

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

NÁVRH A ŘÍZENÍ INVERZNÍHO ROTAČNÍHO KYVADLA

Bc. Dominik Papp

Diplomová práce
2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Dominik Papp**
Osobní číslo: **I13474**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Návrh a řízení rotačního inverzního kyvadla**
Zadávací katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je navržení, sestavení a návrh řízení rotačního inverzního kyvadla. Student k sestavení soustavy použije stavebnici Lego Mindstorms případně obohacenou o další senzory a k návrhu řídicího software použije Matlab.

Teoretická část: Rešerše podobných zařízení, rešerše řídicích algoritmů vhodných k řízení rotačního inverzního kyvadla, popis platformy, možnosti programování NXT v Matlabu.

Praktická část: Sestavení rotačního inverzního kyvadla, dokumentace sestavení, vytvoření ovládacího software, návrh řízení soustavy (SISO systém, případně MIMO systém), vyhodnocení kvality regulace.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

BALÁTĚ, J. Automatické řízení. Praha: BEN, 2003. 654 s. ISBN 80-7300-020-2.
BENEDETTELLI, D. Creating Cool Mindstorms NXT Robots. New York: Springer. ISBN 978-1-59059-966-2

Vedoucí diplomové práce:

Ing. Petr Doležel, Ph.D.

Katedra řízení procesů

Datum zadání diplomové práce:

30. října 2014

Termín odevzdání diplomové práce:

15. května 2015



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 14. listopadu 2014

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 15. 5. 2015

Dominik Papp

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Petru Doleželovi, Ph.D. za vedení práce, cenné rady a připomínky ke zpracování. V neposlední řadě bych chtěl také poděkovat mé rodině a přátelům za podporu během studia.

V Pardubicích dne 15. 5. 2015

Dominik Papp

ANOTACE

Tato práce je zaměřena na identifikaci a řízení rotačního inverzního kyvadla. Teoretická část je věnována rešerši obdobných zařízení, popisu stavebních bloků, experimentální identifikace a vybraných řídicích algoritmů. K sestavení soustavy je použita stavebnice Lego Mindstorms a řízení je realizováno výpočetním software Matlab.

KLÍČOVÁ SLOVA

rotační inverzní kyvadlo, experimentální identifikace, NXT, řízení

TITLE

DESIGN AND CONTROL OF THE ROTARY INVERTED PENDULUM.

ANNOTATION

This thesis is aimed for identification and control of the rotary inverted pendulum. Theoretical part is dedicated research similar devices, description of the building blocks, experimental identification and chosen control algorithms. The system is built by Lego Mindstorms kit and control is realized by computing software Matlab.

KEYWORDS

rotary inverted pendulum, experimental identification, NXT, control

Obsah

Seznam zkratk	10
Seznam symbolů	11
Seznam obrázků	12
Seznam tabulek	14
ÚVOD	15
1 TEORETICKÁ ČÁST	16
1.1 REŠERŠE	16
1.1.1 Furuta pendulum	16
1.1.2 Quanser rotary inverted pendulum.....	16
1.1.3 Quanser rotary double inverted pendulum.....	17
1.1.4 Rotary Inverted Pendulum Trainer for NI ELVIS	17
1.1.5 Jiyue rotary inverted pendulum.....	18
1.2 POPIS PLATFORMY LEGO MINDSTORMS NXT	18
1.2.1 Řídící jednotka NXT brick (NXT Kostka).....	18
1.2.2 NXT interaktivní servo motor	19
1.2.3 NXT snímač natočení.....	20
1.3 PROGRAMOVÁNÍ NXT V PROSTŘEDÍ MATLAB.....	21
1.3.1 Instalace RWTH-Mindstorms NXT Toolbox pro Matlab.....	21
1.3.2 Vytvoření a ukončení spojení.....	24
1.3.3 Komunikace s motorem	25
1.3.4 Komunikace se senzorem.....	27
1.4 EXPERIMENTÁLNÍ IDENTIFIKACE	29
1.4.1 Vyhodnocení ze signálu obecného průběhu.....	30
1.4.2 Vyhodnocení z přechodové charakteristiky	34
1.4.3 Vyhodnocení kvality identifikace	35
1.5 ALGORITMY VHODNÉ PRO ŘÍZENÍ	35
1.5.1 Dopředné řízení.....	36
1.5.2 Zpětnovazební řízení.....	36
1.5.3 Regulátor PID	37
1.5.4 Regulační obvod s pomocnou regulovanou veličinou (kaskádní regulace).....	41
1.5.5 Seřízení regulátoru metodou Ziegler-Nichols.....	41
1.5.6 Metoda požadovaného modelu (Inverze dynamiky).....	42

1.5.7 Vyhodnocení kvality regulace.....	44
2 ŘEŠENÍ.....	45
2.1 POPIS INVERZNÍHO KYVADLA.....	45
2.2 EXPERIMENTÁLNÍ IDENTIFIKACE KYVADLA.....	47
2.2.1 Měření dat	47
2.2.2 Úprava dat před identifikaci.....	48
2.2.3 Identifikace.....	49
2.2.4 Vyhodnocení identifikovaných modelů	53
2.3 URČENÍ PARAMETRŮ REGULÁTORU	54
2.3.1 Nastavení parametrů regulátoru metodou Ziegler-Nichols.....	54
2.3.2 Nastavení parametrů regulátoru metodou požadovaného modelu	55
3 DOSAŽENÉ VÝSLEDKY	57
3.1 VZORKOVACÍ PERIODA	57
3.2 NXT SERVO MOTOR A ANGLE SENZOR	57
3.3 REGULAČNÍ POCHOD.....	58
3.3.1 Experimentální nastavení regulátoru.....	58
3.3.2 Nastavení regulátoru ZN metodou	59
3.3.3 Nastavení regulátoru metodou požadovaného modelu	60
3.4 VYHODNOCENÍ KVALITY REGULAČNÍCH POCHODŮ.....	61
4 ZÁVĚR	62
Literatura	63
Seznam příloh.....	64

Seznam zkratek

A/D	analogově digitální převodník
D/A	digitálně analogový převodník
DOBH	dopředná obdélníková náhrada
EEPROM	elektricky programovatelná i mazatelná paměť
EX	experimentální metoda
LCD	displej tvořený kapalnými krystaly (liquid crystal display)
LICH	lichoběžníková náhrada
PM	metoda požadovaného modelu
PID	proporcionálně integračně derivační regulátor
PSD	proporcionálně sumačně diferenční regulátor
R	regulátor
RJ12	telefonní konektor
RO	regulační obvod
RPM	otáčky na minutu (revolutions per minute)
S	regulovaná soustava
USB	universální sériová sběrnice (universal serial bus)
ZOBD	zpětná obdélníková náhrada
ZV	zpětná vazba
ZN	metoda Ziegler-Nichols

Seznam symbolů

ξ	koeficient tlumení
λ	krok rozmístění rysek na inkrementálním senzoru
a	zesílení otevřeného regulačního obvodu
α	úhel natočení kyvadla
$e(t)$	regulační odchylka
e_{\max}	regulační odchylka při ustálených kmitech.
IAE	kritérium kvality regulace
K_{id}	kvalita identifikace
N	počet bodů měření
n	řád modelu.
R	hodnota spínaná v relé
r_0, P	proporcionální zesílení regulátoru
r_{0k}	kritické zesílení
r_1	derivační konstanta regulátoru
r_{-1}	integrační konstanta regulátoru
T	interval vzorkování, s
T_0	časová konstanta modelu, s
T_I, I	integrační časová konstanta, s
T_K	kritická perioda
T_d	dopravní zpoždění, s.
T_D, D	derivační časová konstanta, s.
$u(t)$	akční zásah
u_{Phi}	úhel natočení ramene
$v_1(t), v_2(t)$	porucha
$w(t)$	žádaná hodnota
$y(t)$	regulovaná veličina
$y_m(t)$	měřená výstupní veličina
$y_s(t)$	simulovaná výstupní veličina
\hat{y}	odhad měřené veličiny

Seznam obrázků

Obr. 1.1 – Futura pendulum (Rexcontrols, 2014).....	16
Obr. 1.2 – Quanser rotary inverted pendulum	17
Obr. 1.3 – Quanser rotary double inverted pendulum	17
Obr. 1.4 – NI ELVIS (Ni Elvis, 2015).....	18
Obr. 1.5 – Jiyue rotary inverted pendulum (Jiyue, 2015)	18
Obr. 1.6 – NXT kostka.....	19
Obr. 1.7 – Vnitřní propojení modulů (Jakeš, 2013).....	19
Obr. 1.8 – Interaktivní servo motor (Phil, 2015)	19
Obr. 1.9 – Snímač natočení (Hitechnic, 2015)	20
Obr. 1.10 – Princip inkrementálního snímače (Ďaďo, 1996)	21
Obr. 1.11 – Postup instalace RWTH Toolboxu	22
Obr. 1.12 – Přenesení souboru MotorControl.rxe	24
Obr. 1.13 – NXT, vytvoření a ukončení spojení.....	25
Obr. 1.14 – NXT, příklad nastavení motoru	26
Obr. 1.15 – NXT, příklad přímého přístupu k motoru.....	27
Obr. 1.16 – NXT, čtení úhlu natočení motoru	27
Obr. 1.17 – NXT, příklad inicializace senzoru	27
Obr. 1.18 – NXT, čtení a převod údajů ze senzoru natočení	29
Obr. 1.19 – Porovnání odezvy reálné soustavy a modelu.....	30
Obr. 1.20 – Metoda nejmenších čtverců	31
Obr. 1.21 – Úprava dat pro identifikaci (Modrlák, str. 18).....	33
Obr. 1.22 – Matlab, fminsearch	33
Obr. 1.23 – Identifikace soustavy s kmitavou odezvou z přechodové charakteristiky (Burý, 2011).....	34
Obr. 1.24 – Schéma otevřeného regulačního obvodu.....	36
Obr. 1.25 – Schéma zpětnovazebního regulačního obvodu.....	37
Obr. 1.26 – Schéma PID regulátoru (standardní tvar)	38
Obr. 1.27 – Spojitý signál (levý), vzorkování (prostřední) a tvarování (pravý).....	39
Obr. 1.28 – Schéma diskrétního regulačního obvodu.....	39
Obr. 1.29 – Způsoby náhrady integrálu (vlevo) a náhrada derivace (vpravo).....	40
Obr. 1.30 – Regulační obvod s pomocnou regulovanou veličinou.....	41
Obr. 1.31 – Ziegler-Nichols metoda (Dušek,2013)	42

Obr. 2.1 – Rameno kyvadla	45
Obr. 2.2 – Kyvadlo	45
Obr. 2.3 – Inverzní rotační kyvadlo	46
Obr. 2.4 – Schéma regulačního obvodu, použité pro měření dat.....	47
Obr. 2.5 – Změřená data	48
Obr. 2.6 – Upravená data	49
Obr. 2.7 – Identifikace MNC	50
Obr. 2.8 – Průběh odchylky měřených dat a modelu získaného metodou MNC	50
Obr. 2.10 – Průběh odchylky měřených dat a modelu získaného metodou pož. modelu.....	51
Obr. 2.9 – Identifikace požadovaným modelem.....	51
Obr. 2.11 – Vyhodnocovaná přechodová charakteristika.....	52
Obr. 2.12 – Zvětšený úsek odečtu hodnot	52
Obr. 2.13 – Identifikace z přechodové charakteristiky.....	53
Obr. 2.14 – Průběh odchylky měřených dat a modelu získaného z přechodové charakteristiky	53
Obr. 2.15 – Relé ve zpětné vazbě	54
Obr. 2.16 – Odezva modelu na relé ve ZV	54
Obr. 3.1 – Statická charakteristika NXT motoru	58
Obr. 3.2 – Odezva NXT motoru na skok do plných otáček.....	58
Obr. 3.3 – Průběh regulačního pochodu po experimentálním seřízení.....	59
Obr. 3.4 – Průběh regulačního pochodu po seřízení metodou Ziegler-Nichols	60
Obr. 3.5 – Průběh regulačního pochodu po seřízení metodou požadovaného modelu.....	61

Seznam tabulek

Tab. 1.1 – Popis registrů senzoru natočení	28
Tab. 1.2 – Seřízení regulátoru metodou Ziegler-Nichols	42
Tab. 1.3 – závislost α a β na relativním překmitu κ	43
Tab. 1.4 – Nastavení parametrů regulátoru metodou požadovaného modelu	43
Tab. 2.1 – Parametry ramene	45
Tab. 2.2 – Parametry kyvadla	46
Tab. 2.3 – Experimentální nastavení regulátorů	47
Tab. 2.4 – Parametry přechodové charakteristiky	52
Tab. 2.5 – Vypočtené parametry přenosu (2.11)	52
Tab. 2.6 – Kritéria pro jednotlivé metody identifikace.....	54
Tab. 2.7 – Odečtené parametry	54
Tab. 2.8 – Parametry PID regulátoru získané ZN metodou.....	55
Tab. 2.9 – Vypočtené parametry regulátoru metodou požadovaného modelu	56
Tab. 3.1 – Doby vykonání funkcí Matlabu	57
Tab. 3.2 – Použité nastavení regulátoru metodou požadovaného modelu.....	61
Tab. 3.3 – Vyhodnocení kvality regulace	61

ÚVOD

Cílem této diplomové práce je sestavení inverzního rotačního kyvadla pomocí stavebnice Lego Mindstorms NXT a jeho řízení ve výpočetním software Matlab. Práce je rozdělena na teoretickou část, část řešení, část s dosaženými výsledky a závěrečné zhodnocení.

Teoretická část je rozdělena do pěti kapitol. V první kapitole je uvedena rešerše obdobných komerčně vyráběných zařízení a jedné studentské práce. Ve druhé kapitole je popsána řídicí jednotka stavebnice Lego Mindstorms NXT, použitý motor a senzor úhlu natočení. Třetí kapitola je pak zaměřena na instalaci ovládacího software a komunikaci stavebnice Lego Mindstorms NXT s výpočtovým software Matlab. Čtvrtá kapitola je zaměřena na metody experimentální identifikace a poslední, pátá, kapitola je určena popisu vybraných řídicích algoritmů.

Část řešení je rozdělena do tří kapitol. V první kapitole je popsáno sestavené inverzní rotační kyvadlo. Druhá kapitola je zaměřena na získání a zpracování naměřených dat a jsou provedeny metody experimentální identifikace pro získání modelu soustavy. Ve třetí kapitole jsou na základě získaných modelů určeny parametry regulačního obvodu.

V část dosažených výsledků jsou čtyři kapitoly. V první kapitole je zdůvodněna volba vzorkovací periody. Ve druhé kapitole je definován úhel horní svislé polohy kyvadla a uvedeny charakteristiky motoru. V poslední kapitole jsou znázorněny grafické průběhy regulačních pochodů dosažených nastavením parametrů regulátoru jednotlivými metodami seřízení.

1 TEORETICKÁ ČÁST

1.1 REŠERŠE

Problém inverzního kyvadla je klasická úloha automatického řízení. Úkolem regulačního obvodu je balancování závaží na tyčce v horní svislé poloze tak, aby kyvadlo nespadlo. Inverzní kyvadlo lze sestavit v mnoha variantách (inverzní kyvadlo na vozíku, balancování na špici, rotační inverzní kyvadlo). V této kapitole je provedena rešerše existujících řešení rotačních inverzních kyvadel.

1.1.1 Furuta pendulum

Futura pendulum na obr. 1.1 je laboratorní model inverzního rotačního kyvadla, který je určený především pro výuku a testování řídicích algoritmů. O řízení motoru a zpracování dat ze senzorů se stará řídicí systém REX. Řídicí systém REX je plně kompatibilní s prostředím Matlab-Simulink, je tedy možné navržený algoritmus nejprve odsimulovat. Na stránkách rexcontrols lze též nalézt interaktivní fyzikální model kyvadla (Rexcontrols, 2014).



Obr. 1.1 – Futura pendulum (Rexcontrols, 2014)

1.1.2 Quanser rotary inverted pendulum

Jedná se o komerčně vyráběné rotační inverzní kyvadlo zobrazené na obr. 1.2. Toto kyvadlo je vhodné k výuce a demonstraci řídicích algoritmů. Kyvadlo tvoří kvalitní hliníkové šasi, ve kterém je uchycen výkonný servo motor. O měření natočení ramene a kyvadla se starají vysoce přesné převodníky s rozlišením 4096 kroků za otáčku. Kyvadlo je jednoduše

propojitelné s prostředím Matlab pomocí Q2-USB převodníku, nebo s prostředím LabView pomocí modulu Q1-cRIO (Quanser, 2015).



Obr. 1.2 – Quanser rotary inverted pendulum (Quanser, 2015)



Obr. 1.3 – Quanser rotary double inverted pendulum (Quanser double, 2015)

1.1.3 Quanser rotary double inverted pendulum

Dvojité rotační kyvadlo společnosti Quanser na obr. 1.3 nabízí další úroveň složitosti. Je určeno pro demonstraci a testování pokročilých řídicích algoritmů, které řeší problémy reálného světa, např. start vícestupňové rakety. Toto kyvadlo nabízí stejné možnosti propojení se softwarem Matlab, či Labview i stejnou hardwarovou výbavu jako Quanser rotary inverted pendulum (Quanser double, 2015).

1.1.4 Rotary Inverted Pendulum Trainer for NI ELVIS

Komerčně vyráběné rotační inverzní kyvadlo od společnosti National Instruments na obr. 1.4, lze efektivně ovládat pomocí softwaru LabView. Obsahuje DC servo motor, snímač natočení motoru s rozlišením 2048 kroků na otáčku a snímač natočení tyče kyvadla s rozlišením 1024 kroků na otáčku (NI Elvis, 2015).

1.1.5 Jiyue rotary inverted pendulum

Jedná se o projekt inverzního rotačního kyvadla studenta Jiyue He. Kyvadlo je znázorněno na obr. 1.5. Otáčení ramena zajišťuje stejnosměrný motor. Úhel natočení motoru je snímán rozlišením 500 kroků na otáčku a úhel natočení kyvadla s rozlišením 1250 kroků na otáčku. Kyvadlo je řízeno mikroprocesorem ATmega 2560 s implementovaným PID algoritmem (Jiyue, 2015).



Obr. 1.4 – NI ELVIS (Ni Elvis, 2015)



Obr. 1.5 – Jiyue rotary inverted pendulum (Jiyue, 2015)

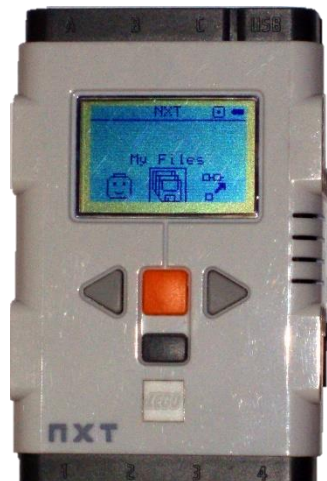
1.2 POPIS PLATFORMY LEGO MINDSTORMS NXT

Stavebnice Lego NXT je určena pro studenty a zájemce o oblasti robotiky. Umožňuje sestavovat a programovat různorodá robotická zařízení. Stavebnice v základní podobě obsahuje programovatelnou inteligentní kostku NXT brick, tři interaktivní servo motory a několik senzorů (ultrazvukový snímač vzdálenosti, snímač zvuku, světla a dotyku).

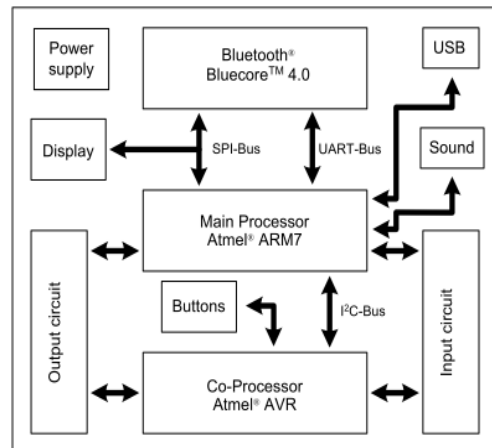
1.2.1 Řídicí jednotka NXT brick (NXT Kostka)

Programovatelná inteligentní NXT kostka na obr. 1.6 je jádrem každé úlohy. Vnitřní uspořádání kostky je složeno z jednotlivých modulů, které mezi sebou komunikují pomocí sběrnice, viz obr. 1.7. O řízení NXT kostky se starají dva mikroprocesory, jeden 32-bitový procesor ARM7. Je to hlavní mikroprocesor, který se stará o komunikaci a vykonávání

uživatelského programu. Druhý je pomocný 8-bitový mikroprocesor Atmel AVR. NXT kostka je vybavena čtveřicí tlačítek a LCD displejem, které umožňují vytvářet velice krátké programy, které slouží především pro ověření funkce senzorů, či motorů. S NXT kostkou lze komunikovat buď pomocí sběrnice USB, nebo pomocí rozhraní bluetooth. Díky rozhraní bluetooth může také více NXT kostek komunikovat mezi sebou. Každá NXT kostka obsahuje 4 vstupní porty, označené číselně 1 – 4, a 3 výstupní porty označené písmeny A,B,C. Motory a senzory se připojují pomocí modifikované verze konektorů RJ12 (Jakeš, 2013).



Obr. 1.6 – NXT kostka



Obr. 1.7 – Vnitřní propojení modulů (Jakeš, 2013)

1.2.2 NXT interaktivní servo motor

Základem NXT servo motoru je stejnosměrný motor s převodovkou. V motoru je vestavěný rotační inkrementální senzor, který slouží pro měření natočení s rozlišením $\pm 1^\circ$.



Obr. 1.8 – Interaktivní servo motor (Phil, 2015)

Rychlost a směr otáčení motoru lze nastavovat v procentech výkonu motoru (motor.Powerlevel) v intervalu $(-100 ; +100)$. Směr otáčení motoru se určuje znaménkem. Kladné hodnoty znamenají, že se motor bude otáčet doleva a pro záporné hodnoty pak doprava. Pro hodnotu 0 se motor zastaví.

Motor se může otáčet nepřetržitě, a nebo otočit o požadovaný úhel. Při zastavování otáčení se motor bude chovat jedním ze tří způsobů:

Coast ... v tomto módu se motor po dosažení požadovaného natočení nechá samovolně, vlivem setrvačnosti, dotočit až do zastavení.

Hold ... v tomto módu se motor automaticky zpomalí ještě před dosažením požadované pozice, tak aby se otáčení zastavilo v požadované poloze s přesností $\pm 1^\circ$ a motor je vypnut.

Hold brake ... v tomto módu je průběh zastavení motoru stejný, jako u módu Hold. Po zastavení motoru je navíc sepnuta aktivní brzda motoru. Tato brzda drží požadovanou pozici i při případném působení vnější síly na motor. Motor v tomto módu odebírá energii, i když se neotáčí (Rwth,2015).

1.2.3 NXT snímač natočení



Obr. 1.9 – Snímač natočení (Hitechnic, 2015)

Tento snímač měří natočení také pomocí rotačního inkrementálního senzoru. Na rozdíl od senzoru zabudovaného v servo motoru, je tento snímač navržen tak, aby vykazoval velmi malé tření. Samotný senzor je umístěn v ose otáčení, takže nejsou zapotřebí žádné ozubené převody.

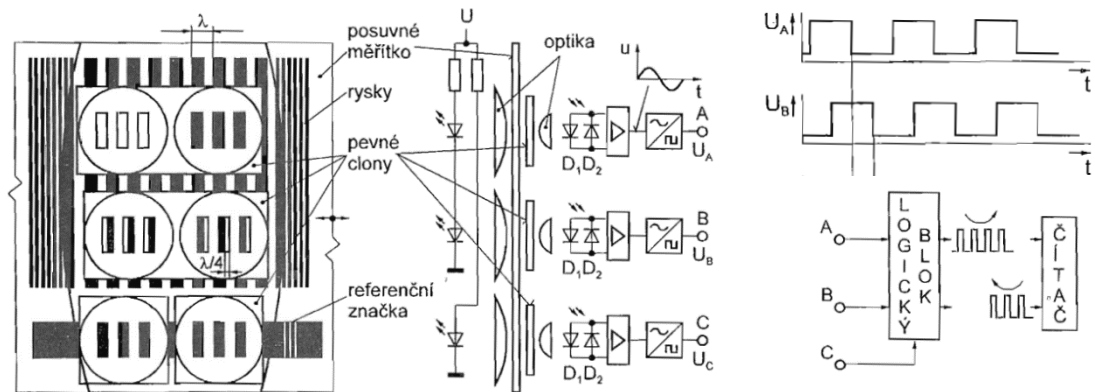
Snímač dokáže měřit kromě aktuální výchylky také celkový úhel otáčení, tato hodnota je uložena v paměti EEPROM. A také rychlost otáčení do 1500 RPM. Přesnost měření úhlu je $\pm 1^\circ$ (Hitechnic, 2015).

Princip inkrementálního rotačního snímače

Základem inkrementálního snímače je průhledný kotouček, nebo pravítko, na kterém jsou pravidelně, s krokem λ , rozmístěny neprůhledné rysky. Tento kotouček je pevně spojen s osou otáčení snímače a je umístěn mezi zdrojem infračerveného světla a fotocitlivým detektorem. Otáčením vzniká přerušovaný světelný paprsek, který je detekován a po následném

tvarování vzniká obdélníkový signál, který je přiveden do čítače, kde inkrementuje (příp. dekrementuje) jeho obsah. Výsledná hodnota natočení je dána počtem impulzů.

Je-li potřeba vyhodnocovat směr otáčení, použijí se dva senzory A a B, vzájemně posunuté o čtvrtinu kroku λ . Třetí senzor (C) se využívá jako nulová značka, k detekci jedné plné otočky (Ďaď'o, 1996).



Obr. 1.10 – Princip inkrementálního snímače (Ďaď'o, 1996)

1.3 PROGRAMOVÁNÍ NXT V PROSTŘEDÍ MATLAB

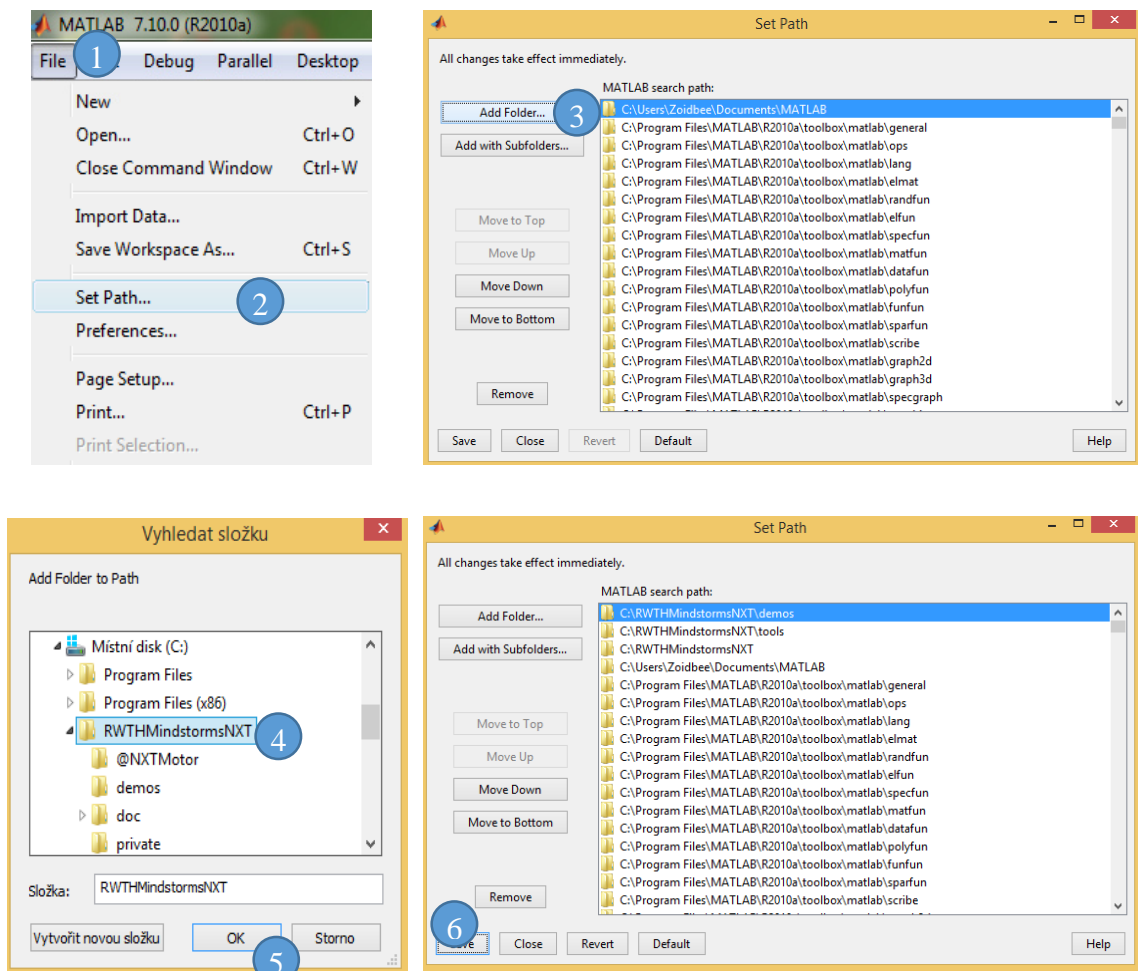
Inteligentní Lego NXT kostku lze programovat pomocí mnoha různých nástrojů v mnoha programovacích jazycích, např. NXT-G, C#, C_NXT, NXT-Python, Lua a další. V mém případě využiji k vytvoření řídicího programu prostředí Matlab. Komunikace mezi NXT kostkou a prostředím Matlab bude zajištěna pomocí „The RWTH - Mindstorms NXT toolbox“. Tento toolbox je vyvíjen na německé universitě RWTH Aachen university a je určený pro řízení soustav Lego Mindstorms NXT pomocí sběrnice USB, nebo rozhraní Bluetooth. Jeho hlavní výhoda spočívá v tom, že lze zkombinovat řízení soustavy s výpočetními výhodami Matlabu.

1.3.1 Instalace RWTH-Mindstorms NXT Toolbox pro Matlab

Instalace RWTH Toolboxu se skládá z několika kroků. Instalace samotného toolboxu, který obsahuje veškeré potřebné funkce pro Matlab. Poté je třeba nainstalovat ovladač NXT kostky pro operační systém. Dalším krokem je přenos souboru do NXT kostky, který umožní interakci s motorem. Posledním krokem je ověření instalace.

1.3.1.1 Instalace RWTH Toolbox

Nejprve je třeba ze stránky www.mindstorms.rwth-aachen.de stáhnout nejnovější soubor toolboxu. V práci je použita aktuální stabilní verze 4.07. Stažený soubor archiv typu *.zip, konkrétně např. RWTHMindstormsNXTv4.07.zip. Tento archiv je nutné rozbalit do pracovního adresáře (např.: C:\RWTHMindstormsNXT). Cestu ke složce RWTHMindstormsNXT, její podsložce RWTHMindstormsNXT\tools a případně také RWTHMindstormsNXT\demos je třeba přiřadit v Matlabu do cest používaných knihoven. To lze provést v menu File (1) → Set path (2). Otevře se nové okno, ve kterém se klikne na tlačítko Add Folder (3), vyhledá se složka RWTHMindstormsNXT (4) a výběr potvrdí stiskem tlačítka OK (5). Stejným způsobem se vyberou podsložky \tools a případně také \demos. Provedené změny se uloží kliknutím na tlačítko Save (6). Nakonec se okno uzavře.



Obr. 1.11 – Postup instalace RWTH Toolboxu

1.3.1.2 Instalace ovladače Lego USB

Na oficiálních stránkách Lega mindstorms <http://www.lego.com/en-us/mindstorms/downloads> je ke stažení ovladač NXT Fantom driver. Stažený archiv je potřeba rozbalit a poté spustit instalační soubor setup.exe, který podle verze operačního systému nainstaluje 32-bitovou, nebo 64-bitovou verzi ovladače. Po dokončení instalace je třeba restartovat počítač.

Objeví-li se při instalaci chybová hláška InstallerError 11006, ovladač lze stáhnout z alternativní stránky <http://www.robotc.net/support/troubleshooting/fantom-driver.php>.

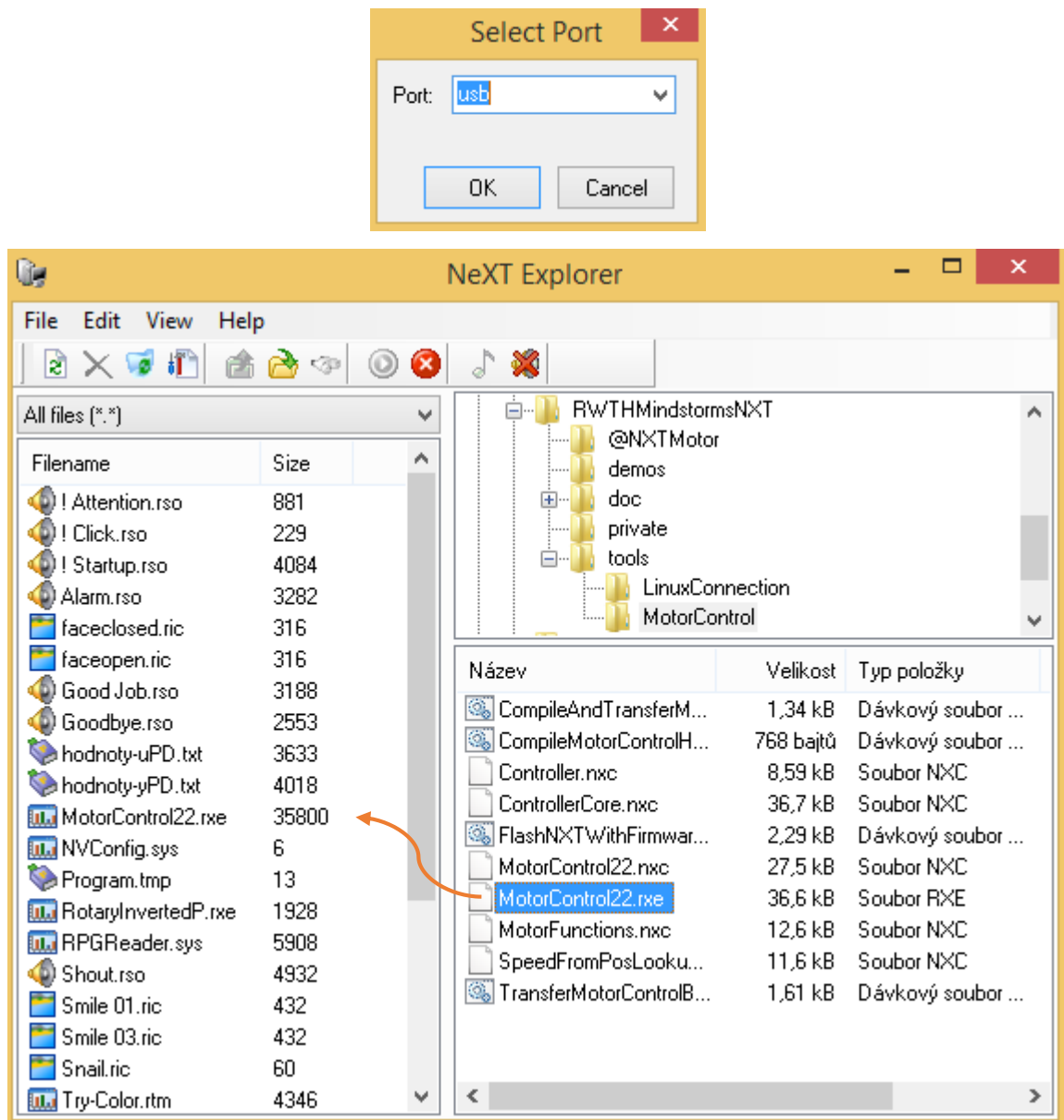
1.3.1.3 Přenesení souboru MotorControl.rxe do Kostky NXT

Tento soubor běží v samotné NXT kostce a zajišťuje komunikaci s aktuálně připojenými motory. Bez tohoto souboru by NXT kostka nereagovala na příkazy k ovládání otáčení motoru, ani čtení úhlu natočení z motoru.

Nejprve je potřeba ze stránky <http://bricxcc.sourceforge.net/utilities.html> stáhnout program NeXTE Explorer, který se použije pro přenos souboru do NXT kostky.

Nejprve se NXT kostka připojí k počítači pomocí USB kabelu a zapne se. Poté se rozbalí stažený archiv a spustí se soubor NeXTE Explorer.exe. Otevře se okno Select port, které vyzývá k zadání portu, kde je NXT kostka připojena. V textovém poli port se ponechá již předvyplněný text **usb** a okno se potvrdí. Poté se otevře hlavní okno NeXTE Exploreru. V levém sloupci je zobrazeny veškeré soubory, které jsou uloženy v NXT kostce. V pravém sloupci se nachází průzkumník souborů na disku počítače. V horní polovině je ve stromové struktuře vypsán seznam složek. Ve spodní polovině je pak vypsán seznam všech souborů v dané složce.

Ve stromovém seznamu se vyhledá naše rozbalená RWTH složka a přejde se do podsložky `\tools\MotorControl` (např. `C:\RWTHMindstormsNXT\tools\MotorControl`). V dolním seznamu se vyhledá soubor **MotorControl.rxe**, který se přetáhne do levého sloupce. Tím dojde ke zkopírování tohoto souboru do vnitřní paměti NXT kostky.



Obr. 1.12 – Přenesení souboru MotorControl.rxe

1.3.1.4 Ověření připojení NXT kostky v Matlabu

Do příkazové řádky Matlabu se zadá příkaz `COM_OpenNXT()`.

Tento příkaz vytváří spojení s NXT kostkou pomocí USB kabelu. Tento příkaz nesmí skončit žádnou chybovou hláškou.

1.3.2 Vytvoření a ukončení spojení

Nové spojení s NXT kostkou připojenou pomocí USB kabelu se vytvoří již zmíněným příkazem `COM_OpenNXT`, který vrací tzv. handle, proměnnou obsahující informace o připojení. Přes tento handle budou následně funkce toolboxu komunikovat s NXT kostkou. Před

začátkem práce s funkcemi toolboxu je třeba definovat defaultní spojení. To zajistí funkce `COM_SetDefaultNXT(h)`, které se předá jako parametr handle na již vytvořené spojení. Tato funkce jej uloží v globální proměnné používané toolboxem (Rwth,2015).

```
h = COM_OpenNXT();  
COM_SetDefaultNXT(h);  
% Spojení s Kostkou se ukončí příkazem  
COM_CloseNXT('all')
```

Obr. 1.13 – NXT, vytvoření a ukončení spojení

1.3.3 Komunikace s motorem

K NXT kostce lze připojit až tři motory naráz, jednotlivé výstupní porty jsou značeny písmeny A, B a C. Z Matlabu lze motor ovládat za pomoci třídy `NXTMotor`, která obsahuje informace o nastavení motoru. Instanci třídy se vytvoří voláním konstruktoru `NXTMotor(port)`. Minimálně se musí předat parametr `port`, ke kterému je motor fyzicky připojen. Parametr `port` lze předat číselně, znakově, nebo symbolickou konstantou. Je-li motor připojen např. na portu A, lze parametr předat číselně (0), znakově 'A', nebo konstantou `MOTOR_A`.

Motoru lze nastavit následující vlastnosti:

Power ... nastavuje rychlost otáčení motoru v procentech (0 – 100 %) a zároveň určuje směr otáčení. Záporná hodnota otáčí motorem doprava a kladná doleva. Nulovou hodnotou se otáčení motoru zastaví. Tato vlastnost musí nabývat celočíselné hodnoty v intervalu $\langle -100; +100 \rangle$. Standardně je nastavena 0.

SpeedRegulation ... je booleovská vlastnost, může nabývat hodnot povoleno (true) a zakázáno (false). Pokud je povolena, motor se automaticky snaží udržet konstantní rychlost otáčení v závislosti na jeho zatížení, tím že firmware NXT kostky interně zvýší hodnotu `motor.Power` (pokud je to možné). Tato volba je využitelná zejména v aplikacích typu dopravník. Při ovládání dvou motorů musí být zakázána. Standardně povolena.

TachoLimit ... definuje úhel ve stupních, o který se má motor otočit a poté se zastavit. Zadaná hodnota musí být celočíselná v intervalu $\langle 0; 999999 \rangle$. Zadáním např. 360 udělá motor jednu otáčku. Nulová hodnota definuje, že se motor točí neustále. Standardně 0.

ActionAtTachoLimit ... definuje chování motoru při dosažení požadovaného otočení (`TachoLimit` musí být různý od nuly). Lze použít jeden ze tří způsobů chování, 'Coast', 'Brake', nebo 'HoldBrake', které jsou blíže popsány kapitole 1.2.2. Standardně 'Brake'.

Vytvoření objektu motoru, nebo úprava některé jeho vlastnosti ještě nevykoná žádnou změnu řízení fyzického motoru. Nově nastavené parametry motoru se musí odeslat do NXT kostky voláním metody `SendToNXT()` z již existujícího objektu motoru. Pro zastavení motoru se volá metoda `Stop()`.

```

%Příklad nastavení motoru
motor = NXTMotor(MOTOR_A);           % Motor je připojen na portu A
motor.SpeedRegulation = false;       % Vypnutá regulace rychlosti
motor.Power = 100;                   % Motor se otáčí maximální rychlostí
motor.TachoLimit = 0;                % Motor se bude nepřetržitě otáčet
motor.ActionAtTachoLimit = 'Hold';   % Nastavení chování při vypnutí motoru

motor.SendToNXT();                   % Odešle nastavení motoru do NXT kostky
motor.Stop();                         % Zastavení motoru

```

Obr. 1.14 – NXT, příklad nastavení motoru

Ovládání motoru pomocí objektu je poměrně časově náročné. Experimentálně bylo zjištěno, že ovládání motoru touto cestou zabere průměrně 16 ms. RWTH toolbox proto poskytuje také funkci pro přímé ovládání motoru:

`DirectMotorCommand(port, power, tachoLimit, speedRegulation, syncedToMotor, turnRatio, rampMode)`

Tato funkce nevyužívá interního programu `MotorControl`, ale posílá příkazy přímo Kostce. Její hlavní výhodou je, že její vykonání je velice rychlé, v průměru její vykonání trvalo 2 ms. Její nevýhodou je, že motor pracuje pouze v režimu `Coast`. Nelze přesně řídit bod zastavení motoru, to však v této práci není třeba. Dále je důrazně zakázáno ovládat tentýž motor pomocí objektu a zároveň pomocí přímého přístupu, neboť tato funkce ignoruje cokoli se právě děje v NXT kostce, což může vést k neočekávanému chování motoru, či pádu programu. Nicméně čtení z motoru při současném ovládání přímým přístupem je povoleno.

První čtyři parametry funkce mají stejný význam jako při použití objektu. Tato funkce navíc nabízí možnosti:

syncedToMotor ... synchronizace s druhým motorem (`MOTOR_A`, `MOTOR_B`).

turnRatio ... synchronizované motory se mohou otáčet každý jinou rychlostí v daném poměru ($-100 ; +100$).

rampMode ... motor může do požadované rychlosti přejít okamžitě, nebo může lineárně zrychlovat, resp. zpomalovat ('off', 'up', 'down').

Kapitola čerpána z (Rwth,2015).

```
% Příklad použití přímého přístupu, roztočení motoru vpravo
DirectMotorCommand(MOTOR_A, -20, 0, 'off', 'off', 0, 'off');
```

Obr. 1.15 – NXT, příklad přímého přístupu k motoru

1.3.3.1 Čtení pozice natočení motoru

Úhel natočení motoru se čte pomocí objektu třídy NXTMotor voláním metody ReadFromNXT(). Metoda vrací objekt obsahující informace o stavu motoru. Pro můj případ je důležitá vlastnost Position, která obsahuje informaci o natočení motoru s přesností $\pm 1^\circ$. Při otáčení motoru se tato hodnota sčítá z každou následující otočkou, např. po dvou otáčkách motoru se přečte úhel otočení 720° (Rwth,2015).

```
% Příklad čtení úhlu natočení z motoru
motor.ReadFromNXT().Position;
```

Obr. 1.16 – NXT, čtení úhlu natočení motoru

1.3.4 Komunikace se senzorem

Senzor může být připojen k jednomu ze čtyřech vstupních portů Kostky (SENSOR_1, SENSOR_2, SENSOR_3, SENSOR_4), komunikace probíhá po sběrnici I2C. Rychlé vyzkoušení funkce senzoru lze provést tak, že se senzor připojí ke NXT kostce na port 1 a v menu NXT kostky se zvolí View → Ultrasonic cm → Port 1. Na displeji se zobrazí měřený úhel, nicméně zobrazená hodnota nebude správná. Protože NXT kostka předpokládá, že je připojené ultrazvukové čidlo. Lze tak ovšem vyzkoušet, zda je připojený senzor funkční.

Před začátkem komunikace se senzorem se musí provést inicializační funkce, která čidlo zapne a připraví ke komunikaci na zadaném portu.

```
NXT_SetInputMode(InputPort, SensorTypeDesc, SensorModeDesc, ReplyMode)
```

InputPort ... vstupní port, ve kterém je připojen senzor (SENSOR_1 – SENSOR_4)

SensorTypeDesc ... typ senzoru ('SWITCH', 'SOUND_DB', 'LOWSPEED_9V' ...)

SensorModeDesc ... jakým způsobem senzor pracuje ('RAWMODE', 'BOOLEANMODE' ...)

ReplyMode ... požadavek potvrzení datového přenosu ('reply', 'dontreply')

```
% Příklad inicializace senzoru
NXT_SetInputMode(Sensor, 'LOWSPEED_9V', 'RAWMODE', 'dontreply');
```

Obr. 1.17 – NXT, příklad inicializace senzoru

Měřená data jsou v senzoru uložena v registru o velikosti 9 Byte, ze kterého lze číst pomocí funkce COM_ReadI2C(Port, RequestLen, DeviceAddress, RegisterAddress).

Funkce vrací vektor přečtených dat ve formátu uint8, nebo v případě neúspěšného čtení, prázdný vektor. Jednotlivé parametry funkce mají význam:

Port ... vstupní port, ke kterému je senzor připojen.

RequestLen ... definuje počet bajtů, které se mají přečíst.

DeviceAddress ... adresa zařízení, použitý snímač natočení má adresu 0x02.

RegisterAddress ... definuje adresu v registru, od které se mají data číst.

Tab. 1.1 – Popis registrů senzoru natočení

Adresa	Obsah
0x41	Mód senzoru
0x42	Aktuální úhel s přírůstkem 2°
0x43	1° doplněk k aktuálnímu úhlu
0x44	32 bitový celkový úhel, horní Bajt
0x45	32 bitový celkový úhel, střední Bajt
0x46	32 bitový celkový úhel, střední Bajt
0x47	32 bitový celkový úhel, spodní Bajt
0x48	16 bitová rychlost otáčení, horní Bajt
0x49	16 bitová rychlost otáčení, spodní Bajt

Senzor se může nacházet v jednot ze tří módů. Po inicializaci automaticky přechází do normálního módu (0x00), ve kterém provádí měření. Druhý mód (0x43) způsobí reset nulové pozice a nahradí ji pozicí aktuálního natočení osy senzoru. Tento údaj je uložen v interní EEPROM paměti senzoru a je tedy uchován i po odpojení napájení. Po tomto módu je potřeba zajistit čekání alespoň 25 ms, pro dokončení operace zápisu do EEPROM. Posledním módem (0x52) je vynulování naakumulovaného celkového úhlu. Po dokončení operace druhého, nebo třetího módu senzor automaticky přechází do prvního módu (Hitechnic,2015).

1.3.4.1 Čtení změřených údajů

Čtení dat ze senzoru se provede voláním funkce COM_ReadI2C. Pro přečtení obsahu celého registru se provede čtení od adresy 0x41, neboli 65 v dekadickém vyjádření. Funkce vrátí vektor dat o velikosti 9 bajtů, které mají uspořádání podle tab. 1.1. Tato „surová“ data je poté nutné převést na jednotlivé hodnoty. Aktuální úhel natočení se získá výpočtem. Úhel natočení je uložen tak, že první bajt obsahuje hodnotu úhlu s přírůstkem dvou a druhý bajt obsahuje jedničkový doplněk. Celkový naakumulovaný úhel se získá tak, že se jeho jednotlivé bajty „složí“ za sebe od nejvyššího po nejnižší a interpretují se jako 32 bitový výsledek. Otáčky

za minutu se opět získají „složením“ horního a spodního bajtu za sebe a interpretováním jako 16 bajtového čísla (Rwth,2015; Hitechnic, 2015).

```
% přečtení celé paměti senzoru
data = COM_ReadI2C(SENSOR_1, 9, 2, 65);

% výpočet aktuálního úhlu natočení
angle = 2*data(2) + data(3);

% výpočet celkově naakumulovaného úhlu
%angleAcc = 0x1000000*data(4) + 0x10000*data(5) + 0x100*data(6) + data(7);
angleAcc = typecast(s(7:-1:4), 'int32'); % zápis pro Matlab

% výpočet rychlosti otáčení
%RPM = 0x100*data(8) + data(9);
RPM = typecast(s(9:-1:8), 'int16'); % zápis pro Matlab
```

Obr. 1.18 – NXT, čtení a převod údajů ze senzoru natočení

1.4 EXPERIMENTÁLNÍ IDENTIFIKACE

Pojmem identifikace se rozumí hledání určení vlastností zkoumané soustavy. Vlastnosti soustavy odpovídají parametrům matematického modelu. Nejprve je tedy třeba stanovit strukturu modelu a poté nalézt vhodné numerické vyjádření jeho parametrů.

Experimentální identifikace využívá k nalezení parametrů modelu data měřená na soustavě, která se poté vybranou metodou identifikují. Změřeným datům odpovídá reakce výstupní veličiny soustavy na vhodný testovací signál. Model popisuje vstupně-výstupní chování soustavy, nalezené parametry ovšem nemají fyzikální význam. Cílem identifikace je nalézt takový model Fid , aby se průběh simulované výstupní veličiny $y_s(t)$ co nejvíce blížil skutečné výstupní veličině $y(t)$.

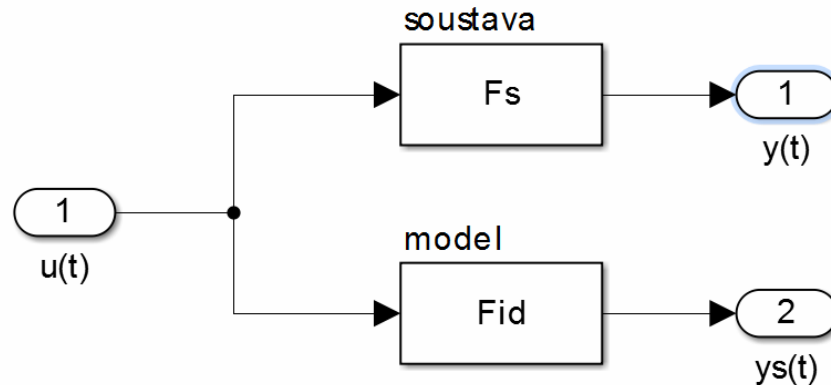
Výhodou experimentální identifikace je, že umožňuje do modelu zahrnout řadu nepříznivých faktorů, jako je změna parametrů soustavy v čase, či šum měřených veličin, které u modelu získaného matematicko-fyzikální analýzou nelze uvažovat. V úloze rotačního inverzního kyvadla jimi jsou např. vůle v motoru, pružnost plastové konstrukce.

Nevýhodou experimentální identifikace je pak samotná nutnost změřených experimentálních dat. Na soustavě tedy musí být proveditelný experiment a sledované veličiny musí být měřitelné.

Pro experiment je důležitá volba testovacího signálu. Testovací signál může být buď deterministický, jehož průběh v čase vyjádřit jako funkci času. Stochastický, náhodný signál

jedinečného a neopakovatelného charakteru (např. bílý šum). Anebo pseudonáhodný, periodický signál, jehož průběh v jedné periodě je znám.

Vyhodnocení dat může probíhat buď off-line metodou, tak že identifikace se provádí všech změřených datech najednou. Anebo online metodou, která identifikuje data postupně. Nalezený model se v průběhu identifikace neustále zpřesňuje. Výhodou této metody je, že není potřeba uchovávat všechna naměřená data (Vrožina, 2012).



Obr. 1.19 – Porovnání odezvy reálné soustavy a modelu

1.4.1 Vyhodnocení ze signálu obecného průběhu

Byl změřen průběh výstupní veličiny $y(t)$ a jemu odpovídající průběh akční veličiny $u(t)$. Data je nejprve třeba vhodně ořezat, tj. odstranit tu část dat, kdy přechází systém z neznámého stavu. Ořezaná data se pak musí transformovat do pracovního bodu, to se provádí odečtením hodnoty prvního prvku od všech prvků ve vektoru dat, pro obě veličiny zvlášť (posun dat do nuly = linearizace v pracovním bodě). Výsledný model platí jen pro okolí pracovního bodu. Ukázka úpravy dat je na obr. 1.21 (Modrlák, 2015).

Metoda nejmenších čtverců (MNC)

Metoda nejmenších čtverců slouží k nalezení optimálních odhadů parametrů modelu. Model se uvažuje ve tvaru obecné diferenciální rovnice (1.1). Odhady parametrů se získají na základě kritéria minimalizace součtu kvadrátů reziduí (1.2), tj. rozdílu skutečné a odhadnuté hodnoty výstupní veličiny. Pro odhady \hat{y} platí rovnice (1.3), tyto odhady jsou zatíženy náhodnou chybou ε .

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-1) + \dots + a_n u(k-n) \quad (1.1)$$

$$K = \sum_{i=n+1}^N [y(i) - \hat{y}(i)]^2 \rightarrow \min, \quad (1.2)$$

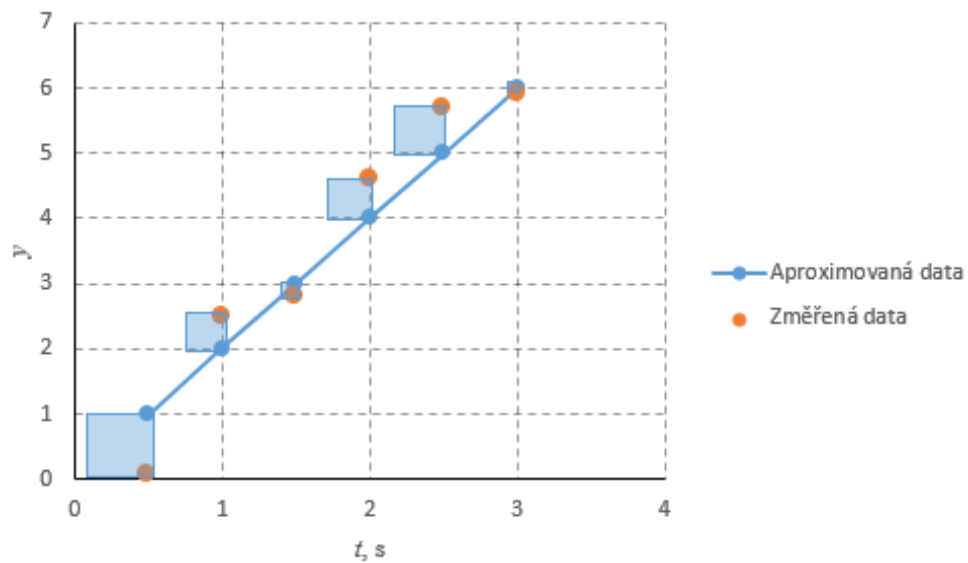
kde N je počet bodů měření,
 n – řád modelu.

$$\hat{\mathbf{y}} = \mathbf{F} \cdot \mathbf{p} + \boldsymbol{\varepsilon}, \quad (1.3)$$

Kritérium lze pro zjednodušení zapsat vektorově, řešením lze dojít ke vztahu pro výpočet parametrů modelu.

$$\mathbf{K} = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) = (\mathbf{y} - \mathbf{F} \cdot \mathbf{p})^T (\mathbf{y} - \mathbf{F} \cdot \mathbf{p}) \Rightarrow \mathbf{p} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{y}, \quad (1.4)$$

kde \mathbf{y} je vektor změřených dat
 $\hat{\mathbf{y}}$ – vektor odhadů
 \mathbf{F} – matice, viz (1.5)
 \mathbf{p} – vektor parametrů a a b .



Obr. 1.20 – Metoda nejmenších čtverců

Změřená data mají tvar

$$\mathbf{y} = [y(1), y(2), \dots, y(N)]^T$$

$$\mathbf{u} = [u(1), u(2), \dots, u(N)]^T$$

Matice \mathbf{F} v rovnici (1.5) je získána z rovnic odhadů $\hat{\mathbf{y}}$, které se získají na základě rovnice (1.1).

$$\hat{y}(k+1) = -a_1 y(k) - \dots - a_n y(k-n+1) + b_1 u(k) + \dots + b_n u(k-n+1)$$

$$\hat{y}(k+2) = -a_1 y(k+1) - \dots - a_n y(k-n+2) + b_1 u(k+1) + \dots + b_n u(k-n+2)$$

$$\hat{y}(k+N) = -a_1 y(k+N-1) - \dots - a_n y(k-n+N) + b_1 u(k+N-1) + \dots + b_n u(k-n+N)$$

Odhad výstupní veličiny \hat{y} se provede od $n+1$ kroku, aby došlo k odstranění prvních n řádků, pro které neexistují naměřená data. (např. pro $n=2$ by byl požadován vzorek $a_2 y(k-1)$, který není k dispozici, neexistuje).

$$\hat{y}(n+1) = -a_1 y(n) - \dots - a_n y(1) + b_1 u(n) + \dots + b_n u(1)$$

$$\hat{y}(n+2) = -a_1 y(n+1) - \dots - a_n y(2) + b_1 u(n+1) + \dots + b_n u(2)$$

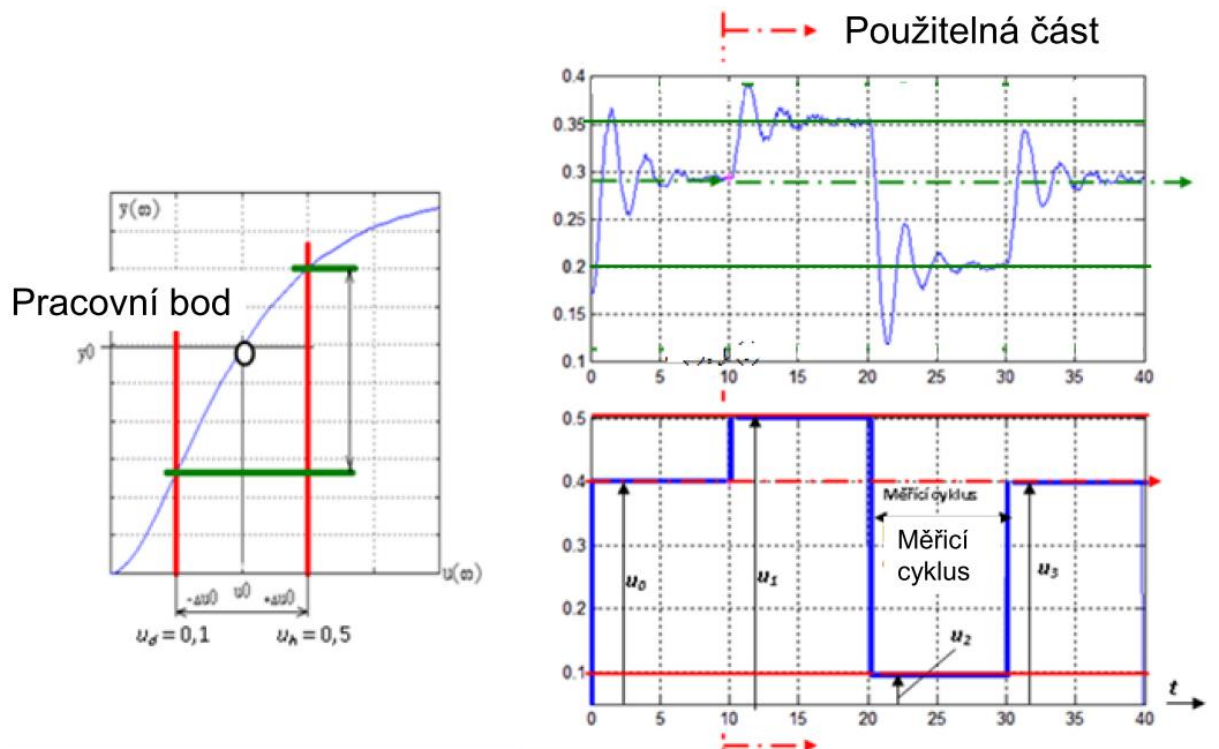
$$\hat{y}(N) = -a_1 y(N-1) - \dots - a_n y(N-n) + b_1 u(N-1) + \dots + b_n u(N-n)$$

(1.5)

$$\begin{bmatrix} \hat{y}(n+1) \\ \hat{y}(n+2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \underbrace{\begin{bmatrix} -y(n) & \dots & -y(1) & u(n) & \dots & u(1) \\ -y(n+1) & \dots & -y(2) & u(n+1) & \dots & u(2) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ -y(k+N-1) & \dots & -y(k-n+N) & u(k+N-1) & \dots & u(k-n+N) \end{bmatrix}}_{\mathbf{F}} \times \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{p}}$$

kde n je rad soustavy,

N – počet vzorků.



Obr. 1.21 – Úprava dat pro identifikaci (Modrlák, str. 18)

Funkce fminsearch

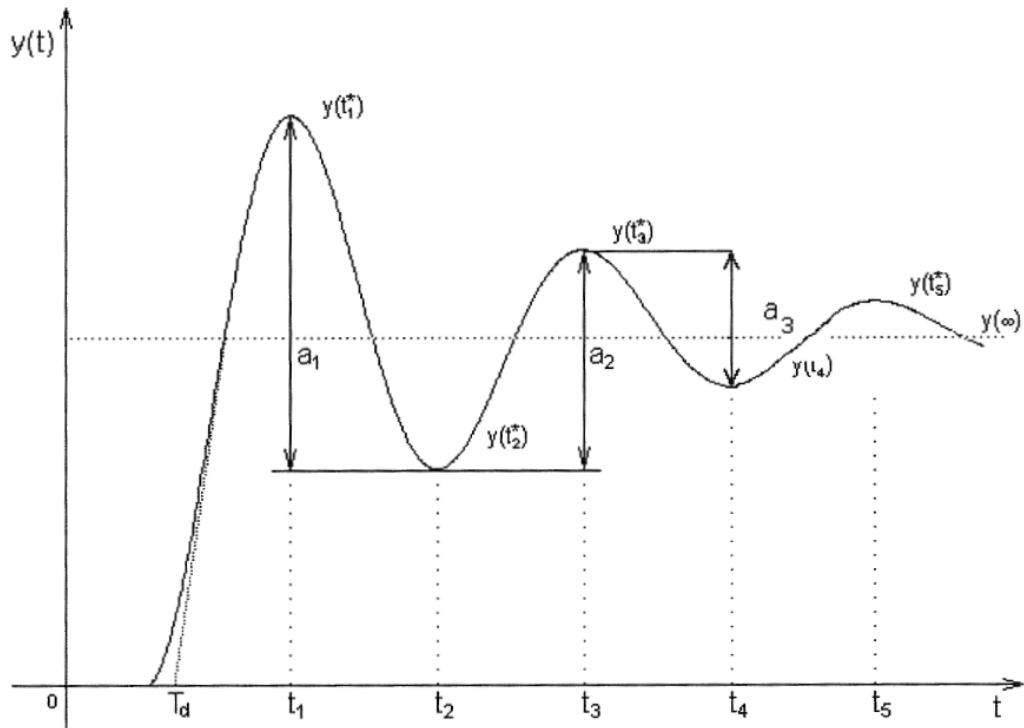
je Matlabovská funkce, která hledá minimum předané funkce více proměnných pomocí derivační metody. Hledání začíná od počátečního odhadu x_0 . Návratovou hodnotou je nalezené minimální řešení (parametry modelu).

```
% hlavička funkce fun
function K = hledanyPrenos(p,u,y,t)
% volání funkce
x = fminsearch(fun,x0);
```

Obr. 1.22 – Matlab, fminsearch

Pro hledání minimálních parametrů modelu musí funkce fun mít jako vstupní parametr odhad parametrů hledaného modelu, vektory vstupní a výstupní veličiny a vektor času. Návratovou hodnotou je kritérium K (Fminsearch, 2015).

1.4.2 Vyhodnocení z přechodové charakteristiky



Obr. 1.23 – Identifikace soustavy s kmitavou odezvou z přechodové charakteristiky (Burý, 2011)

Tato metoda umožňuje identifikovat parametry přenosu soustavy s kmitavou odezvou i dopravním zpožděním. Metoda aproximuje přenos soustavy přenosem:

$$F(s) = \frac{K \cdot e^{-T_d s}}{T_0^2 s^2 + 2\xi T_0 s + 1} \quad (1.6)$$

Proporcionální zesílení K se vypočte podle (1.7), koeficient tlumení ξ se vypočte podle vztahu (1.8), časová konstanta T_0 se vypočte ze vztahu (1.9) a dopravní zpoždění ze vztahu (1.10).

$$K = \frac{y(\infty)}{u(\infty)} \quad (1.7)$$

$$\xi^A = \frac{\ln \frac{a_{i+1}}{a_i}}{\sqrt{\pi^2 + \ln^2 \frac{a_{i+1}}{a_i}}}, \quad \xi^B = \frac{\ln \frac{a_{i+2}}{a_i}}{\sqrt{4\pi^2 + \ln^2 \frac{a_{i+2}}{a_i}}} \quad (1.8)$$

$$T_0 = \frac{1}{\omega} \sqrt{1 - \xi^2}, \quad \text{kde} \quad \frac{1}{\omega} = \frac{1}{\pi \cdot n} (t_{n+1} - t_1) \quad (1.9)$$

$$T_d = \frac{1}{n} \sum_{i=1}^n t_i - \frac{n+1}{2n} \cdot (t_{n+1} - t_1) \quad (1.10)$$

Kapitola čerpána z (Burý, 2011).

1.4.3 Vyhodnocení kvality identifikace

Kvalitou identifikace je myšlena shoda měřeného průběhu s průběhem získaným simulací na základě modelu. Kvalita identifikace bude provedena podle kritéria (1.11). Čím bude menší hodnota kritéria, tím bude model věrněji napodobovat skutečný průběh. Pro K_{id} rovno nule je identifikovaný průběh totožný s měřeným.

$$K_{id} = \frac{\sum_{i=1}^N |y_m - y_s|}{N}, \quad (1.11)$$

kde y_m je měřený průběh výchylky kyvadla, °,
 y_s – simulovaný průběh výchylky kyvadla, °,
 N – počet bodů.

1.5 ALGORITMY VHODNÉ PRO ŘÍZENÍ

V této kapitole budou popsány algoritmy používané v řídicí technice. Základní strukturou v řídicí technice je regulační obvod.

Regulační obvod se skládá ze samotné regulované soustavy **S** (např. inverzní kyvadlo), jejíž dynamické vlastnosti soustavy jsou nejčastěji popsány diferenciální rovnicí, ze které se posléze vyjádří přenos. Přenos je definován jako poměr Laplaceova obrazu výstupní a vstupní veličiny při nulových počátečních podmínkách.

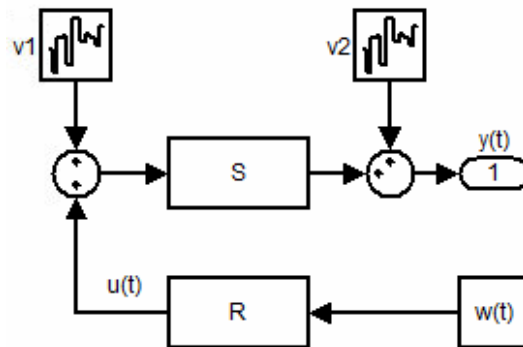
Druhý člen regulačního obvodu je pak regulátor **R**. Regulátor je určitého typu (např. PID, PSD ...) a volbou jeho parametrů lze dosáhnout požadovaný průběh regulované veličiny.

Spojením regulátoru a soustavy pak hovoříme jako o regulačním obvodu. Do regulačního obvodu může také vstupovat porucha, a to jak na výstupu soustavy $v_2(t)$, tak i na vstupu soustavy $v_1(t)$. Úkolem regulátoru je automatické řízení soustavy tak, aby výstupní, regulovaná, veličina $y(t)$ sledovala průběh požadované hodnoty $w(t)$ s nulovou, nebo minimální odchylkou i v případě působení poruchy.

Existují dva základní typy regulačních obvodů, jsou to otevřené (dopředené) a uzavřené (zpětnovazební) regulační obvody.

1.5.1 Dopředné řízení

Dopředné řízení lze použít v těch případech, kdy podrobně známe chování regulované soustavy a také známe průběh poruch. Regulátor vypočítává akční zásah $u(t)$ jen podle požadovaného průběhu žádané hodnoty a nekontroluje skutečnou hodnotu regulované veličiny. Není-li k dispozici měření poruchového signálu (v praxi neproveditelné, nebo velice obtížně), regulátor nedokáže kompenzovat poruchu. Tento způsob řízení se také nazývá ovládání. Lze použít u jednoduchých úloh, jako je např. řízení křižovatky dle známého programu, nebo ohřev na konstantní teplotu přivedením předem známého konstantního napětí (Balátě, 2004).



Obr. 1.24 – Schéma otevřeného regulačního obvodu

1.5.2 Zpětnovazební řízení

Uzavřený regulační obvod vznikne zavedením záporné zpětné vazby od regulované veličiny. Takovému regulační obvodu se nazývá zpětnovazební. Regulátor určuje akční zásah $u(t)$ na základě regulační odchylky $e(t)$, která je definovaná jako rozdíl žádané a skutečné hodnoty regulované veličiny.

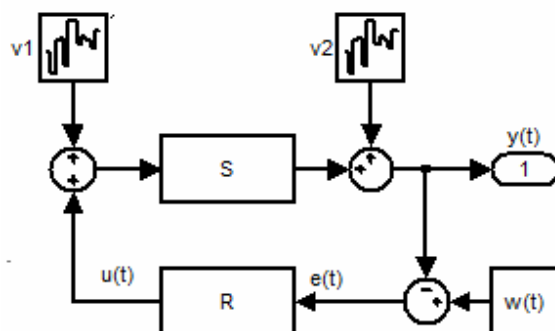
$$e(t) = w(t) - y(t), \quad (1.12)$$

kde $w(t)$ je žádaná hodnota,

$y(t)$ – skutečná hodnota.

Zpětná vazba umožňuje stabilizovat regulační pochod nestabilní soustavy, rychleji dosáhnout požadované hodnoty a odstranit neměřitelné poruchy.

Nevýhodou zpětnovazebního řízení je, že nevhodným nastavením regulátoru, nebo změnou parametrů soustavy, může dojít k nestabilitě jinak stabilní soustavy (Balátě, 2004).



Obr. 1.25 – Schéma zpětnovazebního regulačního obvodu

1.5.3 Regulační PID

Jedná se o nejčastěji používaný regulační obvod v praxi, protože PID regulační obvod je jednoduchý, univerzální a snadno realizovatelný pomocí elektronických komponentů (např. operačními zesilovači). Realizuje akční zásah $u(t)$ na základě znalosti regulační odchylky $e(t)$.

PID regulační obvod se skládá ze tří složek – proporcionální, integrační a derivační. Vynecháním některé složky lze realizovat P, PI, PD, nebo I regulační obvod.

Obecný (paralelní) tvar PID regulačního obvodu je dán vztahem (1.13) a jeho přenos (1.14).

$$u(t) = r_0 e(t) + r_{-1} \int_0^t e(\tau) d\tau + r_1 \frac{de(t)}{dt}, \quad (1.13)$$

$$R(s) = r_0 + \frac{r_{-1}}{s} + r_1 s, \quad (1.14)$$

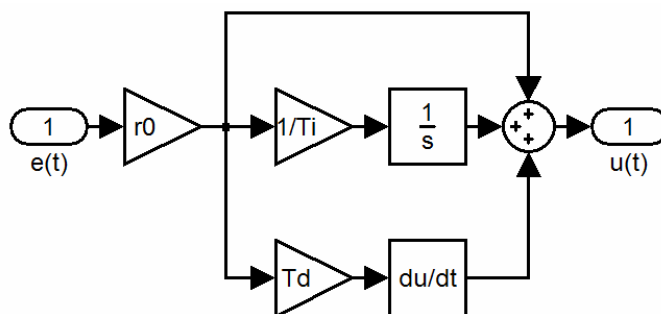
kde r_0 je zesílení regulačního obvodu,
 r_1 – derivační konstanta regulačního obvodu,
 r_{-1} - integrační konstanta regulačního obvodu.

V praxi se spíše používá standardní tvar (1.15) regulačního obvodu, který je jednodušší na realizaci. Jeho přenos je (1.16).

$$u(t) = r_0 \cdot \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right], \quad (1.15)$$

$$R(s) = r_0 \cdot \left[1 + \frac{1}{T_I s} + T_D s \right], \quad (1.16)$$

kde r_0 je proporcionální zesílení regulačního obvodu,
 T_I – integrační časová konstanta, s
 T_D – derivační časová konstanta, s.



Obr. 1.26 – Schéma PID regulátoru (standardní tvar)

Význam jednotlivých složek regulátoru

P složka (Proporcionální)

Proporcionální složka slouží jako zesílení záporné zpětné vazby. Při větším zesílení bude rychlejší regulační děj. Pro velká zesílení může být regulační děj kmitavý, až nestabilní. Samotná P složka zanechává trvalou regulační odchylku.

I složka (Integrační)

Integrační složka zajišťuje dosažení nulové regulační odchylky, po určité době. Zvyšuje ale řád soustavy a tím i zpomaluje regulační děj.

D složka (Derivační)

Derivační složka urychluje regulační děj a snižuje řád soustavy. Akční zásah je generován v závislosti na rychlosti změny regulační odchylky. V ustáleném stavu její vliv vymizí. Její nevýhodou je, že zesiluje i šum, v takovém případě je nutné použít filtraci derivační složky.

Kapitola čerpána z (Balátě, 2004).

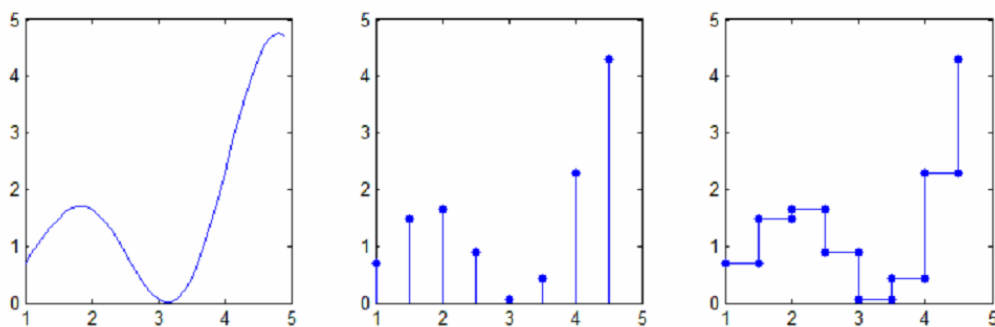
1.5.3.1 Diskretizace PID

Diskrétní řízení pracuje, na rozdíl od spojitého, s diskrétními signály. Spojitý signál se převádí na diskrétní v několika krocích.

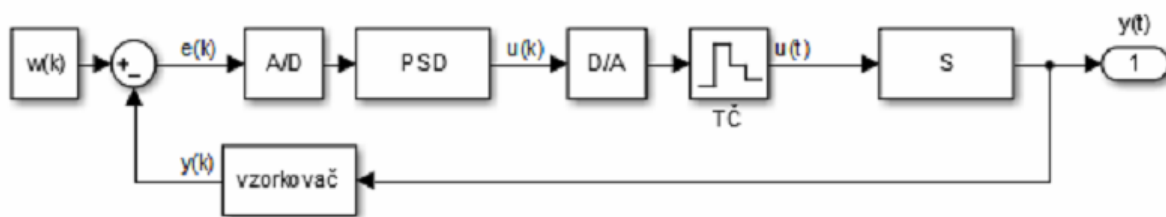
Zprv se signál vzorkuje s danou vzorkovací periodou T . Volba vzorkovací frekvence může podstatně ovlivnit regulační děj. Existuje řada způsobů jak volit vzorkovací periodu (jedna desetina největší časové konstanty soustavy, jedna šestina až jedna patnáctina doby, za kterou přechodová charakteristika dosáhne 95% ustálené hodnoty). Příliš dlouhá vzorkovací perioda destabilizuje regulační děj, neboť při vzorkování dochází ke ztrátě informace. U soustav náchylných na šum nesmí být vzorkovací perioda ani příliš krátká, protože by se s regulovaným signálem vzorkoval i šum. Šum lze do značné míry potlačit vhodnou filtrací (dolní propust',

klouzavý průměr). Obecně lze říci, že vhodná vzorkovací perioda je taková, při které nedojde ke zhoršení kvality regulace o více jak 15% (Balátě, 2004, s. 480).

Navzorkovaný signál se poté převádí A/D převodníkem na posloupnost binárních hodnot, které vstupují do číslicového regulátoru. Výsledný akční zásah je taktéž posloupnost binárních hodnot, které se pomocí D/A převodníku transformují na posloupnost impulzů. Tyto impulzy se poté tvarují, nejčastěji tvarovačem nultého řádu, na schodovitý signál.



Obr. 1.27 – Spojitý signál (levý), vzorkování (prostřední) a tvarování (pravý)



Obr. 1.28 – Schéma diskrétního regulačního obvodu

Při realizaci diskrétního PID regulátoru je nutné nahradit spojitý integrál a derivaci za jejich diskrétní podobu. Takováto regulátor se poté nazývá PSD regulátor (proporcionálně sumačně diferencní).

Integrál lze nahradit sumou. Existují tři varianty, jak takovou náhradu provést. A to buď zpětnou obdélníkovou náhradou (ZOBD) (1.17), dopřednou obdélníkovou náhradou (DOBH) (1.18), anebo lichoběžníkovou náhradou (LICH) (1.19).

$$I(kT) = \int_0^t e(\tau) d\tau \approx T \cdot \sum_{i=1}^k e(iT), \quad (1.17)$$

$$I(kT) = \int_0^t e(\tau) d\tau \approx T \cdot \sum_{i=0}^{k-1} e(iT), \quad (1.18)$$

$$I(kT) = \int_0^t e(\tau) d\tau \approx T \cdot \sum_{i=1}^k \frac{e(iT) + e[(i-1)T]}{2}, \quad (1.19)$$

kde T je interval vzorkování, s,
 N – je celkový počet vzorků.

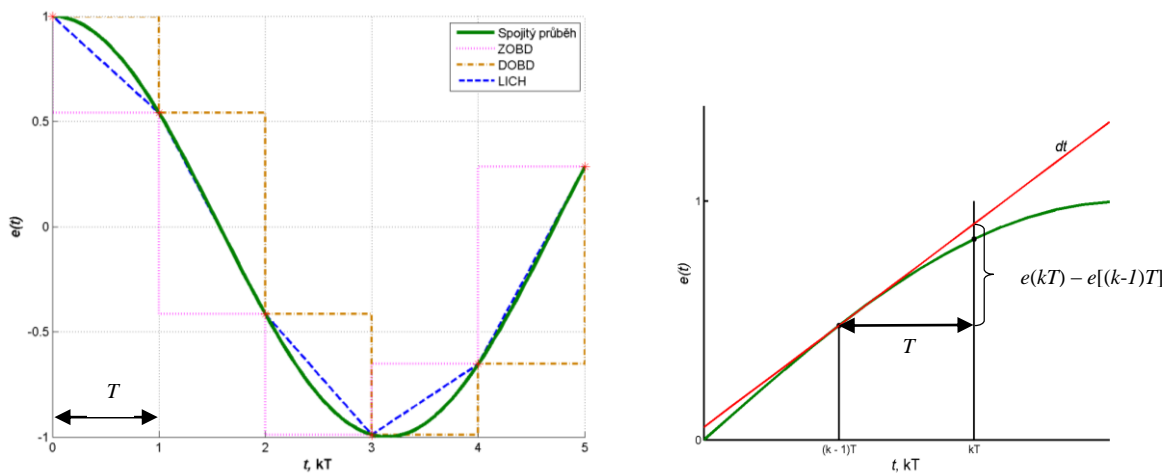
Náhrada derivace se provádí pomocí zpětné diference 1. řádu.

$$D(kT) = \frac{de(kT)}{dt} \approx \frac{e(kT) - e[(k-1)T]}{T}, \quad (1.20)$$

kde T je interval vzorkování, s,
 $e(kT)$ – aktuální regulační odchylka,
 $e[(k-1)T]$ – minulá regulační odchylka.

Rovnici diskrétního PSD regulátoru (1.21) je získána z rovnice PID regulátoru (1.15) nahrazením derivace zpětnou diferencí (1.20) a integrálu metodou ZOBD (1.17).

$$u(kT) = r_0 \cdot \left[e(kT) + \frac{T}{T_I} \cdot \sum_{i=1}^k e(iT) + \frac{T_D}{T} \cdot [e(kT) - e((k-1)T)] \right] \quad (1.21)$$



Obr. 1.29 – Způsoby náhrady integrálu (vlevo) a náhrada derivace (vpravo)

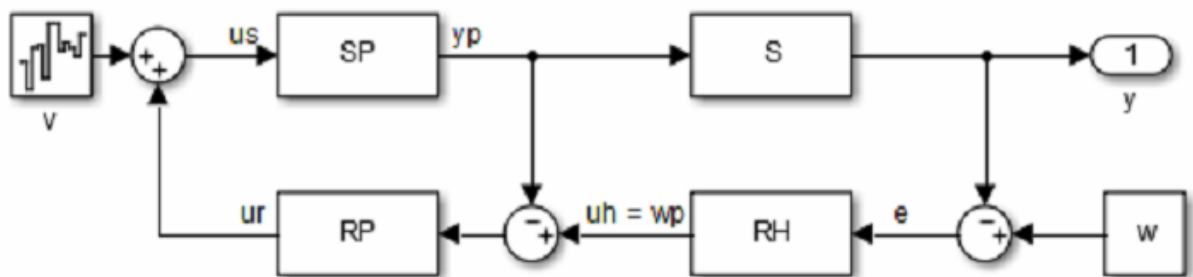
Kapitola čerpána z (Balátě, 2004).

1.5.4 Regulační obvod s pomocnou regulovanou veličinou (kaskádní regulace)

Je jedním z typů rozvětvených regulačních obvodů. Tyto obvody se využívají při regulaci soustav vyšších řádů s cílem dosáhnout lepší stability a kvality regulačního pochodu. Rozvětvené regulační obvody mají četné zastoupení v průmyslových aplikacích.

Regulační obvod s pomocnou regulovanou veličinou, označovaný také jako kaskádní, nebo vlečná regulace, vzniká zavedením pomocné regulované veličiny y_p . Pomocná veličina musí být měřitelná a musí podstatně rychleji reagovat na vstupy poruch, než hlavní regulovaná veličina y . Celý regulační obvod je rozdělen na dva regulační obvody spojené do kaskády, tak, že akční zásah vypočtený hlavním regulátorem RH je vstupem pro pomocný regulátor RP. Regulace v pomocném regulačním obvodě probíhá rychleji, než ve zbývajícím regulačním obvodu a umožňuje tak rychleji reagovat na poruchy v vstupující do soustavy a vykompenzovat jejich vliv dříve, než se projeví na hlavní regulované veličině, výsledná regulace je kvalitnější a stabilnější (Balátě, 2004).

V mém případě inverzního kyvadla tedy pomocný regulátor RP drží výchylku ramene na konstantní pozici, zatímco hlavní regulátor RH mění žádanou hodnotu pozice podle úhlu vychýlení kyvadla.



Obr. 1.30 – Regulační obvod s pomocnou regulovanou veličinou

1.5.5 Seřízení regulátoru metodou Ziegler-Nichols

Tato metoda je založena buď na znalosti přenosu soustavy, nebo experimentálně na reálném zařízení. Nastavení parametrů regulátoru vychází z určení kritických hodnot uzavřeného regulačního obvodu na mezi stability. Postup určení kritických parametrů je následující. Regulátoru se vyřadí vliv integrační a derivační složky, ($T_D = 0$, $T_I = \infty$) a následně se zvyšuje proporcionální zesílení, dokud se soustava nepřivede na mez stability. Na výstupu se objeví ustálené kmity s konstantní amplitudou. Toto zesílení je kritické zesílení r_{0k} . Regulátor lze nahradit pomocí relé, jehož výhodou je, že nemůže dojít k nekontrolovatelnému rozkmitání

soustavy. Kritické zesílení se pak vypočte podle (1.22). Perioda ustálených kmitů se nazývá kritická perioda T_k . Seřízení parametrů regulátoru se provede podle tab. 1.2.

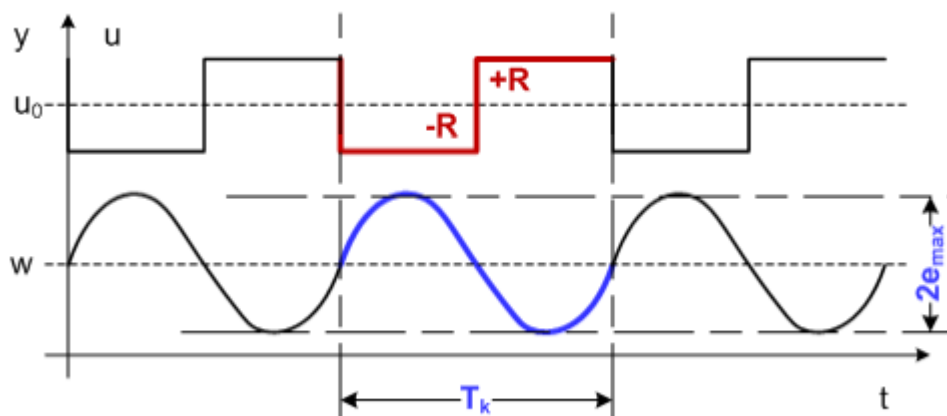
$$r_{0k} = \frac{4R}{\pi \cdot e_{\max}}, \quad (1.22)$$

kde R je hodnota spínaná v relé,

e_{\max} – regulační odchylka při ustálených kmittech.

Tab. 1.2 – Seřízení regulátoru metodou \bar{z}

	r_0	T_I	T_D
P	$0,5 \cdot r_{0k}$	-	-
PI	$0,45 \cdot r_{0k}$	$0,85 \cdot T_k$	-
PD	$0,4 \cdot r_{0k}$	-	$0,05 \cdot T_k$
PID	$0,6 \cdot r_{0k}$	$0,5 \cdot T_k$	$0,125 \cdot T_k$



Obr. 1.31 – Ziegler-Nichols metoda (Dušek,2013)

Kapitola čerpána z (Balátě, 2004).

1.5.6 Metoda požadovaného modelu (Inverze dynamiky)

Jedná se o jednoduchou metodu seřízení parametrů regulátoru na základě znalosti modelu soustavy a požadovaného modelu uzavřeného regulačního obvodu ve tvaru:

$$G_{WY}(s) = \frac{Y(s)}{W(s)} = \frac{a}{s + a \cdot e^{-T_d s}} \cdot e^{-T_d s}, \quad (1.23)$$

kde a je zesílení otevřeného regulačního obvodu,

T_d – dopravní zpoždění, s.

Zesílení otevřeného regulačního obvodu lze získat analyticky pouze pro nekmitavé soustavy a soustavy na mezi stability. Pro ostatní typy soustav byla provedena počítačová simulace a byly stanoveny konstanty α a β , v závislosti na dopravním zpoždění T_d . Tyto konstanty jsou uvedeny v tab. 1.3. Metoda zajišťuje dosažení nulové regulační odchylky a volbu relativního překmitu přechodové charakteristiky v rozmezí 0 – 50 %.

Pro výpočet parametrů regulátoru musí být přenos regulované soustavy v jednom ze základních tvarů, které jsou uvedeny v tab. 1.4.

Tab. 1.3 – závislost α a β na relativním překmitu κ

κ	0	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
α	1,282	0,984	0,884	0,832	0,763	0,697	0,669	0,640	0,618	0,599	0,577
β	2,718	1,944	1,720	1,561	1,437	1,337	1,248	1,172	1,104	1,045	0,992

Tab. 1.4 – Nastavení parametrů regulátoru metodou požadovaného modelu

Přenos soustavy		Doporučené nastavení regulátoru		
		r_0	T_I	T_D
1	$\frac{k_1}{s} \cdot e^{-T_d s}$	$\frac{1}{(\alpha T + \beta T_d)k_1}$	–	–
2	$\frac{k_1}{T_1 s + 1} \cdot e^{-T_d s}$	$\frac{T_I}{(\alpha T + \beta T_d)k_1}$	$T_1 - \frac{T}{2}$	–
3	$\frac{k_1}{s(T_1 s + 1)} \cdot e^{-T_d s}$	$\frac{1}{(\alpha T + \beta T_d)k_1}$	–	$T_1 - \frac{T}{2}$
4	$\frac{k_1}{(T_1 s + 1)(T_2 s + 1)} \cdot e^{-T_d s}$ $T_1 \geq T_2$	$\frac{T_I}{(\alpha T + \beta T_d)k_1}$	$T_1 + T_2 - T$	$\frac{T_1 \cdot T_2}{T_1 + T_2} - \frac{T}{4}$
5	$\frac{k_1}{T^2 s^2 + 2\xi T s + 1} \cdot e^{-T_d s}$ $0,5 \leq \xi \leq 1$	$\frac{T_I}{(\alpha T + \beta T_d)k_1}$	$2\xi T_0 - T$	$\frac{T_0}{2\xi} - \frac{T}{4}$

kde T je vzorkovací perioda, s (pro spojité systémy se uvažuje $T = 0$).

Kapitola čerpána z (Balátě, 2004).

1.5.7 Vyhodnocení kvality regulace

Kvalita regulace bude vyhodnocena kritériem velikosti absolutní regulační plochy – IAE (integral absolute error), které je vhodné pro kmitavé pochody. Opět lze říci, že čím menší bude kritérium IAE , tím bude regulační pochod lepší. V ideálním případě by bylo kritérium nulové, tedy kyvadlo by setrvalo v požadované poloze.

$$IAE = T \sum_{i=1}^N |w(iT) - y(iT)| \approx \int_0^t |w(\tau) - y(\tau)| d\tau, \quad (1.24)$$

kde $w(i)$ je žádaná hodnota natočení kyvadla (vždy 0),

$y(i)$ – měřená hodnota natočení kyvadla,

N – počet změřených bodů,

T – vzorkovací perioda.

Kapitola čerpána z (Ščevík, 2015).

2 ŘEŠENÍ

2.1 POPIS INVERZNÍHO KYVADLA

Pro stavbu rotačního inverzního kyvadla byla použita stavebnice LEGO Mindstorms NXT, ke které bylo nutné dokoupit senzor pro měření úhlu natočení kyvadla. Rameno na obr. 2.1 má délku 11,8 cm a celkovou hmotnost 52 g.



Obr. 2.1 – Rameno kyvadla

Tab. 2.1 – Parametry ramene

Celková délka	11,8 cm
Celková hmotnost	52 g
Hmotnost senzoru	16 g
Rozměry senzoru (d x š x v)	43,3 x 23,8 x 20,1 mm

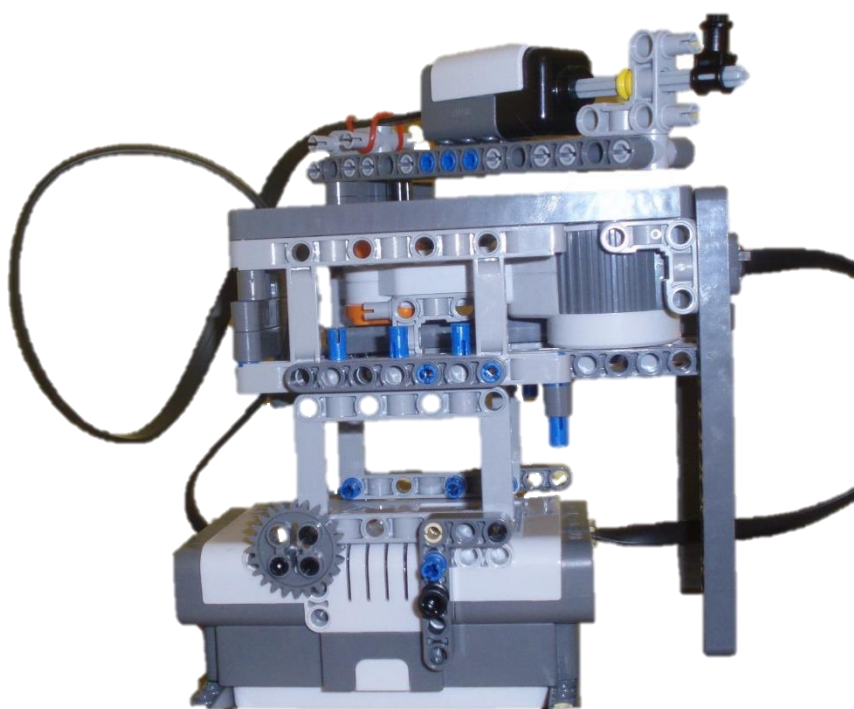
Kyvadlo je tvořeno spojením tří plastických tyčinek o celkové délce 29,5 cm. Na vrcholu je uchyceno závaží o hmotnosti 4 g. Delší tyč kyvadla a těžší závaží jsou lépe regulovatelné, avšak plastické části stavebnice se se zvětšující délkou tyče a hmotností závaží stávají více pružnými a tedy i hůře regulovatelnými.



Obr. 2.2 – Kyvadlo

Tab. 2.2 – Parametry kyvadla

Celková délka	295 mm
Celková hmotnost	12 g
Hmotnost	12 g
Poloměr závaží	15,3 mm
Výška závaží	20 mm
Hmotnost závaží	4 g



Obr. 2.3 – Inverzní rotační kyvadlo

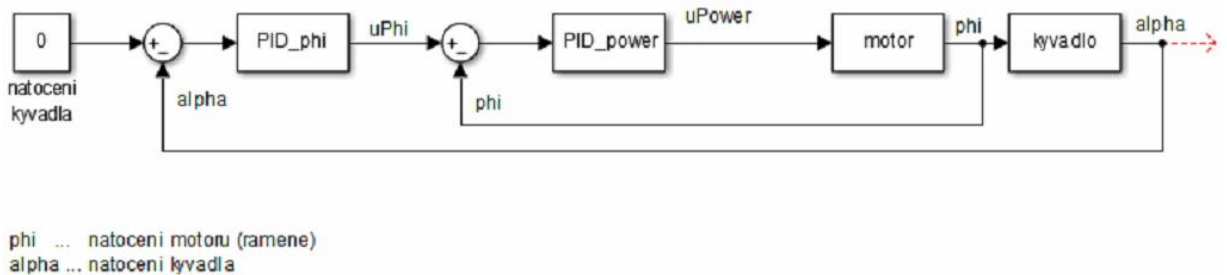
Výsledné rotační inverzní kyvadlo sestavené ze stavebnice LEGO Mindstorms je ukázáno na obr. 2.3. Sestavené kyvadlo vykazuje poměrně velkou vůli mezi převody servo-motoru, osou motoru lze pootočit o 10° aniž by došlo k pohybu samotného motoru. Při pohybu ramene dochází k jeho prohýbání a kmitání ve vertikálním směru. Tyč kyvadla je z pružného materiálu, plastu, a při pohybu dochází k jejímu ohýbání. Z těchto důvodů a nedostatků, bylo rozhodnuto získat model systému pomocí experimentální identifikace.

2.2 EXPERIMENTÁLNÍ IDENTIFIKACE KYVADLA

Experimentální identifikaci kyvadla byla provedena tím způsobem, že nejprve byly experimentálně nastaveny regulátory PID_power a PID_phi z obr. 2.4, tak aby bylo kyvadlo vyregulováno v horní svislé poloze. Naměřená data byla poté upravena a identifikována.

2.2.1 Měření dat

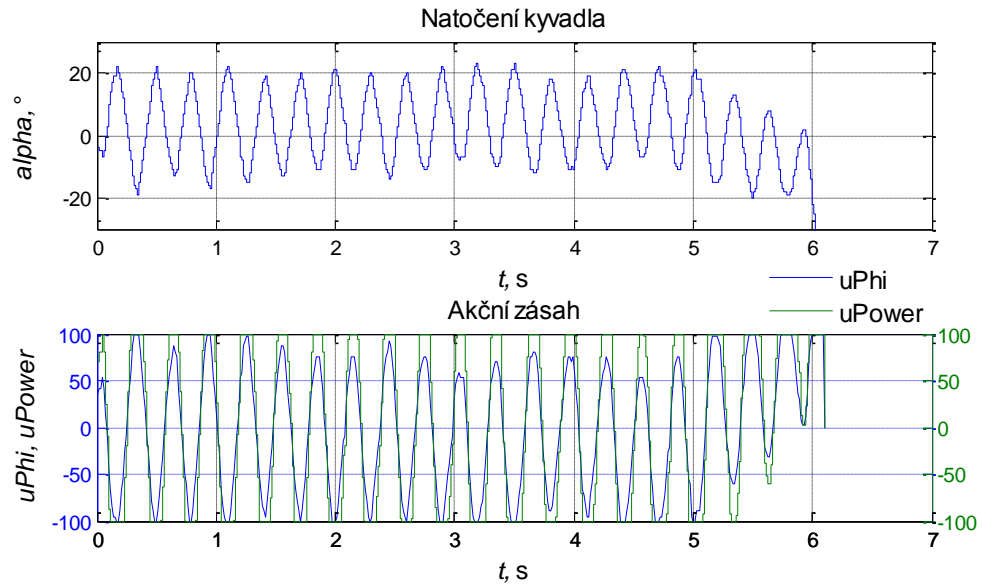
Rotační inverzní kyvadlo je nestabilní astatická soustava. Pro získání dat, kdy se kyvadlo nachází horní svislé pozici, byl použit kaskádní regulační obvod se dvěma PID regulátory. Schéma regulačního obvodu je na obr. 2.4. Vnější regulátor PID_phi realizuje akční zásah na základě aktuálního úhlu natočení kyvadla α . Jeho výstupem je takové natočení ramene u_{Φ} , aby kyvadlo servalo v horní poloze. Regulátor PID_power na základě tohoto požadovaného úhlu natočení ramene určí akční zásah pro motor u_{Power} . Parametry regulátorů byly nastaveny metodou „pokus omyl“. Nejprve jsem nastavil proporcionální zesílení r_0 , při neaktivní integrační i derivační složce. Vzhledem k tomu, že soustava má integrační charakter, byla integrační složka ponechána neaktivní. Poté jsem hledal vhodné nastavení derivační složky. Nejprve jsem nastavil samotný regulátor PID_power tak, aby se motor co nejrychleji natočil do požadované pozice. Poté jsem nastavil regulátor PID_phi. Výsledné hodnoty parametrů obou regulátorů jsou uvedeny v tab. 2.3.



Obr. 2.4 – Schéma regulačního obvodu, použité pro měření dat

Tab. 2.3 – Experimentální nastavení regulátorů

Regulátor	r_0	T_I	T_D
PID_power	1,6	∞	0,005
PID_phi	56	∞	3,7



Obr. 2.5 – Změřená data

2.2.2 Úprava dat před identifikací

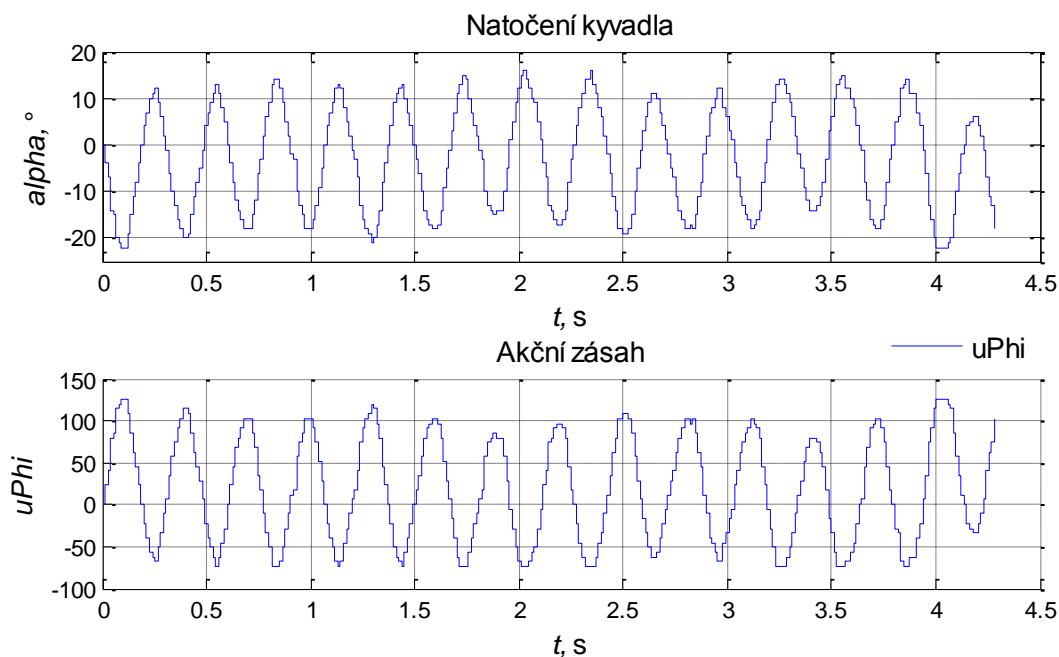
Změřená data na obr. 2.5 je před začátkem identifikace nutné vhodně upravit. Nejprve je nutné data ořezat na obou koncích. Oříznutím na začátku se odstraní ta část dat, kdy bylo kyvadlo ve svislé poloze přidržováno. Odstraní se tím tedy ta část dat, kdy soustava přecházela z neznámého stavu. Ořezáním dat na konci pochodu dojde k odstranění těch dat, kdy došlo k pádu kyvadla.

Nakonec se ořezaná data transformují do pracovního bodu (posun do počátku) odečtením hodnoty prvního změřeného bodu, podle (2.1). Nulový akční zásah tak způsobí nulový výstup.

$$u_{id}(t) = u(t) - u_0 \quad (2.1)$$

$$y_{id}(t) = y(t) - y_0$$

kde u_{id} , y_{id} jsou data použitá pro identifikaci,
 $y(t)$, $u(t)$ – změřená data,
 y_0 , u_0 – první hodnoty dat (pracovní bod).



Obr. 2.6 – Upravená data

2.2.3 Identifikace

Model identifikované soustavy popisuje, o jaký úhel je třeba otočit s ramenem, aby kyvadlo zůstalo v horní poloze. Vstupní data pro identifikace jsou úhel natočení kyvadla $alpha$ a natočení ramene $uPhi$.

Identifikace metodou nejmenších čtverců

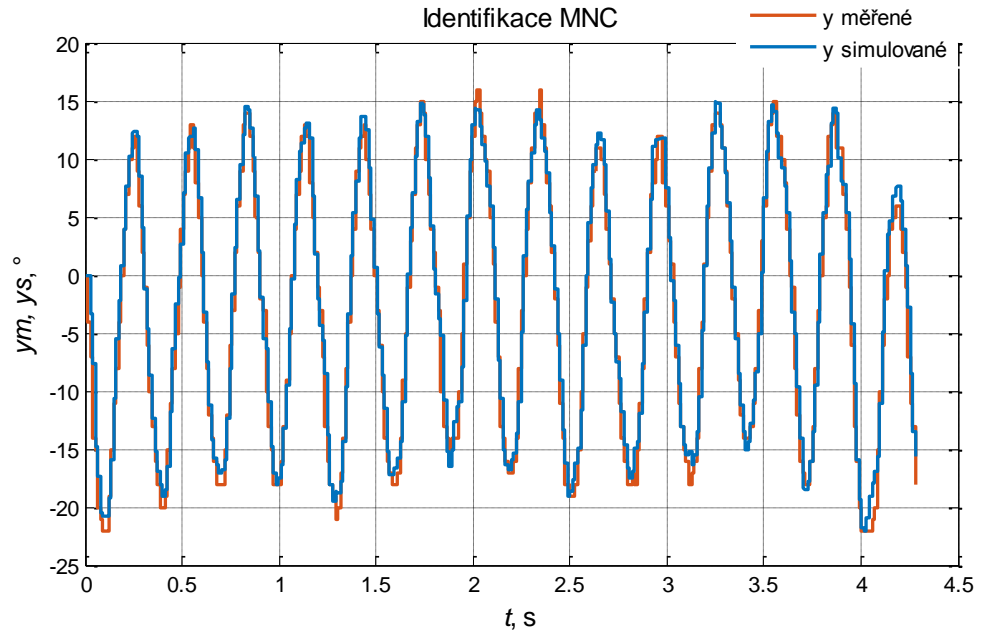
Metodou matematicko-fyzikální analýzy je obvykle získán model čtvrtého řádu (Cpsdg, 2015; Jiyue, 2015), proto byl pro identifikaci metodou nejmenších čtverců také zvolen diskretní model popsáný přenosem 4. řádu (2.2) se vzorkovací periodou $T = 0,0133$ s. Shoda měřeného a simulovaného průběhu je na obr. 2.7, ze kterého je vidět, že model dobře kopíruje naměřená data. Průběh odchylky modelu od měřených dat je na obr. 2.8.

Podle (1.11) byla vypočtena kvalita modelu $K_{id} = 1,188$.

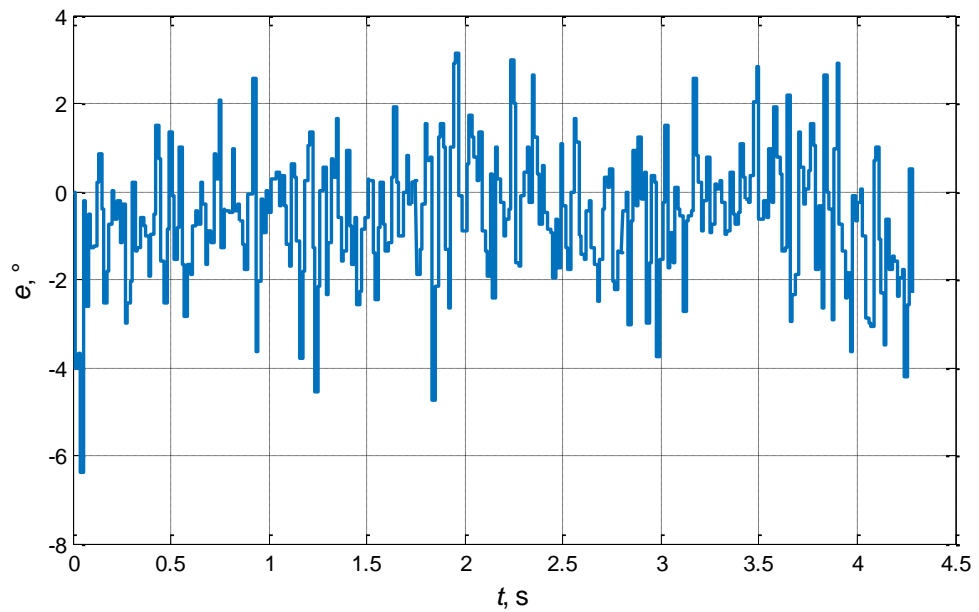
$$F_D(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}, \quad (2.2)$$

kde

$$\begin{aligned} a_1 &= -4,184 \cdot 10^{-1}, & b_1 &= -2,512 \cdot 10^{-3}, \\ a_2 &= -8,606 \cdot 10^{-3}, & b_2 &= -3,511 \cdot 10^{-4}, \\ a_3 &= 1,275 \cdot 10^{-1}, & b_3 &= 5,489 \cdot 10^{-4}, \\ a_4 &= -7,582 \cdot 10^{-2}, & b_4 &= 7,818 \cdot 10^{-4}. \end{aligned}$$



Obr. 2.7 – Identifikace MNC



Obr. 2.8 – Průběh odchylky měřených dat a modelu získaného metodou MNC

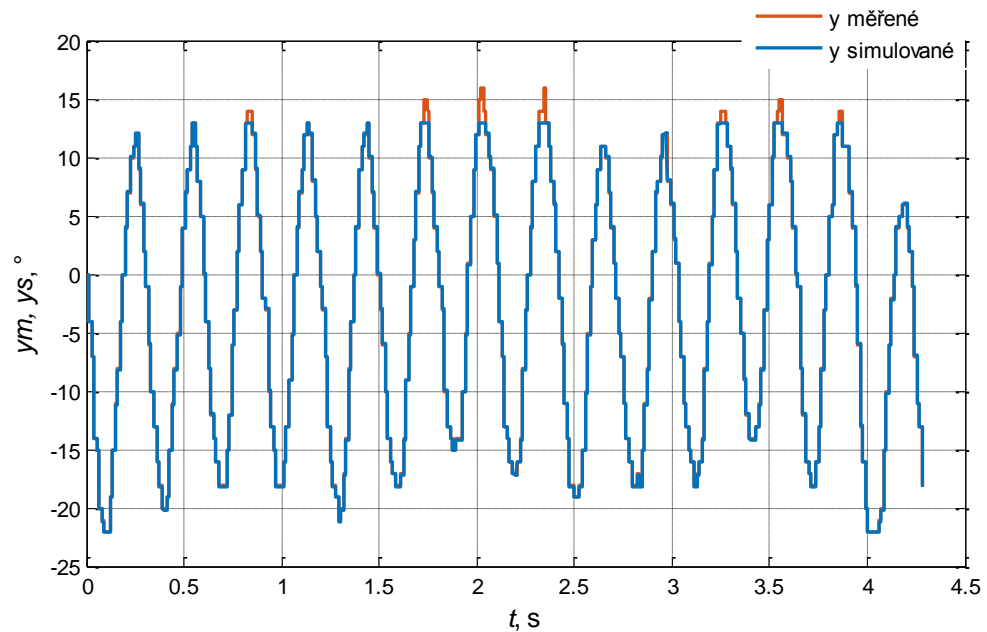
Identifikace pomocí fminsearch

Pro metodu požadovaného modelu nejlépe odpovídá pátý model (2.3) z tab. 1.4. Shoda měřeného a simulovaného průběhu je na obr. 2.9, ze kterého je vidět, že získaný model velice dobře kopíruje naměřená data.

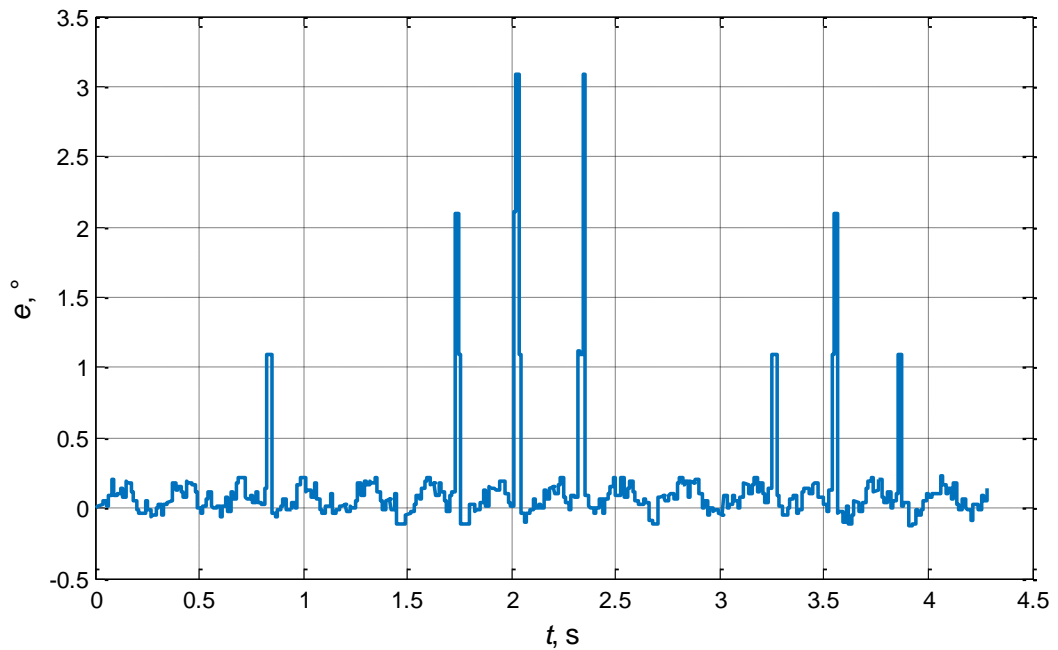
Vypočtená kvalita modelu $K_{id} = 0,153$.

$$\frac{k_1}{T_0^2 s^2 + 2\xi T_0 s + 1} \cdot e^{-T_d s}, \quad (2.3)$$

kde $k_1 = -3,1 \cdot 10^{-3}$, $T_0 = 7,0301 \cdot 10^{-6}$,
 $\xi = 0,7457$, $T_d = 1,6672 \cdot 10^{-4}$.



Obr. 2.9 – Identifikace požadovaným modelem



Obr. 2.10 – Průběh odchylky měřených dat a modelu získaného metodou pož. modelu

Úprava modelu získaného metodou MNC do požadovaného tvaru

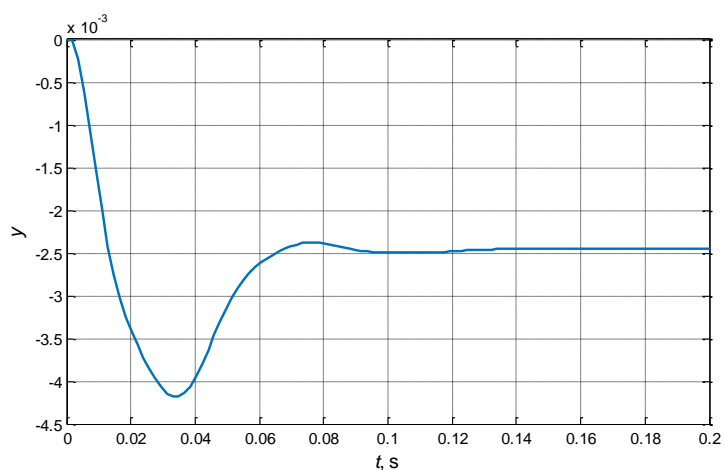
Parametry přenosu (2.3) lze získat vyhodnocením přechodové charakteristiky modelu (2.2) dosazením odečtených parametrů a_1 , a_2 a t_1 , t_2 z obr. 2.12 do vzorců (1.7) až (1.10).

Tab. 2.4 – Parametry přechodové charakteristiky

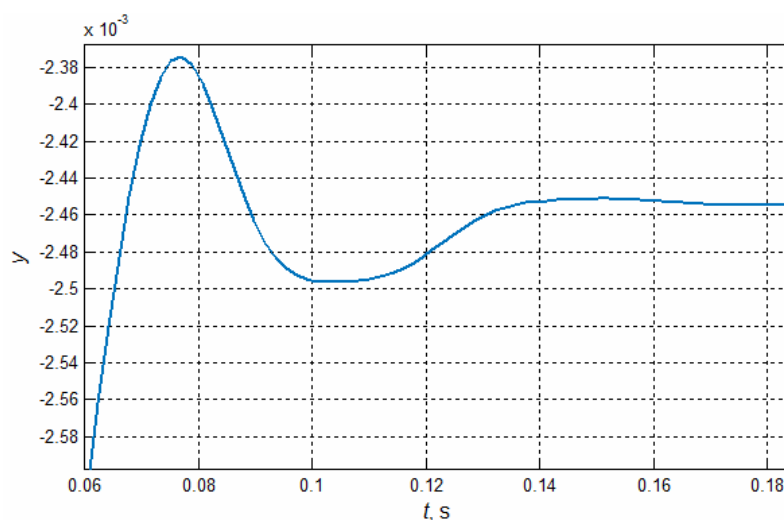
Parametr	Hodnota	Parametr	Hodnota
a_1	$-1,1826 \cdot 10^{-4}$	t_1	$7,8659 \cdot 10^{-2}$
a_2	$-4,4875 \cdot 10^{-5}$	t_2	$1,1518 \cdot 10^{-1}$

Tab. 2.5 – Vypočtené parametry přenosu (2.3)

Parametr	Hodnota	Parametr	Hodnota
k_1	$-2,45 \cdot 10^{-3}$	ζ	0,29
T_0	$1,11 \cdot 10^{-2}$	T_d	$6,03 \cdot 10^{-2}$

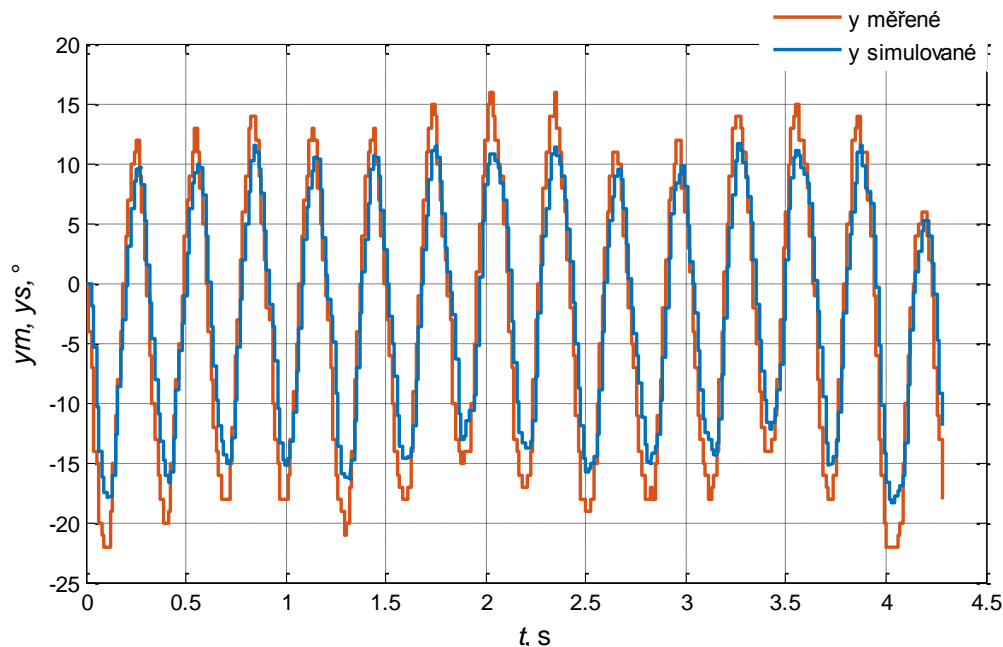


Obr. 2.11 – Vyhodnocovaná přechodová charakteristika

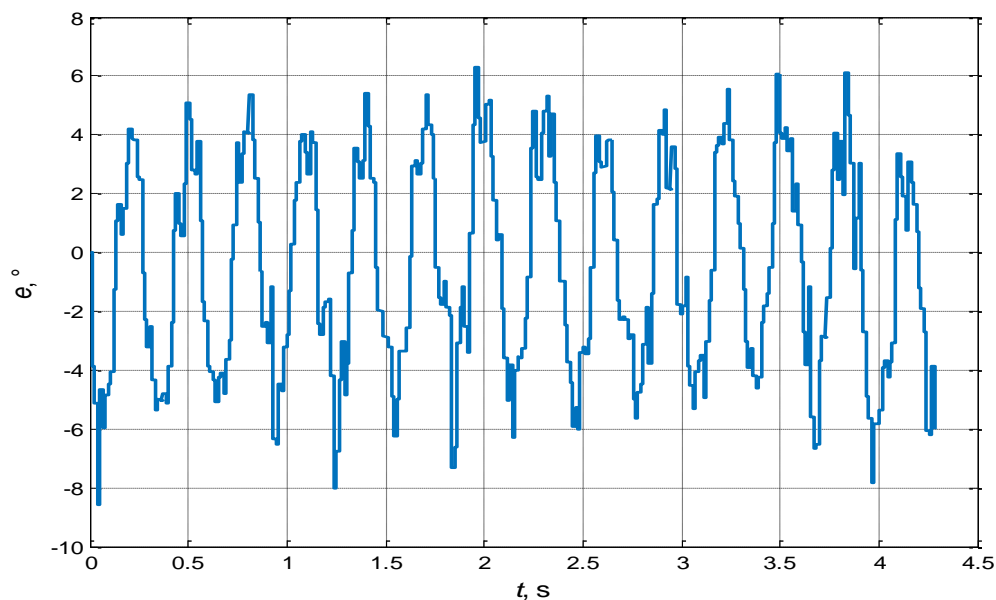


Obr. 2.12 – Zvětšený úsek odečtu hodnot

Protože vypočtené dopravní zpoždění T_d je příliš velké a vnáší do simulované odezvy velký posun, bude jeho hodnota desetinásobně snížena na hodnotu $T_d = 6,03 \cdot 10^{-3}$ s. Shoda měřeného a simulovaného průběhu je na obr. 2.13. Vypočtená kvalita modelu $K_{id} = 3,196$.



Obr. 2.14 – Identifikace z přechodové charakteristiky



Obr. 2.13 – Průběh odchylky měřených dat a modelu získaného z přechodové charakteristiky

2.2.4 Vyhodnocení identifikovaných modelů

Z identifikovaných průběhů a vypočtených hodnot kritérií v tab. 2.6 vyplývá, že nejlepší shoda mezi měřeními a simulovanými daty je u modelu (2.3) získaného funkcí `fminsearch`.

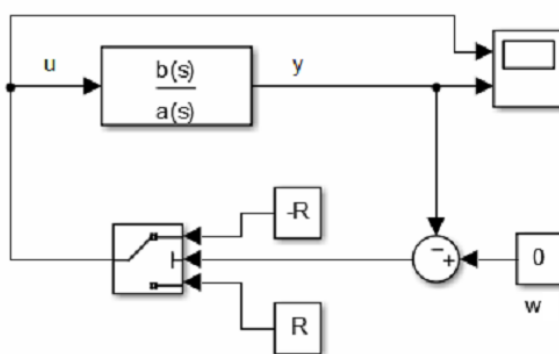
Tab. 2.6 – Kritéria pro jednotlivé metody identifikace

Metoda	Kritérium
MNC	1,188
fminsearch	0,153
Z přechodové char.	3,196

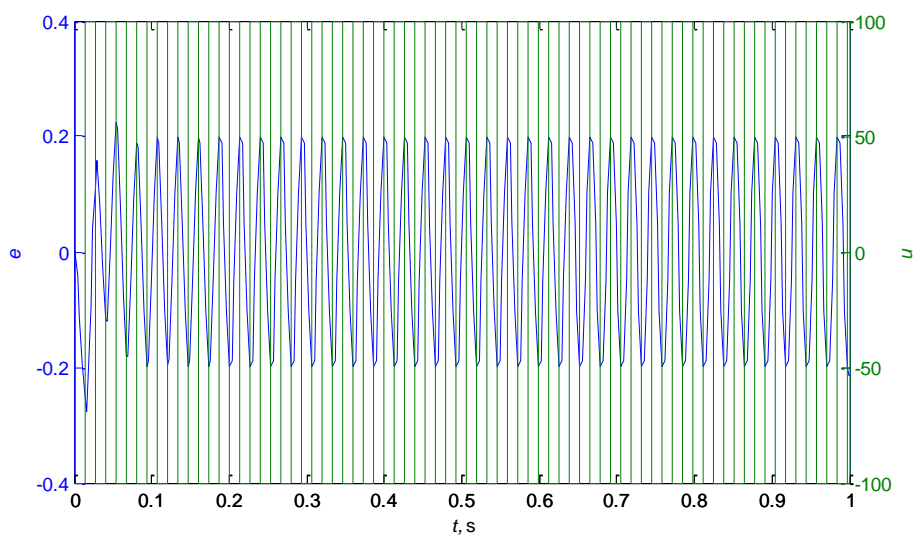
2.3 URČENÍ PARAMETRŮ REGULÁTORU

2.3.1 Nastavení parametrů regulátoru metodou Ziegler-Nichols

Určení parametrů regulátoru z modelu (2.2).



Obr. 2.15 – Relé ve zpětné vazbě



Obr. 2.16 – Odezva modelu na relé ve ZV

Tab. 2.7 – Odečtené parametry

R	100
e_{\max}	0,19679
T_k	$2,66 \cdot 10^{-2}$

Z velikosti regulační odchylky e_{\max} je potřeba vypočíst kritické zesílení r_{0k} dle vztahu (1.22) a následně vypočíst parametry pro PID regulátor podle třetího řádku tab. 1.2.

$$r_{0k} = \frac{4R}{\pi \cdot e_{\max}} = \frac{4 \cdot 100}{\pi \cdot 0,19679} = 647 \quad (2.4)$$

$$P = 0,4 \cdot r_{0k} = 0,4 \cdot 647 = 258,8 \quad (2.5)$$

$$D = 0,05 \cdot T_k = 0,05 \cdot 2,66 \cdot 10^{-2} = 1,33 \cdot 10^{-3} \quad (2.6)$$

Tab. 2.8 – Parametry PID regulátoru získané ZN metodou

P	I	D
258,8	∞	$1,33 \cdot 10^{-3}$

2.3.2 Nastavení parametrů regulátoru metodou požadovaného modelu

Pro metodu požadovaného modelu byla naměřená data identifikována pomocí funkce `fminsearch` pro všech pět tvarů přenosu z tab. 1.4. Nejlépe charakteru soustavy odpovídá pátý tvar přenosu (2.3). Na tento tvar byl také upraven přenos získaný metodou nejmenších čtverců, parametry modelu jsou uvedeny v tab. 2.5. U této soustavy rotačního inverzního kyvadla není možné dosáhnout regulačního pochodu bez překmitu, budou vypočteny parametry pro relativní překmit $\kappa = 25\%$ a 50% . Řízením z Matlabu lze dosáhnout vzorkovací periody $T = 0,0133$ s.

Model získaný identifikací

$$\frac{-3,1 \cdot 10^{-3}}{(7,03 \cdot 10^{-6})^2 s^2 + 2 \cdot 0,75 \cdot 7,03 \cdot 10^{-6} s + 1} \cdot e^{-1,67 \cdot 10^{-4} s} \quad (2.7)$$

Model získaný aproximací z přechodové charakteristiky

$$\frac{-2,45 \cdot 10^{-3}}{(1,11 \cdot 10^{-2})^2 s^2 + 2 \cdot 0,29 \cdot 1,11 \cdot 10^{-2} s + 1} \cdot e^{-6,03 \cdot 10^{-3} s} \quad (2.8)$$

Pro nastavení regulátoru pro překmit $\kappa = 25\%$ jsou z tabulky tab. 1.3 odečteny hodnoty $\alpha = 0,697$ a $\beta = 1,337$. Následně se provede výpočet podle tab. 1.4.

$$P = \frac{I}{(\alpha T + \beta T_d)k_1} = \frac{I}{(0,697 \cdot 0,0133 + 1,337 \cdot 1,67 \cdot 10^{-4}) \cdot (-3,1 \cdot 10^{-3})} = 451,6 \quad (2.9)$$

$$I = 2\xi T_0 - T_s = 2 \cdot 0,75 \cdot 7,03 \cdot 10^{-6} - 0,0133 = -0,0133 \quad (2.10)$$

$$\frac{T_0}{2\xi} - \frac{T_s}{4} = \frac{7,03 \cdot 10^{-6}}{2 \cdot 0,75} - \frac{0,0133}{4} = -0,0033 \quad (2.11)$$

Tab. 2.9 – Vypočtené parametry regulátoru metodou požadovaného modelu

$\kappa, \%$	P	I	D
Model (2.7)			
25	451,6	-0,0133	-0,0033
50	546,9	-0,0133	-0,0033
Model (2.8)			
25	158,4	$-6,7 \cdot 10^{-3}$	$1,56 \cdot 10^{-2}$
50	201,1	$-6,7 \cdot 10^{-3}$	$1,56 \cdot 10^{-2}$

3 DOSAŽENÉ VÝSLEDKY

3.1 VZORKOVACÍ PERIODA

Vzorkovací perioda pro regulaci rotačního inverzního kyvadla by měla být volena co nejnižší. Vzorkovací periodu v tomto případě významně ovlivňují tři faktory a to čtení natočená kyvadla, čtení natočení ramene (potažmo motoru) a nastavení motoru, které může být realizováno buď přes NXTMotor objekt, nebo přímým přístupem. Bylo provedeno 2000 volání příslušných funkcí a do tabulky tab. 3.1 jsou zaznamenány příslušné doby vykonání. Pro ovládání motoru jsem zvolil přímý přístup. Průměrná doba vykonání funkcí je $1,288 \cdot 10^{-2}$ s. S ohledem na celý regulační cyklus jsem zvolil vzorkovací periodu $T = 0,0133$ s.

Tab. 3.1 – Doby vykonání funkcí Matlabu

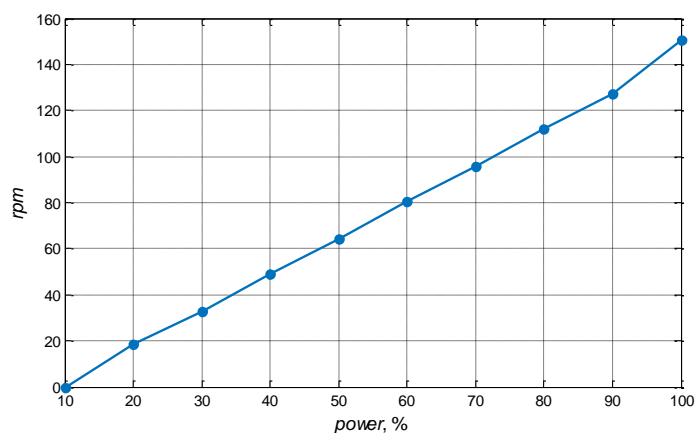
Volaná funkce	Čas, s		
	Průměrná	Minimální	Maximální
Čtení ze senzoru	$8,971 \cdot 10^{-3}$	$7,846 \cdot 10^{-3}$	$3,496 \cdot 10^{-2}$
Čtení z motoru	$2,080 \cdot 10^{-3}$	$1,594 \cdot 10^{-3}$	$5,681 \cdot 10^{-2}$
Ovládání motoru (objekt)	$1,595 \cdot 10^{-2}$	$1,547 \cdot 10^{-2}$	$2,906 \cdot 10^{-2}$
Ovládání motoru (přímo)	$1,827 \cdot 10^{-3}$	$1,466 \cdot 10^{-3}$	$5,405 \cdot 10^{-2}$

3.2 NXT SERVO MOTOR A ANGLE SENZOR

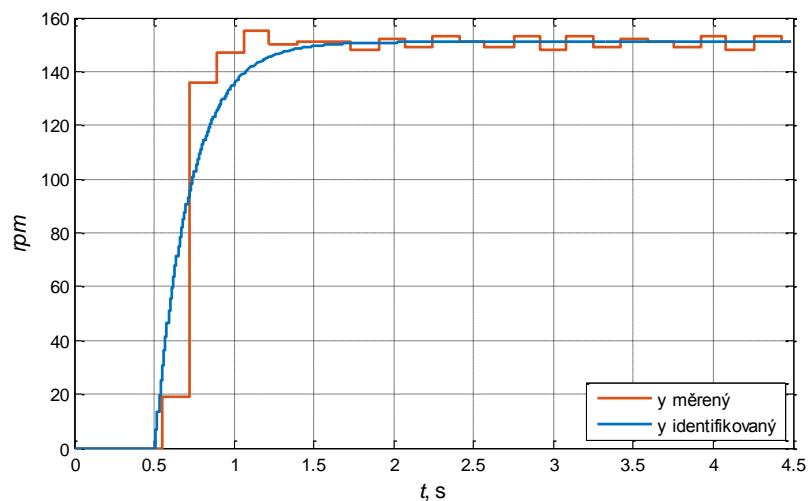
V horní svislé pozici je úhel natočení α roven 227° . Aby bylo možné uvažovat žádanou hodnotu 0° je tento offset odečítán od aktuálně změřené hodnoty.

NXT servo motor zatížený ramenem s kyvadlem má práh necitlivosti $U_0 = 10$. Maximální otáčky motoru dosahují 150 rpm. Závislost rychlosti otáčení na procentní úrovni výkonu motoru je lineární, tato závislost je znázorněna na obr. 3.1. Odezvu motoru na skokovou změnu z úrovně 0% na 100% znázorňuje obr. 3.2. Pro ilustraci byl motor identifikován soustavou prvního řádu s výsledným přenosem.

$$s = \frac{K}{T_0 s + 1} = \frac{1,61}{0,238s + 1} \quad (3.1)$$



Obr. 3.1 – Statická charakteristika NXT motoru



Obr. 3.2 – Odezva NXT motoru na skok do plných otáček

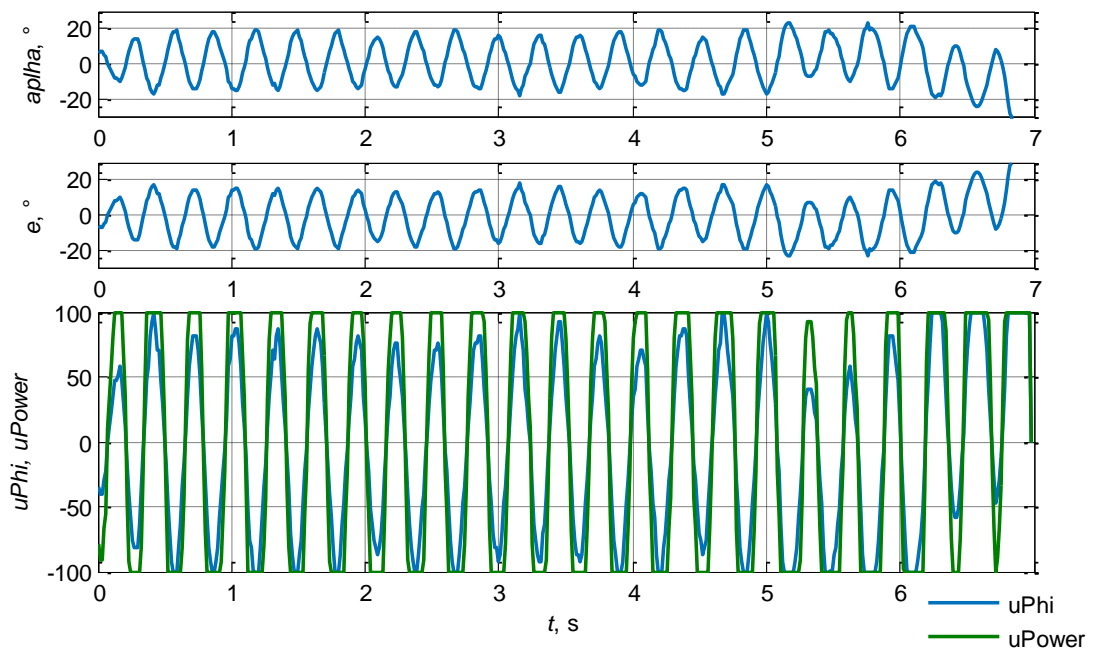
3.3 REGULAČNÍ POCHOD

3.3.1 Experimentální nastavení regulátoru

Na obr. 3.3 jsou zobrazeny průběhy výchylky kyvadla a akčních zásahů při experimentálním nastavení regulátoru. Nastavení regulátorů PID_power a PID_phi jsou uvedeny v tab. 2.3. Nastavení regulátoru PID_power je pro každou metodu nastavení stejné. Nalezené parametry nastavení v následujících metodách jsou vždy pro regulátor PID_phi.

Doba regulace s tímto nastavením se pohybuje okolo 6 sekund. Naměřená data jsou uložena na příloženém CD ve složce Naměřená data\EX pod názvem data_mereni_EX_2.xlsx (resp. data_nastaveni_EX_2.mat).

Vypočtená hodnota kritéria kvality regulace $IAE = 77,38$.



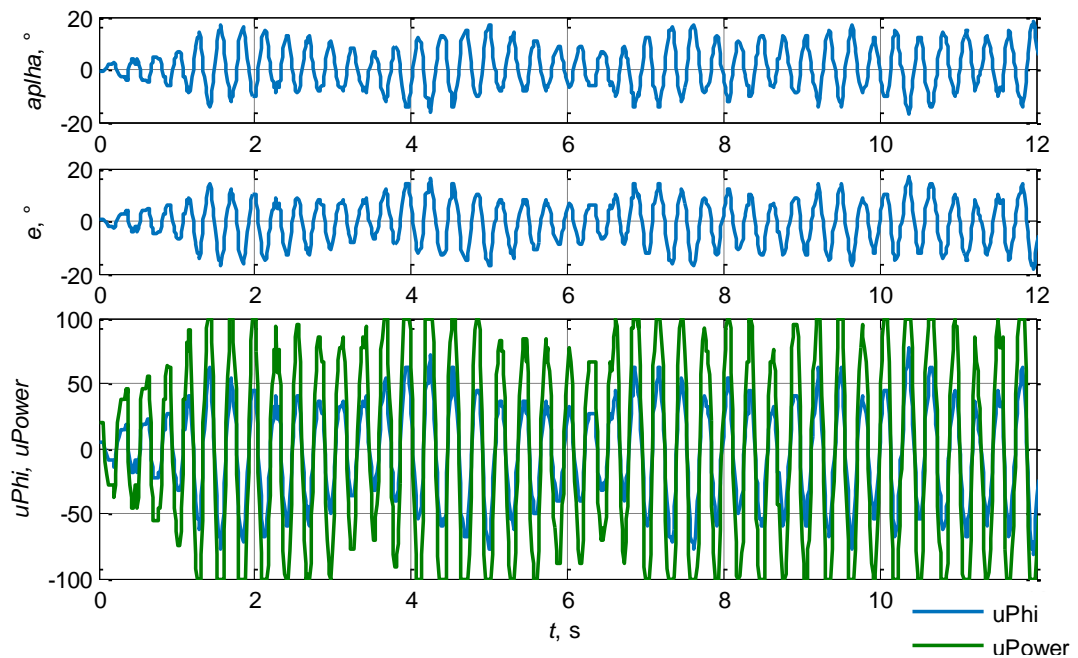
Obr. 3.3 – Průběh regulačního pochodu po experimentálním seřízení

3.3.2 Nastavení regulátoru ZN metodou

Na obr. 3.4 jsou zobrazeny částečné průběhy regulačního pochodu o délce 12 s. Naměřená data z celého regulačního pochodu jsou uložena na přiloženém CD ve složce Naměřená data\ZN pod názvem data_mereni_ZN_1.xlsx (resp. data_nastaveni_ZN_1.mat). Celková doba regulace byla 20 s.

Regulace s tímto nastavením dokáže kyvadlo uregulovat v horní svislé poloze. Byly provedeny tři opakovaní měření. Kyvadlo se pohybuje s odchylkou $\pm 20^\circ$ a maximální doba regulace nepřesahuje 20 s. Pokud kyvadlo začne dosahovat výchylky v okolí 0° , regulátor generuje malé akční zásahy, kdy se kyvadlo nedokáže „přehoupnout“ na opačnou stranu. Rameno se pomalu otáčí jedním směrem, dokud nedojde k pádu kyvadla.

Vypočtená hodnota kritéria kvality regulace $IAE = 45,59$.



Obr. 3.4 – Průběh regulačního pochodu po seřízení metodou Ziegler-Nichols

3.3.3 Nastavení regulátoru metodou požadovaného modelu

Doporučené nastavení regulátoru v tab. 2.9. Vzhledem k tomu, že u nastavení vycházejícího z modelu (2.7) vyšla záporná integrační i derivační složka, byly tyto parametry vynechány (derivační složka položena rovna nule, integrační složka rovna nekonečno). Regulace samotným P-regulátorem takovéto soustavy nebyla možná. U nastavení vycházejícího z modelu (2.8) byla vypnuta pouze integrační složka. Regulační pochod s tímto nastavením nevykazuje dobré výsledky a kyvadlo poměrně rychle padá.

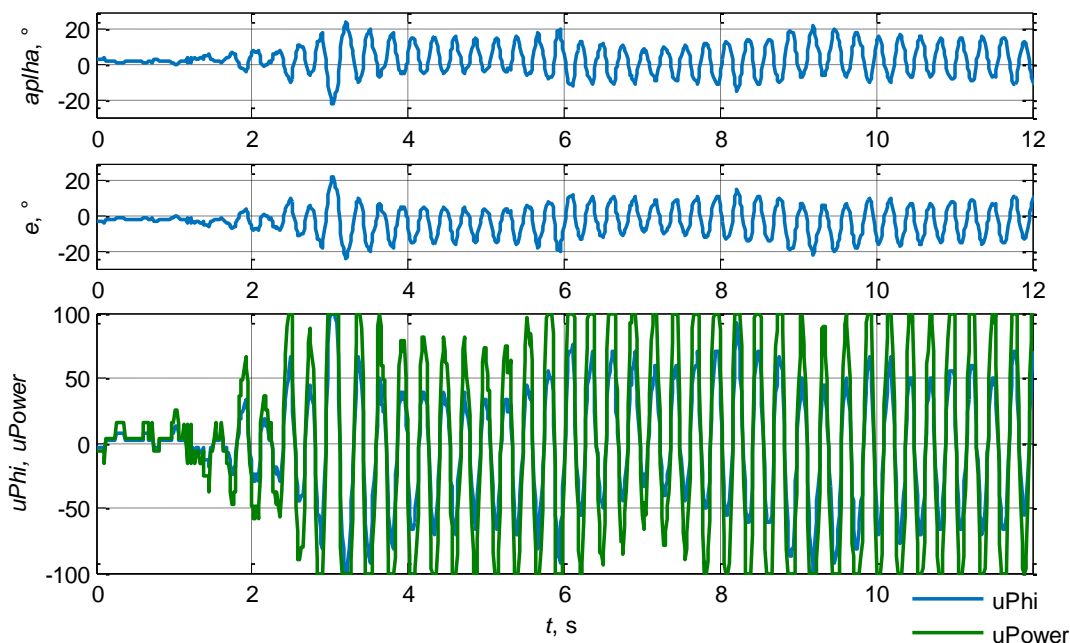
Nejllepší chování bylo dosaženo při nastavení vycházejícího z modelu (2.8) a volbě relativního překmitu $\kappa = 25\%$. Pro zlepšení regulačního pochodu jsem kvůli nedostatečné velikosti doporučeného proporcionálního zesílení tuto hodnotu zvýšil na 305,1. Použité parametry pro nastavení regulátoru PID_phi jsou uvedeny v tab. 3.2.

Na obr. 3.5 jsou částečné průběhy regulačního pochodu o délce 12 s. Naměřená data z celého regulačního pochodu jsou uložena na příloženém CD ve složce Naměřená data\PM pod názvem data_mereni_metodaPM_5.xlsx (resp. data_Nastaveni_PM_5.mat). Celková doba regulace byla přibližně 90 s.

Vypočtená hodnota kritéria kvality regulace $IAE = 43,78$.

Tab. 3.2 – Použité nastavení regulátoru metodou požadovaného modelu

P	I	D
305,1	∞	$1,58 \cdot 10^{-2}$



Obr. 3.5 – Průběh regulačního pochodu po seřízení metodou požadovaného modelu

3.4 VYHODNOCENÍ KVALITY REGULAČNÍCH POCHODŮ

Pro uvedené regulační pochody jsem vypočítal hodnotu kritéria IAE pro část naměřených dat o délce přibližně 7 s. Vypočtené hodnoty jsou uvedeny v tab. 3.3. Z uvedené tabulky vyplývá, že nejlepších výsledků bylo dosaženo při seřízení regulátoru PID_phi metodou požadovaného modelu. Nicméně seřízení pro všechny metody zanechává regulační odchylku o velikosti přibližně $\pm 20^\circ$. Toto chování je pravděpodobně důsledkem velké vůle mezi převody servo motoru a nastavením parametrů regulátoru nebylo možné jej odstranit.

Tab. 3.3 – Vyhodnocení kvality regulace

Metoda	Kritérium	Celková doba regulace, s
Experimentální seřízení	77,38	6,98
Seřízení podle Ziegler-Nichols	45,59	20,38
Seřízení metodou požadovaného modelu	43,78	94,01

4 ZÁVĚR

V této práci byla sestavena soustava rotačního inverzního kyvadla z dílů stavebnice Lego Minstorms NXT. Tato stavebnice neobsahovala senzor natočení, který bylo nutné dokoupit u společnosti HiTechnic. Pro sestavené kyvadlo bylo poté třeba získat model. Data pro identifikaci byla získána tak, že bylo provedeno experimentální seřízení regulátorů a na takto regulované soustavě bylo provedeno měření. Metodou experimentální identifikace byly získány tři modely, které svým chováním odpovídají naměřenému průběhu výchylky kyvadla. Spočtené hodnoty kritéria kvality identifikace jsou uvedeny v tab. 2.6. Pomocí získaných modelů byly stanoveny parametry regulátoru. Pro jednotlivé metody byl poté několikrát spuštěn regulační cyklus a naměřená data uložena. Vypočtená kritéria kvality regulace jsou uvedena v tab. 3.3.

Nejlepší chování bylo dosaženo při seřízení parametrů metodou požadovaného modelu s celkovou dobou regulace přibližně 90 s. Byly získány dva záznamy regulačního pochodu s obdobným výsledkem (PM\data_Nastaveni_PM_4.mat a PM\data_Nastaveni_PM_5.mat). Tyto výsledky ovšem nelze libovolně opakovat, protože kyvadlo je nutné manuálně přemístit do horní svislé polohy a poté záleží, jak se regulátor „chytne“.

Mezi další faktory nepříznivě ovlivňující regulační pochod patří senzor natočení kyvadla i motoru, které mají velmi nízké rozlišení, a to $\pm 1^\circ$, což spolu se vzorkovací periodou snižuje reakční schopnost regulátoru. Mezi převody servo motoru existuje velká vůle, která způsobuje, že nebylo možné dosáhnout menší regulační odchylky. Kyvadlo dosahovalo odchylky až $\pm 20^\circ$. Dalším problémem je fakt, že kyvadlo je tvořeno trojicí plastických tyčinek spojených dohromady, které se při pohybu kyvadla deformují. Kyvadlo tak různě pruží a ohýbá se. Soustava se stává velmi těžce říditelnou.

K výše vyobrazeným datům nebyla zachycena videa. Pro ilustraci byla vytvořena dvě videa zachycující průběh regulačního pochodu, která jsou uložena adresáři Obrázky.

Zdrojové kódy a naměřená data jsou uloženy na příloženém CD.

Literatura

- BALÁTĚ, J. 2004. *Automatické řízení. 2. přeprac. vyd.* Praha: BEN - technická literatura, 654 s. ISBN 8073001489.
- BURÝ, A. 2011. *Modelování a simulace dynamických systémů.* Technická universita Ostrava.
- ĎAĎO, S. 1996. *Senzory a měřicí obvody. 1. vyd.* Praha: ČVUT, 315 s. ISBN 80-01-01500-9.
- DUŠEK, F. 2013. *Teorie řídicích systémů.* Universita Pardubice: Osobní sdělení.
- CPSDG. 2015. *Cyber Physical Systems Design Group. Rotary Inverted Pendulum Example.* University of California. [online]. [cit 2015-04-18]. Dostupné z: <http://cps.ics.uci.edu/research/rotary-inverted-pendulum-example/#modeling>
- CVEJN, J. 2010. *Řízení procesů.* Univerzita Pardubice: Osobní sdělení.
- FMINSEARCH. 2015. *MathWorks documentation.* [online]. [cit 2015-03-29]. Dostupné z: <http://www.mathworks.com/help/matlab/ref/fminsearch.html>
- HITECHNIC. 2015. *NXT Angle Sensor.* [online]. [cit 2015-03-02]. Dostupné z: <https://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NAA1030>
- JAKEŠ, M. 2013. *Řídicí jednotka NXT* [online]. [cit. 2015-02-20]. Dostupné z: <https://lego.zcu.cz/web/ridici-jednotka>
- JIYUE. 2015. *Inverted rotary pendulum.* [online]. [cit. 2015-02-17]. Dostupné z: http://dasl.mem.drexel.edu/~jiyueHe/rotary_inverted_pendulum/index.html
- MODRLÁK, O.; HUBKA, L. 2015. *Identifikace systémů: Přednáška 1* [online]. Technická univerzita v Liberci [cit. 2015-03-29]. Dostupné z: <http://www.fm.tul.cz/esf0247/index.php?download=973>
- NI ELVIS. 2015. *Rotary Inverted Pendulum Trainer for NI ELVIS.* [online]. [cit. 2015-02-17]. Dostupné z: <http://www.ni.com/white-paper/10865/en/>
- QUANSER. 2015. *Rotační inverzní kyvadlo.* [online]. [cit. 2015-02-17]. Dostupné z: http://www.quanser.com/Products/rotary_pendulum
- QUANSER DOUBLE. 2015. *Dvojité rotační inverzní kyvadlo.* [online]. [cit. 2015-02-17]. Dostupné z: http://www.quanser.com/Products/rotary_double_pendulum
- PHIL, 2015. *NXT motor internals.* [online]. [cit 2015-04.13]. Dostupné z: <http://www.philohome.com/nxtmotor/nxtmotor.htm>
- REXCONTROLS. 2015. *Furuta Pendulum.* [online]. [cit. 2015-02-17]. Dostupné z: <https://www.rexcontrols.cz/mechatronicke-modely-furuta-pendulum>
- RWTH. 2015. *Functions by Category.* [online]. [cit 2015-03-02]. Dostupné z: <http://www.mindstorms.rwth-aachen.de/documents/downloads/doc/version-4.03/categories.html>
- ŠČEVÍK, P. 2015. *Číslicová regulace.* Technická univerzita Ostrava. [online]. [cit. 2015-04-15]. Dostupné z: <http://books.fs.vsb.cz/cislicovaregulace/>
- VROŽINA, M.; JANČÍKOVÁ, Z.; DAVID, J. 2012. *Identifikace systémů.* Technická univerzita Ostrava.

Přílohy

A – CD

B – Obrázky kyvadla

C – Zdrojové kódy

Příloha k diplomové práci

Návrh a řízení inverzního rotačního kyvadla

Dominik Papp

OBSAH CD

Obsah adresáře

Vlastní text práce je zde uložen ve formátech docx a pdf:

PappD_NávrhAŘízeníInverzníhoRotačníhoKyvadla_PD_2015.docx,

PappD_NávrhAŘízeníInverzníhoRotačníhoKyvadla_PD_2015.pdf.

Identifikace – tento adresář obsahuje zdrojové kódy pro Matlab, které byly použity pro identifikaci soustavy.

Obrázky – tento adresář obsahuje obrázky kyvadla a dva ilustrační video záznamy.

Naměřená data – tento adresář obsahuje naměřená data z regulačních pochodů.

Regulace - tento adresář obsahuje zdrojové kódy pro Matlab, které byly použity pro regulaci soustavy.

Příloha k diplomové práci

Návrh a řízení inverzního rotačního kyvadla

Dominik Papp

OBRÁZKY KYVADLA

Seznam obrázků

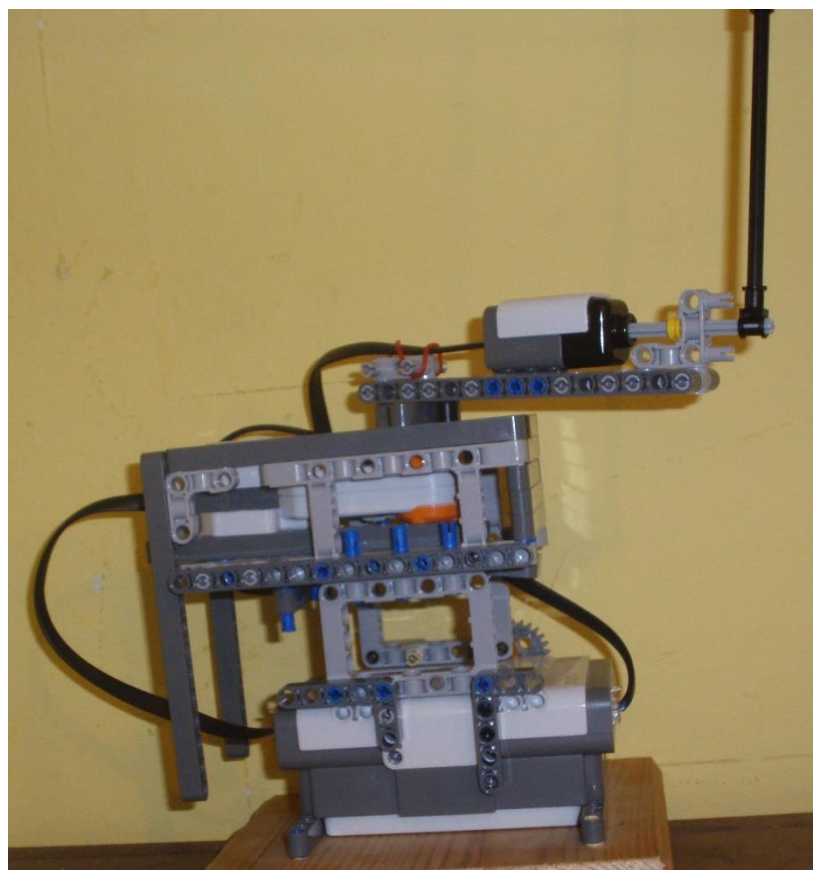
Obr. B.1 – Inverzní kyvadlo, pohled z boku.....	B-2
Obr. B.2 – Inverzní kyvadlo, pohled shora.....	B-2
Obr. B.3 – Inverzní kyvadlo, bližší pohled z boku.....	B-3



Obr. B.1 – Inverzní kyvadlo, pohled z boku



Obr. B.2 – Inverzní kyvadlo, pohled shora



Obr. B.3 – Inverzní kyvadlo, bližší pohled z boku

Příloha k diplomové práci

Návrh a řízení inverzního rotačního kyvadla

Dominik Papp

ZDROJOVÉ KÓDY

```

classdef PIDregulator < handle
properties (SetAccess = private, Hidden=true)
    P; % zesileni regulatoru
    I; % integracni konstanta
    D; % derivacni konstanta
    Ts; % interval vzorkovani
    sumE; % suma regulacnich odchylek
    uMax; % maximalni akcni zasah
    uMin; % minimalni akcni zasah
    U0; % kdy se zacne motor tocit
end
methods
function this = PIDregulator(P, I, D, Ts, U0, uMin, uMax)
    this.setP(P);
    this.setI(I);
    this.setD(D);
    this.setTs(Ts);
    this.setU0(U0);
    this.setUMin(uMin);
    this.setUMax(uMax);
    this.sumE = 0;
end
function setP(this, P)
    this.P = P;
end
function setI(this, I)
    if(I <= 0),
        I= inf;
        msgbox({'Integrační konstanta musí být kladná a
nenulová (I > 0). Nastaveno I = inf'});
    end;
    this.I = I;
end
function setD(this, D)
    if(D<0),
        D=0;
        msgbox({'Derivační konstanta musí být kladná, nebo
rovna nule (D >= 0). Nastaveno D = 0'});
    end;
    this.D = D;
end
function setTs(this, Ts)
    if(Ts<=0),
        Ts=1;
        msgbox({'Vzorkovací perioda nesmí být záporná, ani
nulová(Ts > 0). Nastaveno Ts = 1'});
    end;
    this.Ts = Ts;
end
function setU0(this, u0)
    this.U0 = u0;
end

```

```

end
function setUMin(this, uMin)
    this.uMin = uMin;
end
function setUMax(this, uMax)
    this.uMax = uMax;
end
function u = evalControlAction(this, w, y)
    Eakt = w - y;
    this.sumE = this.sumE + Eakt;
    u = (this.P*Eakt + (this.Ts/this.I)*this.sumE + ...
        (this.D/this.Ts)*Eakt);
    u = sign(u)*(this.U0 + abs(u)); % prah necitlivosti
                                   % <-U0;U0>

    % limitace
    if (u > this.uMax)
        u = this.uMax;
    elseif (u < this.uMin)
        u = this.uMin;
    end
    u = int16(u);
end
end
end

```

```

classdef AngleSensor < handle
properties (SetAccess = private)
    Offset; % offset do nulove polohy kyvadla
end
methods
    %% konstruktor
    function this = AngleSensor(Sensor, Offset)
        this.setOffset(Offset);
        % inicializace cidla
        NXT_SetInputMode(Sensor, 'LOWSPEED_9V', 'RAWMODE',
'dontreply');
        pause(0.2); % pauza pro zapnuti senzoru
    end;
    %% setter
    function setOffset(this, Offset)
        this.Offset = Offset;
    end;
    %% getter
    function Offset=getOffset(this)
        Offset = this.Offset;
    end;
    %% mereni uhlu ve stupnich
    function angle = measureDeg(this)
        %          senzor , adr zar, pocet BYTE, adr reg
        data = COM_ReadI2C(SENSOR_1, 2, 2, 66);
    end;
end

```



```

        angle = (2*double(data(1))+double(data(2))) -
this.Offset;
    end;
    %% mereni uhlu v radianech
    function angle = measureRad(this)
        angle = deg2rad(this.measureDeg());
    end;
    %% mereni uhlu, celkoveho uhlu a otacek
    function [angle, AccAngle, RPM] = measureAll(this)
        data=COM_ReadI2C(SENSOR_1, 8, 2, 66);
        angle=double(data(1))*2+double(data(2));
        AccAngle = typecast(data(6:-1:3), 'int32');
        RPM = typecast(data(8:-1:7), 'int16');
    end;
end
end

function data=spustRegulacniSmycku(PID_power, PID_phi,
Ts,vykresleni)
%% Open connetion
h = COM_OpenNXT();
COM_SetDefaultNXT(h);
%% Initialize motor:
motor = NXTMotor(MOTOR_A, 'Power', 0, 'TachoLimit', 0,
'SpeedRegulation', false);
motor.ActionAtTachoLimit = 'brake';%'HoldBrake'; 'coast',
'brake'
motor.ResetPosition();
%% Initialize sensor
angleSensor = AngleSensor(SENSOR_1 ,226);
%% regulacni smycka
k = 0; % pocitadlo pruchodu cyklem
delka = 3600; % delka regulacniho pochodu, sec
y_motor=zeros(20000,1); % uhel natoceni motoru
y_kyvadlo=y_motor; % uhel natoceni kyvadla
u_power=y_motor; % akcni zasah jdouci do motoru (POWER)
u_phi=y_motor; % akcni zasah znamenajici o kolik
otocit motorem (Phi)
disp('zacatek regulace');
start=tic; % ukonceni po uplynuti max doby
regulace
while true
    k = k + 1;
    % precte uhel natoceni motoru
    y_motor(k) = deg2rad(motor.ReadFromNXT().Position);
    % precte uhel natoceni kyvadla
    y_kyvadlo(k)= angleSensor.measureRad();

    % ukonci reg. pri odchylce > 45 deg (1 rad)
    if(abs(y_kyvadlo(k)) > 1), motor.Stop('off'); break; end;

```

```

    % regulace kyvadla, u_phi je natoceni ramene
    u_phi(k) = PID_phi.evalControlAction(0, y_kyvadlo(k));
    u_power(k) = PID_power.evalControlAction(u_phi(k),
y_motor(k)); % vypocteni akcniho zasahu
    % nastaveni nove hodnoty motoru
    DirectMotorCommand(MOTOR_A, u_power(k), 0, 'off', 'off', 0,
'off');
    while (toc(start) < k * Ts)
        pause(0.0001);
    end %synchronizace periody vzorkovani
    if (toc(start) > delka)
        break; % po dosazeni doby regulace ukonci smycku
    end
end
motor.Stop('off');
%% vykresleni
if vykresleni == 1
    T = 0:Ts:Ts*(k-1);
    figure('color',[1 1 1]);
    subplot(211),stairs(T,(rad2deg(y_kyvadlo(1:k))));
title('vychylka kyvala');grid
    subplot(212),stairs(T,u_power(1:k),'color',[0,0.5,0],
'linewidth',2), hold on
    stairs(T,u_phi(1:k), 'color',[0 0 1]); hold off
    legend('uPower','uPhi','Location','best');
    title('akcni zasah motoru'),grid
% print -dpng Images/pochod7.png;
end
data.u_phi=u_phi(1:k);
data.u_power=u_power(1:k);
data.y_kyvadlo=y_kyvadlo(1:k);
data.y_motor=y_motor(1:k);
%% Close connection
COM_CloseNXT(h);
disp('konec regulace')

%% Check toolbox installation
% verify that the RWTH - Mindstorms NXT toolbox is installed.
if verLessThan('RWTHMindstormsNXT', '3.00');
    error(strcat('This program requires the RWTH - Mindstorms
NXT Toolbox version 3.00 or greater. Go to
http://www.mindstorms.rwth-aachen.de and follow the
installation instructions!'));
end%if

%% Clear and close
COM_CloseNXT all, clear all, close all, clc
%%
vykresleni=0;
ulozData=1;

```

```

filenameMatlab='data_Nastaveni_EX_3.mat';
filename='C:\Users\Zoidbee\Desktop\data_mereni_EX_3.xlsx';
%% .....Regulator.....
Ts = 0.0133; %interval vzorkovani
U0 = 10;13; % zacne se otacet Power=10
uMin = -100; uMax = 100;
% PID_power by mel zustat nemenny, pri tomto nastaveni byly
mereny data, ze kterych byly urceny modely
PID_power = PIDregulator(1.6, inf, 0.005, Ts, U0, uMin,
uMax);
% Nastaveni regulatoru PID_phi
% experimentalni nastaveni
%PID_phi = PIDregulator(56, inf, 3.7, Ts, 0, uMin, uMax);

% nastaveni pomoci ZN metody
% PID_phi = PIDregulator(258.8, inf, 1.33e-3, Ts, 0, uMin,
uMax); % vypoctene nastaveni

% nastaveni metodou pozadovaneho modelu
PID_phi = PIDregulator(300.55, inf, 1.51e-2, Ts, 0, uMin,
uMax); % vypoctene nastaveni
%% spusteni regulacni smycky
data=spustRegulacniSmycky(PID_power, PID_phi, Ts, vykresleni);

%% ulozeni dat do excelu a promenou data.mat
if ulozData == 1
    T=0:Ts:Ts*(length(data.u_phi)-1);
    dataUloz=[T',data.u_phi, data.u_power, data.y_kyvadlo,
data.y_motor];
    xlsxwrite(filename,dataUloz);

    save(filenameMatlab,'data');
end

%% vykresleni
if vykresleni == 1
    T=0:Ts:Ts*(length(data.u_phi)-1);
    figure('color',[1 1 1]);
    subplot(211),stairs(T,(rad2deg(data.y_kyvadlo))),
title('vychylka kyvala');grid
    subplot(212),stairs(T,data.u_power,'color',[0,0.5,0],
'linewidth',2), hold on
    stairs(T,data.u_phi, 'color',[0 0 1]); hold off
    legend('uPower','uPhi','Location','best');
    title('akcni zasah motoru'),grid
% print -dpng Images/pochod7.png;
end

```