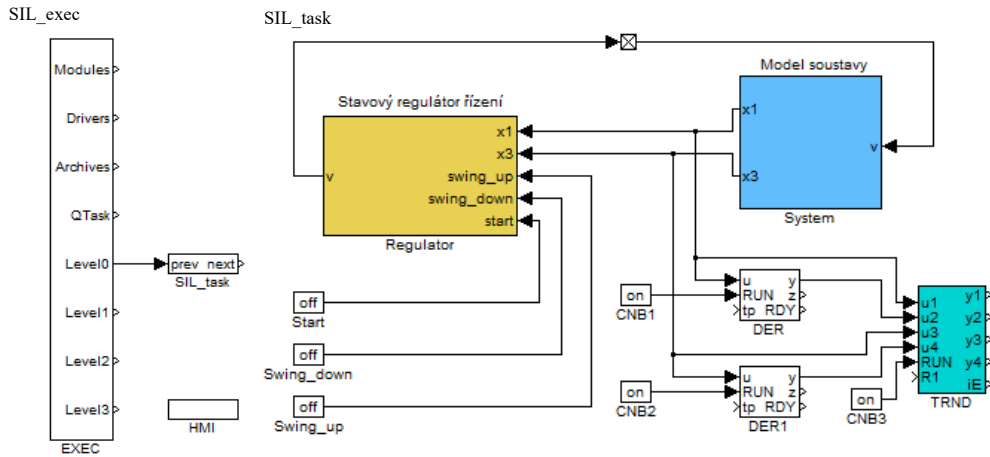


Obrázek 24: Model soustavy s PID regulátorem

5.2.1 SIL simulace pro inverzní kyvadlo

Jak již bylo uvedeno v předešlých kapitolách, u simulace SIL je algoritmus řízení i model soustavy spouštěn na stejném počítači. Stěžejními prvky tohoto typu simulace je regulátor (stavový regulátor řízení) a systém (model soustavy). Na obrázku 24 jsou vidět ještě další bloky, jako je TRND a DER.

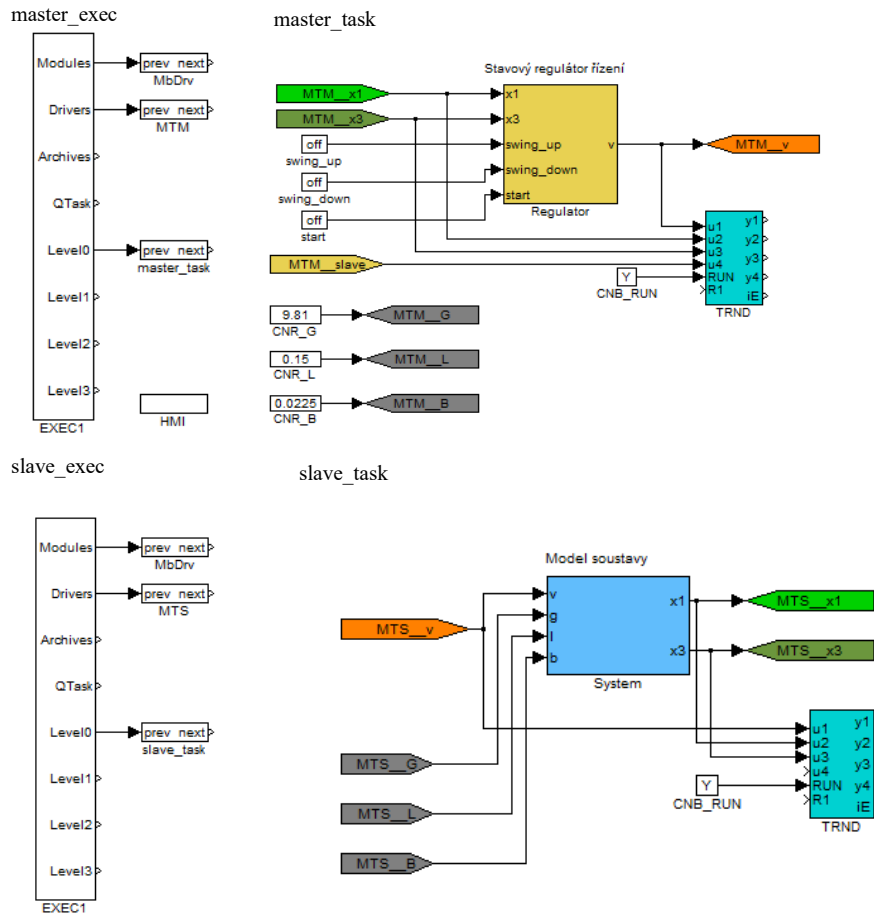
Blok TRND slouží pouze k zobrazení signálů a nemá žádný vliv na funkci systému. Blok DER je použit k derivaci signálu polohy vozíku a polohy kyvadla, čímž byla získána jejich aktuální rychlost.



Obrázek 25: Blokové schéma SIL simulace pro inverzní kyvadlo

5.2.2 PIL simulace pro inverzní kyvadlo

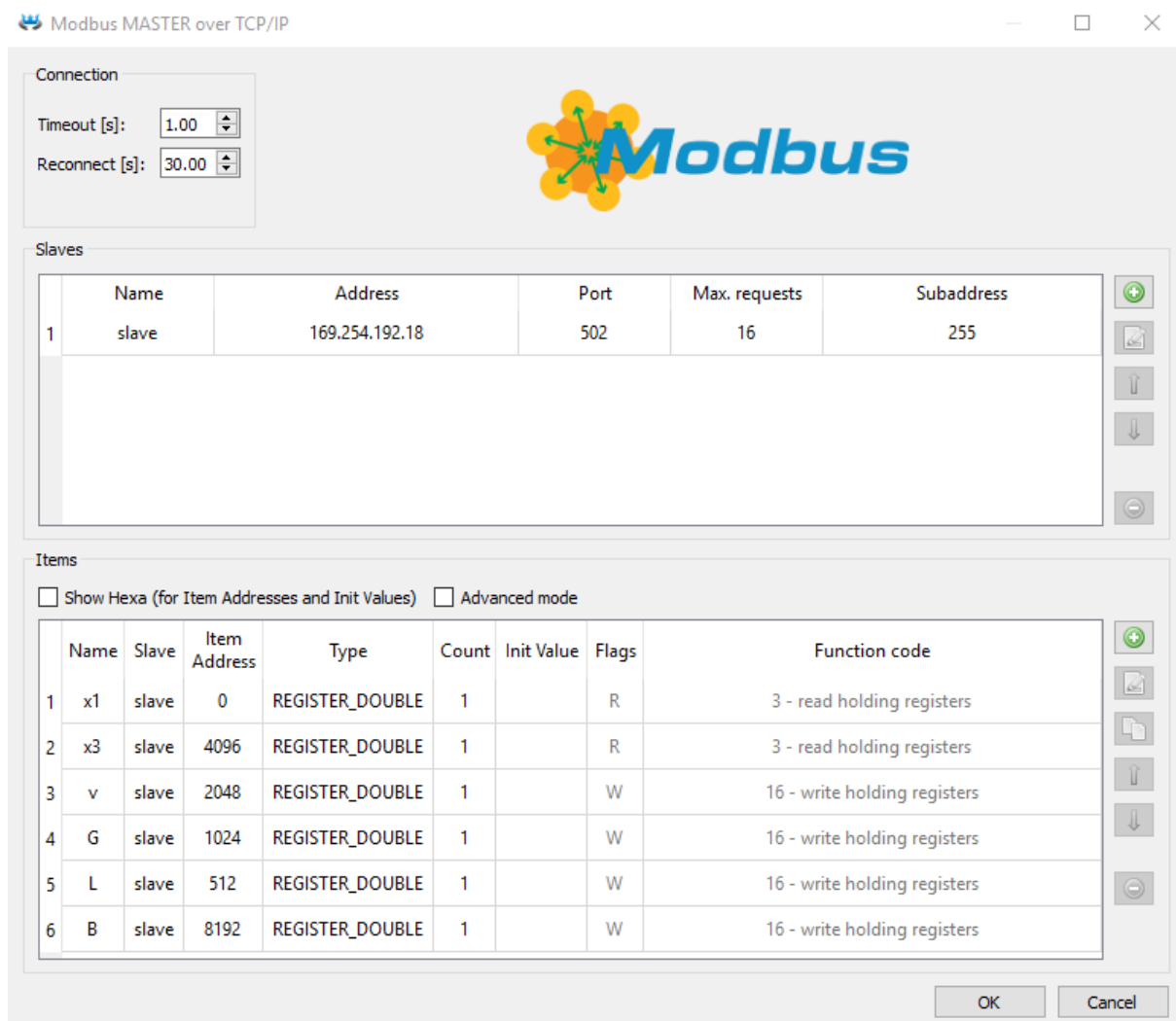
U Simulace typu PIL je důležité, že regulátor již pracuje na cílovém zařízení. Jedním z důležitých výstupů může být informace o tom, jaká je zapotřebí dráha pro kyvadlo o určitých parametrech, a to dokonce ještě před vyrobením samotného kyvadla. Tím se dá předejít zničení zařízení nebo dokonce případnému zranění. U PIL simulace je potřeba také nastavit parametry komunikace.



Obrázek 26: Blokové schéma PIL simulace inverzního kyvadla pro master a slave

Pro komunikaci regulátoru s modelem soustavy se využívá komunikačního protokolu Modbus, který je založených na principu předávání dat mezi klientem a serverem. Komunikace probíhá přes ethernetové rozhraní a lze ji nastavit jednoduše přímo v REX designeru (obrázek 27).

Wi-fi rozhraní se jevílo jako nevhodný způsob komunikace vzhledem k dlouhé době odezvy. Regulátor nebyl schopen dostatečně včas reagovat na změny soustavy a docházelo k rozkmitání celé soustavy.

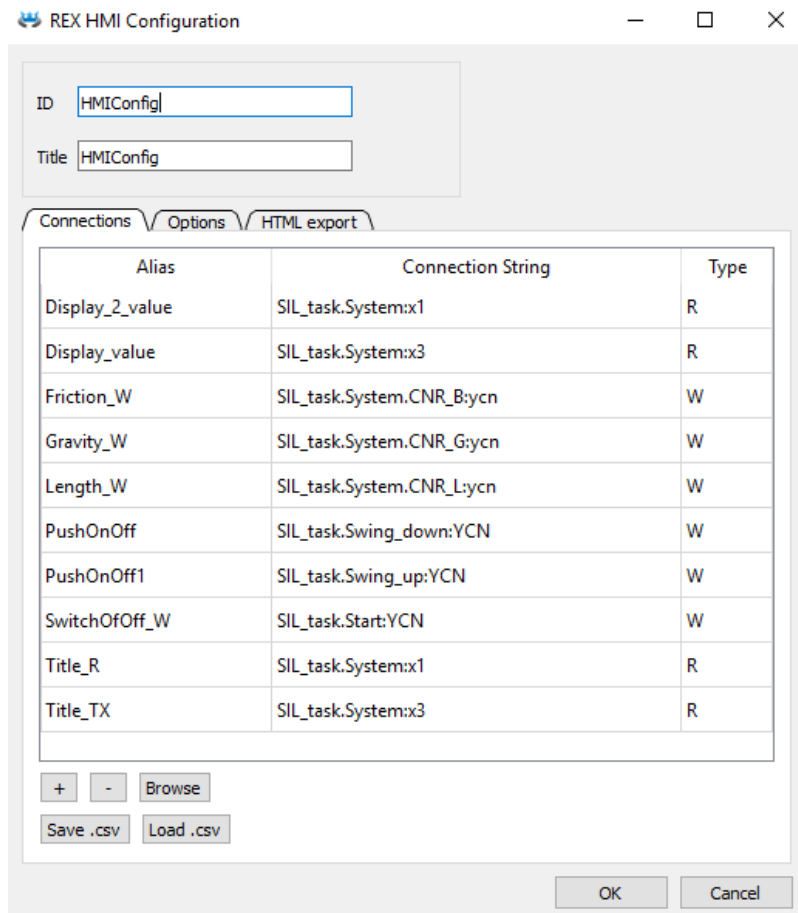


Obrázek 27: Nastavení komunikace prostřednictvím Modbus

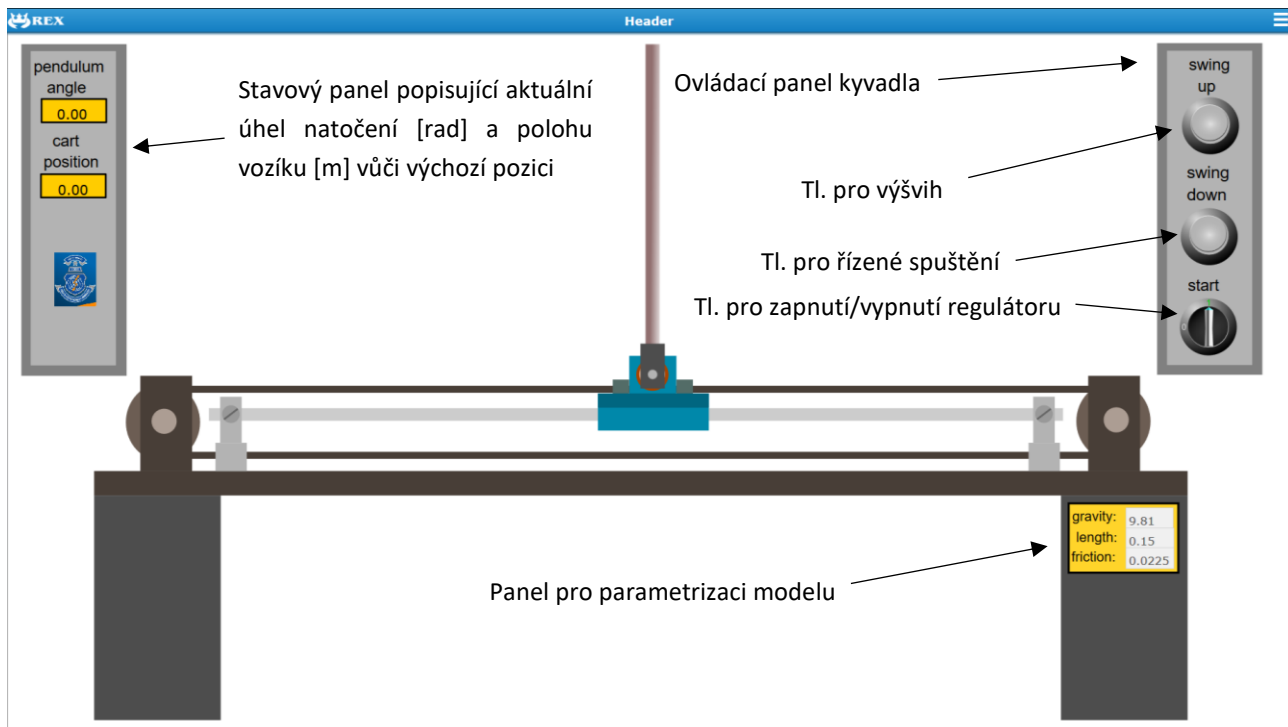
5.3 Vizualizace

Pro účely SIL a PIL simulace byla vytvořena vizualizace v nástroji HMI Designer, který pracuje na bázi Inkscape. Signál polohy ramene je navázán na rotační pohyb ramene kyvadla (rotate) a signál polohy vozíku je navázán na pohyb celého vozíku i s ramenem kyvadla v ose x (translate x). Podrobnější přiřazení proměnných, pro simulaci typu PIL lze vidět na obrázku 28.

Pro ovládání výšvihu a spuštění kyvadla pak slouží tlačítka v postranním panelu. Aktuální hodnotu polohy vozíku a ramene zobrazuje displej na levé straně vizualizace. Celková vizualizace kyvadla, včetně jeho popisu, je zobrazena na obrázku 29.



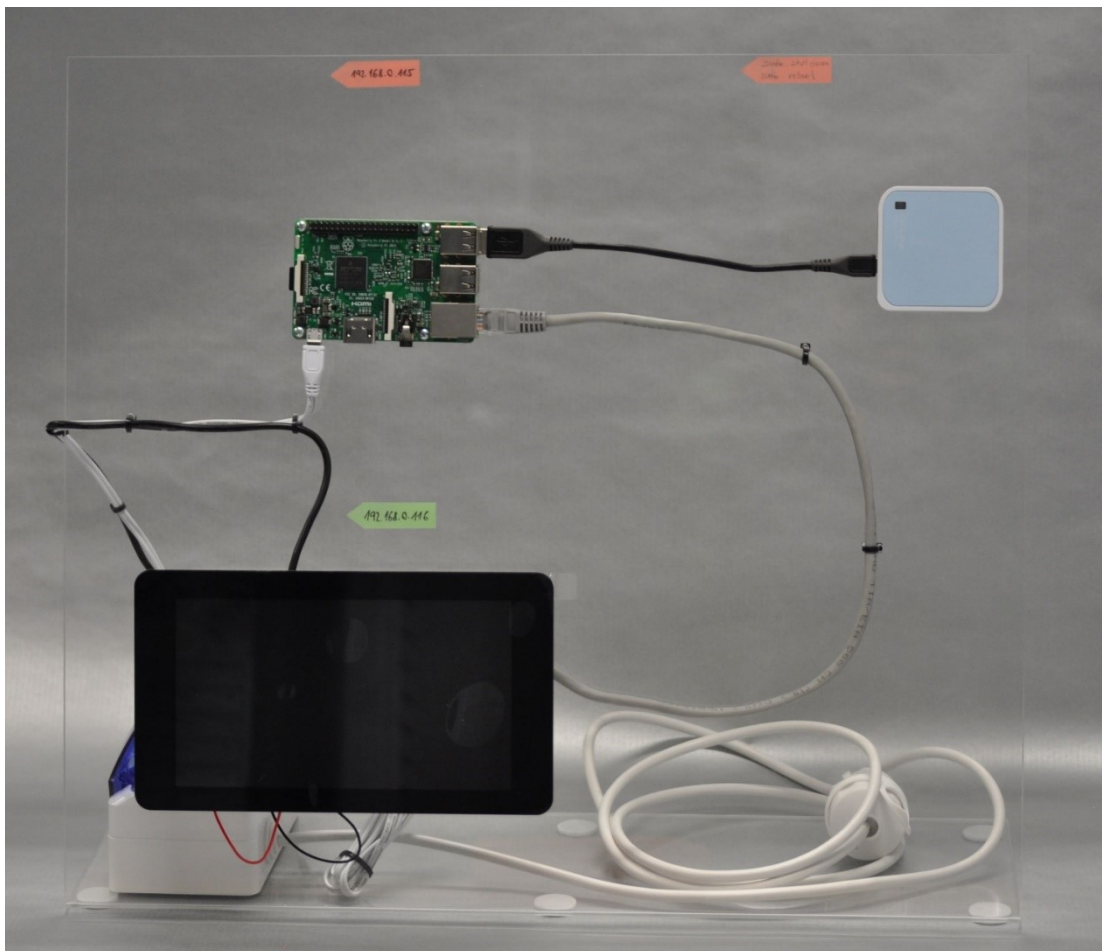
Obrázek 28: Ukázka přiřazení proměnných



Obrázek 29: Ukázka vizualizace inverzního kyvadla

5.4 Realizace testovacího stojanu

Pro praktickou ukázkou PIL simulace byl vytvořen výukový stojan z plexiskla. Na stojanu jsou umístěny dva mikrokontrolery Raspberry Pi 3, 7“ dotykový display a Wi-fi router TP-Link TL-WR702N, který slouží ke snadnější komunikaci mezi modelem a programátorem. Na dotykovém displeji lze zobrazit vizualizaci kyvadla a stejně tak lze provést příkaz k vyšvihnutí nebo řízenému spuštění kyvadla. Fotografie praktické realizace lze vidět na obrázku 30.



Obrázek 30: Ukázka praktického řešení PIL simulátoru

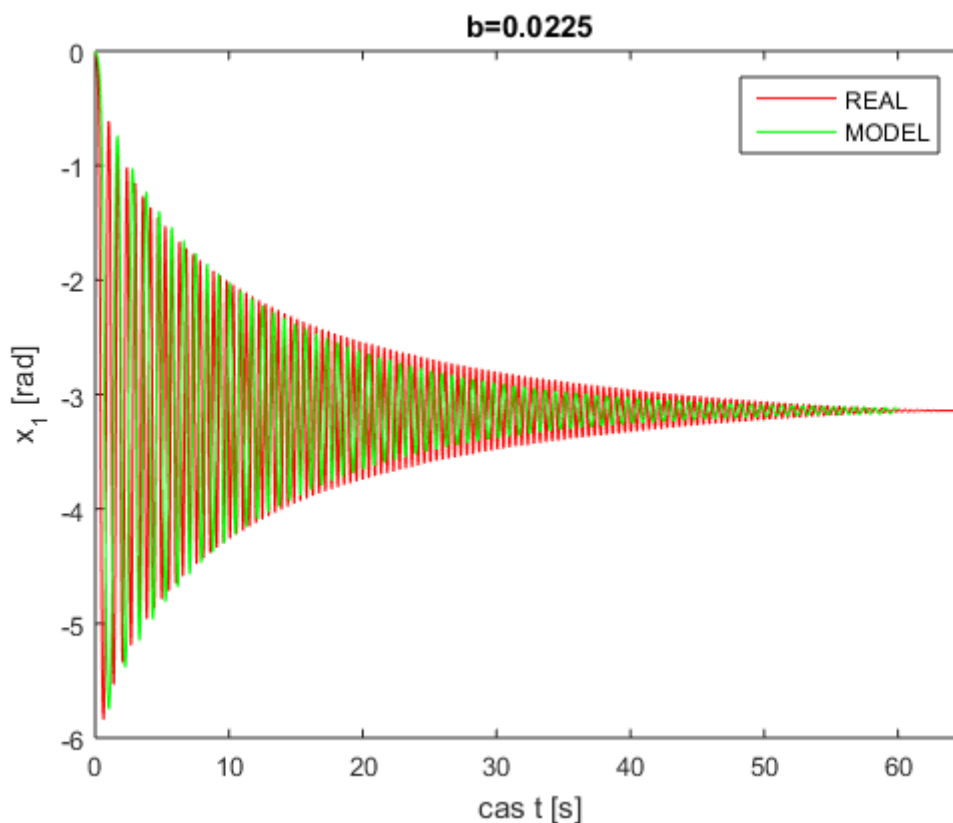
6 Ověření činnosti SIL a PIL simulátoru

Ověření činnosti SIL a PIL simulátoru spočívá v porovnání získaných dat z reálného měření s daty získané SIL a PIL simulací. Za stavové veličiny kyvadla se v tomto případě považují poloha ramene (x_1), rychlost ramene (x_2), poloha vozíku (x_3) a rychlost vozíku (x_4). Ze všeho nejdřív však bylo potřeba určit koeficient tření b .

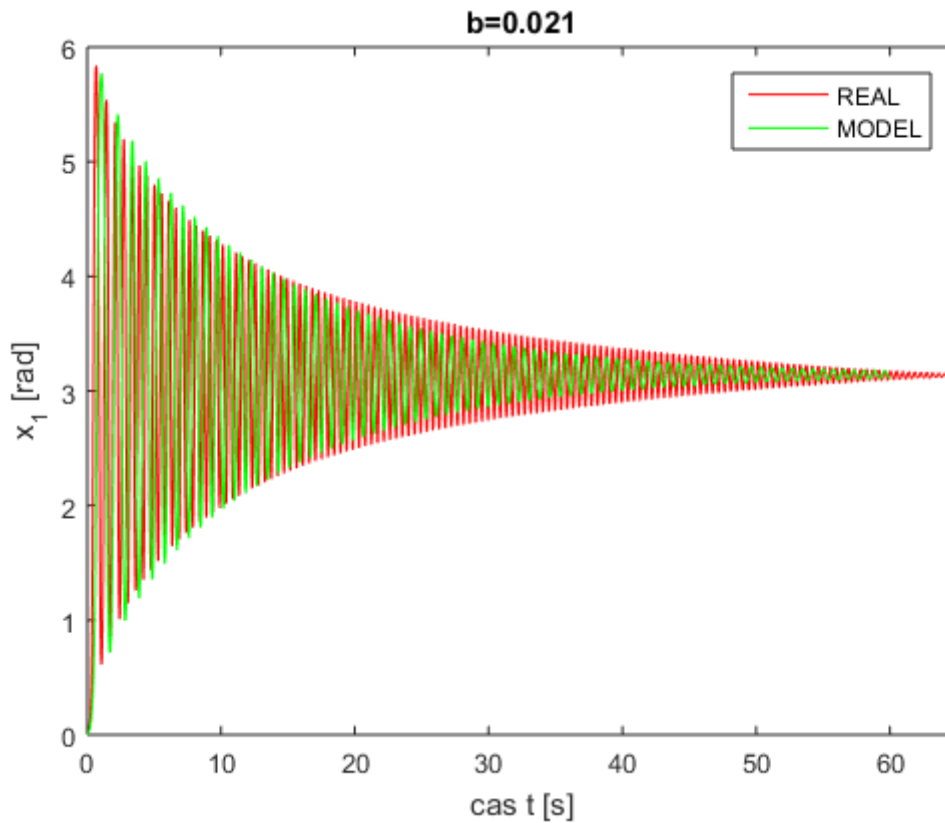
6.1 Určení koeficientu tření b

Pro potřeby realizace modelu soustavy bylo nutné co nejpřesněji určit koeficient tření b . Ten byl určen experimentálně a jako nejadekvátnější hodnota se jevila pro $b = 0.0225$. Na obrázku 31 a 32 lze vidět obálky signálů (zelené) pro dva různé koeficienty. Čím je hodnota b menší, tím menší je strmost signálu. Jinými slovy, čím bude koeficient útlumu větší, tím rychleji dojde k útlumu kmitání. Červená křivka zobrazuje data získaná z reálného měření.

Při detailním zkoumání křivky lze vidět, že tzv. pseudoperiodu kmitů lze stanovit velmi obtížně, což souvisí s tím, že model i reálné měření by mělo začínat teoreticky od nulového úhlu, prakticky od určité velmi malé počáteční hodnoty úhlu, který je ve skutečnosti téměř nemožné nastavit stejně pro model i reálný pokus. Zároveň, tato počáteční hodnota velmi ovlivňuje další průběh, proto zelená a červená křivka na sobě nikdy nebudou ležet, a dokonce jsou někdy i v proti fázi. Nicméně pro stanovení hodnoty koeficientu útlumu je důležitá zejména její obálka.



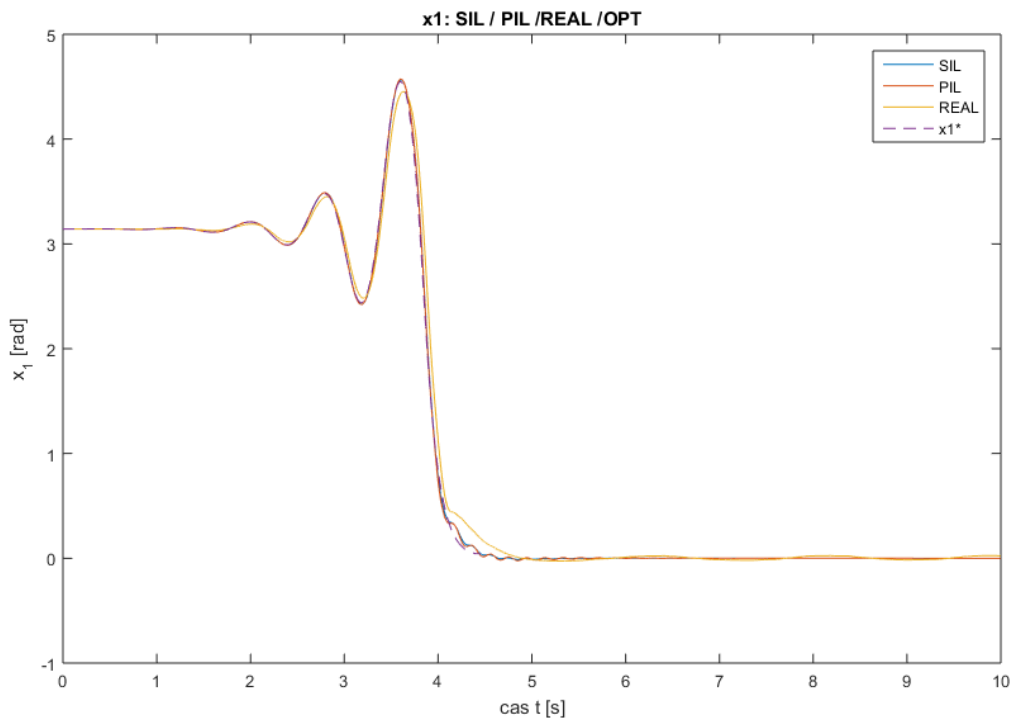
Obrázek 31: Neřízené spuštění kyvadla pro $b = 0.0225$



Obrázek 32: Neřízené spuštění kyvadla pro $b = 0.0210$

6.2 Ověření výšvihu a regulace v horní poloze

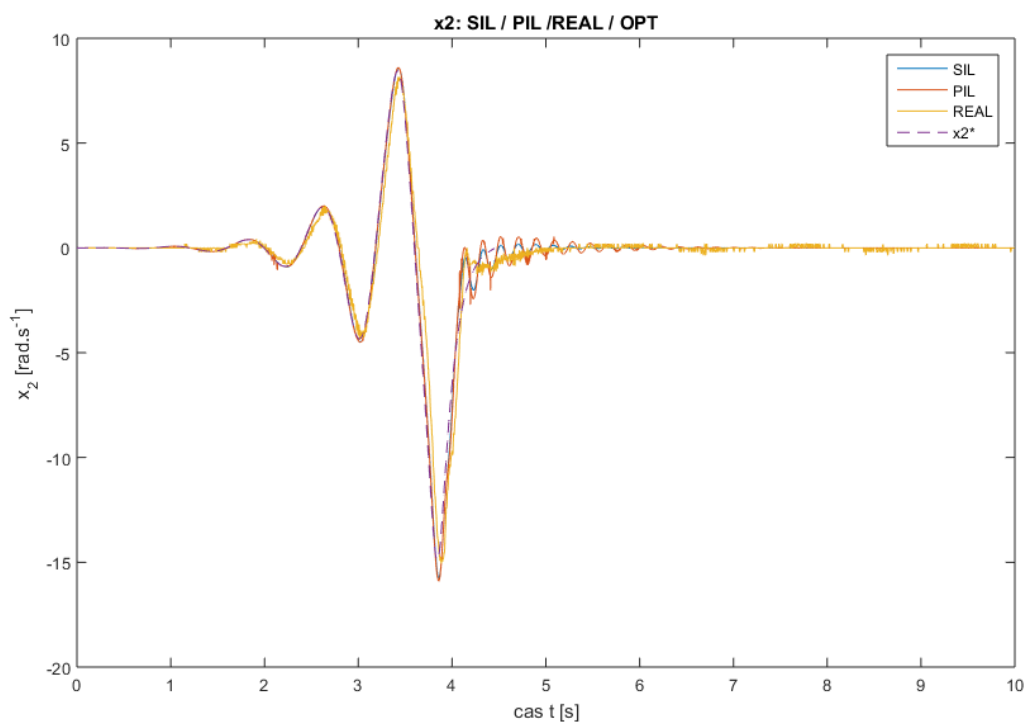
Pro všechny stavové veličiny byla vedoucím práce vypočtená hodnota optimálního průběhu veličin. Tyto veličiny, označeny symbolem $(x1^*, x2^*, x3^*, x4^*)$, byly vypočteny speciální metodou založenou na numerických simulacích systémů a na metodě časové symetrie systému upraveného pro systémy se třením. Metoda se však v současné době stále rozvíjí a dosud nebyla publikována.



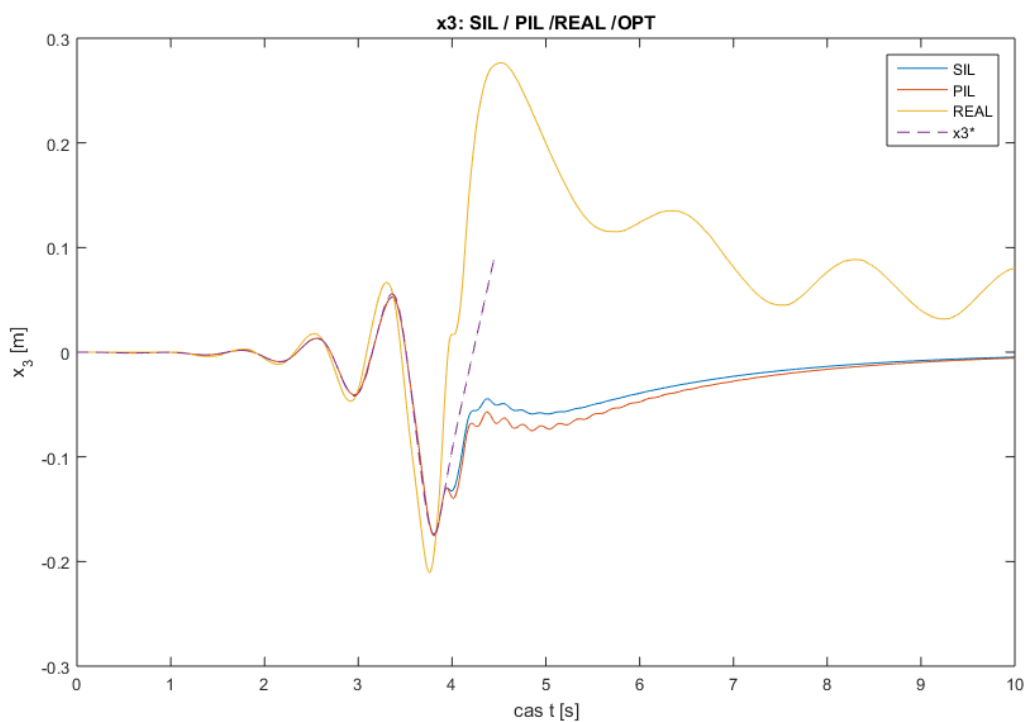
Obrázek 33: Srovnání x_1 (poloha ramene)

Při porovnání průběhů polohy ramene a jeho rychlosti (obrázek 29 a 30), lze konstatovat velice dobrou adekvátnost modelu, jelikož průběhy těchto veličin jsou téměř identické. Naopak u polohy vozíku a jeho rychlosti (obrázek 31 a 32) lze vidět, že simulované průběhy se liší od reálného průběhu. Tato skutečnost je způsobena faktory, jenž souvisí s fyzikální realitou:

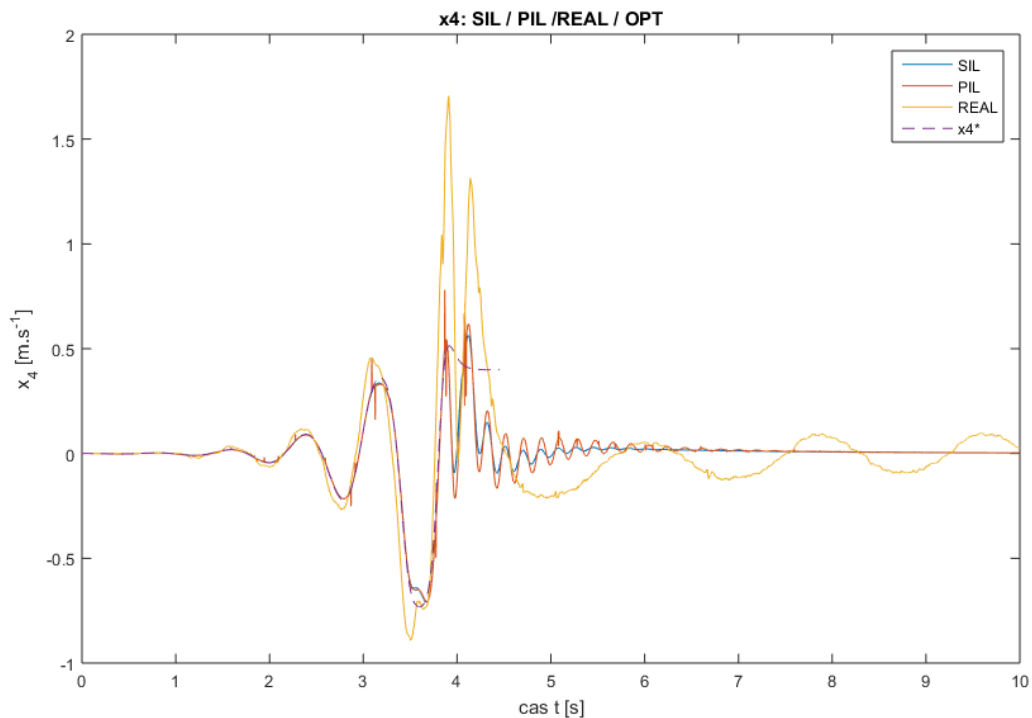
- veličina x_3 je nesrovnatelně citlivější na odchylky od optimálních trajektorií. To znamená, že například velmi malá odchylka ramene (x_1) od vertikální polohy způsobí výraznou odchylku na poloze vozíku x_3 .
- proměnné tření
- nelinearita motoru
- reálný rychlostní regulátor



Obrázek 34: Srovnání x_2 (rychlost ramene)



Obrázek 35: Srovnání x_3 (poloha vozíku)



Obrázek 36: Srovnání x_4 (rychlost vozíku)

Optimální řízení a optimální trajektorie $x_1^* - x_4^*$ byly vypočteny do $T_f = 4.45$ s. Použitá metoda výpočtu výšvihu vede na takové optimální řízení a takové optimální trajektorie, jejichž konečná hodnota v čase T_f je blízká nule pro x_1 a x_2 , nikoliv však pro x_3 a x_4 . Protože z hlediska koncepce řízení, tak jak bylo uvedeno v kapitole 4, se musí všechny stavové veličiny ustálit na nulové hodnotě, lze pro čas $t > T_f$ předepsat optimální trajektorie na nulové hodnoty.

Teoreticky by se předepisování trajektorií na nulovou hodnotu mělo nastartovat právě v čase T_f , jelikož hodnoty x_1 a x_2 v tomto čase jsou přibližně nulové. V reálných podmínkách je T_f nastavena o něco níže, cca na 4.046 s, jelikož u reálného průběhu by pro teoretickou hodnotu $T_f = 4.45$ s přesáhla veličina x_3 fyzikální mez danou konstrukcí kyvadla – dráhu, která je k dispozici.

Z průběhu x_3 lze vidět mírné odchylky po celou dobu výšvihu. Přibližně v čase 3.92 s, kde x_1 nabývá hodnoty $\frac{\pi}{2}$, se soustava nachází v tzv. „singulárním bodě“, tedy v okamžiku, kde nelze žádným akčním zásahem u ovlivnit polohu ramene (x_1). V tomto okamžiku se rameno nachází vodorovně a pohybem vozíku nelze dosáhnout změny úhlu ramene. Tato skutečnost má vliv na další průběh, protože v tomto okamžiku a jeho blízkém okolí, stavový regulátor pracuje s veličinami x_3 a x_4 , které samotné jsou v regulátoru jako nestabilní (odchylka od požadované hodnoty se zvětšuje, když nejsou zapojeny další stavy), proto od tohoto okamžiku dochází k výraznému odchýlení od optimální trajektorie. Regulátor se teoreticky snaží předepsat nekonečně velký akční zásah, ale vlivem použité metody výpočtu výšvihu, respektující omezené řízení, je hodnota akčního zásahu konečná. V kombinaci s ostatními uvedenými faktory, které se týkají fyzikální reality, dochází na veličině x_3 k rozdílu oproti očekávání.

Rozdíl v průběhu x_3 mezi modelem a realitou však není zásadní pro návrh stavového regulátoru ani pro výpočet výšvihu. Nenarušuje koncepci PIL modelu, který primárně slouží k ověření výšvihu ramene což v tomto případě funguje velice dobře.

Adekvátnost modelu by se zlepšila zejména těmito opatřeními:

- použití výkonnějšího motoru (přiblížení se ideálnímu regulátoru rychlosti)
- použití jiných optimálních trajektorií $x_1^* - x_4^*$, které by končily na nulové hodnotě
- zavedením nové stavové proměnné x_5 , která by byla ve významu integrálu z dráhy vozíku (x_3), a která by se penalizovala v účelové funkci při výpočtu LQR regulátoru

Za těchto okolností by se dal vytvořený PIL model použít v módu, kdy by nejen ověřoval výšvih a regulaci ramene, ale i předikoval chování vozíku a vypočítával konstrukční parametry jiného modelu inverzního kyvadla. Například při zadání nové délky ramene by tento model odpověděl na otázku, jakou dráhu je nutné mít k dispozici pro výšvih takového kyvadla.

7 Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat SIL a PIL simulátor pro výukový fyzikální model inverzního kyvadla, seznámit se s řídicím systémem REX, se zaměřením na platformu Raspberry Pi, a dále seznámit se s technikami real-time modelování. Dalším úkolem bylo vytvořit vizualizaci pro oba typy simulací, a nakonec ověřit činnosti SIL a PIL simulátoru s reálnými daty naměřenými na fyzikálním modelu regulované soustavy.

Úvodní část této práce je věnovaná popisu systému REX, který slouží pro návrh a realizaci komplexních algoritmů automatického řízení. Je zde však uveden i popis mikrokontroléru Raspberry Pi 3, který byl použit pro účely simulace typu PIL.

Kapitola, která následuje po úvodní části se zabývá problematikou modelováním a simulací. Jsou zde popsány jednotlivé typy simulací, které byly aplikovány na motivační příklad, na němž jsem se snažil přiblížit více tuto problematiku. Jako motivační příklad bylo zvoleno sériové zapojení dvou RC článků.

K sestavení modelu pro účely simulace bylo potřeba nejprve vytvořit matematický model fyzikálního modelu inverzního kyvadla, čemuž je věnována kapitola 3. V této kapitole je ukázáno odvození modelu a následné sestavení stavového popisu inverzního kyvadla, které bylo poté využito pro sestavení modelu pro simulaci typu SIL a PIL.

Aby model mohl úspěšně fungovat bylo potřeba navrhnout a vytvořit stavový regulátor řízení, což ovšem nebylo cílem této práce, nicméně algoritmus řízení byl dodán vedoucím bakalářské práce. Z důvodu komplexního pojetí této práce jsem zde zařadil i kapitolu, která se zabývá právě koncepcí řízení inverzního kyvadla.

V praktické části bakalářské práce jsem se věnoval již samotnému návrhu a realizaci SIL a PIL simulátoru pro inverzní kyvadlo. Jako vývojové prostředí, pro oba typy simulace byl zvolen řídicí systém REX, který dokáže pracovat na mikrokontroleru Raspberry Pi. K oběma modelům byla v prostředí Inskcape vytvořena vizualizace, která názorně zobrazuje stavy jednotlivých stavových veličin pomocí vizuálního modelu inverzního kyvadla na vozíku, přičemž chování této vizualizace odpovídá chování reálného modelu. Tato vizualizace je do jisté míry interaktivní, jelikož umožňuje parametrizaci modelu pro konstanty b , l , g čímž umožňuje testovat příslušné algoritmy řízení.

Pro účely simulace typu PIL byl vytvořen výukový model, který je tvořen jak mikrokontrolérem regulátoru, tak mikrokontrolérem pro matematický model soustavy. Model byl vybaven 7“ dotykovým displejem na kterém lze zobrazit a ovládat vizualizaci inverzního kyvadla.

V závěru práce jsem pak provedl porovnání stavových veličin získaných z jednotlivých simulací s daty získanými z reálného měření. Při posouzení průběhů polohy a rychlosti ramene, lze konstatovat, že adekvátnost modelu byla velice dobrá, protože průběhy veličin, získaných simulací prakticky kopírují průběhy veličin z reálného měření a z vypočteného průběhu optimální trajektorie. Poněkud menší adekvátnost platila pro polohu a rychlost vozíku.

Tento fakt je zapříčiněn vícero faktory, které jsou podrobněji popsány v kapitole 6, nicméně daly by se shrnout do zhruba čtyř bodů. Jedna z příčin, proč tomu tak je, je skutečnost, že tření, které je u reálného modelu není statické, ale dynamické. Ve skutečnosti se také pracuje s reálným rychlostním regulátorem, který má omezený výkon, dále motor použitý v reálném modelu vykazuje nelinearitru a poslední bod se

týká výrazné citlivosti polohy vozíku na změnu polohy kyvadla. To znamená, že na malou odchylku polohy ramene reaguje regulátor výraznou změnou polohy vozíku.

Typické použití vytvořeného modelu PIL může být následující. Uživatel si zvolí libovolnou cílovou platformu (může se jednat o PLC, Raspberry Pi, PC apod.) na níž bude realizováno řízení. Bude-li mít k dispozici vypočtené trajektorie, může celkem snadno ověřit funkčnost takového řešení. Model tedy odpovídá na otázku, zda se podaří výšvih a jaká je potřebná dráha vozíku pro výšvih.

Literatura

- [1] [WSC '98]. ED. BY D. J. MEDEIROS .. *1998 Winter Simulation Conference proceedings: December 13-16, 1998, Washington, D.C.* New York [u.a.]: Association for Computing Machinery, 1998. ISBN 0780351347.
- [2] *About SIL and PIL Simulations* [online]. MathWorks, 2015 [cit. 2017-01-14]. Dostupné z: <https://lost-contact.mit.edu/afs/cs.stanford.edu/pkg/matlab-r2015b/matlab/r2015b/help/ecoder/ug/about-sil-and-pil-simulations.html>
- [3] *Automatizace a automatizační technika: systémové pojetí automatizace.* Brno: Computer Press, 2014. ISBN 978-80-251-3628-7.
- [4] *Design Two Degree of Freedom PID Controllers* [online]. MathWorks, 2017 [cit. 2017-03-29]. Dostupné z: https://de.mathworks.com/help/slcontrol/ug/designing-two-degree-of-freedom-pid-controllers.html?requestedDomain=www.mathworks.com#bsg5g_b
- [5] KRÁL, Petr. *Regulace Inverzního kyvadla*, 2012. Diplomová práce (Ing.). Vysoká škola báňská – Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky. Katedra kybernetiky a biomedicínského inženýrství, 2012-06-06.
- [6] MUGHAL, Asif Mahmood. *Real time modeling, simulation and control of dynamical systems.* New York, NY: Springer Science Business Media, 2016. ISBN 978-331-9339-054.
- [7] OZANA, Stepan, Martin PIES a Radovan HAJOVSKY. *Computation of Swing-up Signal for Inverted Pendulum Using Dynamic Optimization* [online]. s. 301 [cit. 2017-01-15]. DOI: 10.1007/978-3-662-45237-0_29. Dostupné z: http://link.springer.com/10.1007/978-3-662-45237-0_29
- [8] *Raspberry Pi 3 Model B – Newest Version* [online]. Bells Yew Green, 2016 [cit. 2016-12-30]. Dostupné z: <https://www.pi-supply.com/product/raspberry-pi-3-model-b-newest-version/>
- [9] Raspberry Pi 3 Model B – Raspberry Pi. *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi* [online]. Cambridge: Raspberry Pi Foundation, 2016 [cit. 2016-12-30]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [10] RaspbianAbout – Raspbian. *FrontPage – Raspbian* [online]. 2016 [cit. 2016-12-30]. Dostupné z: <https://www.raspbian.org>
- [11] REX pro Raspberry Pi. *REX Controls – Pokročilá automatizace, měření a regulace* [online]. Plzeň: REX Controls, 2016 [cit. 2016-12-30]. Dostupné z: <https://www.rexcontrols.cz/ridici-system-rex-raspberry-pi>
- [12] Řídicí systém REX pro pokročilé řízení strojů a procesů. *REX controls* [online]. 2016 [cit. 2016-12-28]. Dostupné z: http://www.rexcontrols.cz/media/documents/promotion/Rex_prospekt.pdf
- [13] Řídicí systém REX. *REX controls* [online]. 2016 [cit. 2016-12-28]. Dostupné z: <https://www.rexcontrols.cz/rex>
- [14] Simulace MIL – Model in the loop. *REX controls* [online]. Plzeň: REX Controls [cit. 2017-01-14]. Dostupné z: <https://www.rexcontrols.cz/simulace-mil>

- [15] Simulace PIL – Hardware in the loop. *REX controls* [online]. Plzeň: REX Controls [cit. 2017-01-15]. Dostupné z: <https://www.rexcontrols.cz/simulace-hil>
- [16] Simulace PIL – Processor in the loop. *REX controls* [online]. Plzeň: REX Controls [cit. 2017-01-15]. Dostupné z: <https://www.rexcontrols.cz/simulace-pil>
- [17] Simulace SIL – Software in the loop. *REX controls* [online]. Plzeň: REX Controls [cit. 2017-01-14]. Dostupné z: <https://www.rexcontrols.cz/simulace-sil>
- [18] *Testen von Embedded Systems: Hardware in the Loop (HIL) Testing* [online]. In: KIRNER, Raimund. Wien: Technische Universität Wien, 2008, s. 15 [cit. 2017-01-14]. Dostupné z: https://ti.tuwien.ac.at/cps/teaching/courses/testing_emb_sys/Documents/old-lecture-slides/tes6_hil_testing.pdf/view
- [19] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi: uživatelská příručka*. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
- [20] *Začínáme se systémem REX na platformě Raspberry PI: Uživatelská příručka*. Verze 2.10.8. Plzeň: REX Controls, 2016.

Přílohy

Příloha I: Odvození matematického modelu systému druhého řádu.....	1
Příloha II: Stavový regulátor pro SIL v prostředí REX.....	3
Příloha III: Stavový regulátor pro PIL v prostředí REX	4
Příloha IV: CD-ROM disk	5

Příloha I: Odvození matematického modelu systému druhého řádu

Rovnice popisující obvod:

$$R_1 I_1 + \frac{1}{C_1 s} (I_1 - I_2) = U$$

$$\frac{1}{C_1 s} (I_1 - I_2) + R_2 I_2 + \frac{1}{C_2 s} I_2 = 0$$

$$Y = \frac{1}{C_2 s} I_2$$

Sestavení matic:

$$I_1 \left(R_1 + \frac{1}{C_1 s} \right) + I_2 \left(-\frac{1}{C_1 s} \right) = U$$

$$I_1 \left(-\frac{1}{C_1 s} \right) + I_2 \left(\frac{1}{C_1 s} + R_2 + \frac{1}{C_2 s} \right) = 0$$

$$\begin{bmatrix} R_1 + \frac{1}{C_1 s} & -\frac{1}{C_1 s} \\ -\frac{1}{C_1 s} & \frac{1}{C_1 s} + R_2 + \frac{1}{C_2 s} \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} U \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} R_1 + \frac{1}{C_1 s} & -\frac{1}{C_1 s} \\ -\frac{1}{C_1 s} & \frac{1}{C_1 s} + R_2 + \frac{1}{C_2 s} \end{bmatrix}^{-1} \cdot \begin{bmatrix} U \\ 0 \end{bmatrix}$$

$$I_2 = \frac{C_2 U s}{R_1 C_1 R_2 C_2 s^2 + R_1 C_1 s + R_2 C_2 s + R_1 C_2 s + 1}$$

dosazení I_2 do vztahu $Y = \frac{1}{C_2 s} I_2$

$$Y = \frac{U}{R_1 C_1 R_2 C_2 s^2 + R_1 C_1 s + R_2 C_2 s + R_1 C_2 s + 1}$$

Vyjádření přenosu soustavy:

$$G = \frac{Y}{U} = \frac{1}{R_1 C_1 R_2 C_2 s^2 + R_1 C_1 s + R_2 C_2 s + R_1 C_2 s + 1}$$

Po dosazení hodnot:

$$G = \frac{1}{22,841s^2 + 14,758s + 1}$$

Převedení na Frobenienův tvar:

$$22,841\ddot{y} + 14,758\dot{y} + y = u$$

$$\ddot{y} + \frac{14,758}{22,841}\dot{y} + \frac{1}{22,841}y = \frac{1}{22,841}u$$

$$\ddot{x} + \frac{14,758}{22,841}\dot{x} + \frac{1}{22,841}x = u$$

Substituce pro Frobeniův kanonický tvar

$$x = x_1$$

$$\dot{x} = \dot{x}_1 = x_2$$

$$\ddot{x} = \dot{x}_2$$

$$y = \frac{1}{22,841}x$$

Po substituci:

$$\ddot{x} + \frac{14,758}{22,841}\dot{x} + \frac{1}{22,841}x = u$$

$$\dot{x}_2 + \frac{14,758}{22,841}x_2 + \frac{1}{22,841}x_1 = u$$

$$y = \frac{1}{22,841}x_1$$

Sestavení rovnic pro odvození matic:

$$\dot{x}_1 = 0x_1 + x_2 + 0u$$

$$\dot{x}_2 = -\frac{1}{22,841}x_1 - \frac{14,758}{22,841}x_2 + u$$

Matice Frobeniova tvaru:

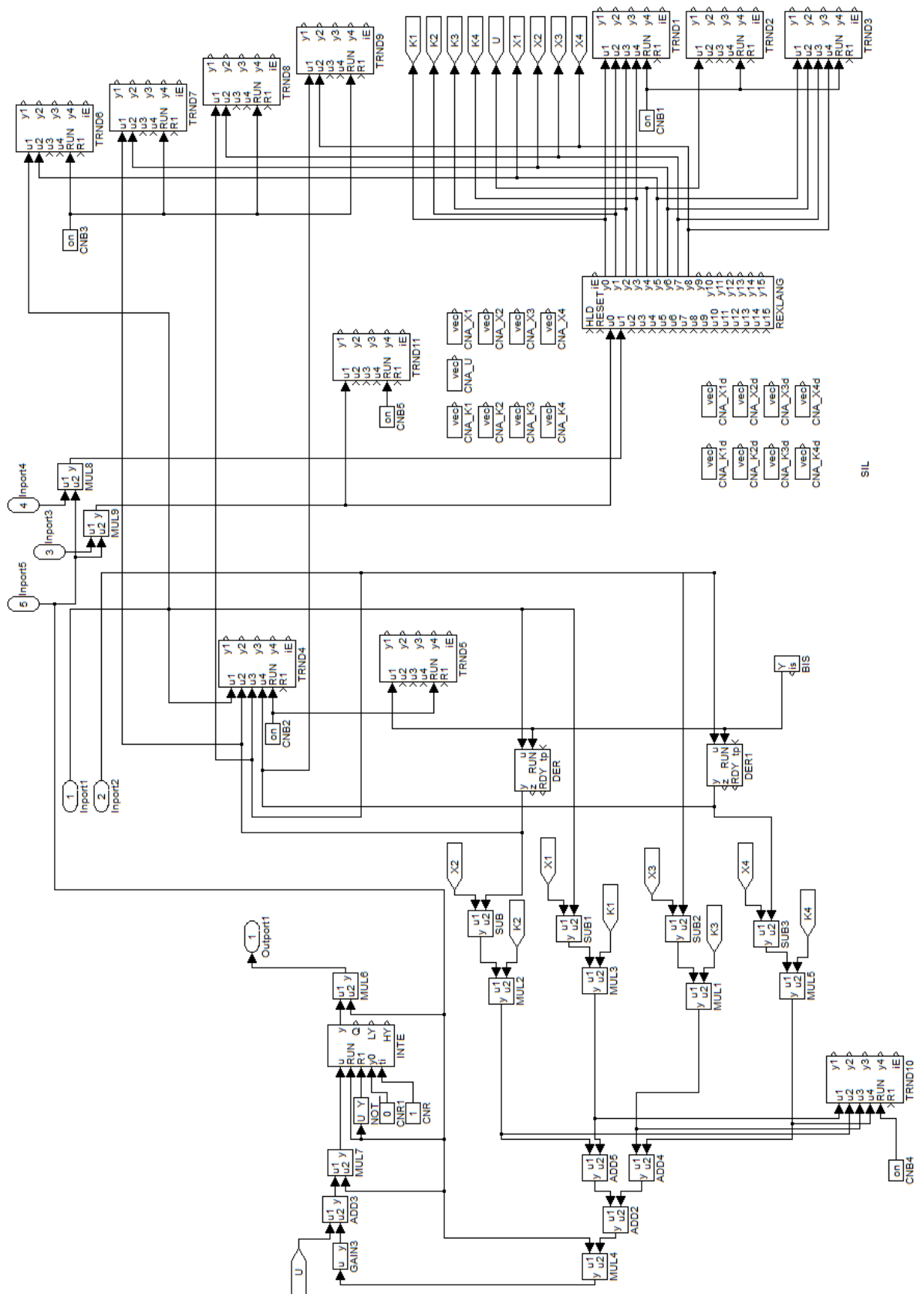
$$A = \begin{pmatrix} 0 & 1 \\ -\frac{1}{22,841} & -\frac{14,758}{22,841} \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

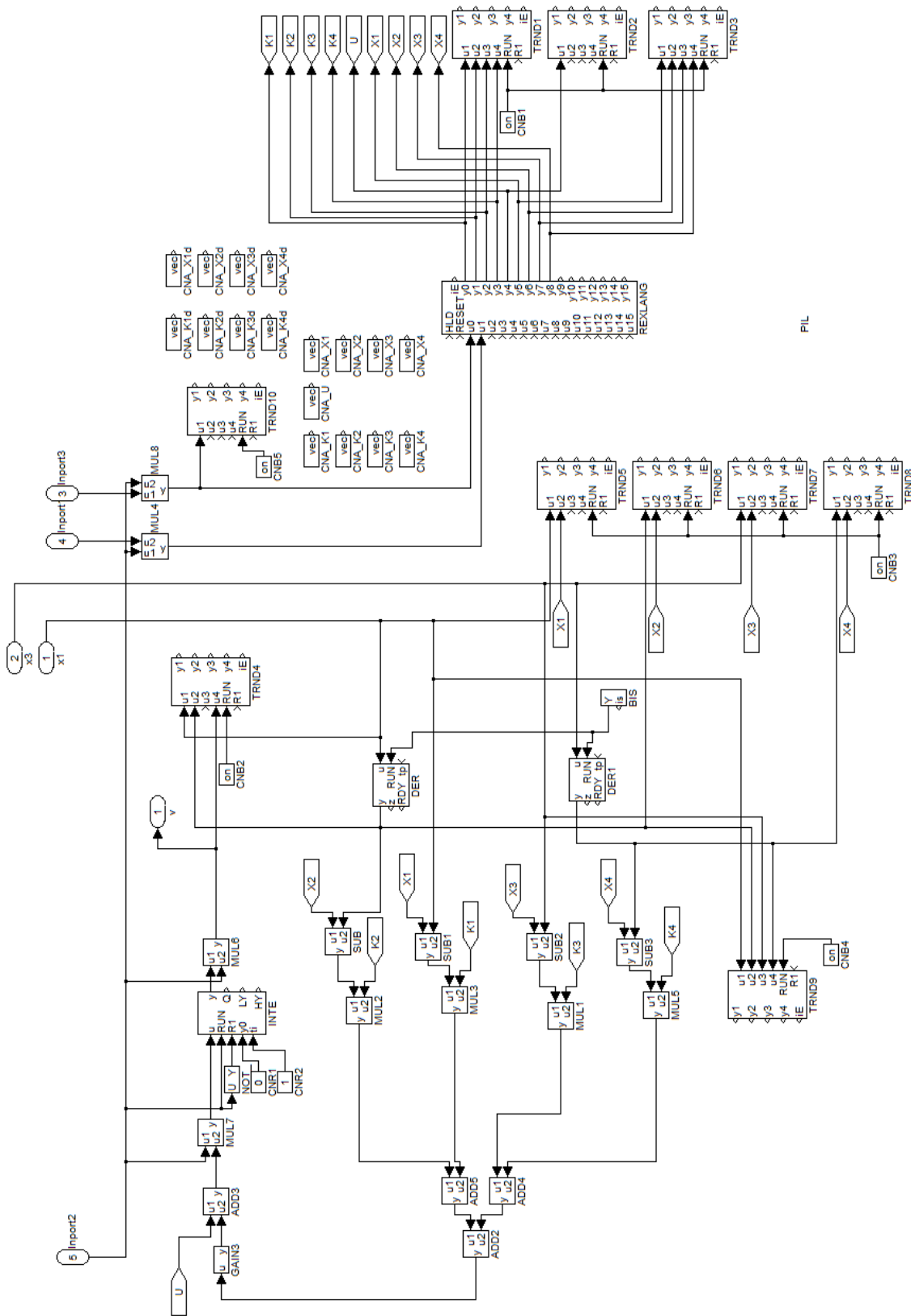
$$C = \begin{pmatrix} \frac{1}{22,841} & 0 \end{pmatrix}$$

$$D = (0)$$

Příloha II: Stavový regulátor pro SIL v prostředí REX



Příloha III: Stavový regulátor pro PIL v prostředí REX



Příloha IV: CD-ROM disk

Součástí této bakalářské práce je CD-ROM disk, který obsahuje:

- Bakalářskou práci ve formátu PDF
- adresář SIL – soubory pro simulaci v REX včetně vizualizace
- adresář PIL – soubory pro simulaci v REX včetně vizualizace